

# Técnicas de Clustering - 2



# Métodos jerárquicos



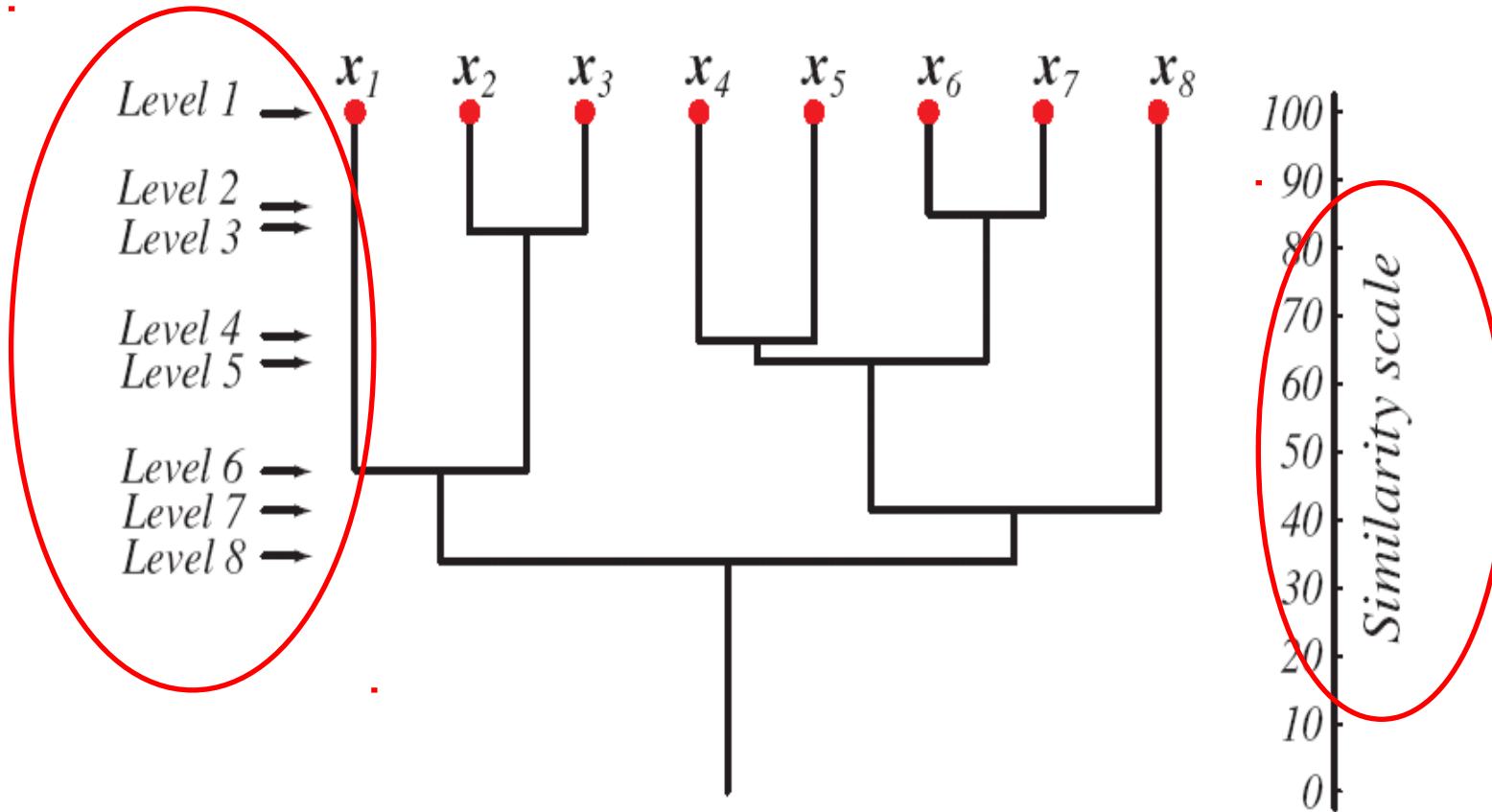
# Introducción

- Hay problemas con jerarquías naturales
  - Taxonomía natural: Reinos, phyla, clases, familias...
- Datos donde hay estructuras a distintos “niveles”
  - Cluster dentro de los clusters, iterativamente
- Más que una división “plana” en muchos clusters, es interesante reconstruir la jerarquía a partir de los datos

# Objetivo

- El clustering jerárquico busca organizar los datos en una serie de particiones anidadas
- En cada nivel, los datos agrupados son más similares entre sí que si se los compara con los de otros grupos

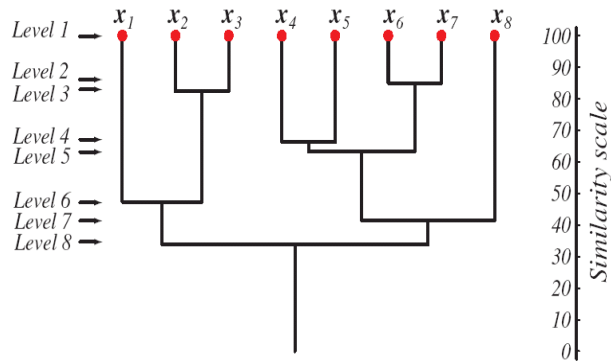
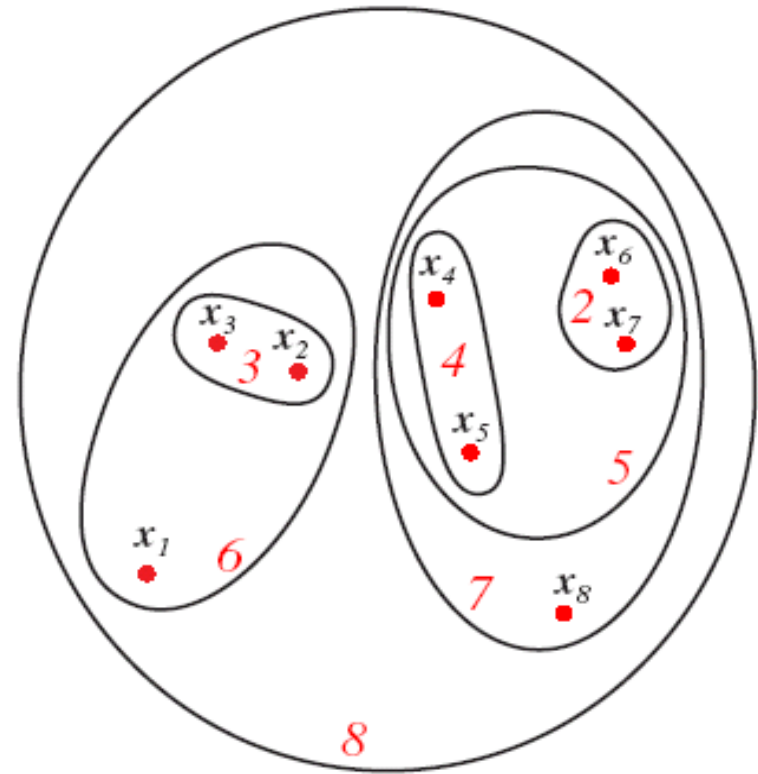
# Representación: Dendrogramas



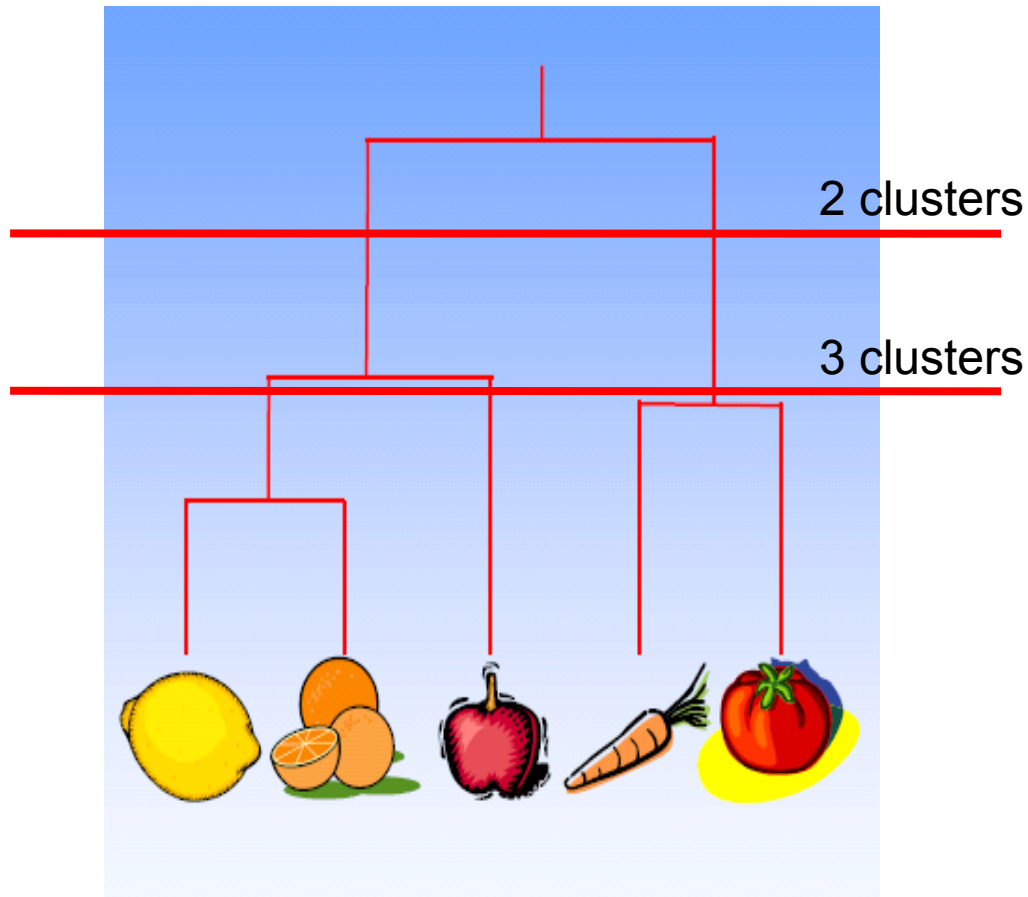
Un dendrograma es un diagrama de divisiones que representa una jerarquía de categorías, basado en general en el grado de similitud o número de características compartidas entre los elementos - Wikipedia

# Otras representaciones

Diagrama de Venn mostrando los mismos datos que el dendrograma anterior. Puede ser menos evidente la distancia entre los grupos.



# Cantidad de clusters

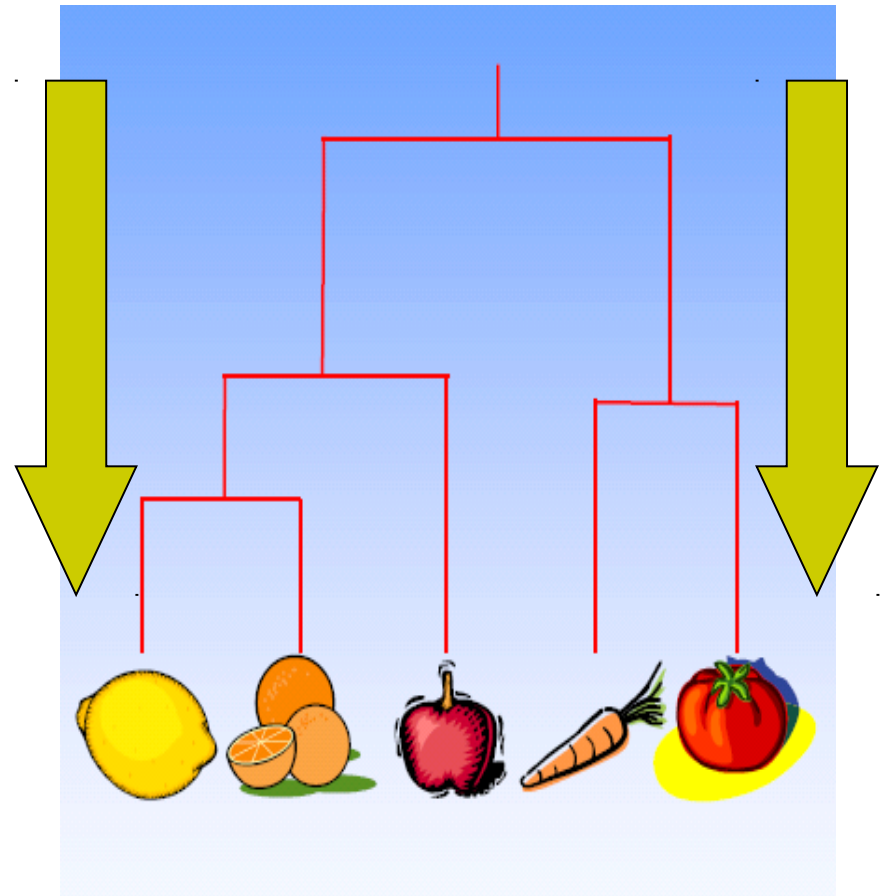


Cortando el árbol al nivel deseado se obtiene un clustering de los datos.

Los sub-árboles conectados a ese nivel forman los clusters

# Estrategias

- Top-Down o divisiva
  - Muy poco usada
  - Comienza con todos los puntos en un cluster
  - A cada paso divide todos los clusters en dos partes de acuerdo a un criterio a optimizar
  - Termina con todos los puntos separados
  - Mejor para pocos clusters

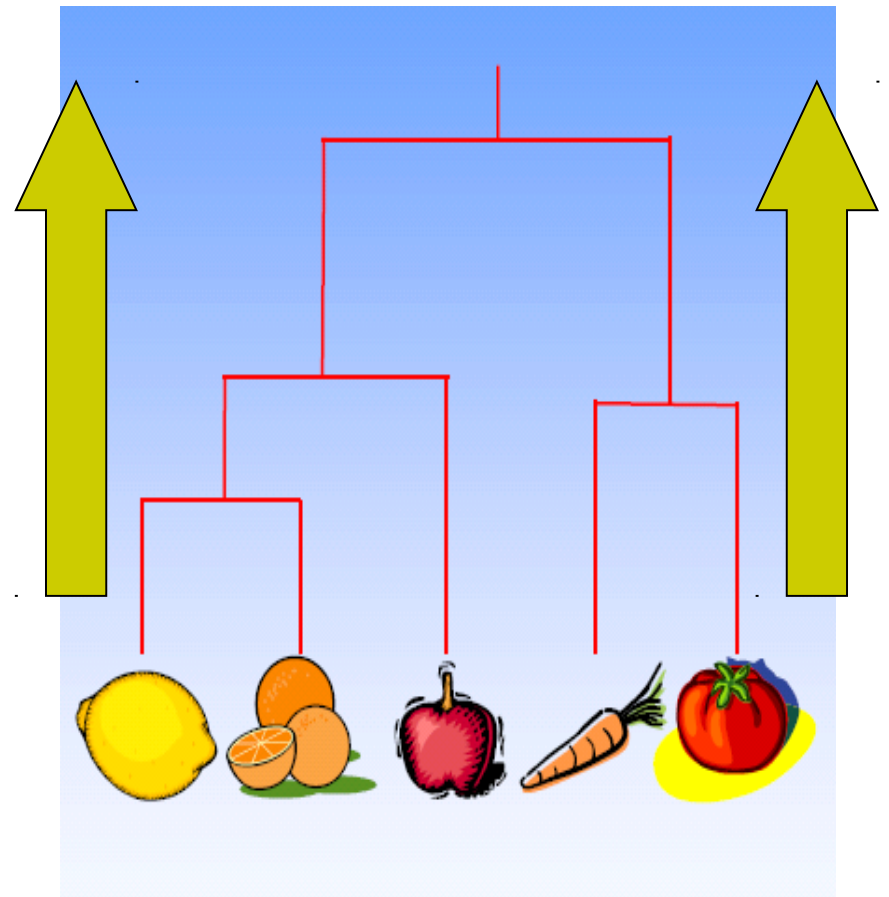




# Estrategias

- Bottom-Up o aglomerativa

- Comienza con todos los puntos separados (clusters individuales)
- A cada paso busca los dos clusters más similares (de acuerdo a algún criterio) y los une
- Termina con todos los puntos en un solo cluster



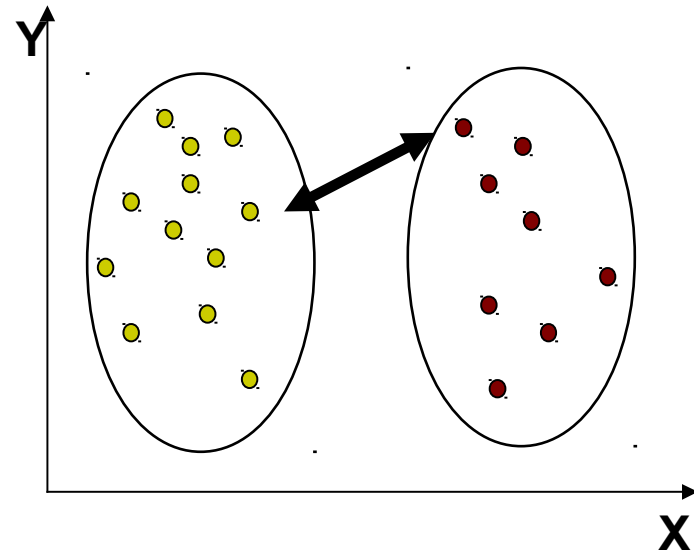
**Ésta es la que usamos de aquí en más**

# Criterios de similitud

- A cada paso tenemos que evaluar qué clusters juntar.
- Necesitamos una medida de similitud entre clusters (no es lo mismo que entre datos).
- Distintas medidas dan como resultado distintas jerarquías (distintas soluciones).

# Single Linkage

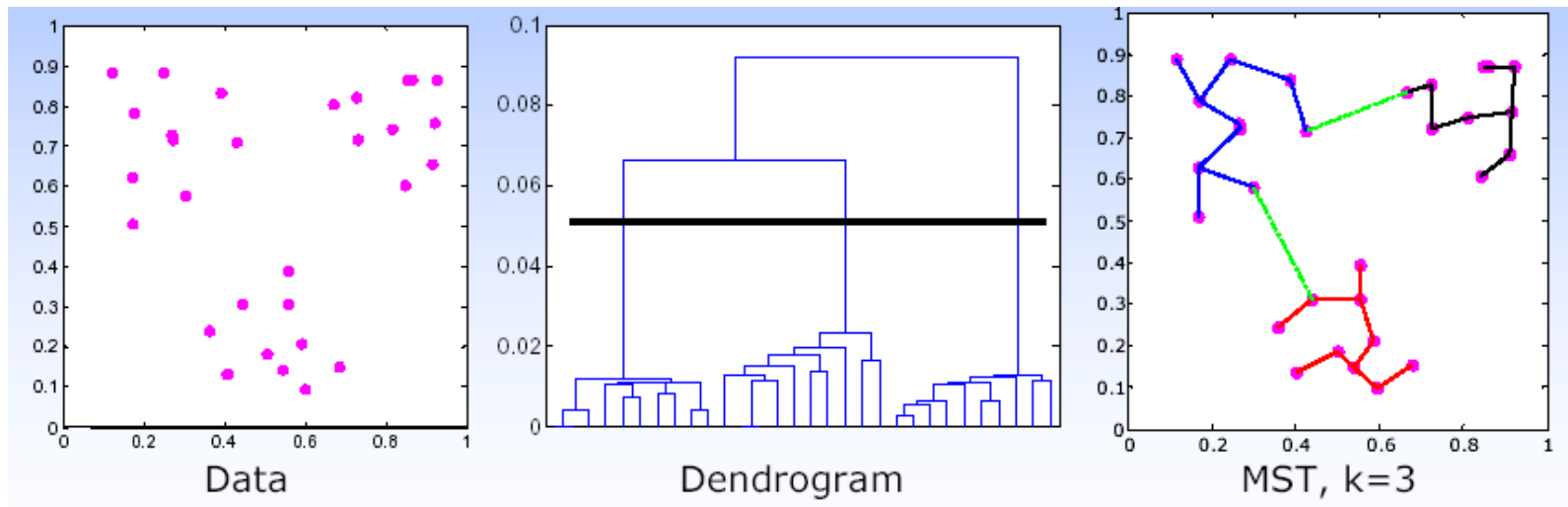
- Medida de distancia: distancia mínima entre pares de puntos, uno en cada cluster.
- Agrupa “vecinos de vecinos”
- Busca “alta conectividad”, no grupos compactos.



$$d_{min}(\mathcal{D}_i, \mathcal{D}_j) = \min_{x_t \in \mathcal{D}_i} \min_{x_s \in \mathcal{D}_j} d(x_s, x_t)$$

# Single Linkage es...

- Juntar cada grupo con el punto más cercano, empezando de puntos separados...Suenan a algo?
- SL produce el Minimum spanning tree de los datos
  - Algoritmo de Prim
- Cortando las conexiones más largas se encuentran los clusters

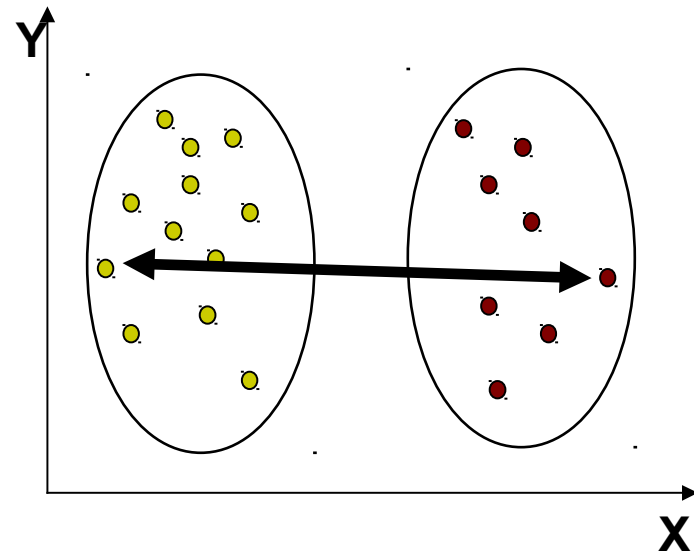


# Single linkage

- Problemas
  - Muy dependiente de outliers
  - Ineficiente:  $O(n^2)$  como mínimo
  - No puede corregir errores previos, malas asignaciones

# Complete Linkage

- Medida de distancia: distancia máxima entre pares de puntos, uno en cada cluster.
- Agrupa “conjuntos completamente vecinos”
- Busca grupos compactos, estilo k-means.



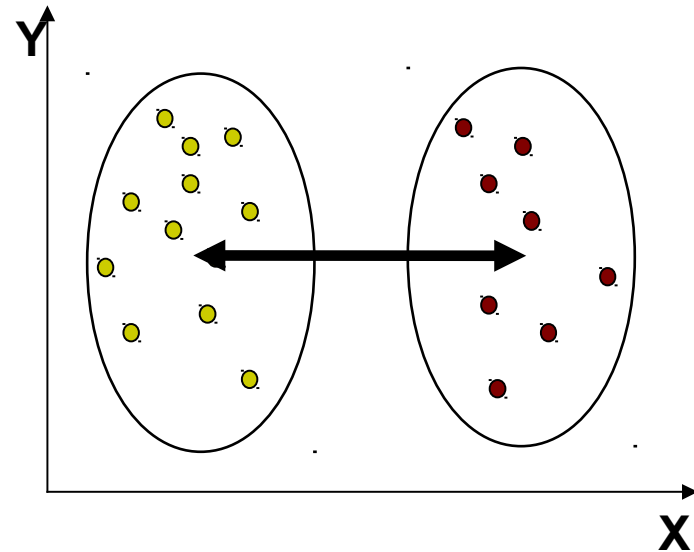
$$d_{max}(\mathcal{D}_i, \mathcal{D}_j) = \max_{x_t \in \mathcal{D}_i, x_s \in \mathcal{D}_j} d(x_s, x_t)$$

# Complete linkage

- Dependiente de outliers (pero diferente)
- Determinista
- Pero:
  - Ineficiente:  $O(n^2)$  como mínimo
  - No puede corregir errores previos, malas asignaciones
  - No tiene el poder de SL de formar clusters con formas arbitrarias.

# Average Linkage

- Medida de distancia: media de la distancia entre pares de puntos, uno en cada cluster.
- Intermedia entre las dos anteriores
- Busca grupos conectados y compactos.



$$d_{average}(\mathcal{D}_i, \mathcal{D}_j) = \text{average}_{x_t \in \mathcal{D}_i, x_s \in \mathcal{D}_j} d(x_s, x_t)$$



# Otro criterio

- Criterio de Ward
  - Busca los cluster que al juntarlos dan el menor aumento en la suma de distancia cuadrada entre sus componentes
  - Equivalente a k-means

# Práctica

- Ver códigos en R.
- Hay problemas para comparar con k-means y todos los linkages entre sí.

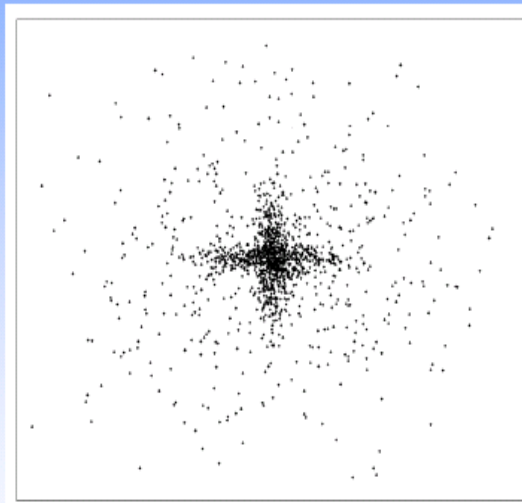
# Resumen

- Vimos los dos métodos de clustering más comunmente usados:
  - K-means (PAM): divisivo, clusters compactos
  - Single/Aver/Complete linkage: construyen dendogramas con distintos criterios
- En el 95% de las aplicaciones reales se usa uno de los dos.
- Ambos son:
  - Heurísticas simples, sin gran teoría detrás
  - Fáciles de implementar
  - En la práctica los dos dan resultados “razonables” muchas veces
- Son base de métodos más elaborados

# Mensaje hasta acá

- No existe “el mejor” método de clustering
- Todos parten de imponer una estructura
- Si aciertan, entonces el resultado es bueno

No hay algoritmos que puedan resolver estos dos problemas eficientemente



Mixture of 3 Gaussians



Two “half rings”

# Otros métodos



# SOM

- Self-Organizing Map. Desarrollado por T. Kohonen en los '80
- Basado en el mapeo que hace la corteza visual del cerebro
- Idea: Tengo un “mapa”, un conjunto de centros con una topología dada (generalmente una grilla equiespaciada en 1, 2 o hasta 3 dimensiones).

# SOM

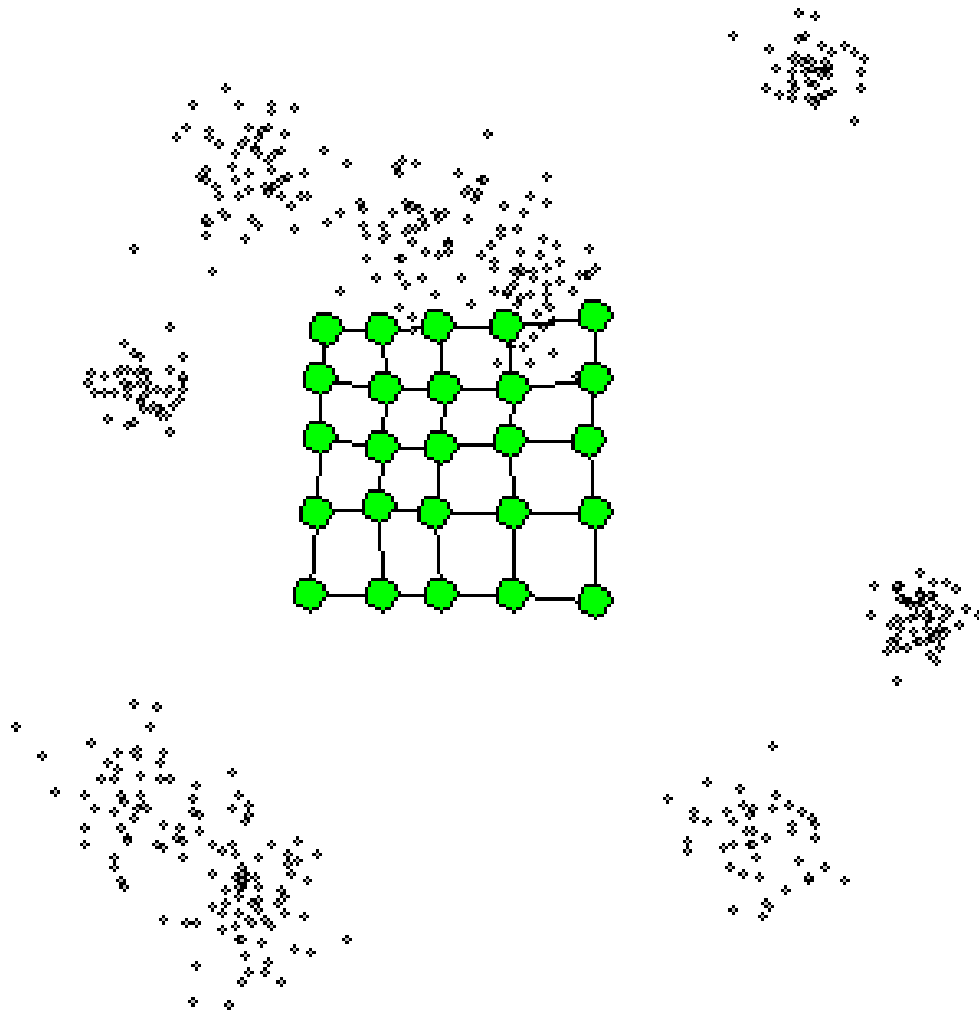
- Trata de ajustar el mapa a los datos, respetando la topología (que se deforme lo menos posible)
- Los nodos del mapa son centros de clusters que compiten entre sí para estar cerca de los datos.

# SOM

- Algoritmo simple:
  - Tomo un punto al azar
  - Busco el nodo más cercano
  - Muevo el nodo (y sus vecinos directos en la grilla) un paso dado en la dirección del punto
  - Itero

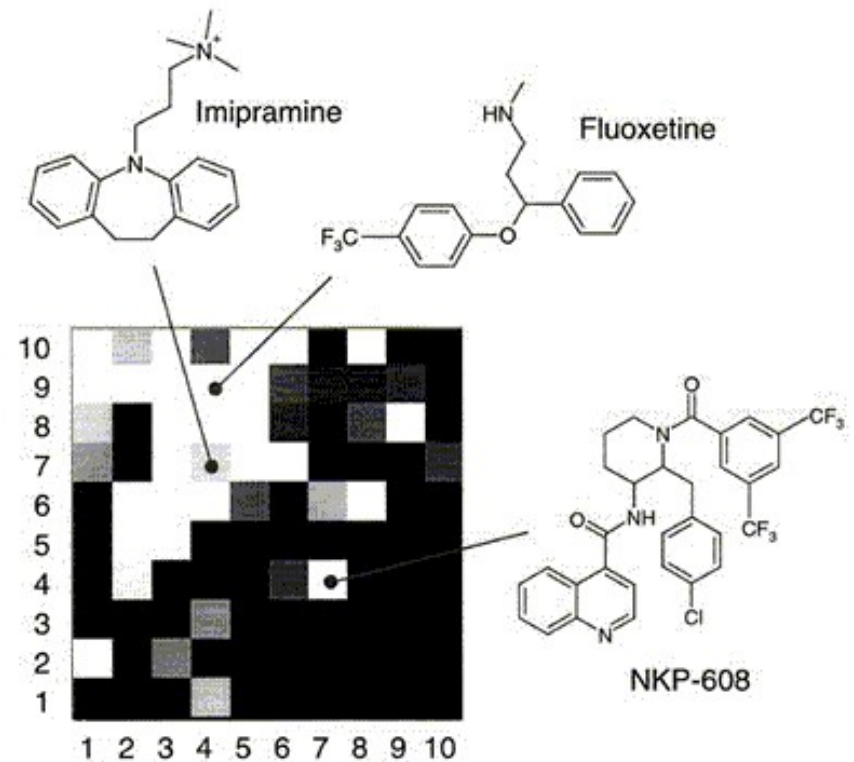


# SOM Ejemplo bi-dimensional



# SOM - Análisis

- Una vez que se ajusto la grilla a los datos, se suele analizar el resultado inversamente:
  - Se grafica el mapa en su estructura original
  - Se muestran en el gráfico los puntos del problema analizado que quedaron asignados a cada nodo
  - Se usa un código de grises para indicar distancias entre los nodos???? Ideas de vacío????



# Model-based methods

- Propone un modelo funcional para los datos
  - Por ejemplo, una mezcla de distribuciones Gaussianas.
- Busca ajustar los parámetros del modelo a los datos
  - Típicamente, usando Expectation-Maximization
- Se regula la complejidad del modelo propuesto de acuerdo a la cantidad de datos
  - Iguales o distintas covarianzas, etc

# Métodos de ensamble

- Propone juntar la información de un montón de soluciones de clustering diversas.
  - Juntar evidencia: Si un par de puntos están siempre juntos, dejarlos así. Y lo contrario.
  - Usa una matriz de evidencias como nueva matriz de distancias y clusterizarla
  - Distintas formas de crear las soluciones de clustering
  - Distintos métodos para clusterizarla

# Métodos espectrales

- Pensar el problema como el de encontrar componentes disjuntas en un grafo
  - Construir la matriz de similaridad del grafo
    - Grafo de k-vecinos
    - Medida de similaridad apropiada (gausiana)
  - Calcular una proyección PCA de esa matriz en bajas dimensiones
  - Buscar clusters en ese espacio
  - Muy efectivo!