DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Clustering jerárquico

© Fernando Berzal, berzal@acm.org

Clustering jerárquico

- Métodos jerárquicos
  - DIANA [Divisive Analysis]
  - AGNES [Agglomerative Nesting]
- Extensiones
  - BIRCH
    [Balanced Iterative Reducing and Clustering Using Hierarchies]
  - CURE
    [Clustering Using Representatives]
  - Chameleon
    (Hierarchical Clustering Using Dynamic Modeling)
  - ROCK
    [RObust Clustering using linKs]

Métodos de agrupamiento

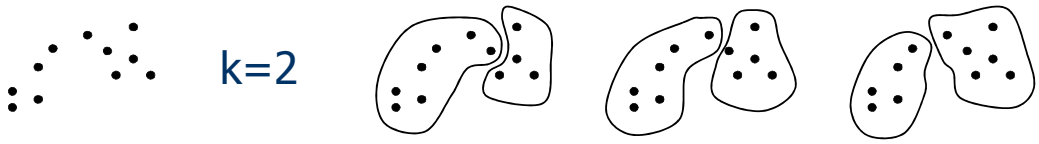
Familias de algoritmos de clustering:

- **Agrupamiento por particiones**
  k-Means, PAM/CLARA/CLARANS, BFR
- **Métodos basados en densidad**
  DBSCAN, Optics, DenClue
- **Clustering jerárquico**
  Diana/Agnes, BIRCH, CURE, Chameleon, ROCK

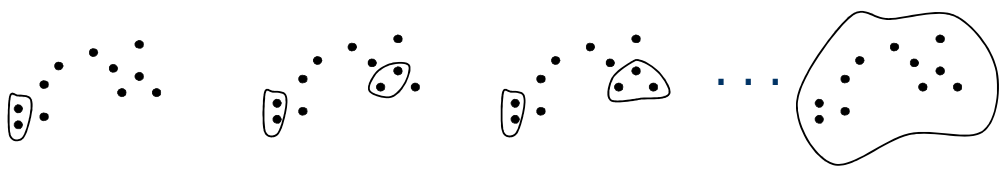
...

Métodos de agrupamiento

Clustering por particiones (suele fijarse k)

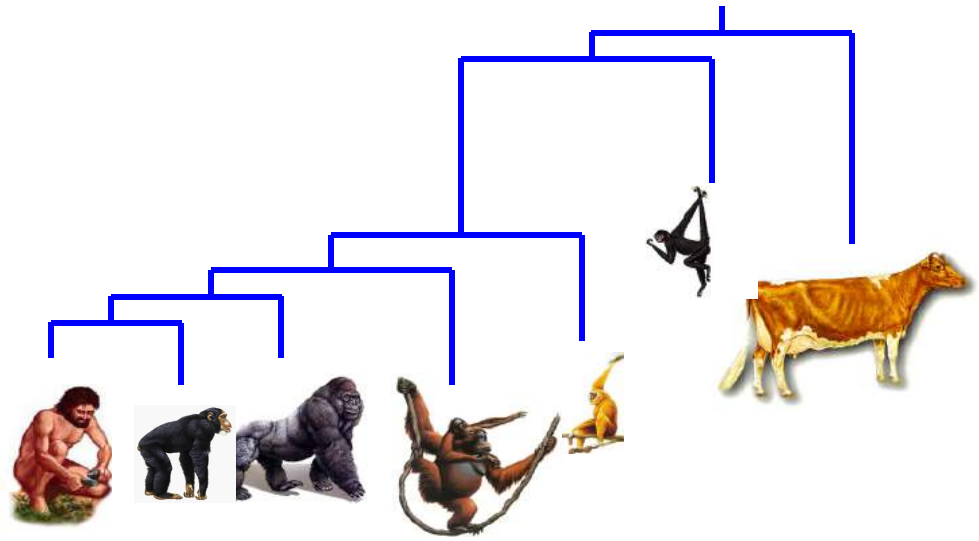


Clustering jerárquico (no se fija k)



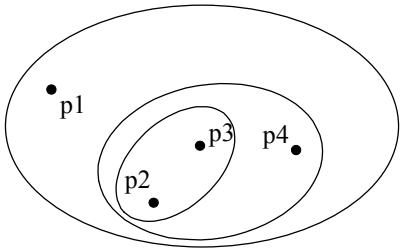
Se obtiene como resultado final
un conjunto de agrupamientos.

Clustering jerárquico

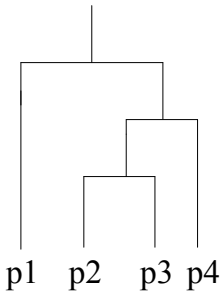


DENDROGRAMA: La similitud entre dos objetos viene dada por la "altura" del nodo común más cercano.

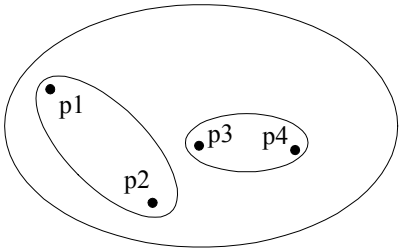
Clustering jerárquico



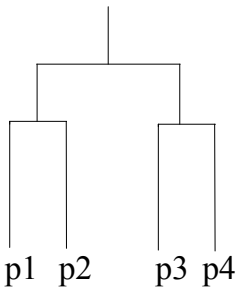
Tradicional



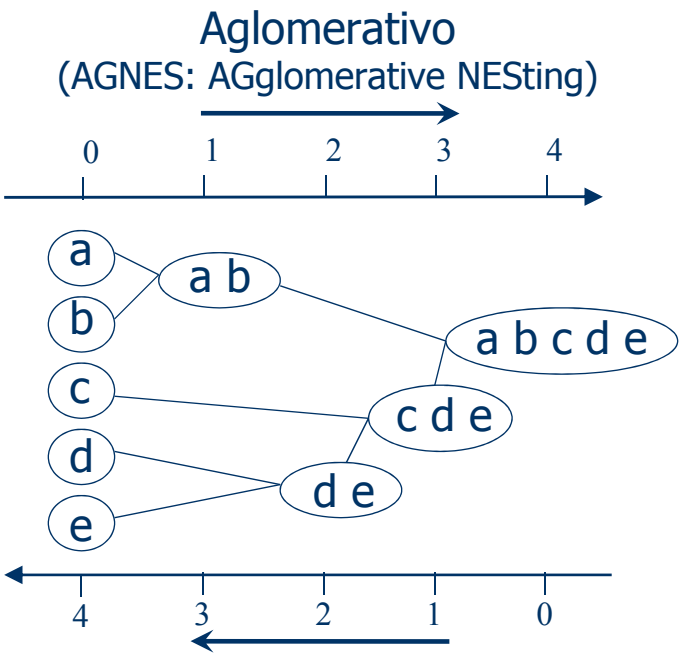
DENDROGRAMA



No tradicional



Clustering jerárquico

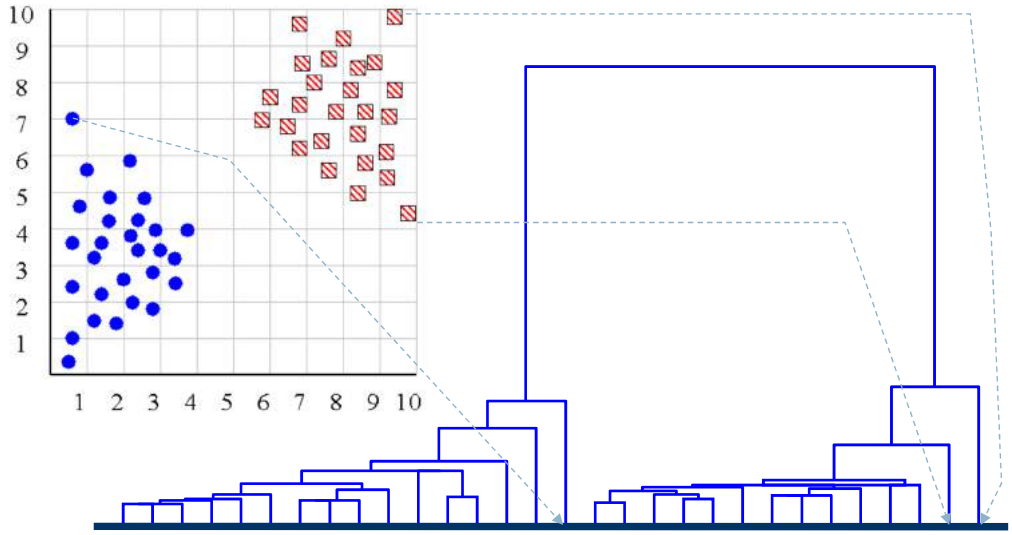


Clustering jerárquico

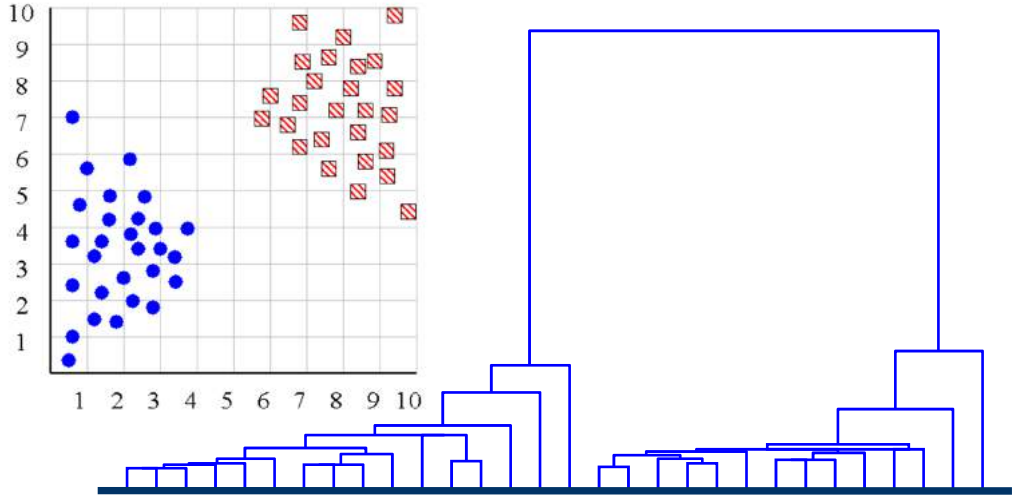
Dos tipos de técnicas de clustering jerárquico

- **Técnicas aglomerativas**
  - Comenzar con cada caso como cluster individual.
  - En cada paso, combinar el par de clusters más cercano hasta que sólo quede uno (o k).
- **Técnicas divisivas**
  - Comenzar con un único cluster que englobe todos los casos de nuestro conjunto de datos.
  - En cada paso, partir un cluster hasta que cada cluster contenga un único caso (o queden k clusters).

Clustering jerárquico



Clustering jerárquico



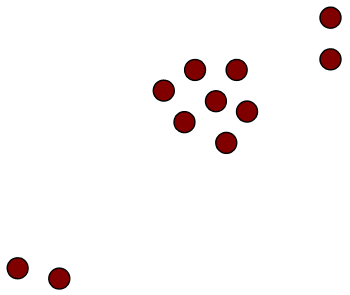
El **dendrograma** nos puede ayudar
a determinar el número adecuado de agrupamientos
(aunque normalmente no será tan fácil).

Clustering jerárquico

Ejemplo

Construir el correspondiente dendrograma.

¿Cuál es el número ideal de clusters?

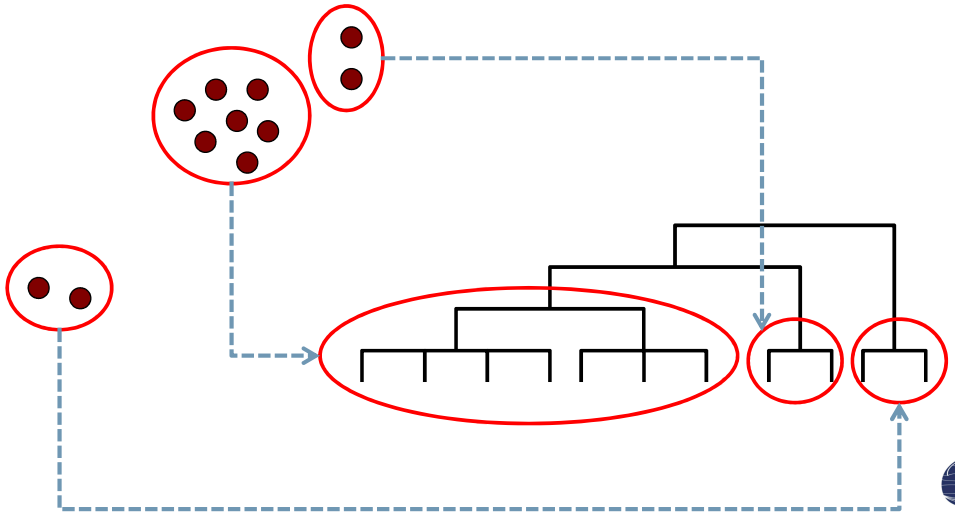


Clustering jerárquico

Ejemplo

Construir el correspondiente dendrograma.

¿Cuál es el número ideal de clusters?

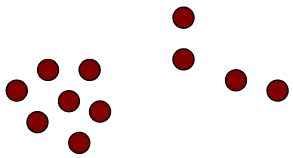


Clustering jerárquico

Ejemplo

Construir el correspondiente dendrograma.

¿Cuál es el número ideal de clusters?

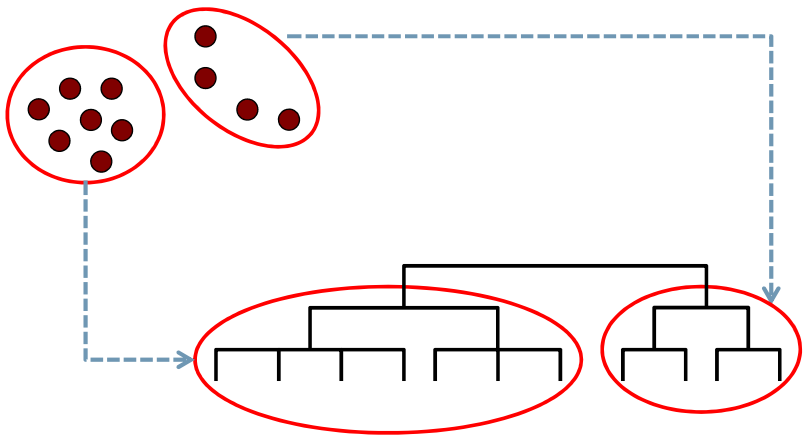


Clustering jerárquico

Ejemplo

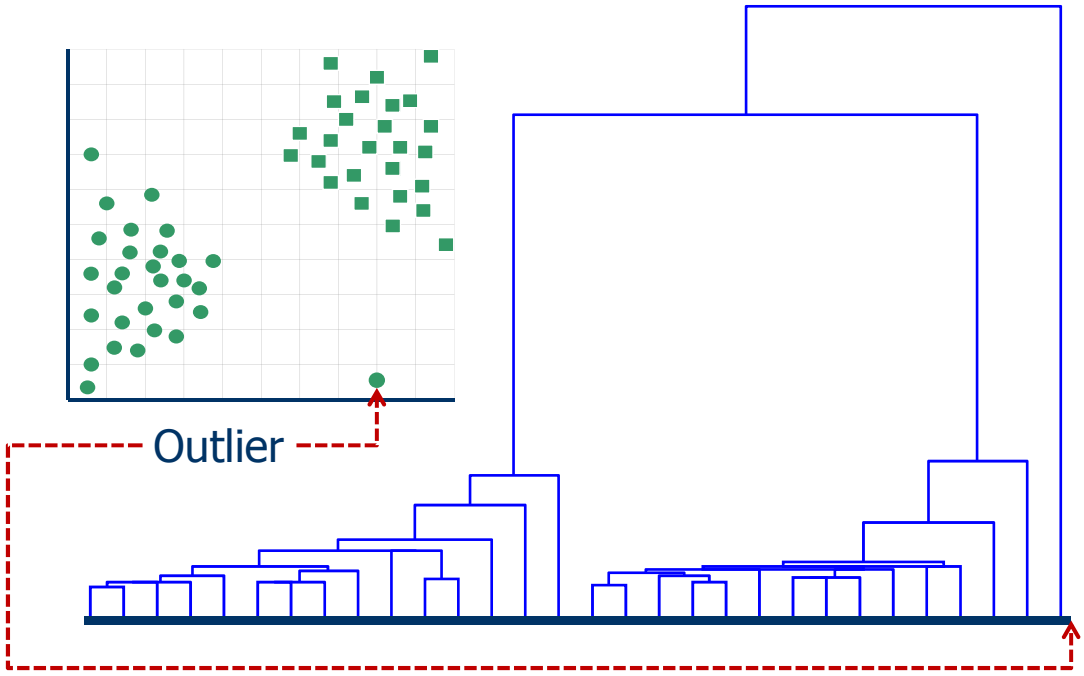
Construir el correspondiente dendrograma.

¿Cuál es el número ideal de clusters?



Clustering jerárquico

Detección de la presencia de outliers:



Clustering jerárquico

Algoritmo básico (aglomerativo)

Calcular la matriz de similitud/distancias
Inicialización: Cada caso, un cluster
Repetir
    Combinar los dos clusters más cercanos
    Actualizar la matriz de similitud/distancias
hasta que sólo quede un cluster

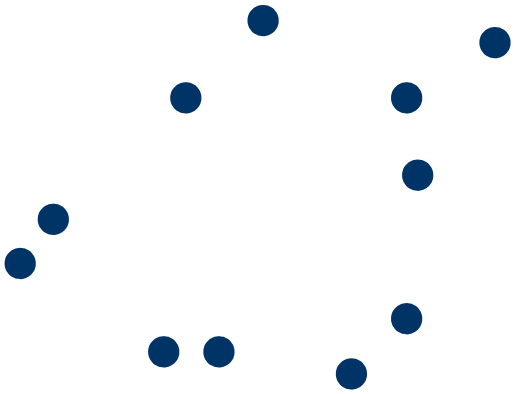
- Estrategia de control irrevocable (greedy):
  Cada vez que se unen dos clusters,
  no se reconsidera otra posible unión.

Clustering jerárquico

Inicialización:

Clusters de casos individuales y matriz de distancias

	p1	p2	p3	p4	p5
p1					
p2					
p3					
p4					
p5					

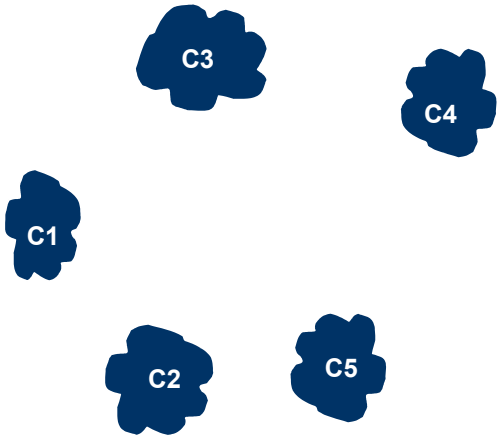


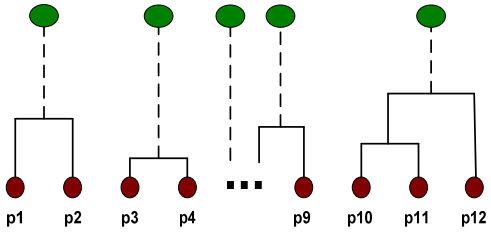
Clustering jerárquico

Tras varias iteraciones:

Varios clusters…

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

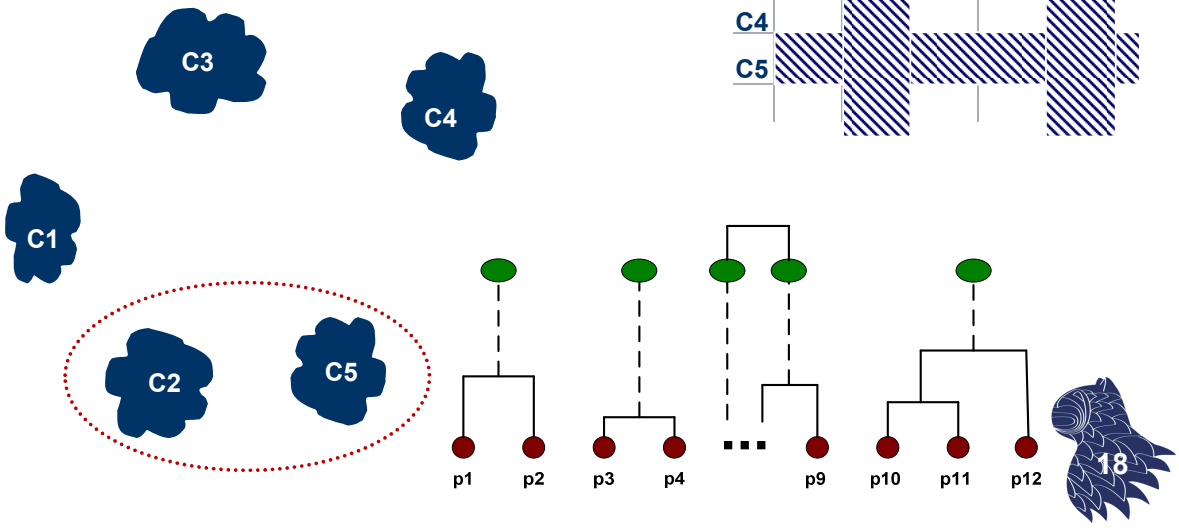




Clustering jerárquico

Combinamos los clusters 2 y 5
y actualizamos la matriz de distancias
¿cómo?

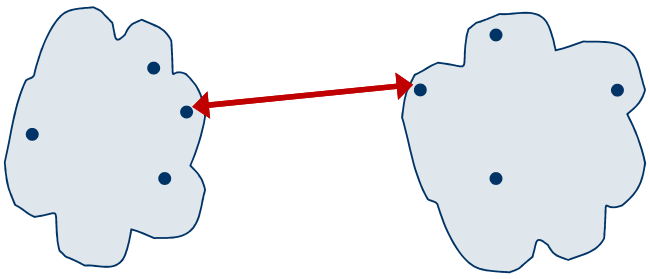
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					



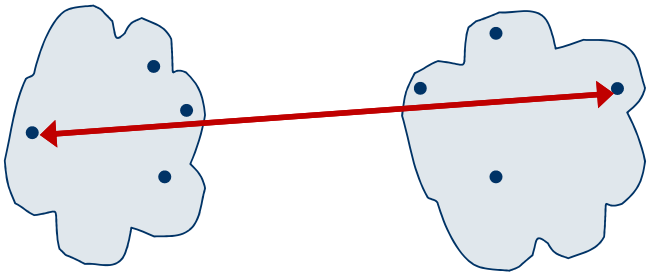
Clustering jerárquico

¿Cómo se mide la distancia entre clusters?

- MIN
  single-link



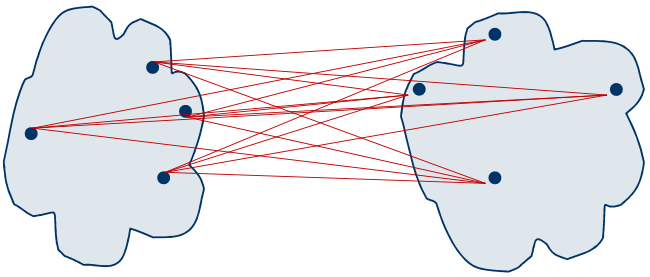
- MAX
  complete
  linkage
  (diameter)



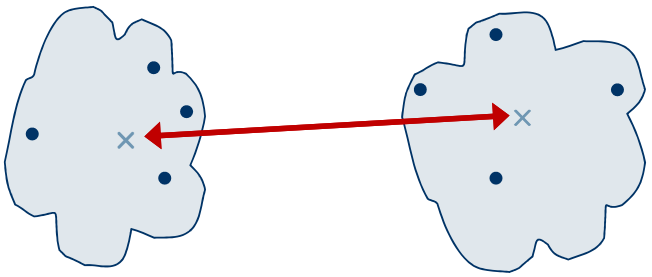
Clustering jerárquico

¿Cómo se mide la distancia entre clusters?

- Promedio

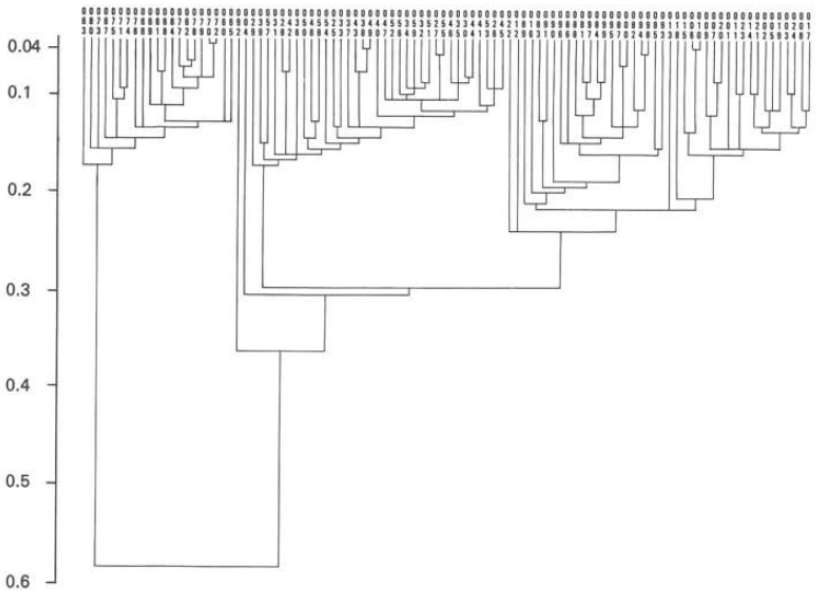


- Centroides
  p.ej. BIRCH



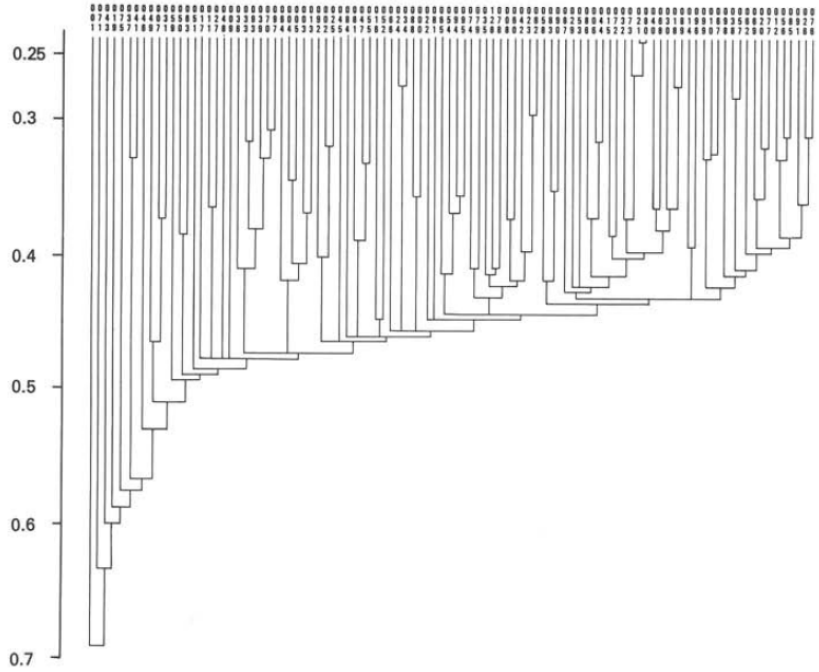
Clustering jerárquico

Datos sintéticos (4 clusters): Single-link



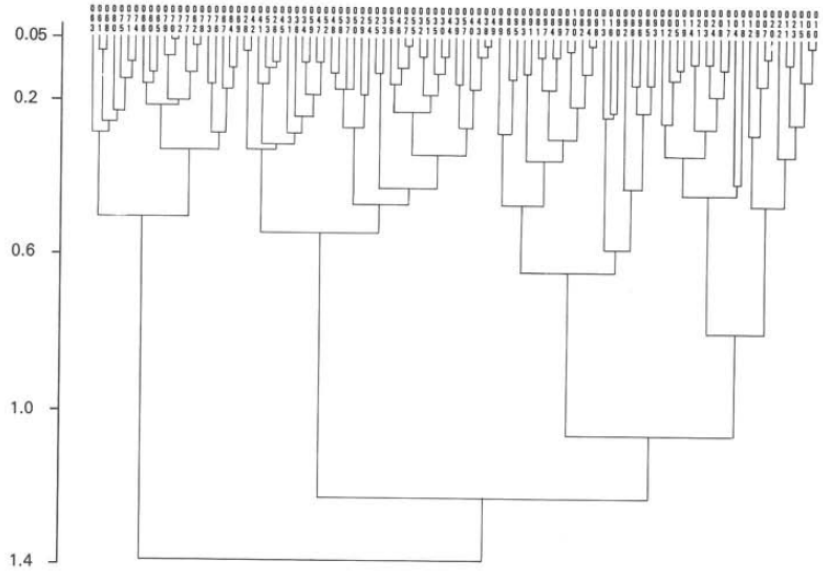
Clustering jerárquico

Datos sintéticos (aleatorios): Single-link



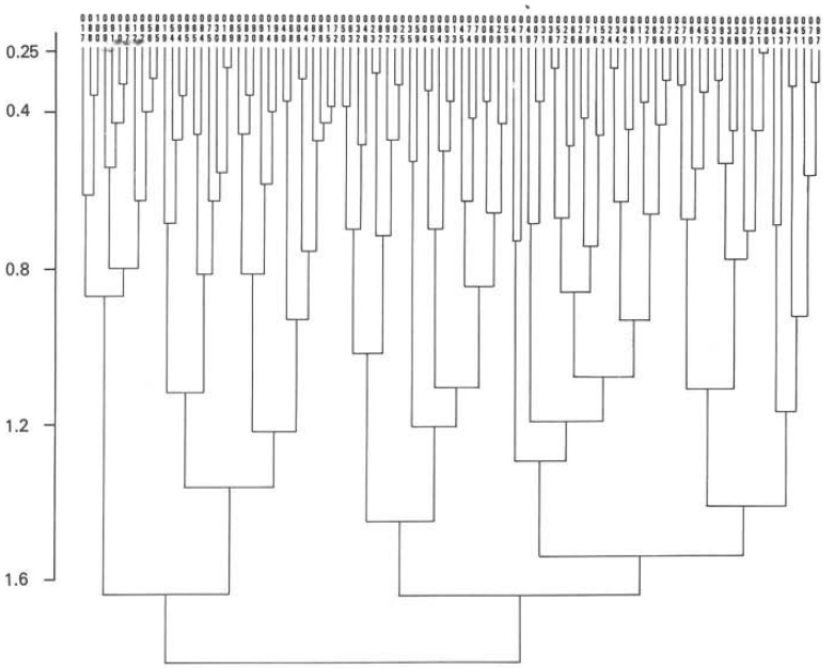
Clustering jerárquico

Datos sintéticos (4 clusters): Complete-link



Clustering jerárquico

Datos sintéticos (aleatorios): Complete-link



Clustering jerárquico

Ejercicio

Utilizar un algoritmo aglomerativo de clustering jerárquico para agrupar los datos descritos por la siguiente matriz de distancias:

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

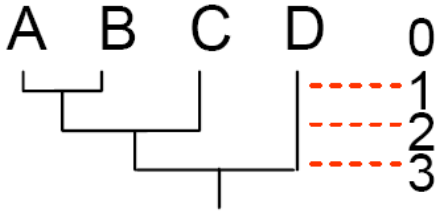
Variantes:

- **Single-link** (mínima distancia entre agrupamientos).
- **Complete-link** (máxima distancia entre agrupamientos).

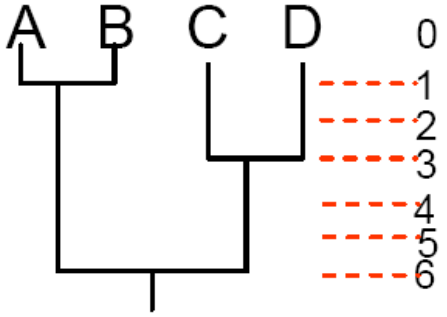
Clustering jerárquico

Ejercicio resuelto

- Single-link



- Complete-link



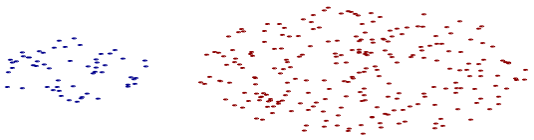
Clustering jerárquico

DEMO

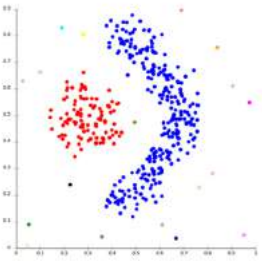
http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/AppletH.html

Clustering jerárquico

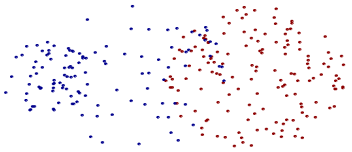
Single-link funciona bien
donde lo hace el k-means:



Single-link permite detectar
clusters no elípticos:

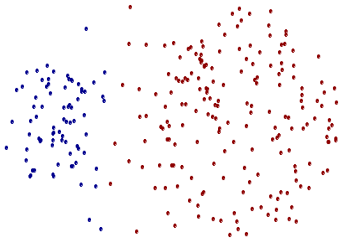


Single-link es demasiado
sensible a la presencia de ruido:

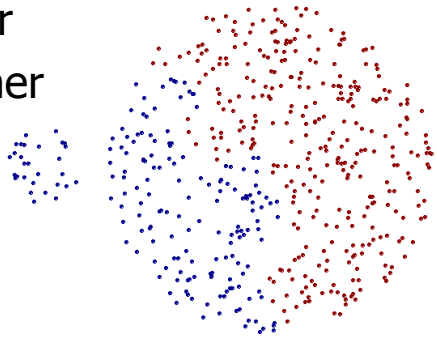


Clustering jerárquico

Complete-link es menos
sensible a la presencia de ruido:



Complete-link tiende a romper
los clusters grandes y a obtener
clusters globulares:



Clustering jerárquico

Principal inconveniente del clustering jerárquico

Baja escalabilidad $\geq O(n^2)$

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

Clustering jerárquico

Algoritmos representativos de clustering jerárquico

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

✓ SPSS: Two-Step Clustering (variante de BIRCH)

BIRCH

Se basa en una estructura de datos jerárquica llamada **CF Tree [Clustering Feature Tree]**

FASE 1:
Se construye un árbol CF inicial en memoria (compresión "multi-nivel" de los datos que intenta preservar la estructura de los agrupamientos del conjunto de datos original).

FASE 2:
Se utiliza un algoritmo de clustering arbitrario para agrupar los nodos hoja del árbol CF (p.ej. cualquier algoritmo jerárquico aglomerativo).

BIRCH

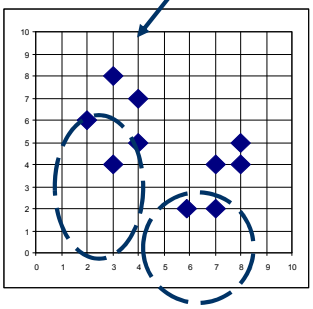
Clustering Feature (CF): CF = (N, LS, SS)

N: Number of data points

LS: linear sum of N points $\sum_{i=1}^{N} X_i$

SS: square sum of N points $\sum_{i=1}^{N} X_i^2$

CF = (5, (16,30),(54,190))

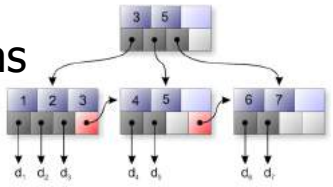


(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

- Con 2d+1 valores numéricos se resumen estadísticamente los datos de un cluster.

BIRCH

- El árbol CF es un **árbol balanceado** en el que los nodos internos almacenan las sumas de los CFs de sus descendientes.



- Un árbol CF tiene dos parámetros: el número máximo de hijos de un nodo y el diámetro máximo de los subclusters almacenados en las hojas del árbol.

- Para cada punto, se encuentra la hoja más cercana, se añade a su cluster y se actualiza su CF. Si el diámetro del subcluster es mayor que el diámetro máximo admitido, se divide la hoja (y puede que sus ascendientes).

BIRCH

Ventaja

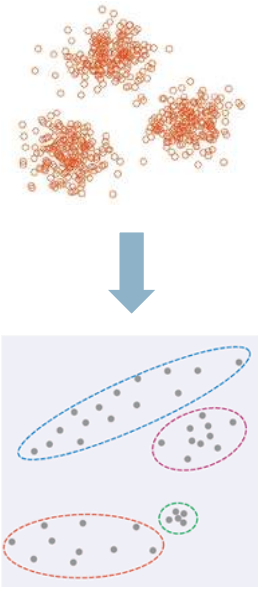
- Escalabilidad lineal **O(n)**: Se construye un buen conjunto de clusters con un simple recorrido del conjunto de datos y se puede mejorar su calidad con unos pocos recorridos adicionales.

Limitaciones

- Sólo es válido para datos de tipo numérico.
- Sensible al orden de presentación de los datos.
- Como fijamos el tamaño de los clusters asociados a las hojas del árbol CF, los clusters pueden no resultar naturales...

CURE [Clustering Using Representatives]

Extensión del algoritmo de las k medias
para clusters de formas y tamaños arbitrarios



Problema de K-Means & BFR

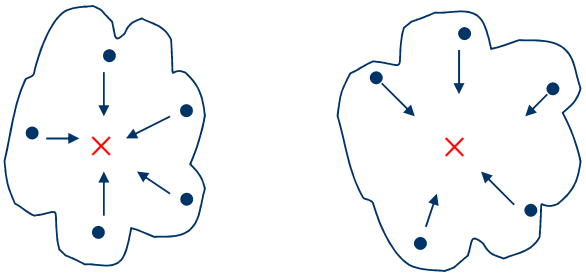
Asumen que los clusters están normalmente distribuidos en cada dimensión (ejes fijos).

Solución de CURE

Usando la distancia euclídea y un conjunto de puntos representativos para representar cada cluster, se permite que los clusters adopten cualquier forma.

CURE [Clustering Using Representatives]

Representantes



- Los representantes de cada cluster se construyen a partir de una muestra de puntos del cluster, tan dispersa como sea posible.
- A partir de la muestra, se crean los representantes moviendo los puntos de la muestra hacia el centroide del cluster (por ejemplo, un 20%).

CURE [Clustering Using Representatives]

Algoritmo en dos fases

Fase 1

- Se selecciona una muestra del conjunto de datos que quepa en memoria principal.
- Se agrupan esos puntos utilizando un algoritmo jerárquico (combinando los puntos/clusters más cercanos).
- Se selecciona un conjunto de representantes para cada uno de los clusters obtenidos

Fase 2

- Se recorre el conjunto completo de datos y cada punto se asigna al cluster más cercano (se busca el representante más cercano al punto y el punto se asigna al cluster del representante).

CURE [Clustering Using Representatives]

Pseudocódigo

For every cluster u (each input point),

u.mean = mean of the points in the cluster
u.rep = set of c representative points of the cluster (initially c = 1 since each cluster has one data point).
u.closest = the cluster closest to u.

All the input points are inserted into a k-d tree T

Treat each input point as separate cluster,

compute u.closest for each u and then insert each cluster into a heap Q (clusters are arranged in increasing order of distances between u and u.closest).

CURE [Clustering Using Representatives]

Pseudocódigo

while size(Q) > k

Remove the top element of Q (say u) and merge it with its closest cluster u.closest (say v) and compute the new representative points for the merged cluster w.

Remove u and v from T and Q.

For all the clusters x in Q,
update x.closest and relocate x

insert w into Q

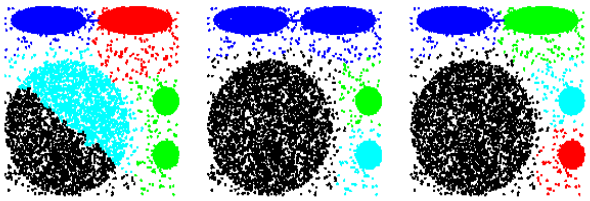
CURE [Clustering Using Representatives]

Complejidad del algoritmo

Tiempo	**O (n^2 log n)**
Espacio	**O (n)**

No se puede aplicar directamente sobre grandes conjuntos de datos, por lo que se suele muestrear el conjunto de datos.

CURE [Clustering Using Representatives]



a) BIRCH b) MST METHOD c) CURE

CURE

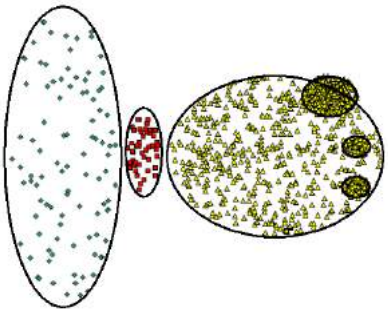
a) BIRCH

b) MST METHOD

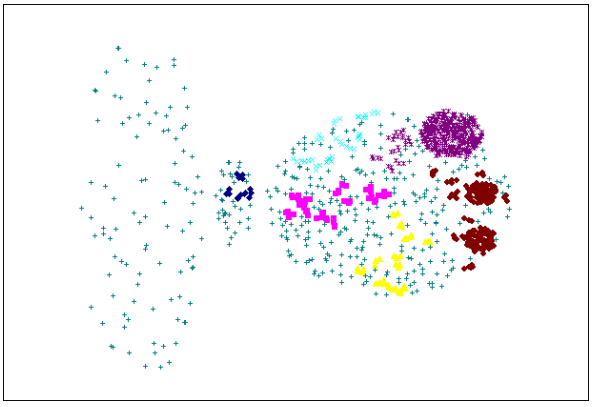
c) CURE

CURE [Clustering Using Representatives]

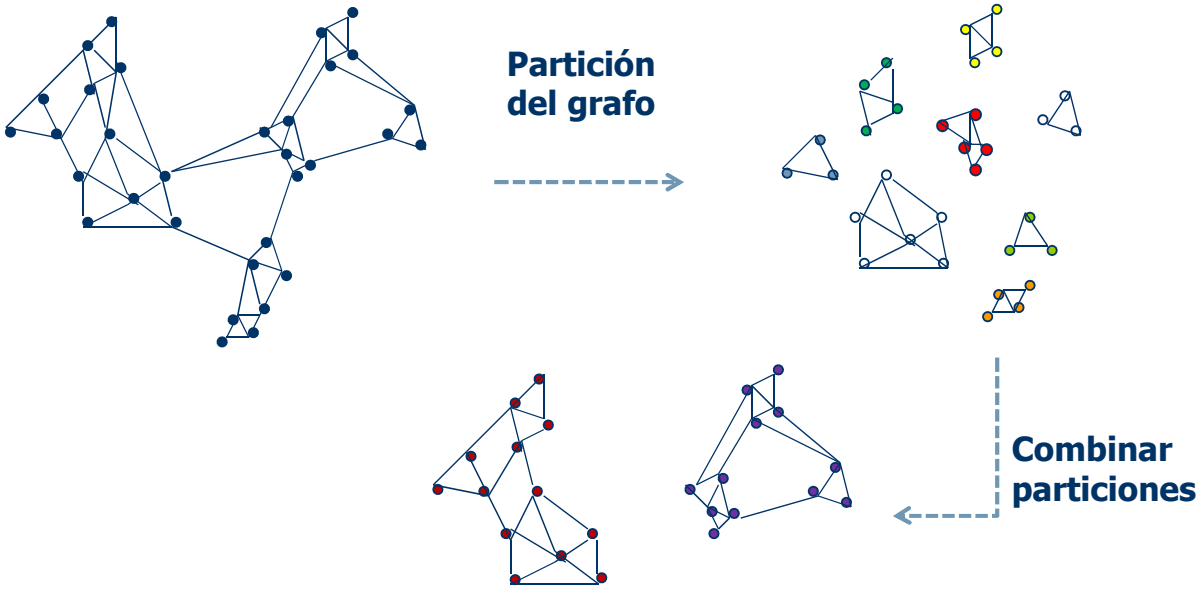
Agrupamientos con distintas densidades



CURE



CHAMELEON



CHAMELEON

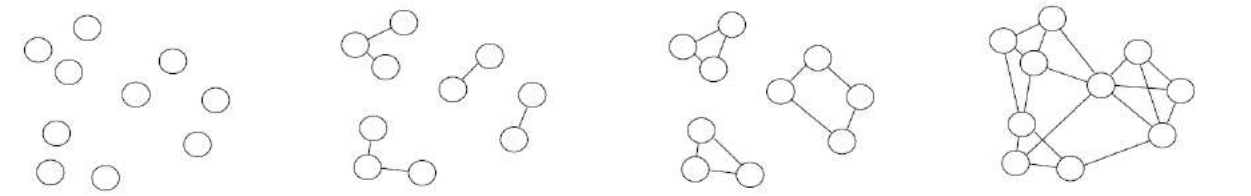
CHAMELEON

Similitud basada en un modelo "dinámico":
dos clusters se combinan sólo si su interconectividad
y su proximidad son altas con respecto a
la interconectividad de los clusters y
la cercanía de los puntos dentro de los clusters

- FASE 1: Algoritmo de particionamiento de grafos (número elevado de clusters relativamente pequeños)
- FASE 2: Algoritmo aglomerativo de clustering jerárquico (combinando los subclusters obtenidos por el algoritmo de particionamiento de grafos).

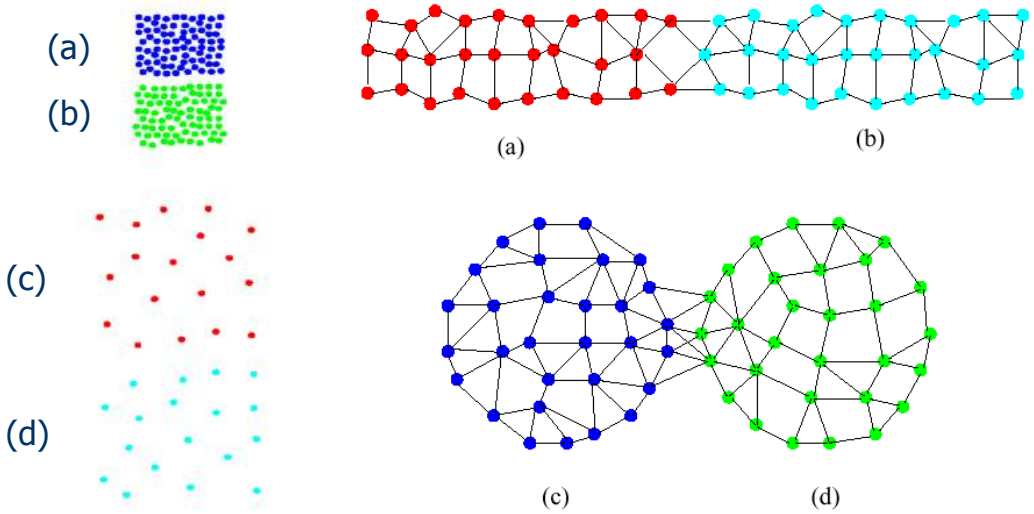
CHAMELEON

k-NN graphs [k-nearest graphs]



(a) Original Data in 2D (b) 1-nearest neighbor graph (c) 2-nearest neighbor graph (d) 3-nearest neighbor graph

CHAMELEON



- Single-link o CURE combinarían (a) y (b)
- Usando el promedio se mezclarían (c) y (d)

CHAMELEON

Interconectividad

- ***EC*$_{\{Ci,Cj\}}$**
  Interconectividad absoluta entre C_i y C_j
  (suma de los pesos de las aristas que conectan vértices de C_i con vértices de C_j).
- ***EC*$_{Ci}$**
  Interconectividad interna de un cluster C_i :
  (tamaño de su bisector de corte mínimo [min-cut], i.e. suma de los pesos de las aristas que dividen el grafo en dos partes aproximadamente iguales)
- ***RI***
  Interconectividad relativa $RI(C_i, C_j) = \frac{|EC_{\{C_i,C_j\}}|}{\frac{|EC_{C_i}|+|EC_{C_j}|}{2}}$

CHAMELEON

Proximidad

- ***RC*(C_i,C_j)**
  Proximidad relativa entre dos clusters (cercanía absoluta entre los dos clusters normalizada con respecto a la cercanía interna de los clusters)

$$RC(C_i, C_j) = \frac{\overline{S}_{EC_{\{C_i,C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\overline{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\overline{S}_{EC_{C_j}}}$$

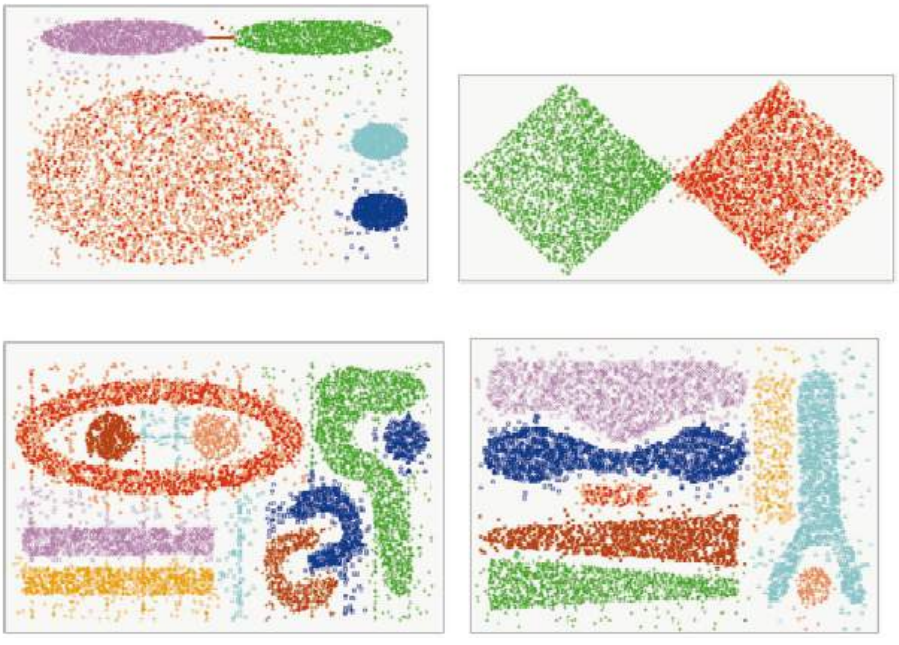
- $\overline{S}_{EC_{C_i}}$ es la media de los pesos de las aristas que pertenecen al bisector de corte mínimo del cluster.
- $\overline{S}_{EC_{\{C_i,C_j\}}}$ es la media de los pesos de las aristas que conectan vértices de un cluster con vértices del otro.

CHAMELEON

Algoritmo jerárquico aglomerativo

- Se combinan sólo aquellos pares de clusters cuyos valores de interconectividad RI y cercanía RC estén por encima de valores fijados por el usuario.
- Se mezcla el par de clusters que maximice una función que combina RI y RC.

CHAMELEON

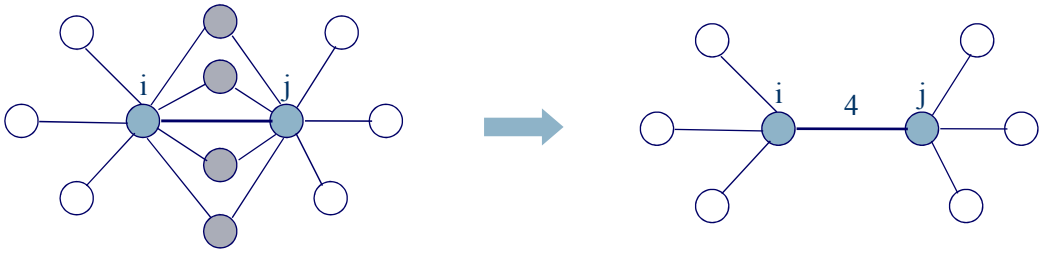


CHAMELEON

ROCK [RObust Clustering using linKs]

Número de vecinos compartidos

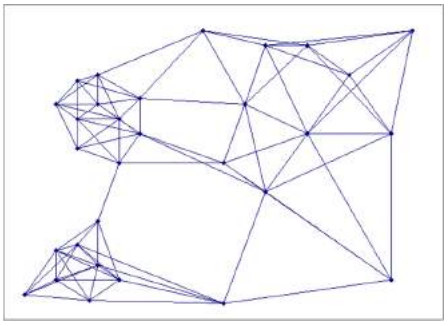
Una medida de similitud más:



ROCK [RObust Clustering using linKs]

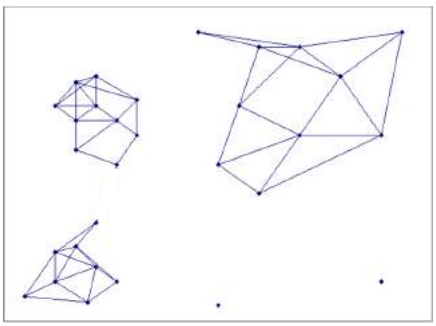
SNN Graph

Grafo de vecinos compartidos



Grafo ponderado

(similitudes entre puntos vecinos)



Grafo SNN

(peso de las aristas = número de vecinos compartidos)

ROCK [RObust Clustering using linKs]

- Algoritmo de clustering jerárquico para conjuntos de datos con atributos booleanos y categóricos.
- Un par de puntos se consideran vecinos si su similitud está por encima de un umbral preestablecido.

Algoritmo

1. Obtener una muestra del conjunto de datos.
2. Calcular el valor del enlace entre cada par de puntos (transformar similitudes originales, calculadas usando el coeficiente de Jaccard, en similitudes que reflejen el número de vecinos compartidos).
3. Usar un algoritmo aglomerativo usando el número de vecinos compartidos como medida de similitud.

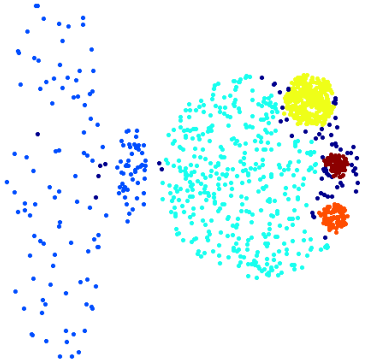
Jarvis-Patrick algorithm

- Se encuentran los k vecinos más cercanos de cada punto (esto es, se eliminan todos menos los k enlaces más fuertes de cada punto).
- Un par de puntos está en el mismo cluster si comparten más de T vecinos y cada punto están en la lista de k vecinos del otro punto.

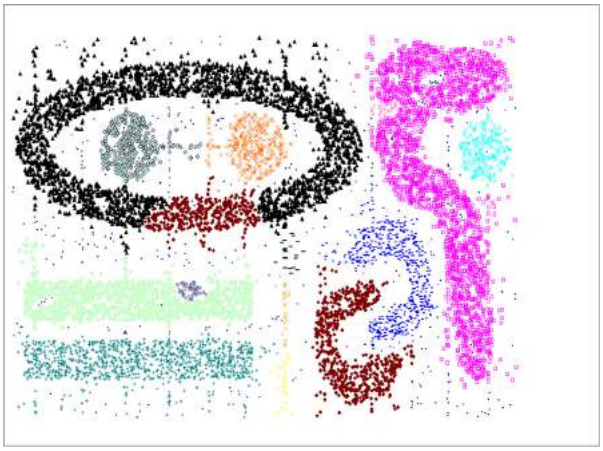
Algoritmo frágil

6 vecinos compartidos de 20

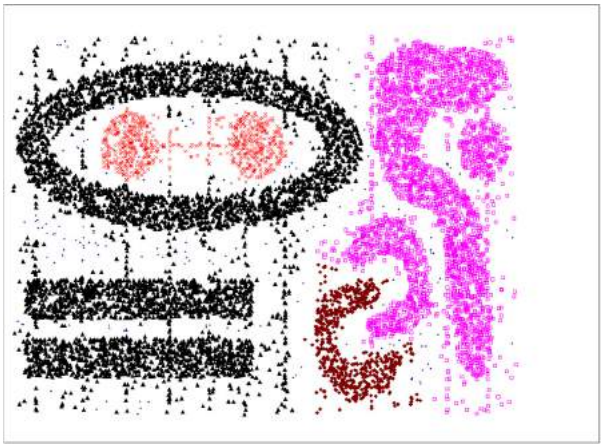
OK!



Jarvis-Patrick algorithm



Umbral T más bajo
que no mezcla clusters



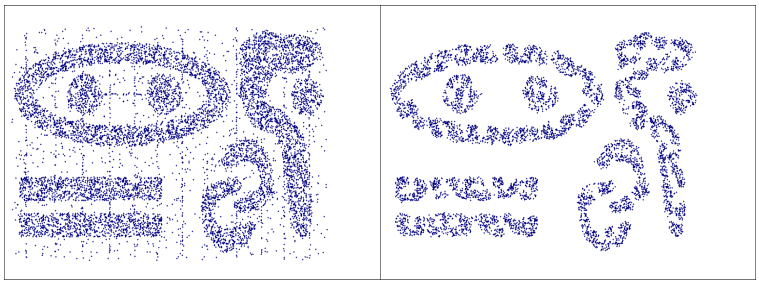
Umbral T-1

SNN Clustering

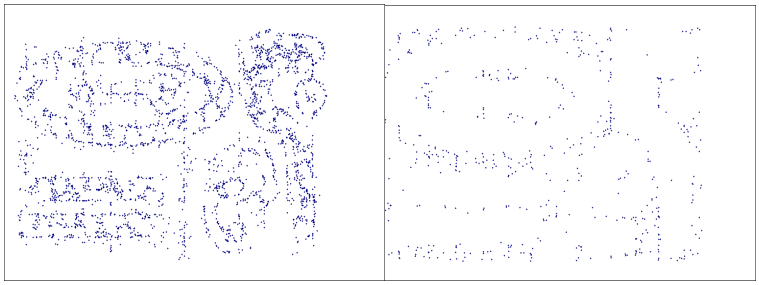
- Se calcula la matriz de similitud y nos quedamos sólo con los k vecinos más similares, a partir de los cuales construimos el grafo de vecinos compartidos SNN.
- Calculamos la **densidad SNN** de cada punto (usando un parámetro Eps dado por el usuario, el número de puntos que tienen una similitud igual o mayor que Eps para cada punto).

SNN Clustering

Densidad SNN



Puntos originales Densidad alta



Densidad media Densidad baja

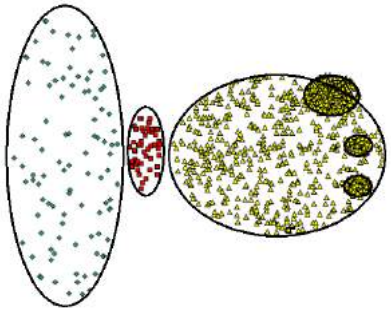
SNN Clustering

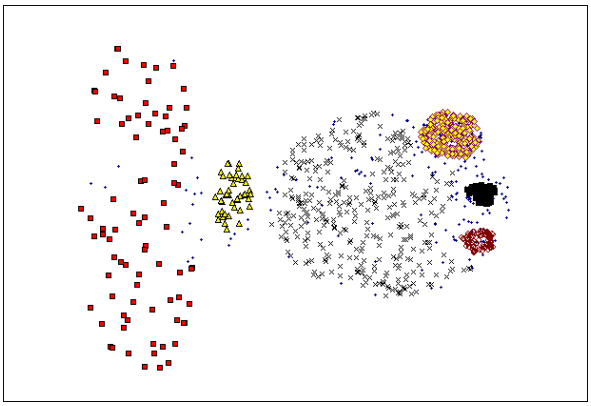
- Encontramos los puntos "core" (todos los puntos con densidad SNN mayor que MinPts, otro parámetro dado por el usuario).
- Agrupamos los puntos "core" (dos puntos a distancia Eps o menor se colocan en el mismo cluster).
- Descartamos todos los puntos no-"core" que estén a distancia mayor que Eps de un punto "core" (los consideramos ruido).
- Asignamos los puntos restantes al cluster del "core" más cercano.

NOTA: Son los mismos pasos del algoritmo DBSCAN…

SNN Clustering

Funciona con clusters de distinta densidad



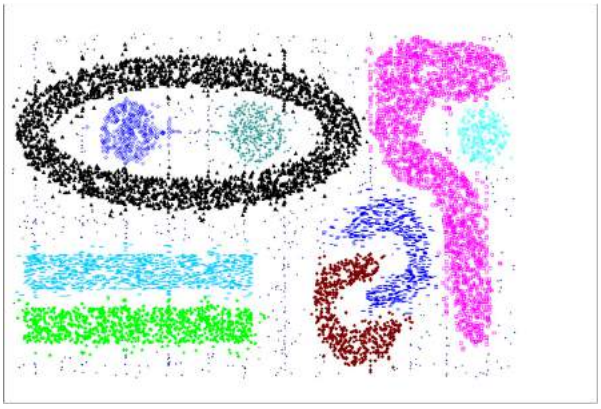


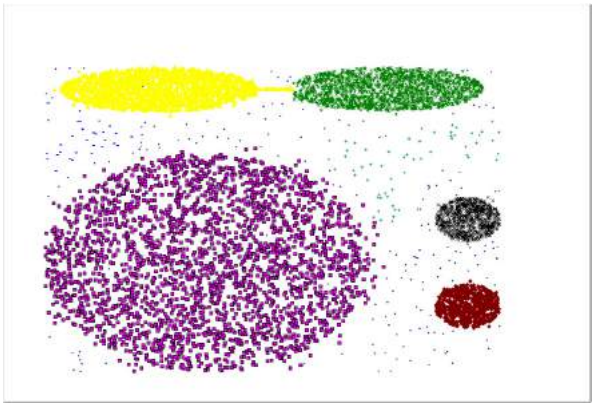
Puntos originales

SNN Clustering

SNN Clustering

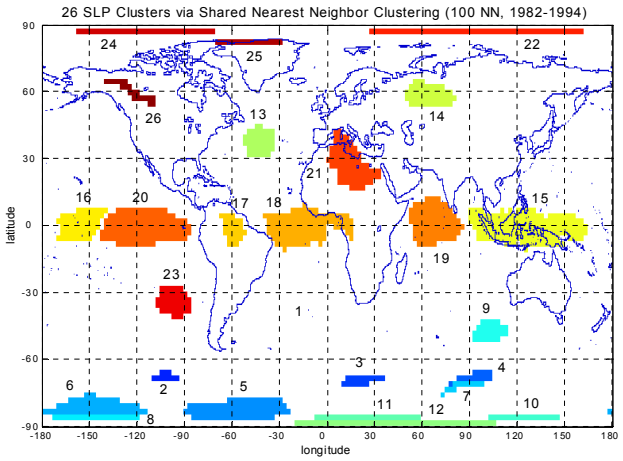
Detecta clusters de formas complejas

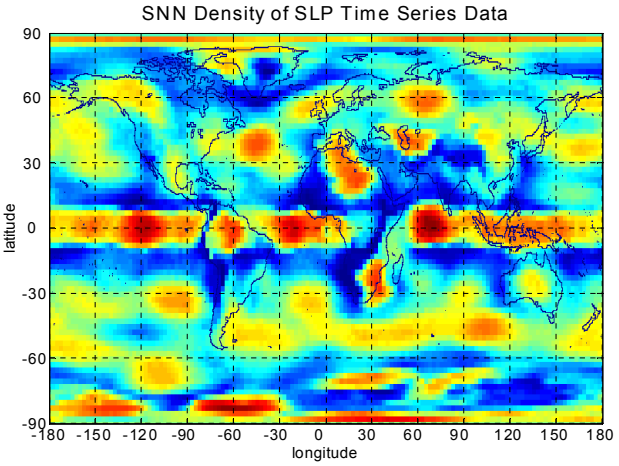




SNN Clustering

Se puede aplicar a datos espacio-temporales





SNN Clustering

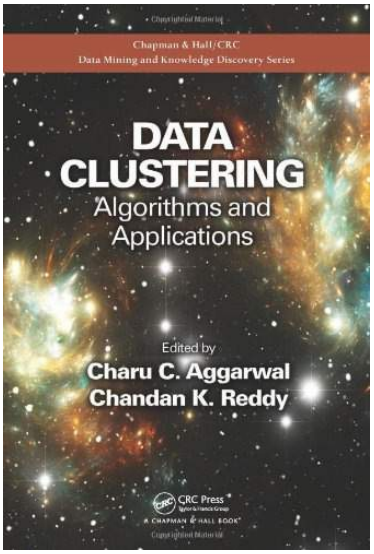
Complejidad

O (n * tiempo necesario para encontrar los vecinos a distancia Eps)

- En el peor caso, **O (n^2)**
- Si no hay demasiadas dimensiones, se pueden emplear estructuras de datos para encontrar los vecinos más cercanos, p.ej. R*trees, k-d trees…

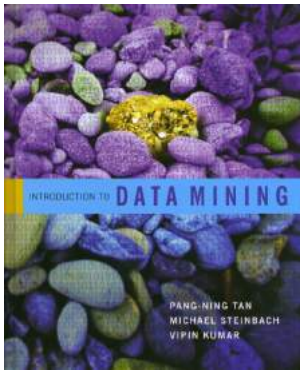
Bibliografía

- Charu C. Aggarwal & Chandan K. Reddy (editors): **Data Clustering: Algorithms and Applications**. Chapman & Hall / CRC Press, 2014. ISBN 1466558210.

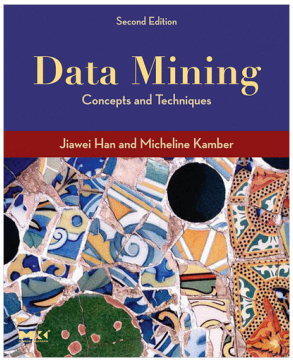


Bibliografía - Libros de texto

- Pang-Ning Tan, Michael Steinbach & Vipin Kumar: **Introduction to Data Mining** Addison-Wesley, 2006. ISBN 0321321367 [capítulos 8&9]



- Jiawei Han & Micheline Kamber: **Data Mining: Concepts and Techniques** Morgan Kaufmann, 2006. ISBN 1558609016 [capítulo 7]



Bibliografía - Algoritmos

BIRCH

- Tian Zhang, Raghu Ramakrishnan & Miron Livny: **BIRCH: An efficient data clustering method for very large databases**. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (SIGMOD'96) & ACM SIGMOD Record 25(2):103-114, 1996. DOI 10.1145/235968.233324

CURE

- Sudipto Guha, Rajeev Rastogi & Kyuseok Shim: **CURE: An efficient clustering algorithm for large databases**. Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data (SIGMOD '98) & ACM SIGMOD Record 27(2):73-84, 1998 DOI 10.1145/276304.276312
- Sudipto Guha, Rajeev Rastogi & Kyuseok Shim: **CURE: An efficient clustering algorithm for large databases**. Information Systems 26(1):35-58, March 2001. DOI 10.1016/S0306-4379(01)00008-4

Bibliografía - Algoritmos

CHAMELEON

- George Karypis, Eui-Hong (Sam) Han & Vipin Kumar: **Chameleon: Hierarchical Clustering Using Dynamic Modeling**. IEEE Computer 32(8):68-75, August 1999. DOI 10.1109/2.781637

ROCK

- Sudipto Guha, Rajeev Rastogi & Kyuseok Shim: **ROCK: A robust clustering algorithm for categorical attributes**. Proceedings of the 15th International Conference on Data Engineering (ICDE'99), Sydney, Australia, March 1999.
- Sudipto Guha, Rajeev Rastogi & Kyuseok Shim: **ROCK: A robust clustering algorithm for categorical attributes**. Information Systems 25(5):345-366, July 2000. DOI 10.1016/S0306-4379(00)00022-3

Apéndice: Notación O

El impacto de la eficiencia de un algoritmo…

n	10	100	1000	10000	100000
O(n)	10ms	0.1s	1s	10s	100s
$O(n \cdot \log_2 n)$	33ms	0.7s	10s	2 min	28 min
$O(n^2)$	100ms	10s	17 min	28 horas	115 días
$O(n^3)$	1s	17min	12 días	31 años	32 milenios