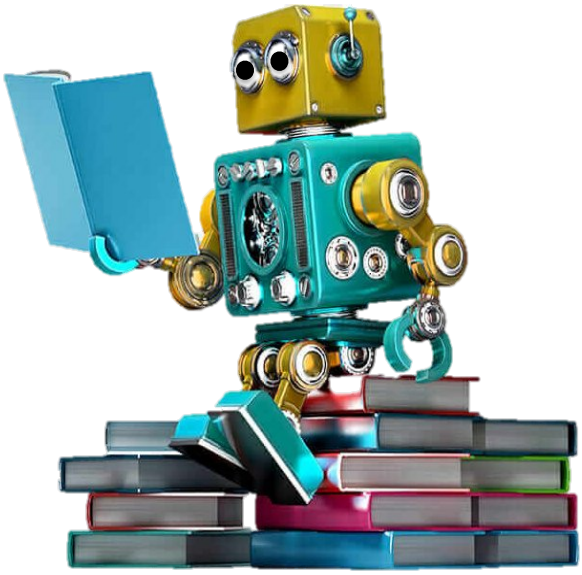
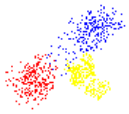
Clustering

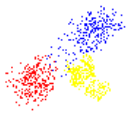


Quienes se parecen, se juntan. *(Refrán francés).*

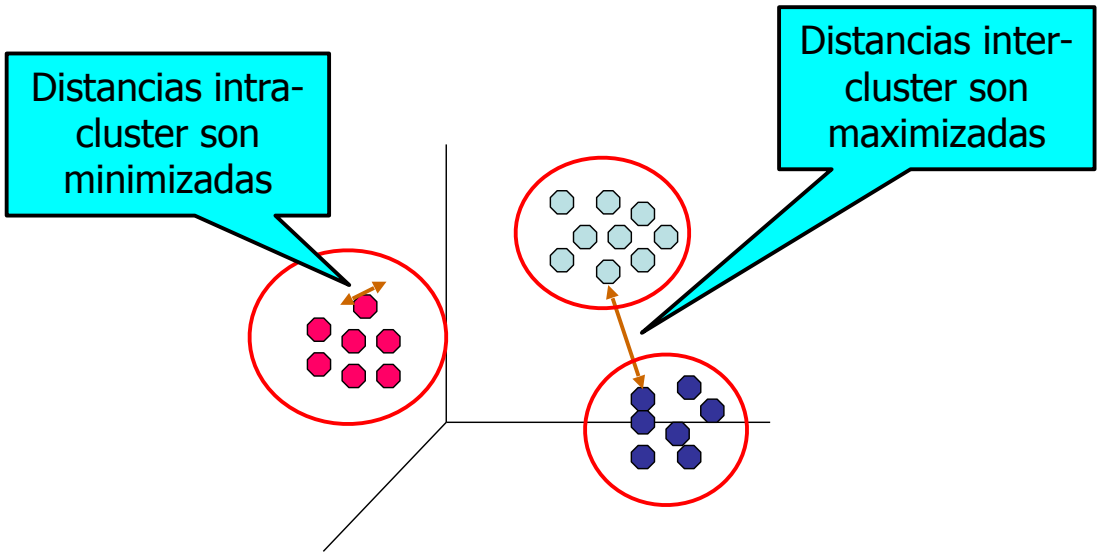


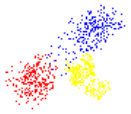
Clustering

- 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?

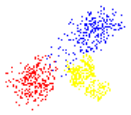
¿Cuántos clusters?

6

2

4

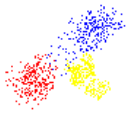
AMBIGUEDAD



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

Clustering de lluvias en Australia



Ej: Clusters en imágenes

Procedimiento: convertir imagen a espacio apropiado, color, gradientes, textura, movimiento, etc. y luego aplicar mean-shift

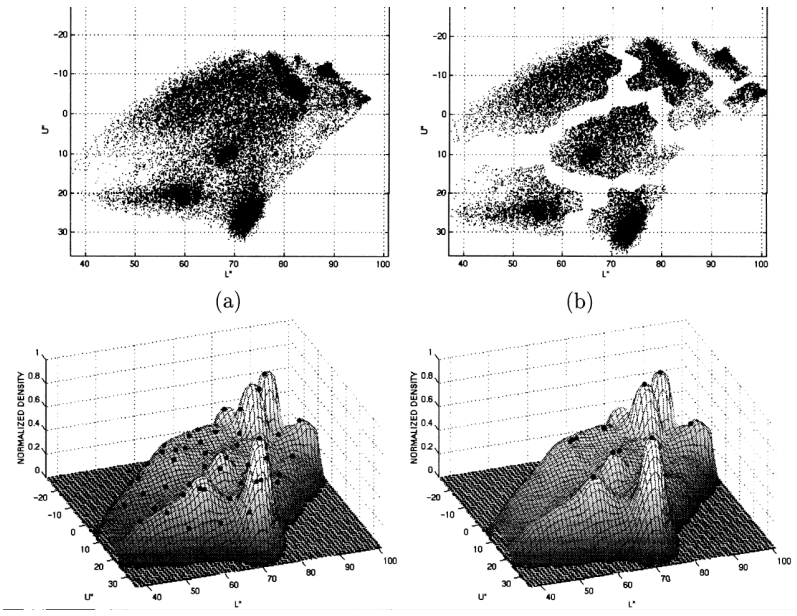
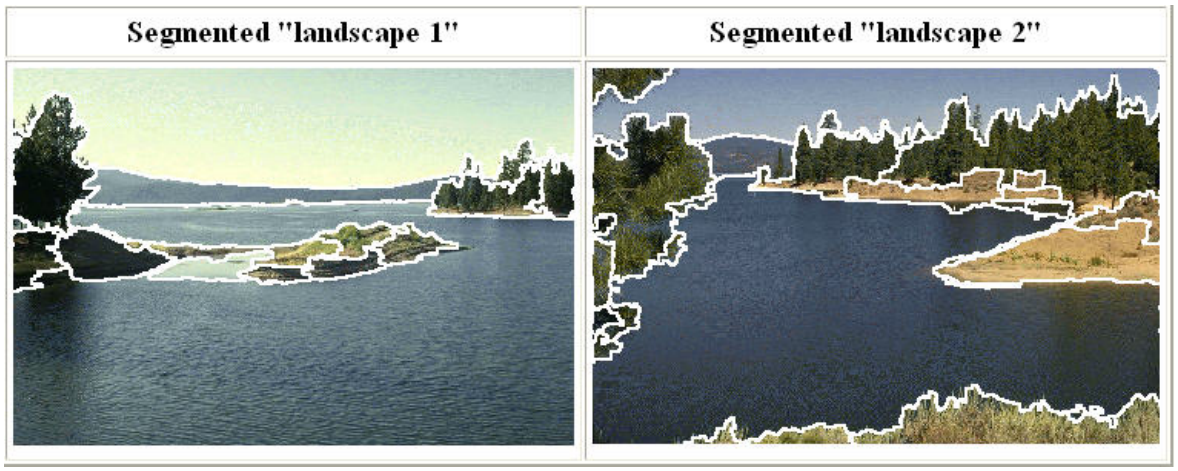
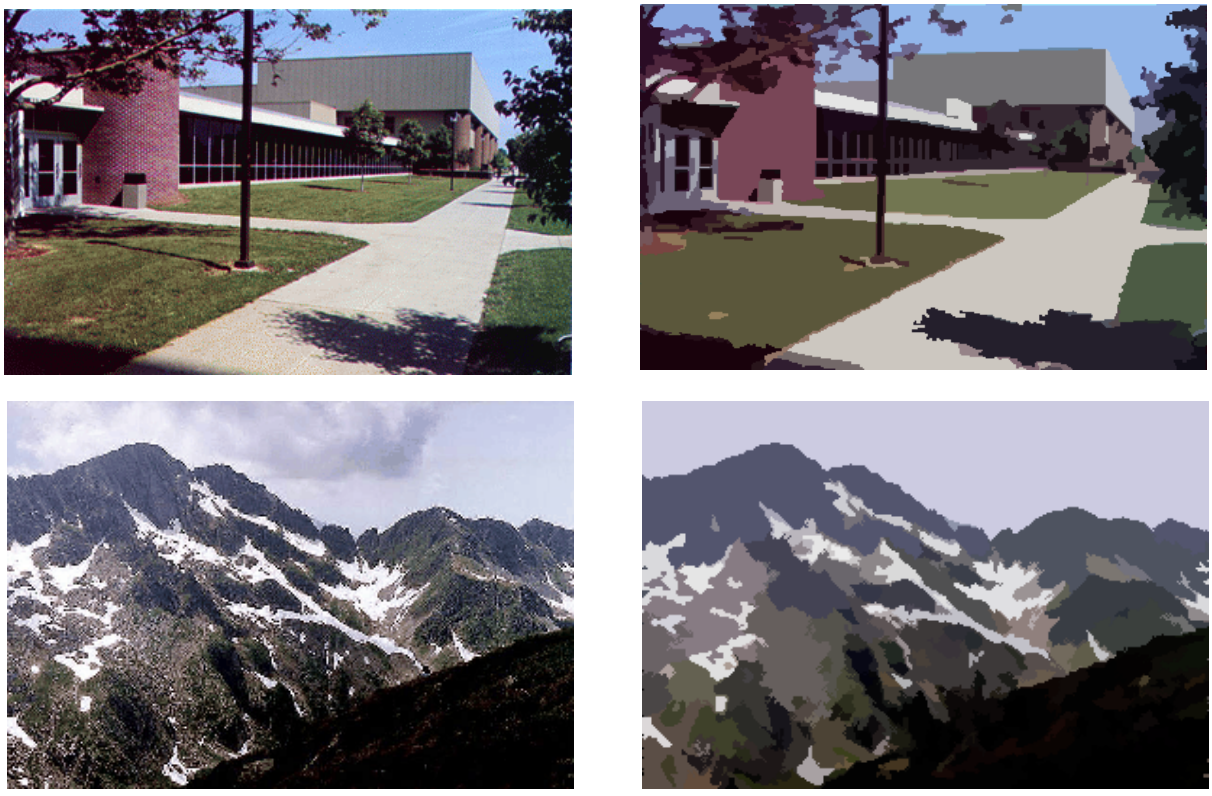


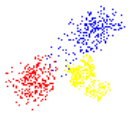
*Image From: Dorin Comaniciu and Peter Meer, Distribution Free Decomposition of Multivariate Data, Pattern Analysis & Applications (1999)2:22–30



http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

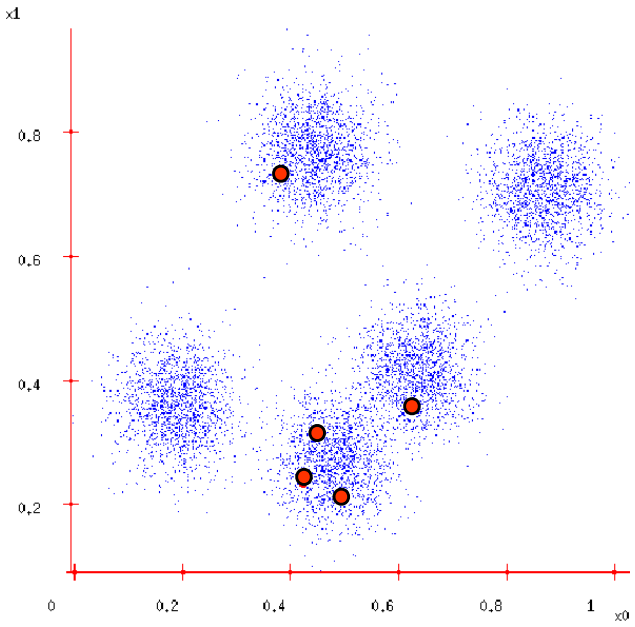


http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html



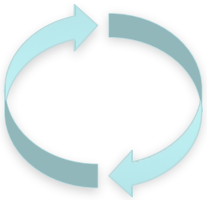
K-Means

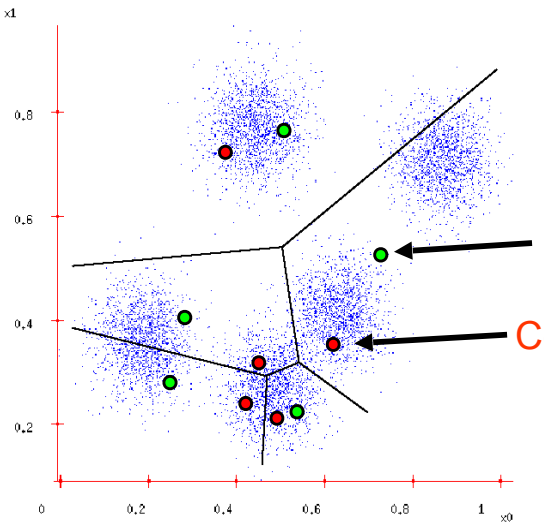
- "Adivinar" cuántos clusters son
- Dar centros iniciales aleatoriamente



K-Means

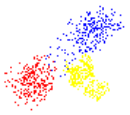
- Asignar a cada punto el centro más cercano
- Cada centro calcula el centroide de los puntos que fueron asignados a él
- Los centros se desplazan a dichos centroides



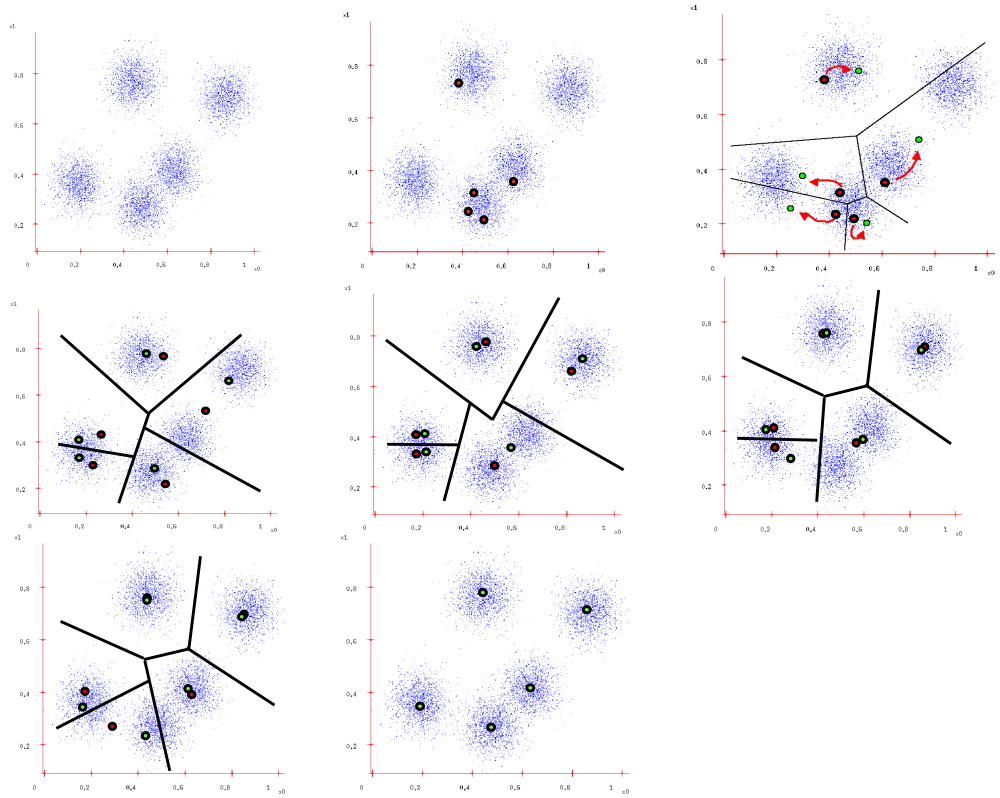


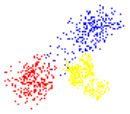
Nuevo Centroide

Centro original



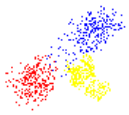
K-Means





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

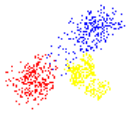
Problema: No siempre se converge a la posición óptima de los centros.

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.



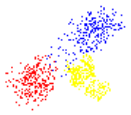
Mean Shift

- 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

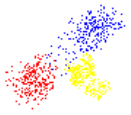
- Inicializar K medias $x_i, \quad i = 1:k$
- Calcular nueva media con (*h: bandwith*):

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

Kernel Gaussiano
(1 ejm de muchos)

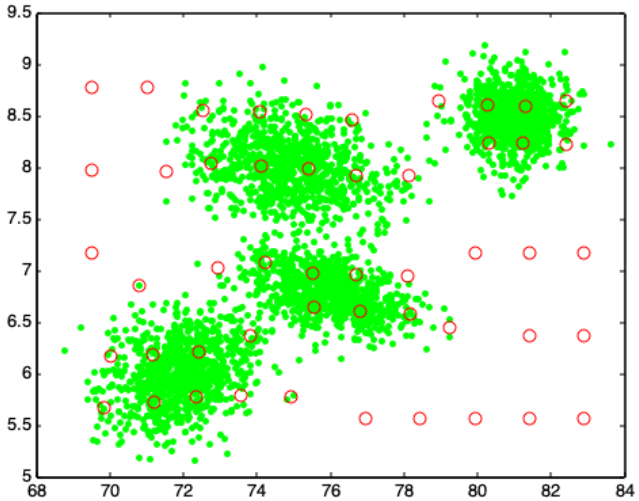
$$g\left(\left\|\frac{x - x_i}{h}\right\|\right) \equiv g(x_i; x, h) = e^{\frac{-|x - x_i|^2}{h^2}}$$

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

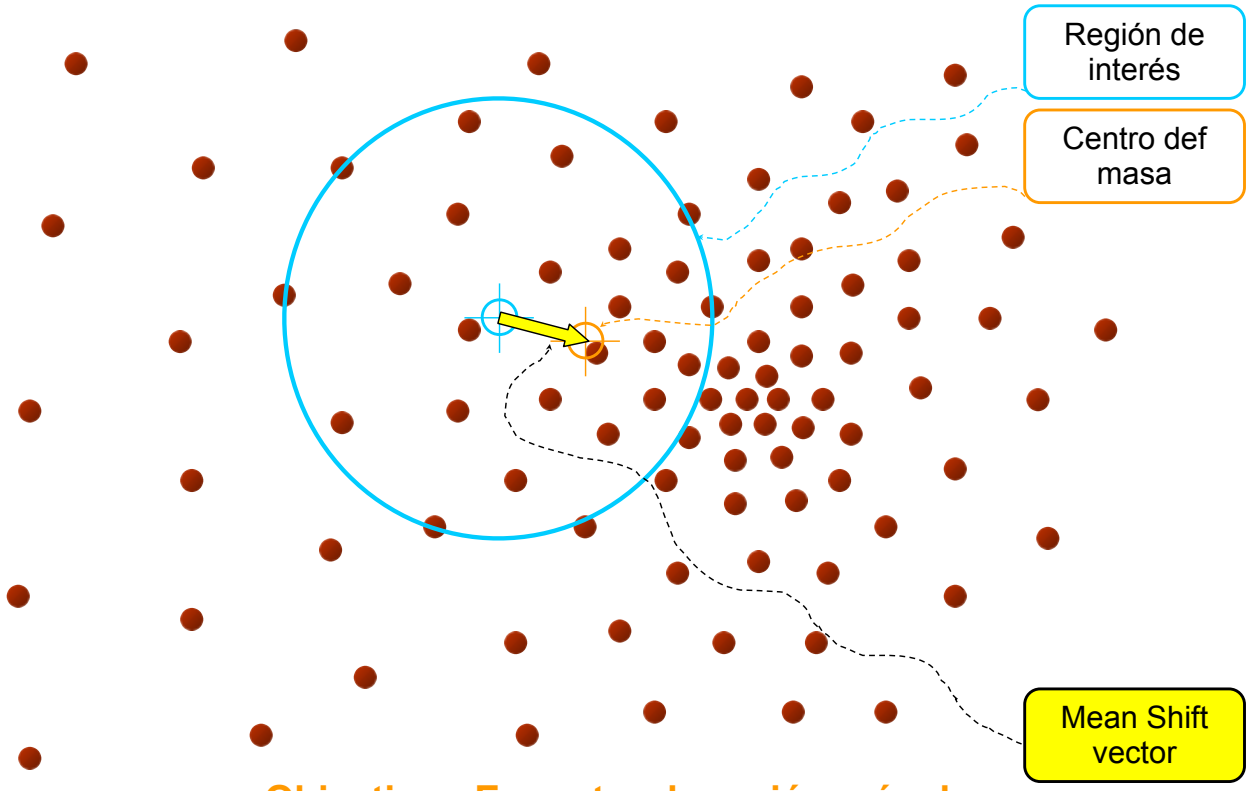


Mean Shift

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

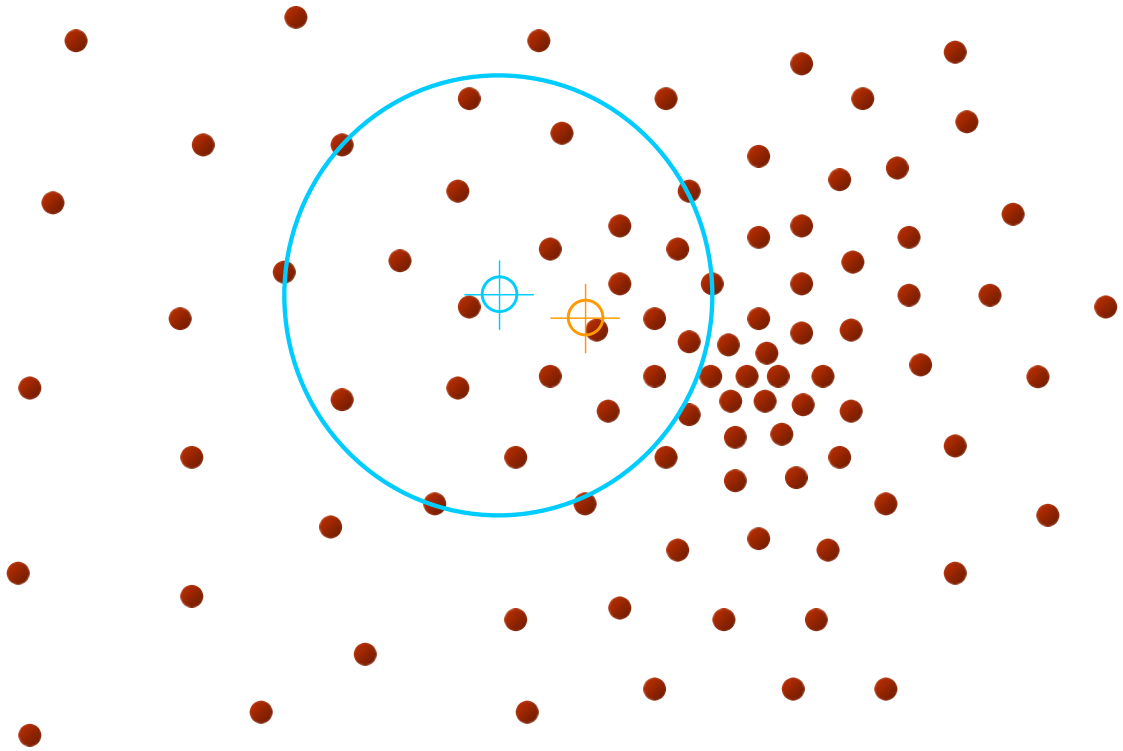


Mean Shift

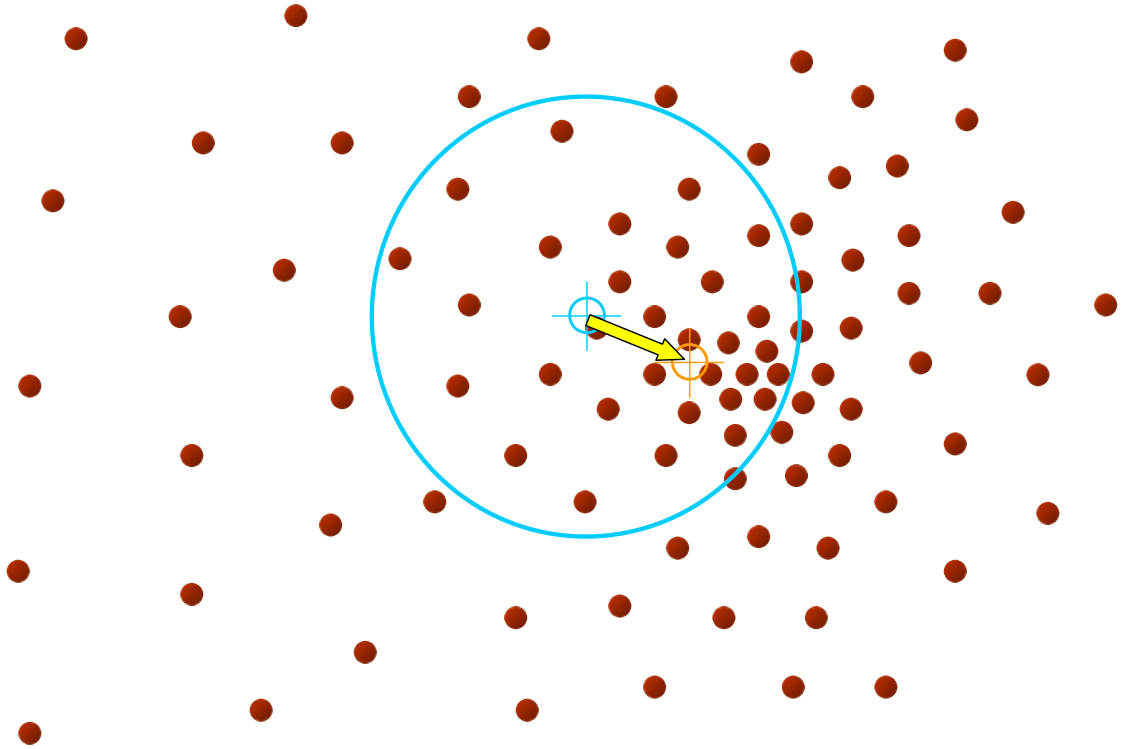


Objectivo : Encontrar la región más densa

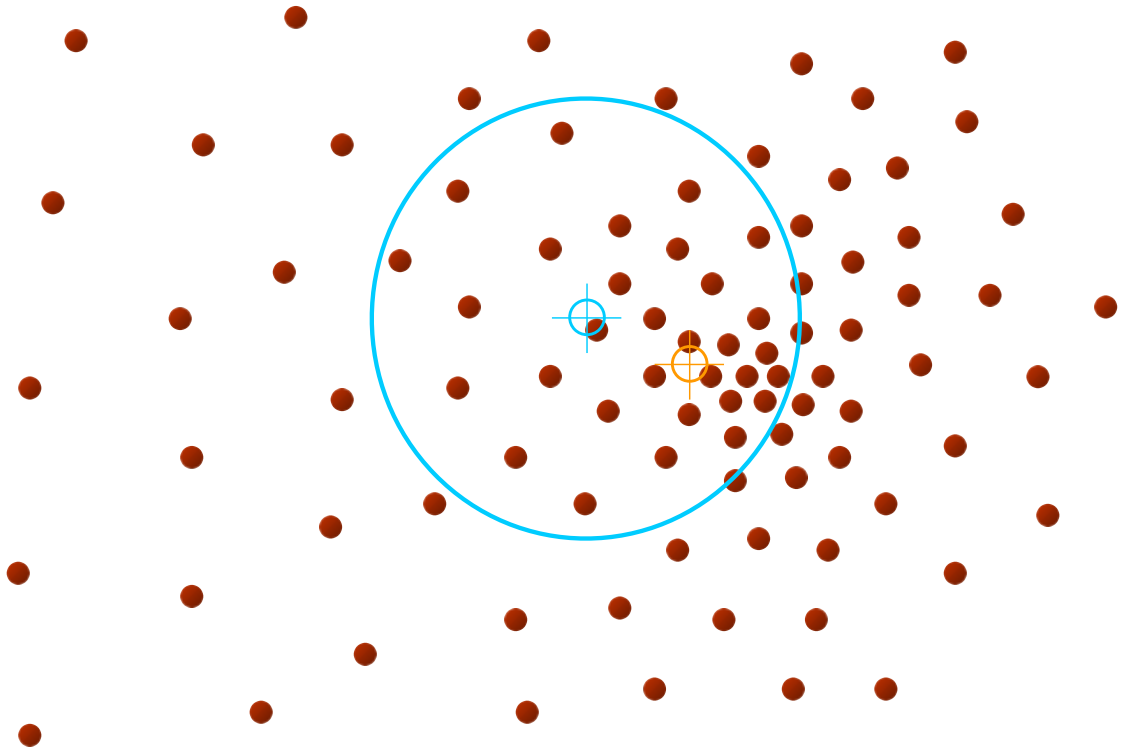
Mean Shift



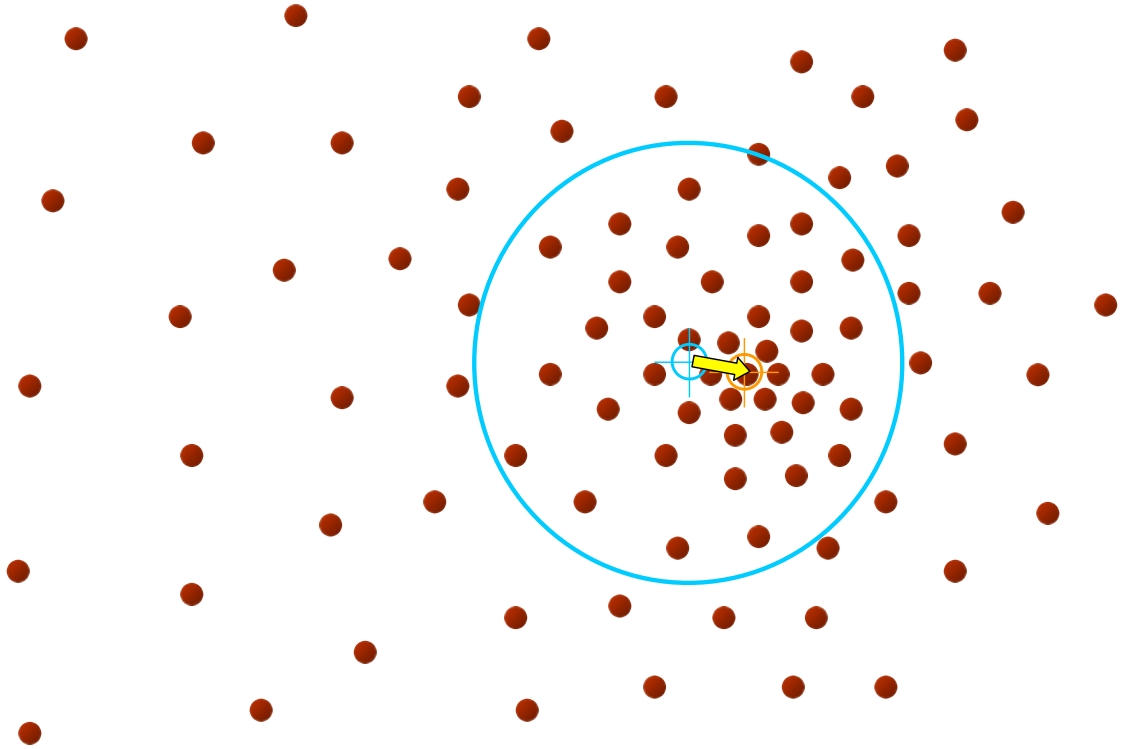
Mean Shift



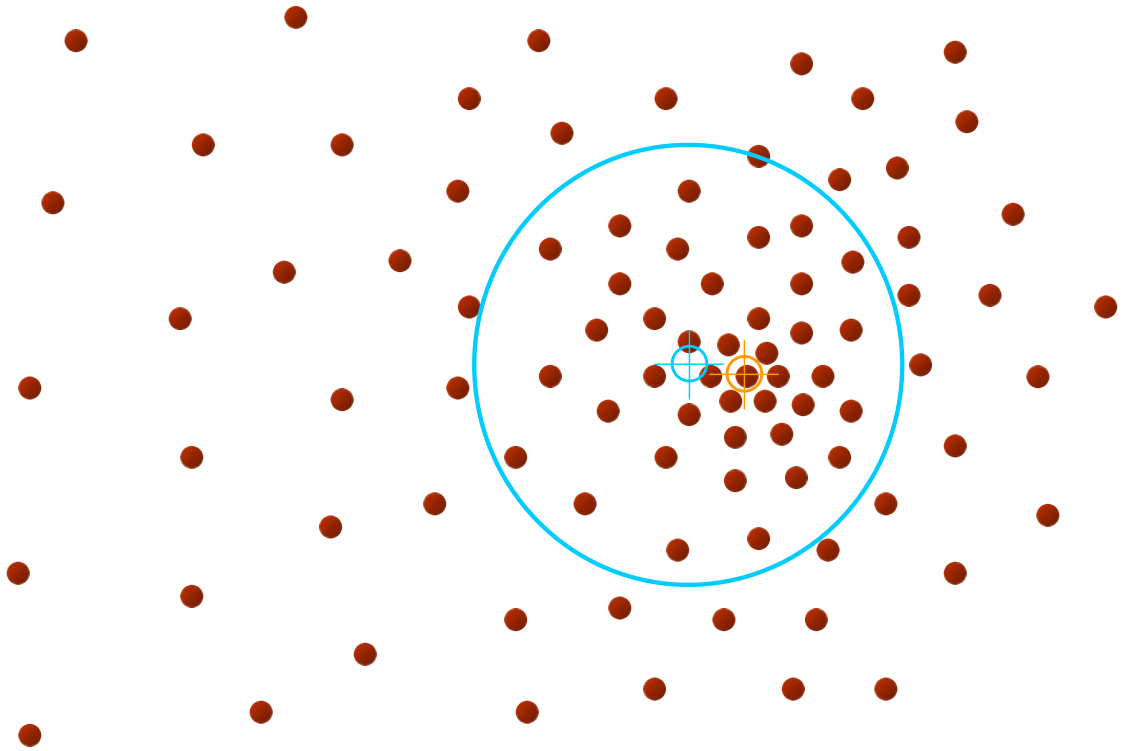
Mean Shift



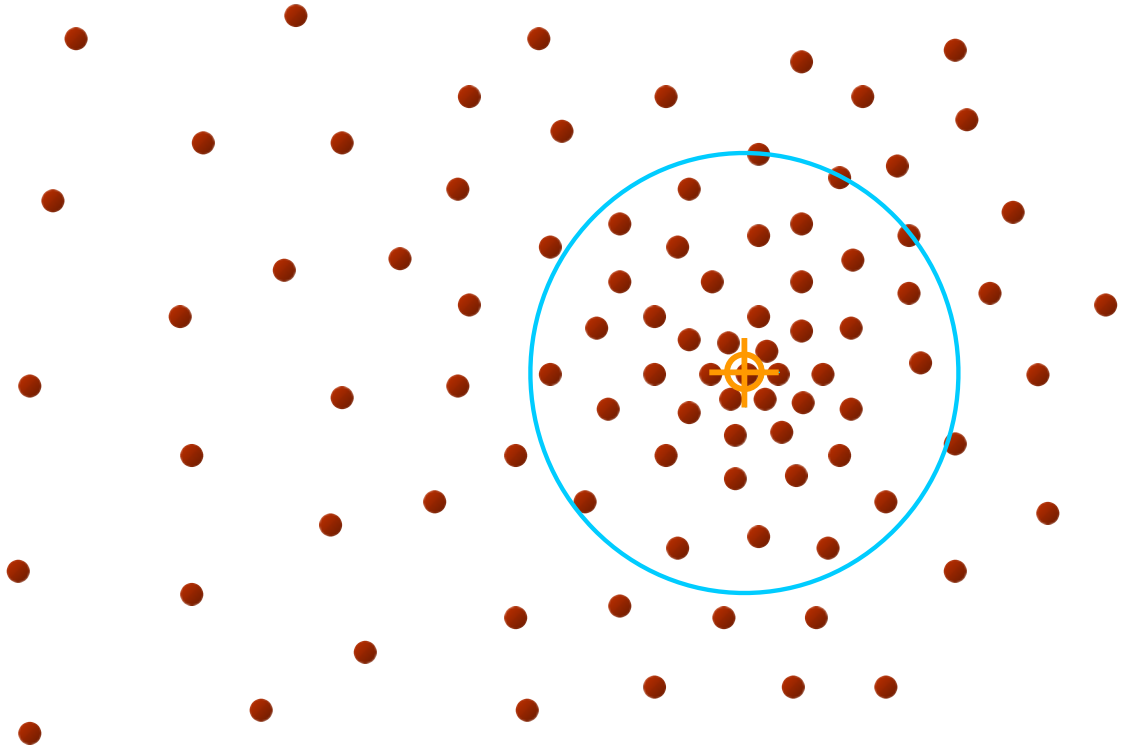
Mean Shift

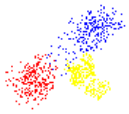


Mean Shift



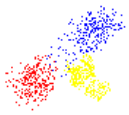
Mean Shift





EM (Expectation Maximization) y GMM

- Sea una base de datos D cuyos datos siguen una distribución dada por una mezcla de distribuciones Gaussianas
- Supongamos todo conocido menos las medias, i.e. los centros o posiciones de las gaussianas



EM (Expectation Maximization) y GMM

$x_1, x_2, \ldots x_R \sim$(i.i.d) $N(\mu, \sigma^2)$

$$P(\text{ data } | \boldsymbol{\mu}_1 \ldots \boldsymbol{\mu}_k) = p(x_1 \ldots x_R | \mu_1 \ldots \mu_k)$$

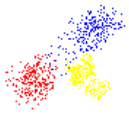
$$= \prod_{i=1}^{R} p(x_i | \mu_1 \ldots \mu_k)$$

$$= \prod_{i=1}^{R} \sum_{j=1}^{k} p(x_i | w_j, \mu_1 \ldots \mu_k) P(w_j)$$

$$= \prod_{i=1}^{R} \sum_{j=1}^{k} K \exp\left(-\frac{1}{2s^2}(x_i - \mu_j)^2\right) P(w_j)$$

Probabilidad de que Xi haya sido generada por la componente j de la mezcla

Probabilidad de componente j



EM (Expectation Maximization) y GMM

$$\frac{\partial}{\partial \mu_i} \log \text{Prob}(\text{data} | \mu_1 \ldots \mu_k) = 0$$

Derivando y usando algebra...

$$\mu_j = \frac{\sum_{i=1}^{R} P(w_j / x_i, \mu_1 \ldots \mu_k) x_i}{\sum_{i=1}^{R} P(w_j / x_i, \mu_1 \ldots \mu_k)}$$

EM (Expectation Maximization) y GMM

$$P(w_j | x_i, \mu_1 .. \mu_k)$$

Prob. de xk bajo la componente i

Prob.a priori de que un punto haya sido generado por la componente i en el tiempo t

Algún valor inicial: $\lambda_t = \{ \mu_1(t), \mu_2(t) \dots \mu_c(t) \}$

Expectation

$$p(w_i | x_k, \lambda_t) = \frac{p(x_k | w_i, \lambda_t) p(w_i | \lambda_t)}{p(x_k | \lambda_t)} = \frac{p(x_k | w_i, \mu_i(t)) * p_i(t)}{\sum_{j=1}^{c} p(x_k | w_j, \mu_i(t)) * p_j(t)}$$

Responsabilidad de componente I (o tambien posteriori)

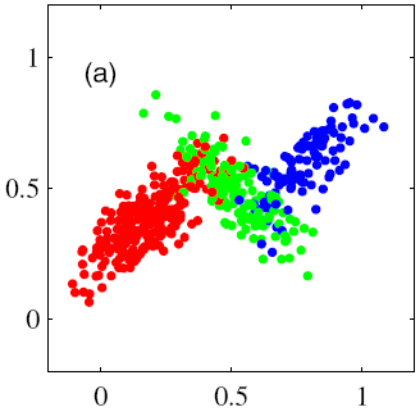
Maximization

$$\mu_i(t+1) = \frac{\sum_k p(w_i | x_k, \lambda_t) * x_k}{\sum_k p(w_i | x_k, \lambda_t)}$$

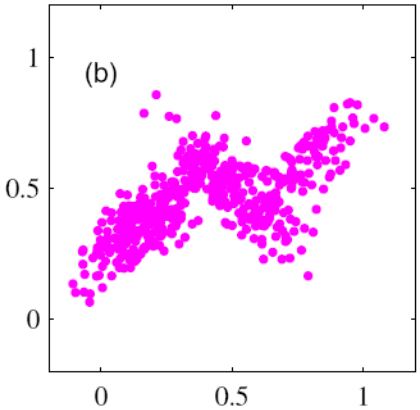
Prob. de xk bajo toda la mezcla de Gaussianas

EM-GMM (detalles)

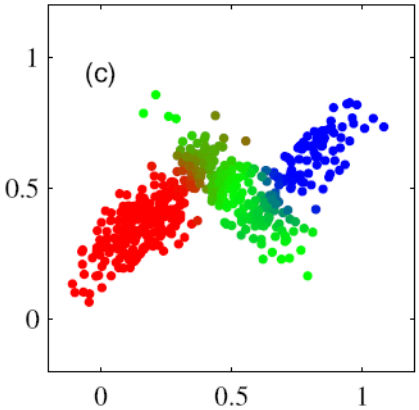
- El proceso termina cuando el loglikelihood total no varia o se hay maximo de iteraciones.
- La responsabilidad me indica que un dato pertenece a varios clusters en *distinta* medida (soft).



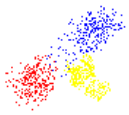
Datos sabiendo su real cluster



Datos en problema

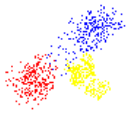


Datos usando GMM (soft)



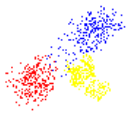
Revisando mas: taxonomía

- 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



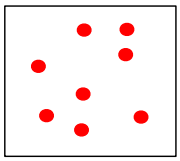
Clustering Jerárquico

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

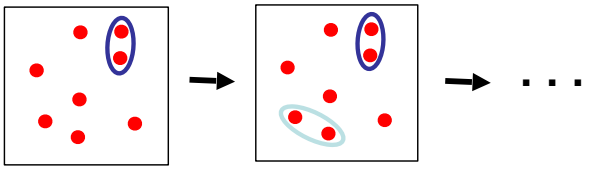


Clustering Aglomerativo

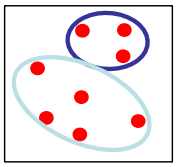
- Inicio: cada punto es un cluster

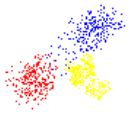


- Iterativamente unir los dos clusters más cercanos

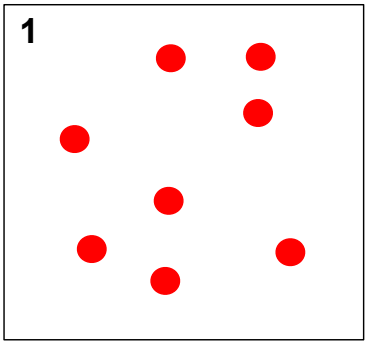


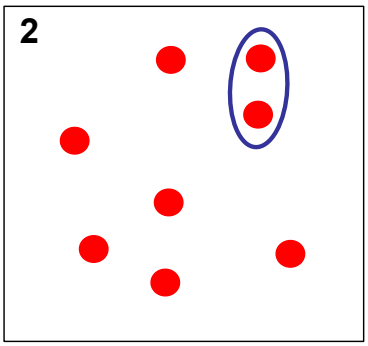
- Parar cuando distancia entre los clusters a unir supera algún umbral pre-determinado

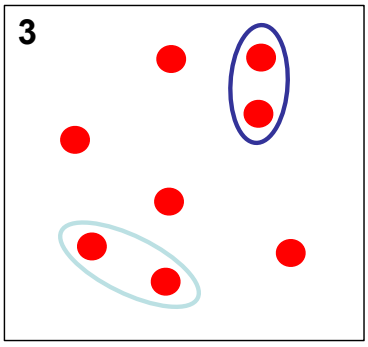


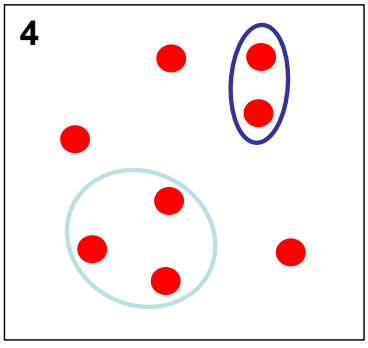


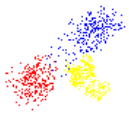
Clustering Aglomerativo



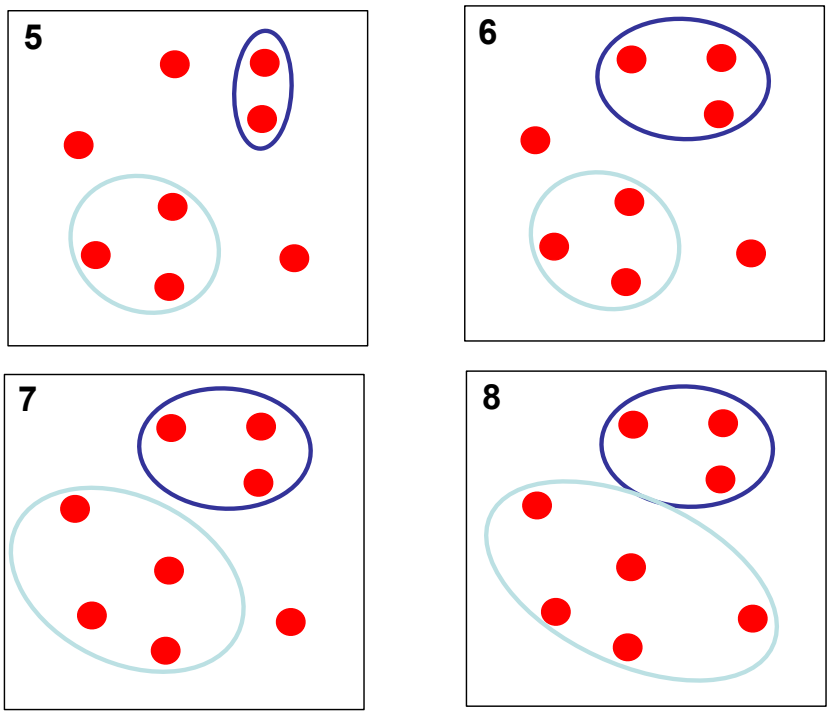


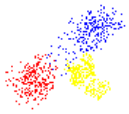






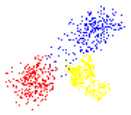
Clustering Aglomerativo



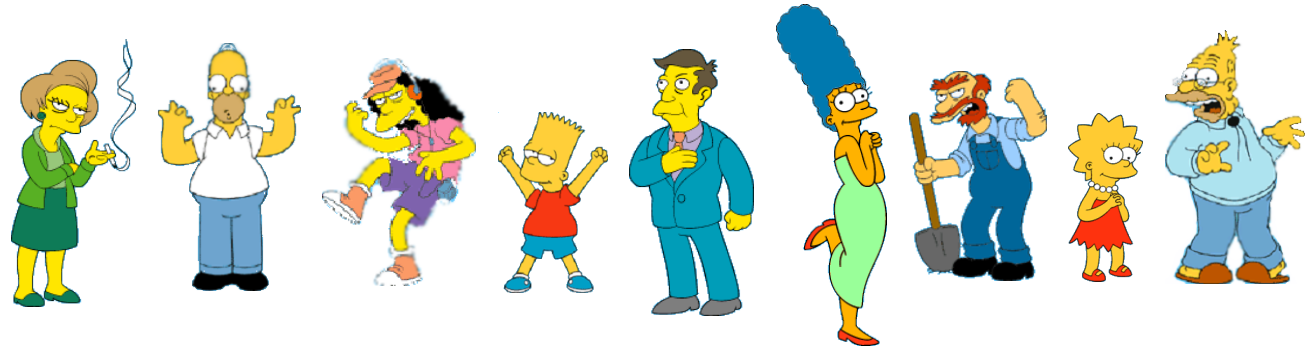


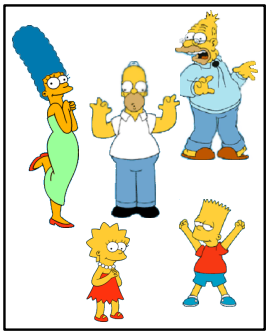
Clustering Aglomerativo

- 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



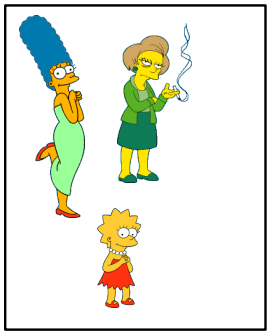
Medidas de similaridad





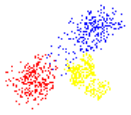
La familia Simpson

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)

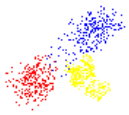
$$D(C_a, C_b) = Max\{d(i,j)\}, i \in C_a, j \in C_b$$

Distancia entre medias (mean distance)

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

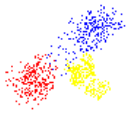
Distancia promedio entre pares (av. pairwise dist.)

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



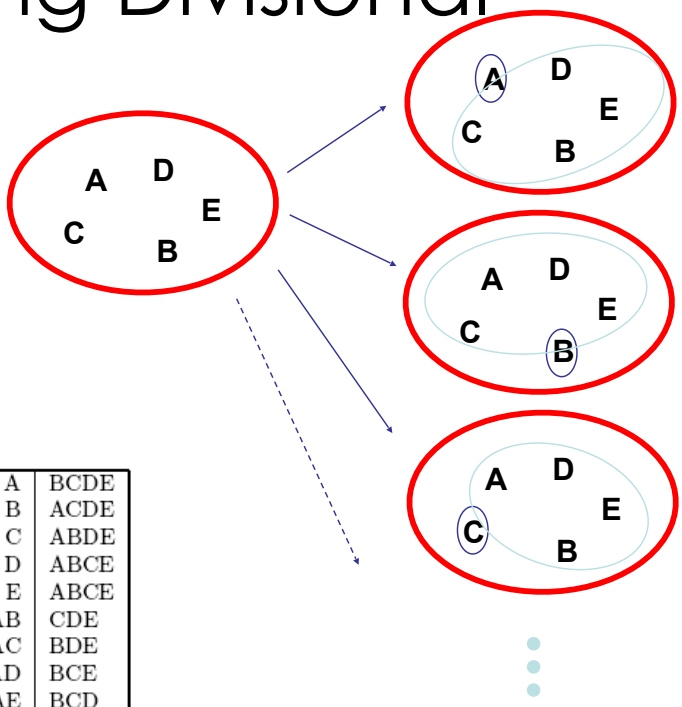
Clustering Divisional

- 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

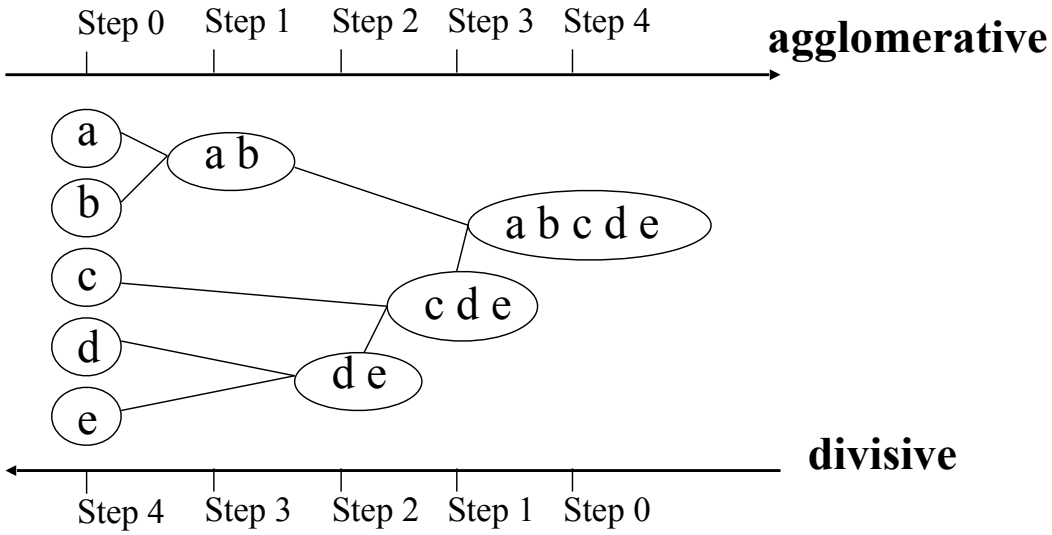


Posibles splits

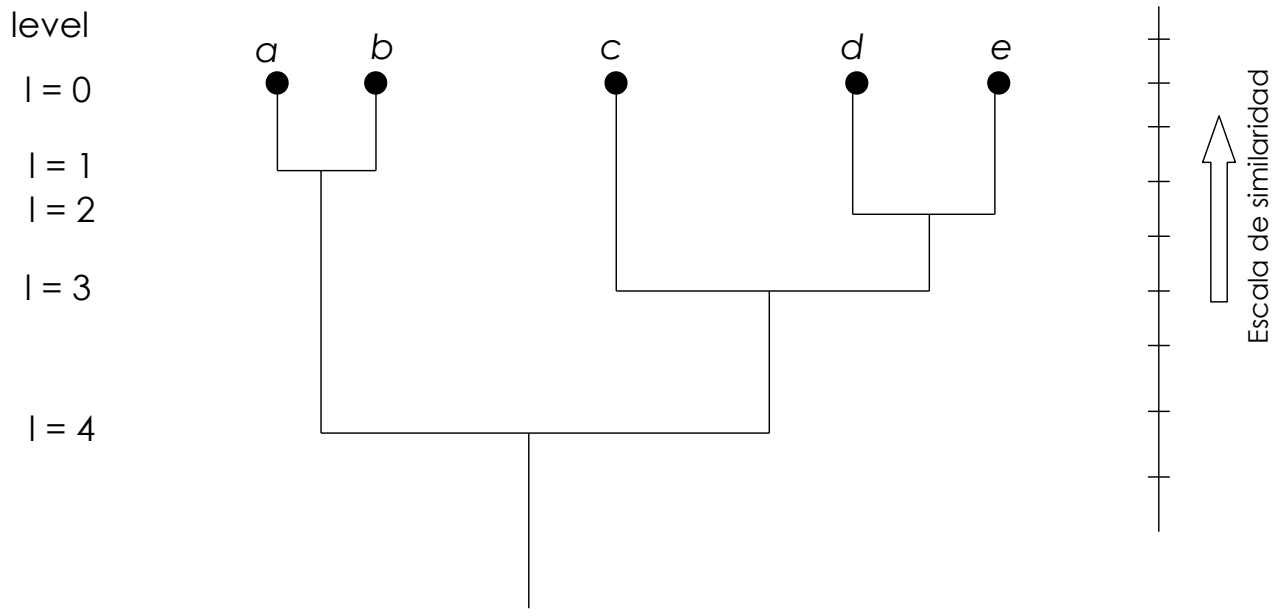
A	BCDE
B	ACDE
C	ABDE
D	ABCE
E	ABCE
AB	CDE
AC	BDE
AD	BCE
AE	BCD
BC	ADE
BD	ACE
BE	ACD
CD	ABE
CE	ABD
DE	ABC

Complejidad muy alta,
Por eso algoritmo k-Means

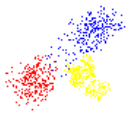
Clustering Divisional

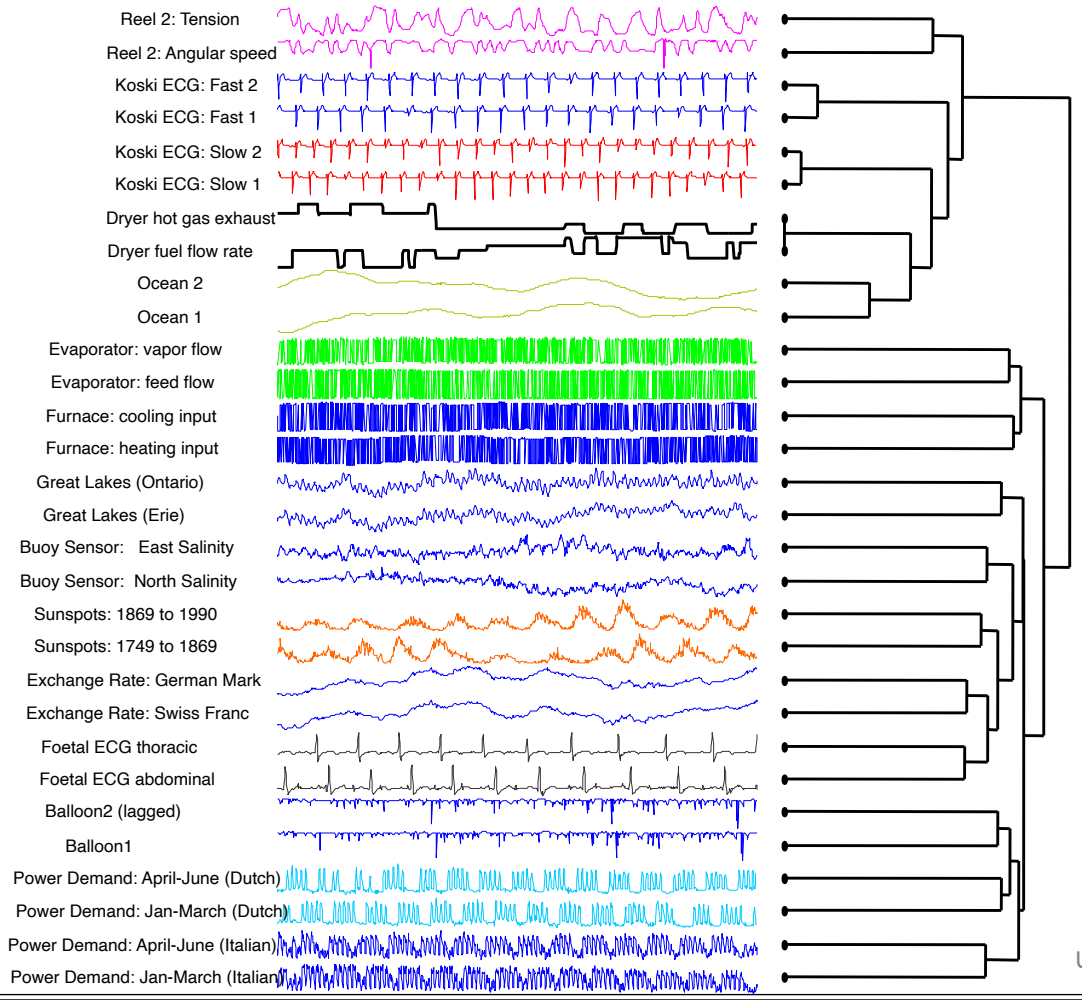


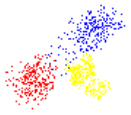
Dendograma



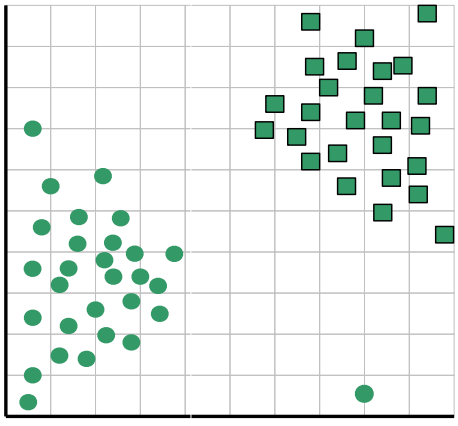
Estructura de árbol utilizada para representar el proceso de clustering jerárquico



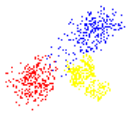




Dendograma y detección de outliers



Outlier



Clustering sobre nominales

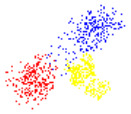
- Sea el siguiente caso para transacciones:

$A = (1,0,0,0,0)$ $\quad |A-B| = \sqrt{2}$

$B = (0,0,0,0,1)$ $\quad |A-C| = \sqrt{3}$

$C = (1,1,1,1,0)$ $\quad |B-C| = \sqrt{5}$

- Si los tratamos de forma numérica tenemos que se mezclaría A y B…
- Sin embargo no comparten ningún ítem, mientras que A y C si.
- **Se requiere nueva métrica!.**



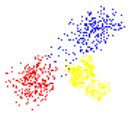
Coef. de Jaccard y Links

- El coeficiente de Jaccard es :

$$J(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

Recomendable para transacciones

- Otro concepto es el de link: si el punto A es vecino de B y el punto C es vecino de B, entonces A y C estan *linked*.
- La lógica es que si 2 puntos están en el mismo cluster entonces deben tener muchos vecinos en comun.



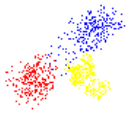
Links y función de calidad

- Dos puntos son vecinos, si *sim*(p_1,p_2)> θ (umbral), donde *sim* se refiere a metrica de similaridad (Coef. Jaccard, euclidiana , etc).
- *link*(p_1,p_2) = numero de vecinos que comparten.
- El criterio de calidad de un clustering es dado por:

$$E_l = \sum_{i=1}^{k} n_i \sum_{p_q, p_r \in C_i} \frac{link(p_q, p_r)}{n_i^{1+2f(\theta)}}$$

Ni : tamaño de cluster i
f(θ): es una funcion tal que $n^{f(\theta)}$ es el nro esperado de vecinos

- Donde para transacciones f(θ)=(1- θ)/(1+ θ).

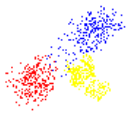


Rock (Robust Clustering using links)

- Luego podemos ver el algoritmo ROCK, que se basa en clustering jerárquico que explora el concepto de links para datos con atributos categóricos
- Usa medida de similaridad:

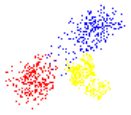
$$g(C_i, C_j) = \frac{link[C_i, C_j]}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}}$$

- Donde el link para clusters se define tal como en el clustering aglomerativo vistas (single link, etc).



Detalles para ROCK

- Los datos nominales se pueden traspasar a datos tipo transaccion, donde cada valor de la variable nominal genera una nueva variable transaccional.
- Los outliers deben tener pocos vecinos, asi son identificables.
- Para base de datos muy grandes, se aplica este algoritmo sobre una muestra de la base de datos.



Experimentos con ROCK

- Base de datos de Market basket de 114586 transacciones :

# Cluster	1	2	3	4	5	6
# Transactions	9736	13029	14832	10893	13022	7391
# Items	19	20	19	19	22	19

# Cluster	7	8	9	10	Outliers
# Transactions	8564	11973	14279	5411	5456
# Items	19	21	22	19	116

- Se halló 5% como outliers (clusters de 1 elemento).
- Es aprox. $O(s^2)$ donde s es tamaño de muestra.