UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Grado Inteligencia Artificial

Advanced Clustering – Parte II

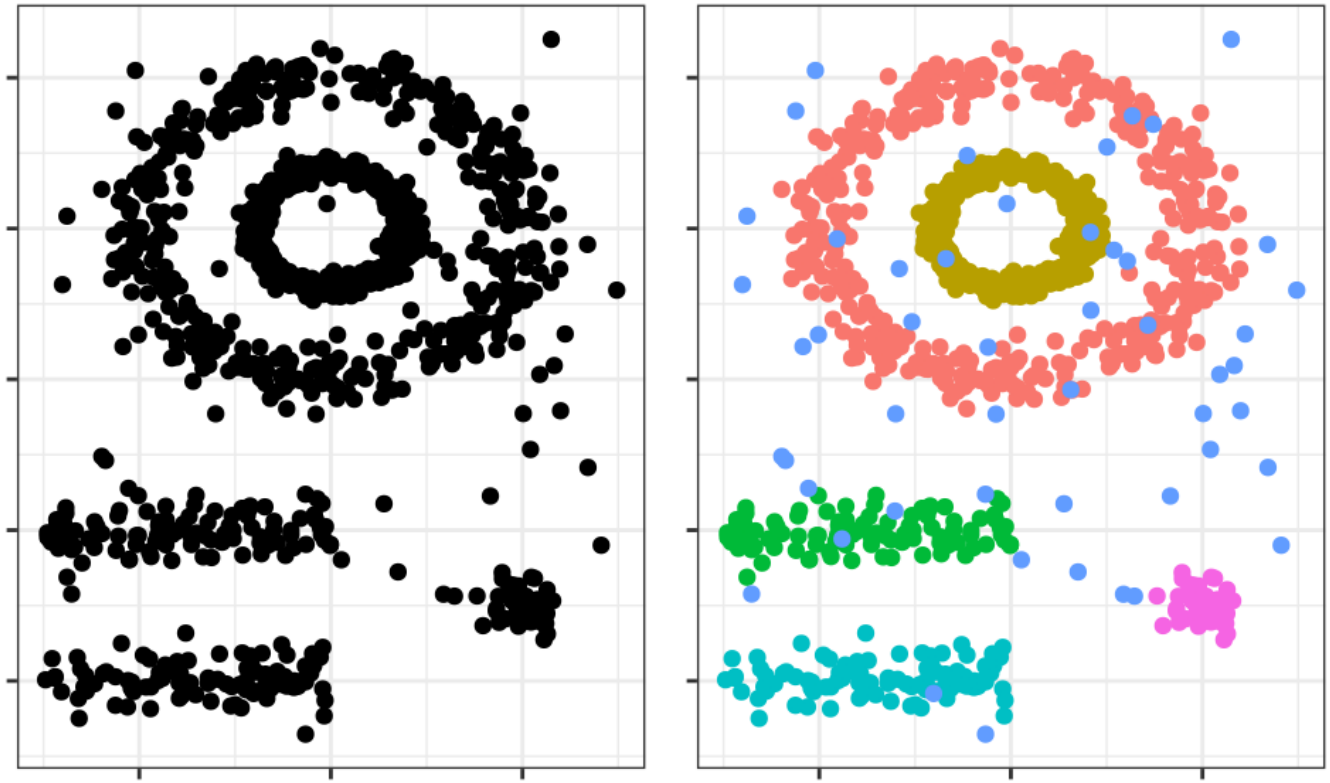
Prof. Dante Conti

Prof. Sergi Ramírez

Credits: Prof. K. Gibert (CURE, Section)

DBSCAN (Density-based spatial clustering of applications with noise) – Idea Introductoria

Forma de identificar clusters siguiendo el modo intuitivo como lo hace el cerebro humano, identificando regiones con alta densidad de observaciones separadas por regiones de baja densidad.



DBSCAN – Idea Introductoria

Los métodos de clustering como k-means, hierarchical, k-medoids son eficaces encontrando agrupaciones con forma esférica o convexa que no contengan un exceso de outliers o ruido, pero fallan al tratar de identificar formas arbitrarias.

DBSCAN evita este problema siguiendo la idea de que, para que una observación forme parte de un cluster, tiene que haber un mínimo de observaciones vecinas dentro de un radio de proximidad y de que los clusters están separados por regiones vacías o con pocas observaciones.

DBSCAN – Definiciones y Algoritmo

El algoritmo DBSCAN necesita dos parámetros:

Epsilon (ϵ) que define la región vecina a una observación, también llamada ϵ -neighborhood.

-Minimum points (V) (minPts): número mínimo de observaciones dentro de la región epsilon.

Empleando estos dos parámetros, cada observación del set de datos se puede clasificar en una de las siguientes tres categorías:

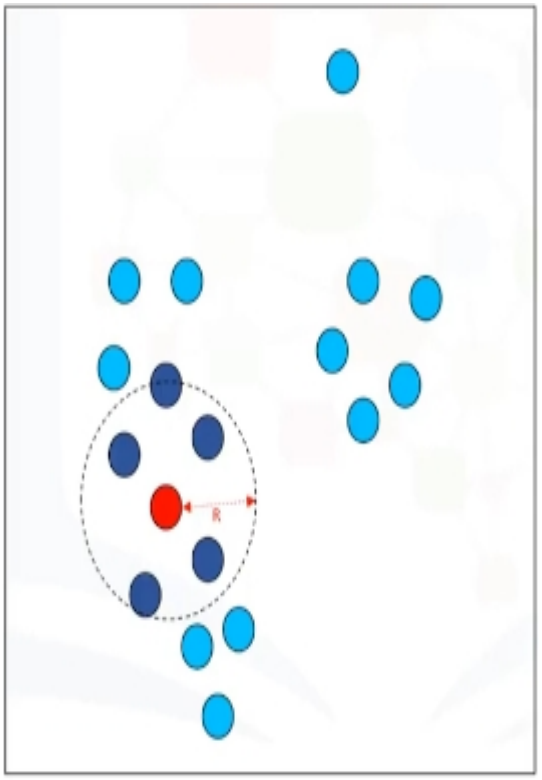
Core point: observación que tiene en su ϵ-neighborhood un número de observaciones vecinas igual o mayor a minPts.

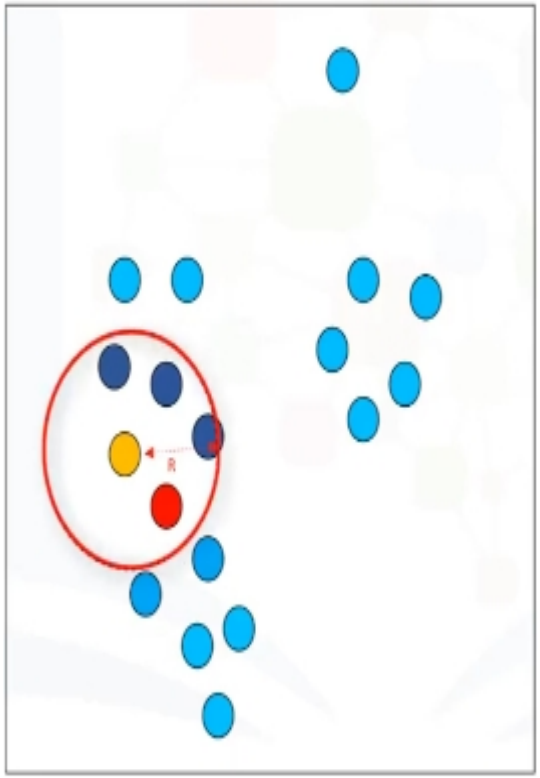
Border point: observación que no satisface el mínimo de observaciones vecinas para ser core point pero que pertenece al ϵ-neighborhood de otra observación que sí es core point.

Noise u outlier: observación que no es core point ni border point.

DBSCAN – Definiciones y Algoritmo

Core point, border point and outlier point. Aqui, epsilon=2, V=6





DBSCAN – Definiciones y Algoritmo

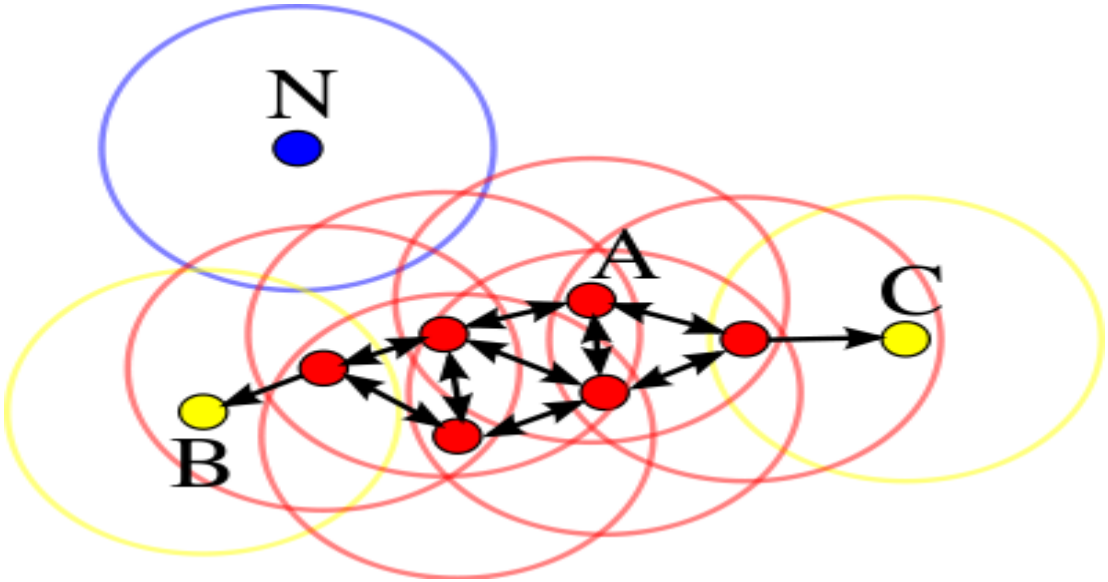
Directamente alcanzable (direct density reachable): una observación A es directamente alcanzable desde otra observación B si A forma parte del ϵ -neighborhood de B y B es un core point. Por definición, las observaciones solo pueden ser directamente alcanzables desde un core point.

Alcanzable (density reachable): una observación A es alcanzable desde otra observación B si existe una secuencia de core points que van desde B a A.

Densamente conectadas (density conected): dos observaciones A y B están densamente conectadas si existe una observación core point C tal que A y B son alcanzables desde C.

DBSCAN – Definiciones y Algoritmo

La siguiente imagen muestra las conexiones existentes entre un conjunto de observaciones si se emplea minPts=4
. La observación A y el resto de observaciones marcadas en rojo son core points, ya que todas ellas contienen al menos 4 observaciones vecinas (incluyéndose a ellas mismas) en su ϵ-neighborhood. Como todas son alcanzables entre ellas, forman un cluster. Las observaciones B y C no son core points pero son alcanzables desde A a través de otros core points, por lo tanto, pertenecen al mismo cluster que A
. La observación N no es ni un core point ni es directamente alcanzable, por lo que se considera como ruido.



Algoritmo

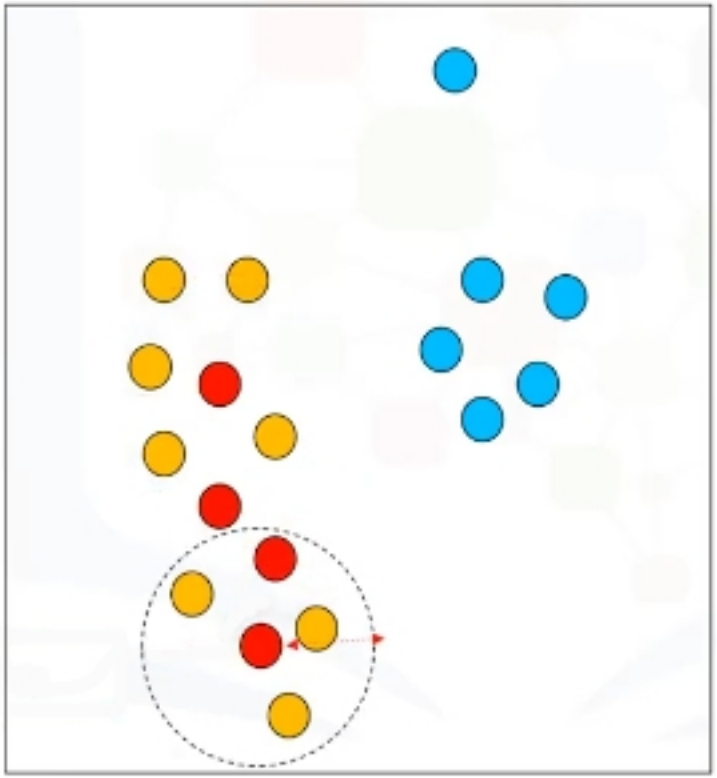
Para cada observación xi calcular la distancia entre ella y el resto de observaciones. Si en su ϵ-neighborhood hay un número de observaciones ≥minPts marcar la observación como core point, de lo contrario marcarla como visitada.

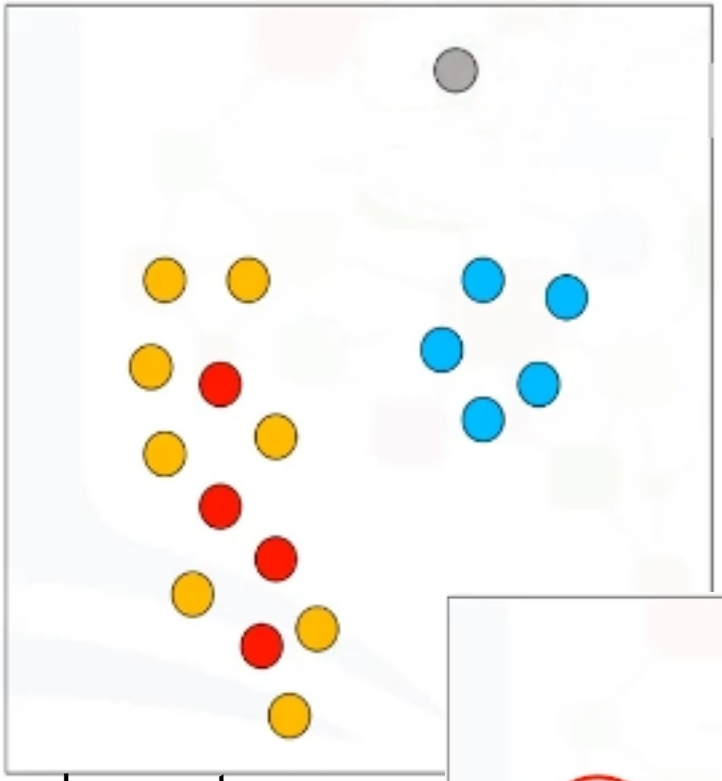
Para cada observación xi marcada como core point, si todavía no ha sido asignada a ningún cluster, crear uno nuevo y asignarla a él. Encontrar recursivamente todas las observaciones densamente conectadas a ella y asignarlas al mismo cluster.

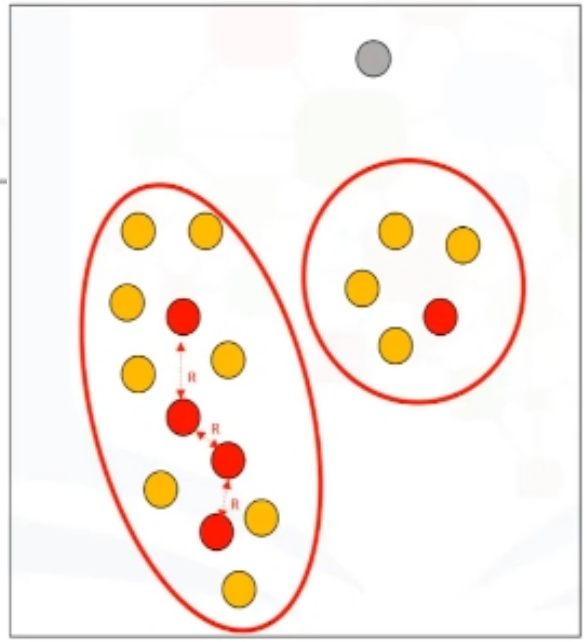
Iterar el mismo proceso para todas las observaciones que no hayan sido visitadas.

Aquellas observaciones que tras haber sido visitadas no pertenecen a ningún cluster se marcan como outliers.

DBSCAN – Algoritmo







El paso final consiste en conectar todos los puntos centrales que son vecinos y ubicarlos en el mismo clúster. Como resultado, todo clúster cumple dos propiedades: todos los puntos que forman parte de un mismo clúster están densamente conectados entre ellos y, si una observación A es densamente alcanzable desde cualquier otra observación de un clúster, entonces A también pertenece al clúster

DBSCAN – Fitting

Selección de parámetros

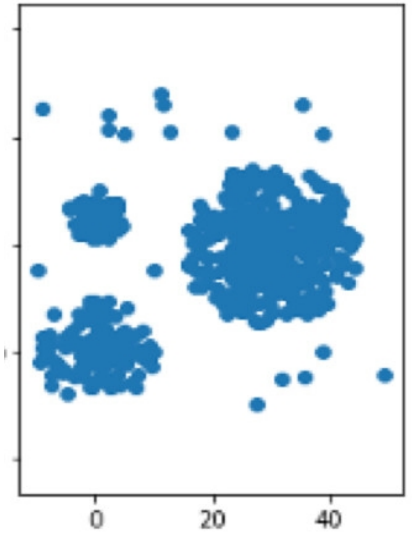
Como ocurre en muchas otras técnicas estadísticas, en DBSCAN no existe una forma única y exacta de encontrar el valor adecuado de epsilon (ϵ) y minPts.

minPts
Cuanto mayor sea el tamaño del set de datos, mayor debe ser el valor mínimo de observaciones vecinas. Se recomienda no bajar nunca de 3. Si los datos contienen niveles altos de ruido, aumentar minPts favorecerá la creación de clusters significativos menos influenciados por outliers.

epsilon: una buena forma de escoger el valor de ϵ es estudiar las distancias promedio entre las k=minPts observaciones más próximas. Al representar estas distancias en función de ϵ, el punto de inflexión de la curva suele ser un valor óptimo. Si el valor de ϵ escogido es muy pequeño, una proporción alta de las observaciones no se asignarán a ningún cluster, por el contrario, si el valor es demasiado grande, la mayoría de observaciones se agruparán en un único cluster.

DBSCAN – (Important Tips)

1) The basic idea behind the density-based clustering approach is derived from an intuitive human clustering method. For example, looking at the figure, one can easily identify three clusters along with various noise points, due to differences in point density.



2)Typically, the people who work most with DBSCAN take the minimum point twice of the dimensionality of the data.

3) Once the minimum point has been chosen, we can continue with epsilon. One method to obtain this value is to calculate the nearest neighbor distances in a matrix of points. The idea is to calculate the average of the distances from each point to its nearest neighbors. The value of k will be specified by the user and corresponds to the minimum points. Next, these k distances are plotted in ascending order. The goal is to determine the knee, which corresponds to the optimal epsilon parameter.

DBSCAN – (Important Tips)

Ventajas de DBSCAN

No requiere que el usuario especifique el número de clusters.

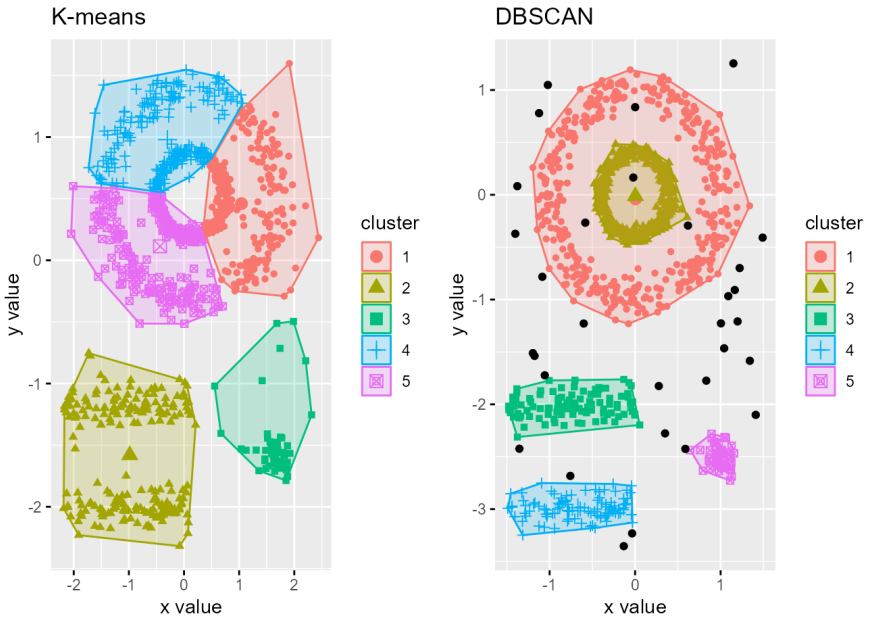
Es independiente de la forma que tengan los clusters, no tienen por qué ser circulares.

Puede identificar outliers, por lo que los clusters generados no se influenciados por ellos.

Desventajas de DBSCAN

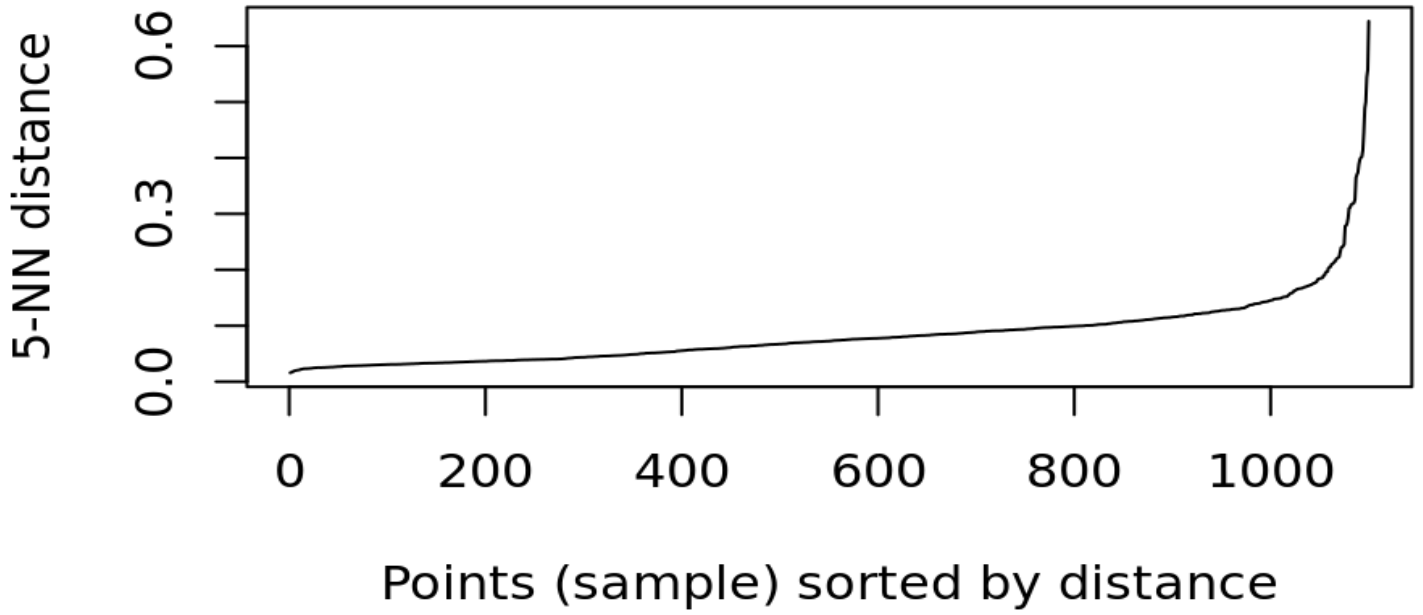
No es un método totalmente determinístico: los border points que son alcanzables desde más de un cluster pueden asignarse a uno u otro dependiendo del orden en el que se procesen los datos.

No genera buenos resultados cuando la densidad de los grupos es muy distinta, ya que no es posible encontrar los parámetros ϵ y minPts que sirvan para todos a la vez.



DBSCAN – (Important Tips)

```
# Selección del valor óptimo de epsilon. Como valor de minPts se emplea 5.
dbscan::kNNdistplot(datos, k = 5)
```



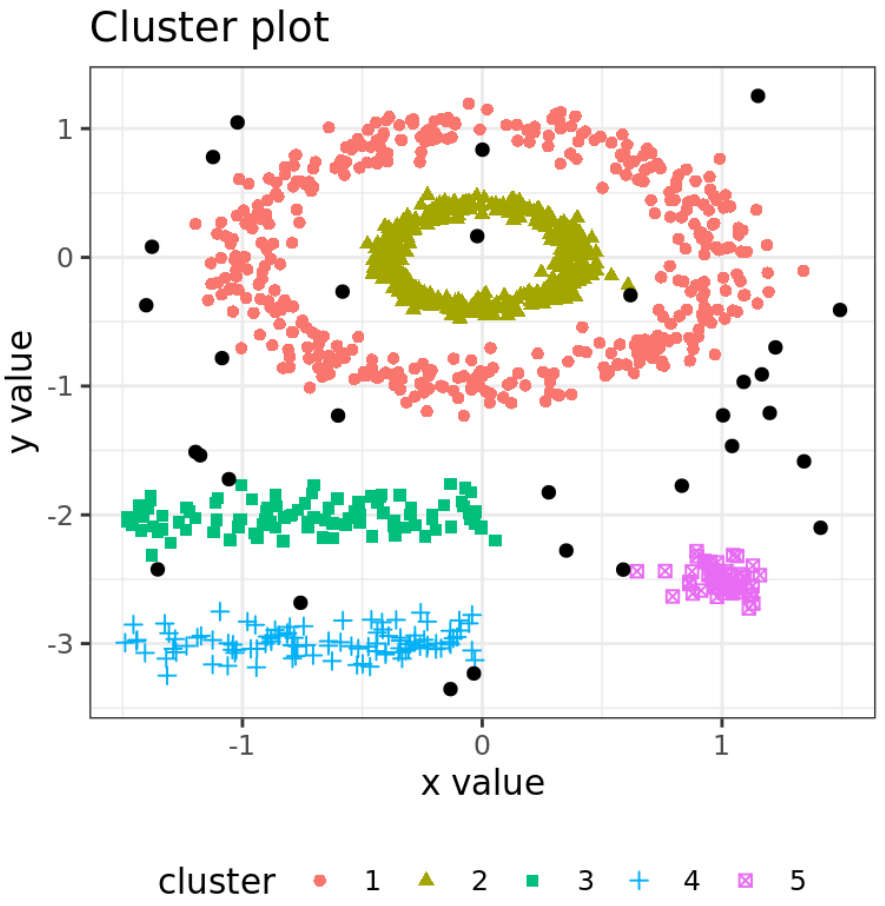
La curva tiene el punto de inflexión en torno a 0.15, por lo que se escoge este valor como epsilon para DBSCAN.

DBSCAN – (Important Tips)

```
set.seed(321)

# DBSCAN con epsilon = 0.15 y minPts = 5
dbscan_cluster <- fpc::dbscan(data = datos, eps = 0.15, MinPts = 5)

# Resultados de la asignación
head(dbscan_cluster$cluster)
```



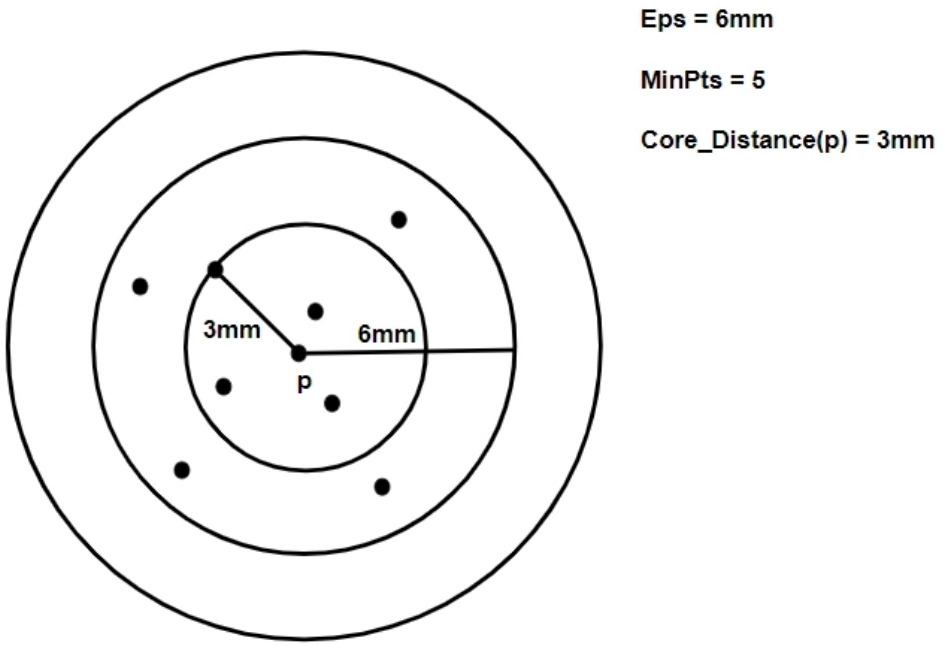
OPTICS (Ordering Points To Identify Cluster Structure) – Idea Introductoria

DBSCAN algorithm assumes the density of the clusters as constant, whereas the OPTICS algorithm allows a varying density of the clusters.

OPTICS adds two more terms to the concept of the DBSCAN algorithm, i.e.:

Core Distance
Reachability Distance

Core distance → It is the minimum value of radius required to classify a given point as a core point



OPTICS (Ordering Points To Identify Cluster Structure) – Idea Introductoria

Reachability Distance

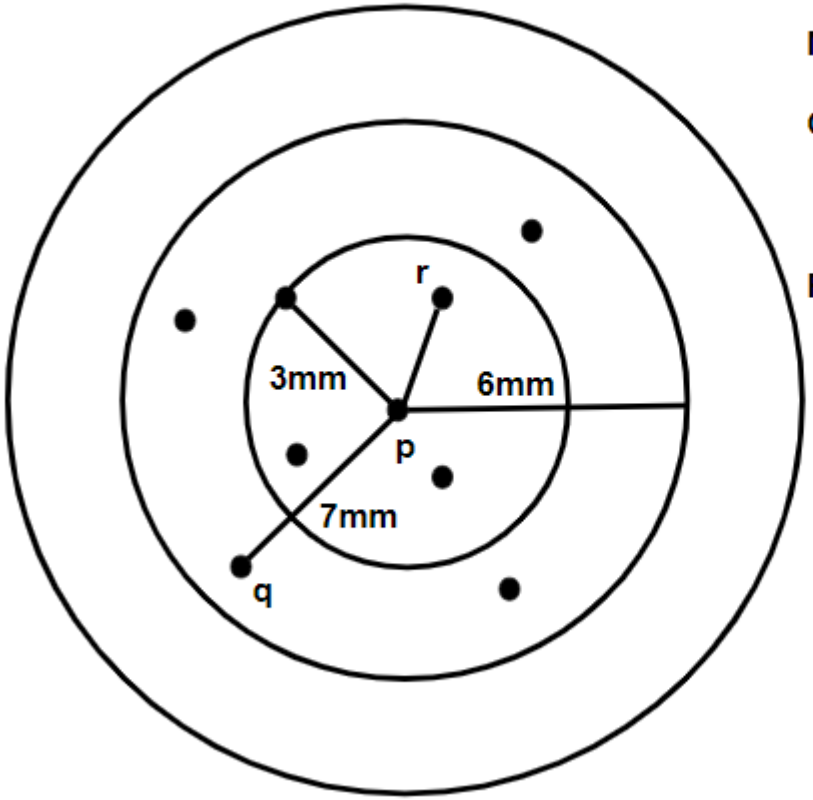
It's determined in relation to another data item, q .The Euclidean Distance(or any other distance metric) between two points p and q, and the Core Distance of p is the Reachability distance between p and q.

Calculating Reachability Distance Between Two Points

First, we will calculate the reachability distance between p and q. In the figure below, we can see that point q is outside the core distance of point p, so to calculate the reachability distance between p and q is the euclidean distance between p and q, i.e., 7mm.

The reachability distance between p and r is the core distance of point p, as r lies within the core distance. Hence the reachability distance between p and r will be 3mm.

OPTICS (Ordering Points To Identify Cluster Structure) – Idea Introductoria



Eps = 6mm

MinPts = 5

Core_Distance(p) = 3mm

Reachability_Distance(q,p) = 7mm

Reachability_Distance(r,p) = 3mm

OPTICS (Algoritmo)

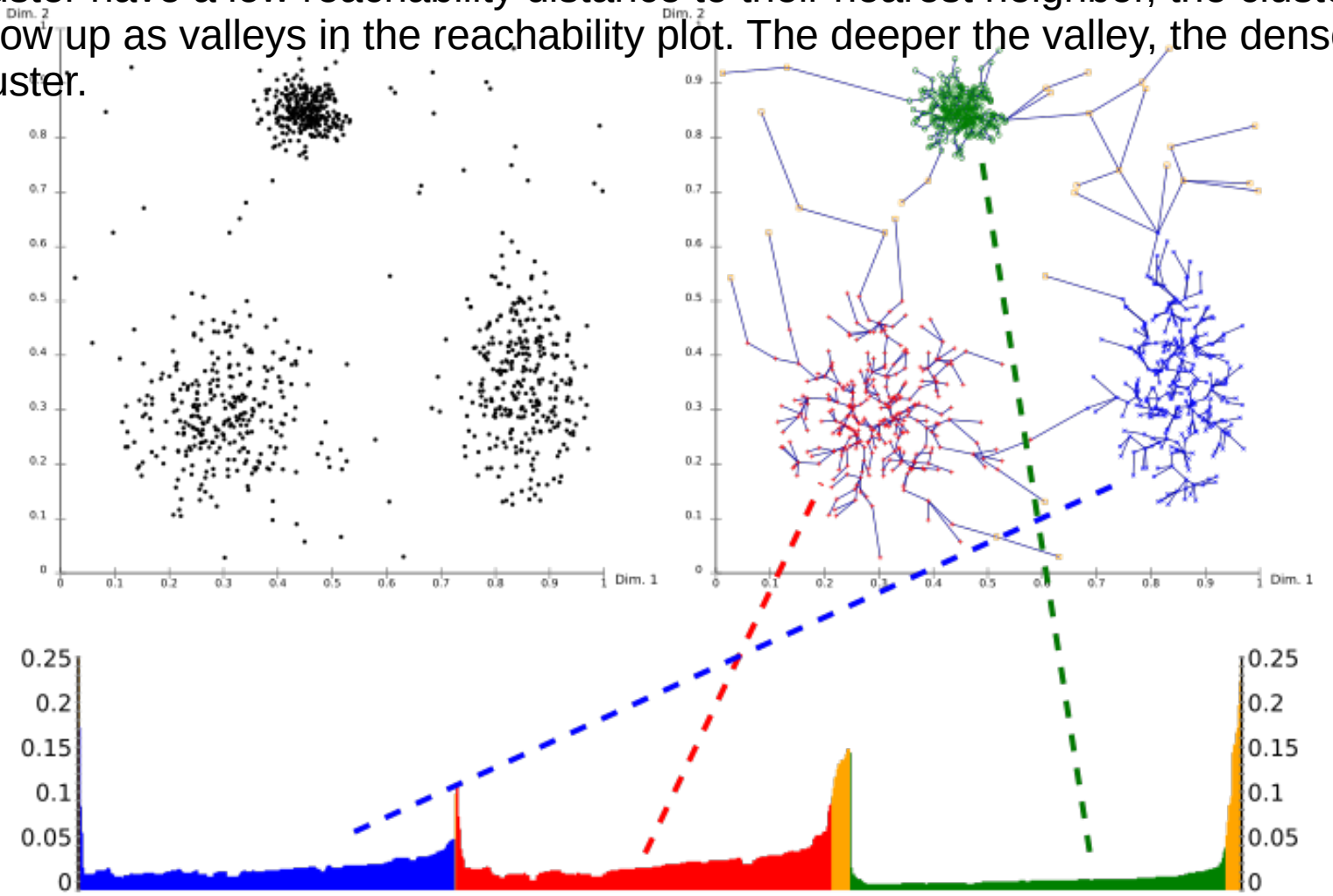
$$\text{core-dist}_{\varepsilon, MinPts}(p) = \begin{cases} \text{UNDEFINED} & \text{if } |N_\varepsilon(p)| < MinPts \\ MinPts\text{-th smallest distance in } N_\varepsilon(p) & \text{otherwise} \end{cases}$$

$$\text{reachability-dist}_{\varepsilon, MinPts}(o, p) = \begin{cases} \text{UNDEFINED} & \text{if } |N_\varepsilon(p)| < MinPts \\ \max(\text{core-dist}_{\varepsilon, MinPts}(p), \text{dist}(p, o)) & \text{otherwise} \end{cases}$$

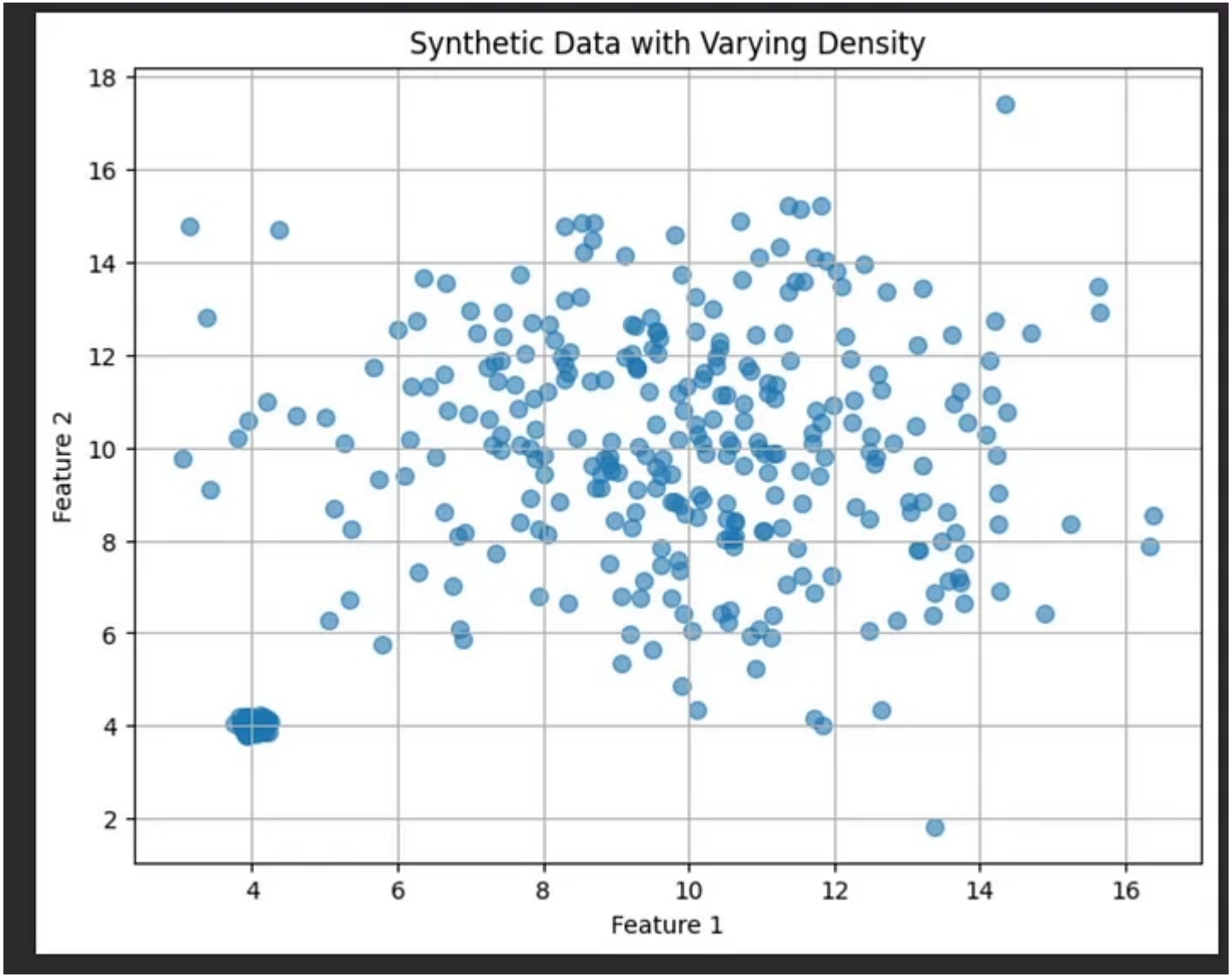
If p and o are nearest neighbors, p and o belong to the same cluster.

OPTICS (Reachability-plot)

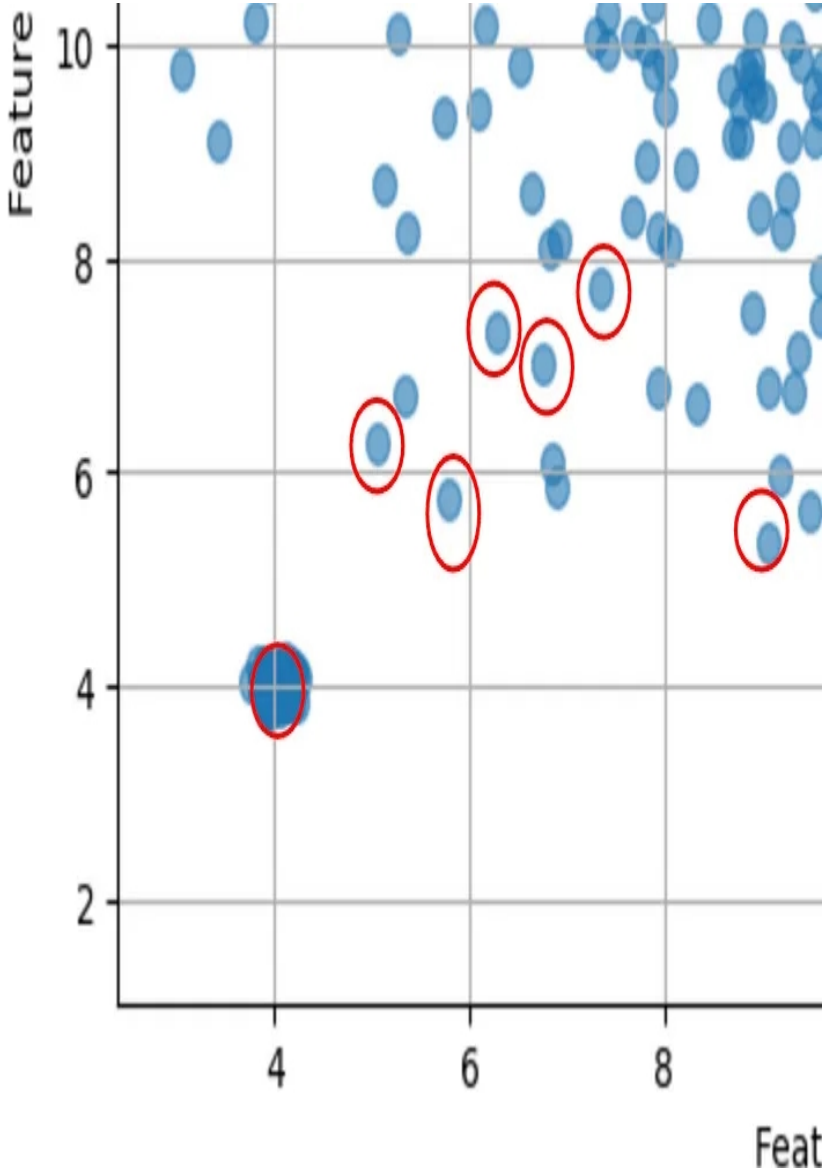
It is a 2D plot, with the ordering of the points as processed by OPTICS on the x-axis and the reachability distance on the y-axis. Since points belonging to a cluster have a low reachability distance to their nearest neighbor, the clusters show up as valleys in the reachability plot. The deeper the valley, the denser the cluster.

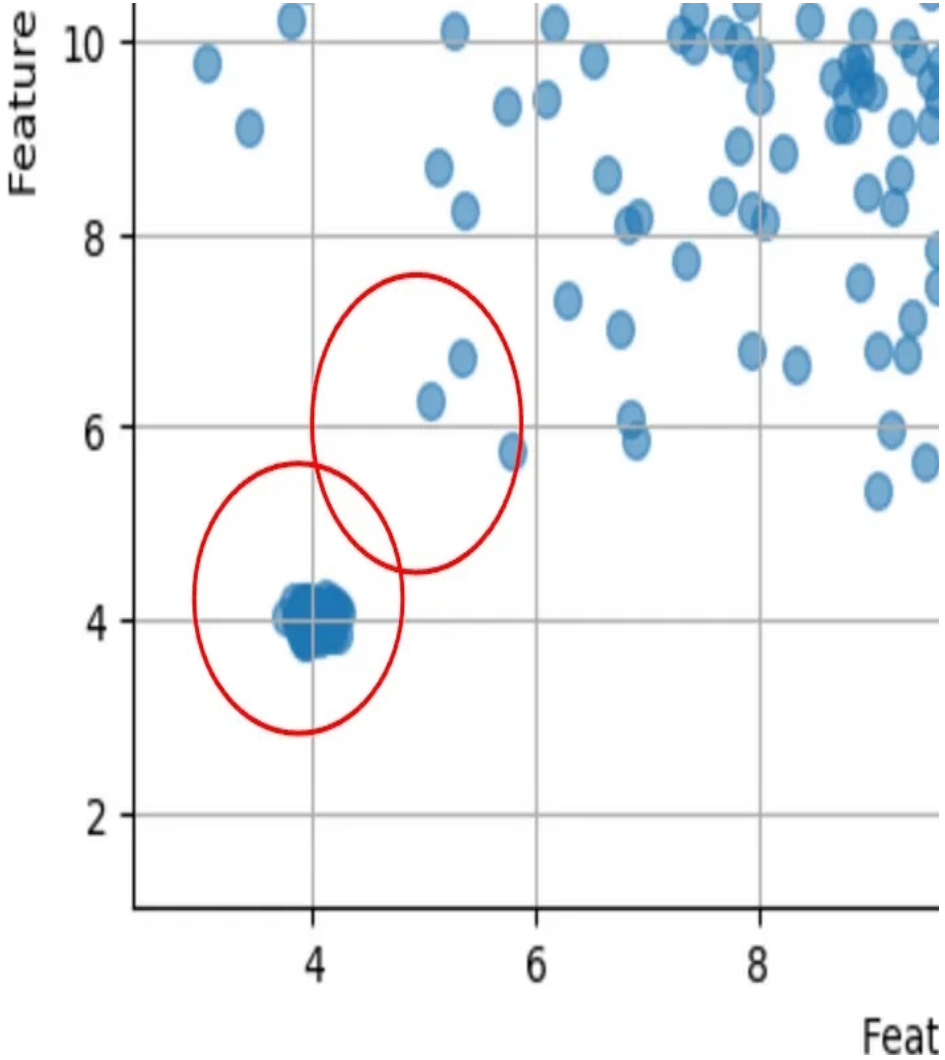


OPTICS (Ejemplo)

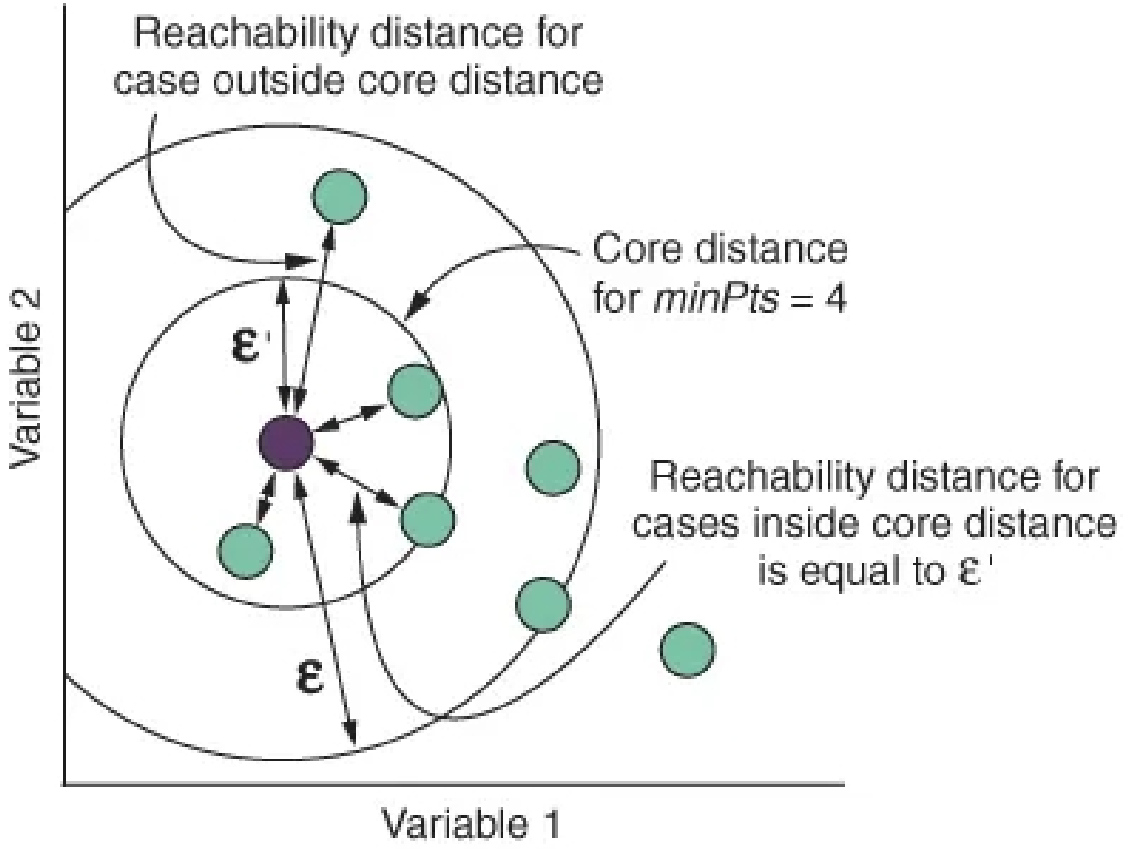


OPTICS (Ejemplo)

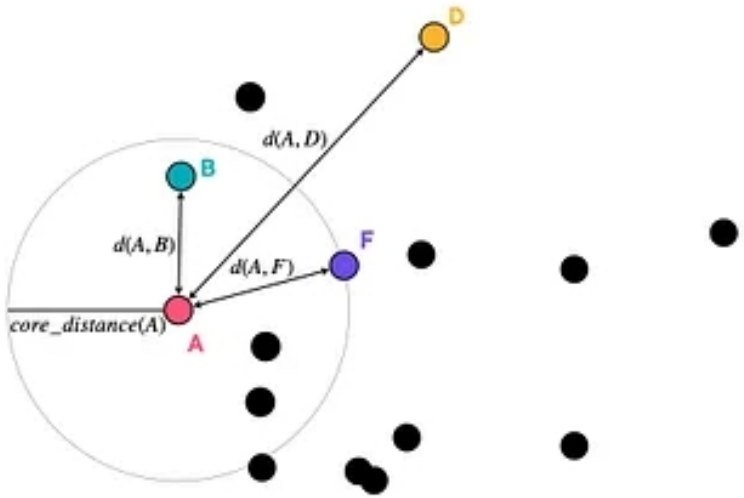




OPTICS (Ejemplo)



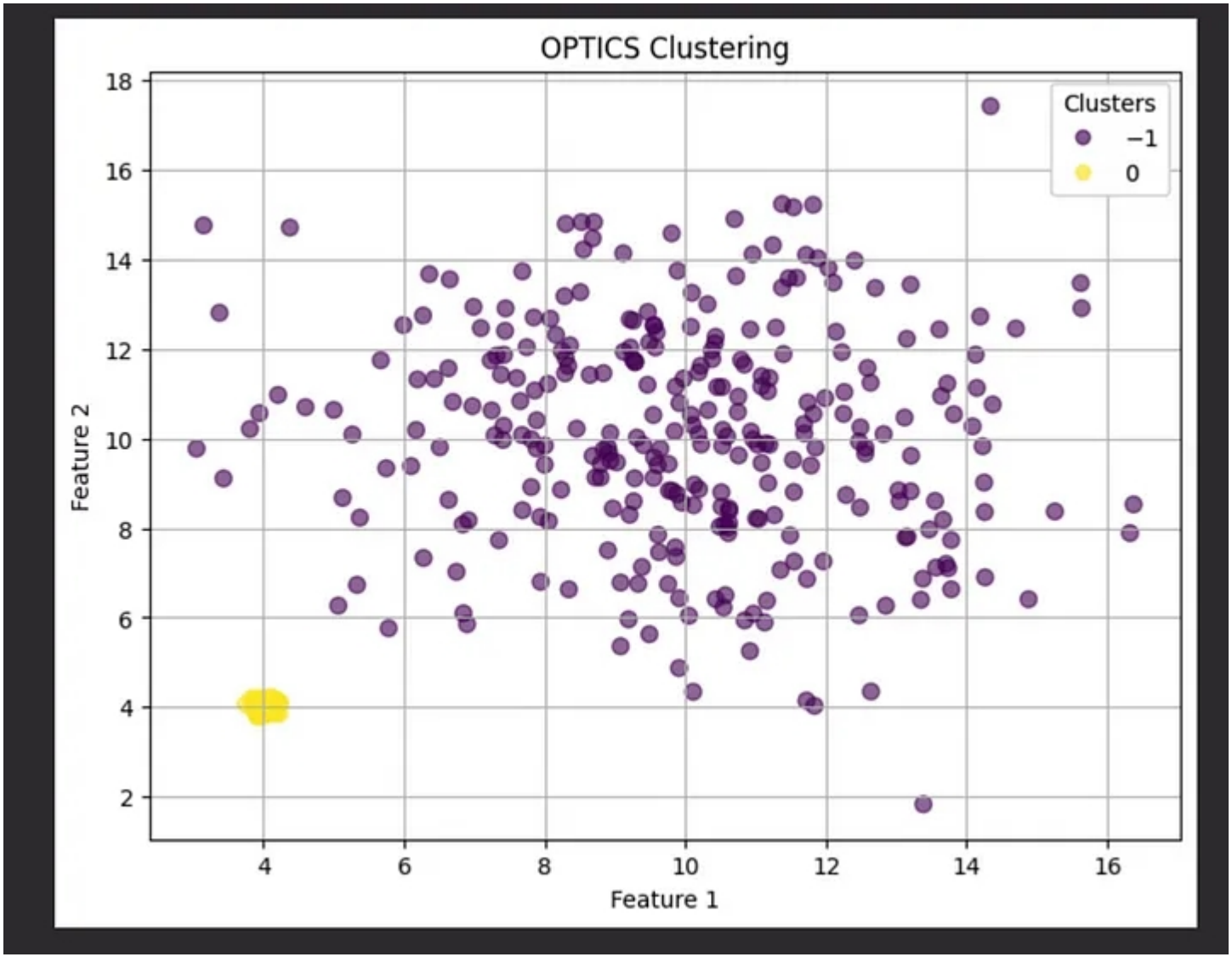
OPTICS (Ejemplo)



$minPoints = 5$

$$\left.\begin{array}{l} d(A,F) = 0.4489609 = core\_distance(A) \\ d(A,B) = 0.3452897 < core\_distance(A) \\ d(A,D) = 0.9740773 > core\_distance(A) \end{array}\right\} \Longrightarrow \begin{array}{l} reachability\_distance(B,A) = core\_distance(A) = 0.4489609 \\ reachability\_distance(F,A) = core\_distance(A) = 0.4489609 \\ reachability\_distance(D,A) = d(A,D) = 0.9740773 \end{array}$$

OPTICS (Ejemplo)

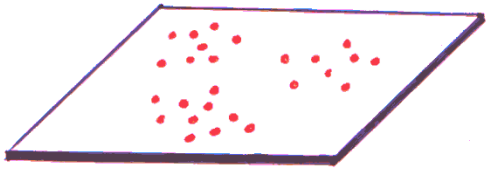


Clustering very large data sets

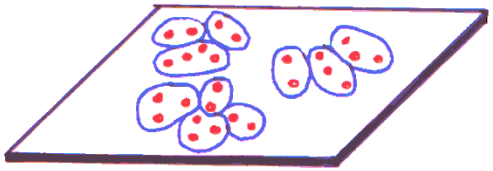
FIB UPC UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

1. Perform *m* (=2 or 3) times a k-means algorithm (with a large value for K, K≈14)
2. Form the crosstable of m obtained partitions
3. Calculated the centroids of the (non empty) cells of the crosstable
4. Perform a Hierarchical Clustering of the centroids weighted with the number of individuals per cell
5. Decide the number of classes present in your data
6. Consolidate your clustering

Clustering very large data sets

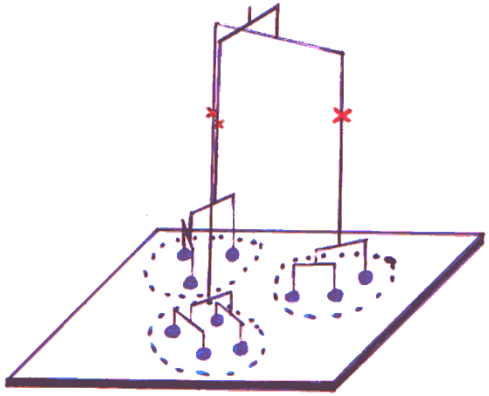


DATOS A CLASIFICAR



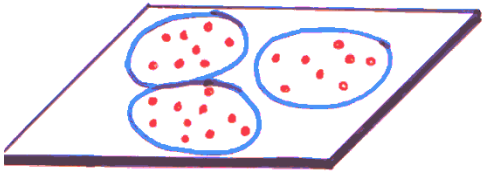
ETAPA 1

PARTICION CON UN NUMERO MUY GRANDE DE CLASES



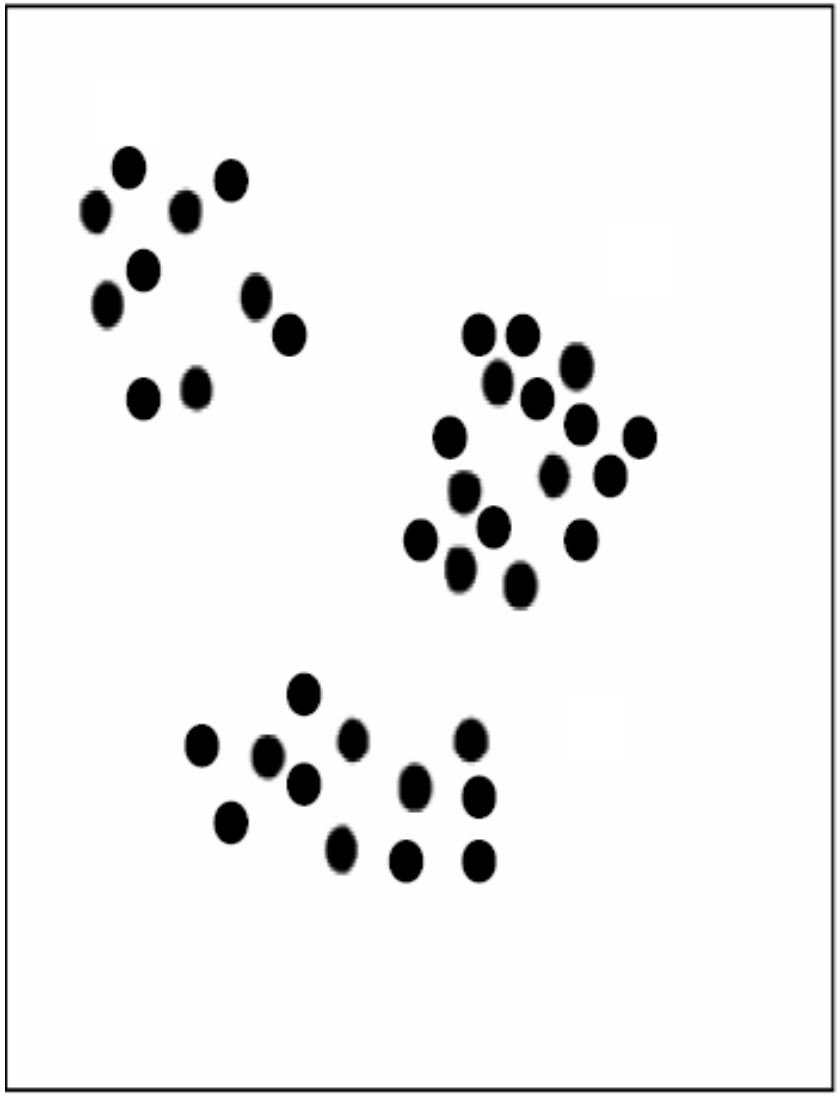
ETAPA 2

CLASIFICACION JERARQUICA DE LOS CENTROS DE LAS CLASES Y CORTE DEL ARBOL



DATOS CLASIFICADOS

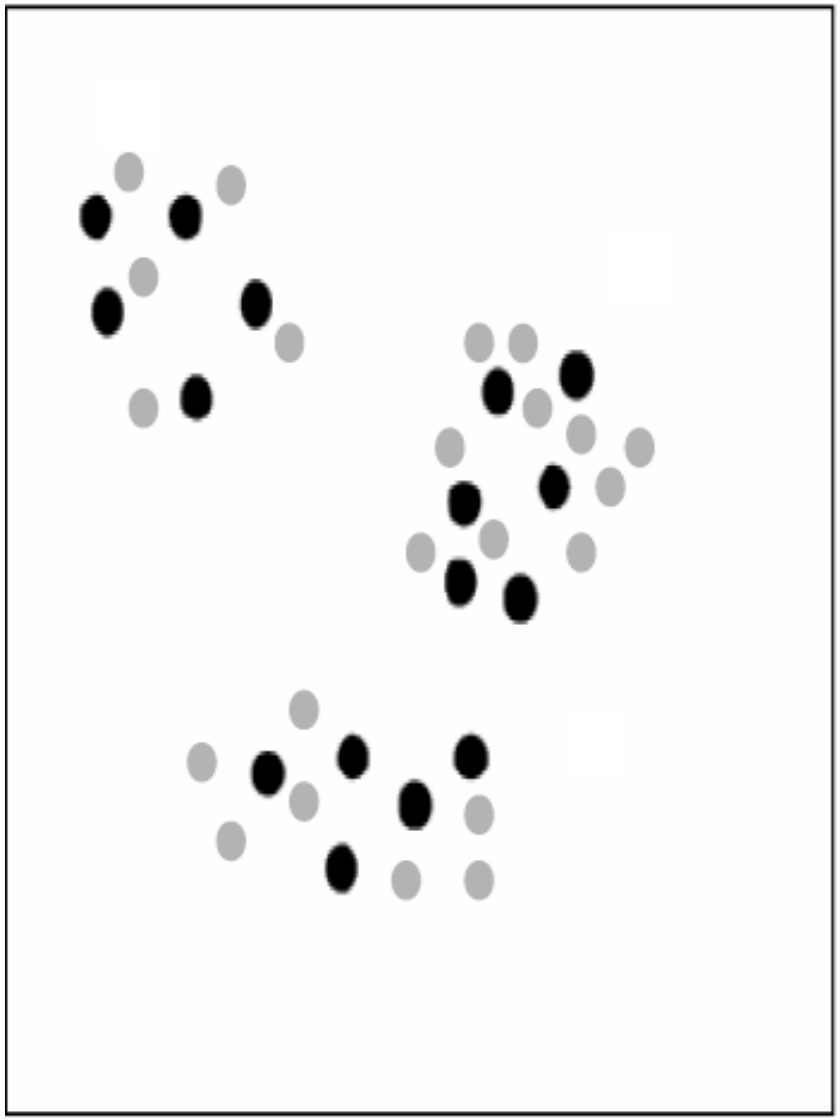
CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



Accelerates hierarchical clustering

- Pick a reduced random sample

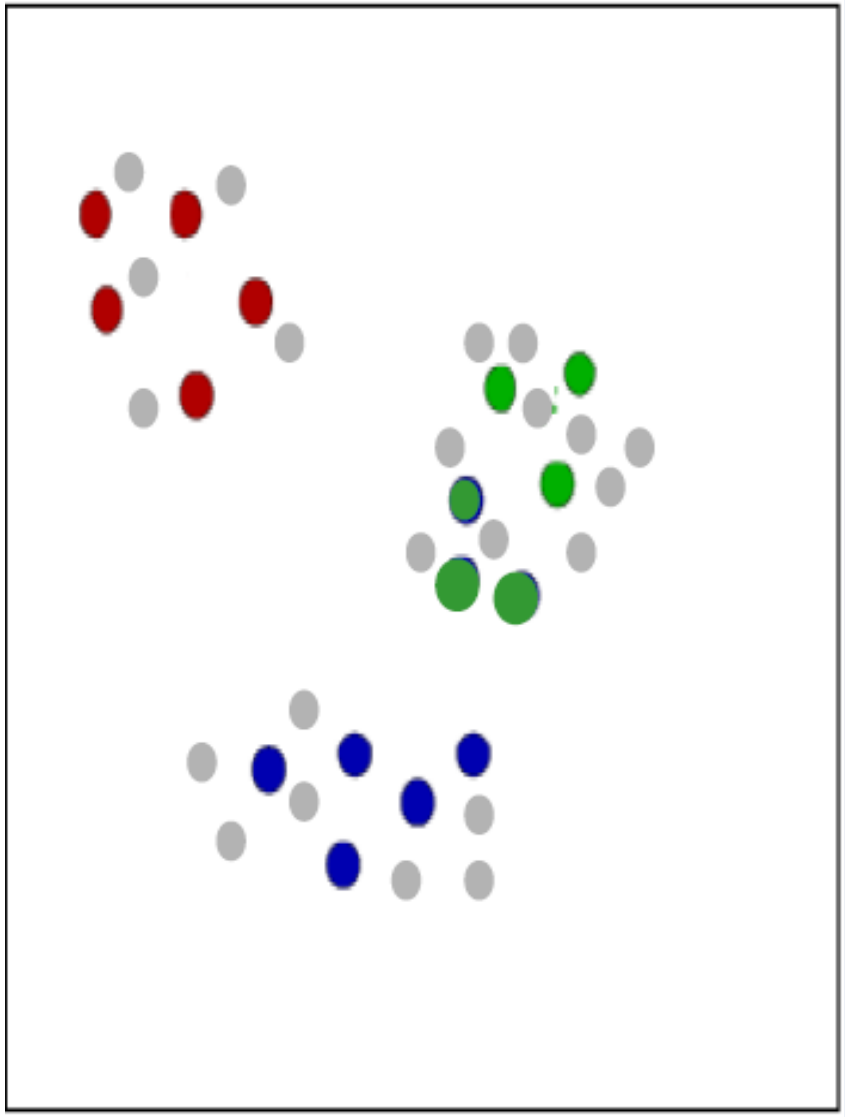
CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



Accelerates hierarchical clustering

- Pick a reduced random sample

CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



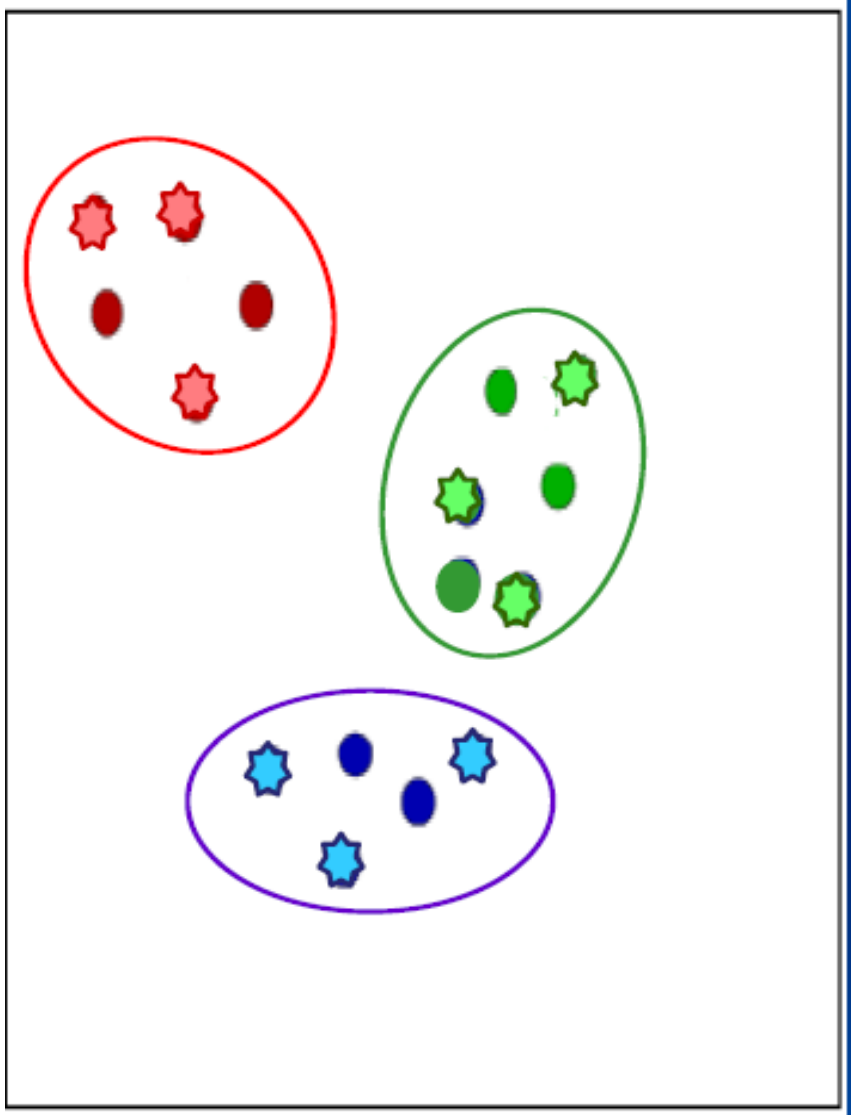
Accelerates hierarchical clustering

- Pick a reduced random sample
- Hierarchical clustering of the sample
  (euclidean distance and single linkage)

  Cut the tree and find classes

Clustering very large data sets

CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



Accelerates hierarchical clustering

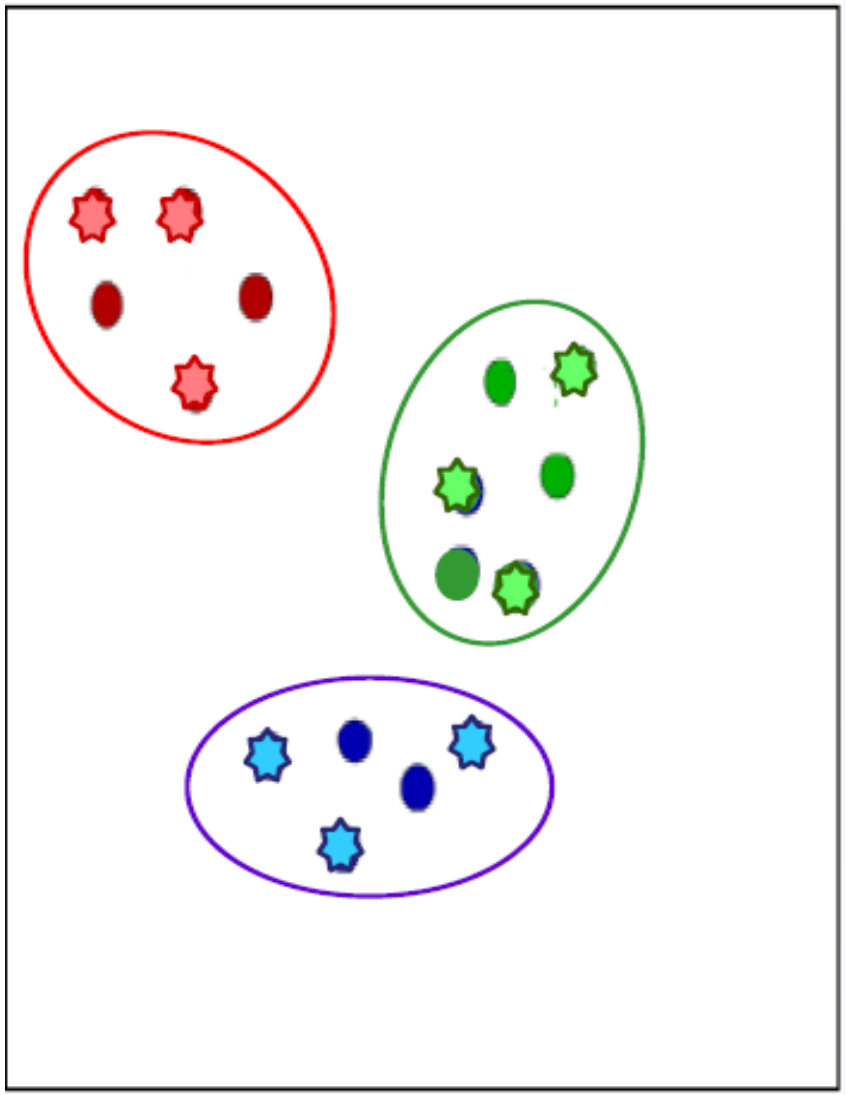
- Pick a reduced random sample
- Hierarchical clustering of the sample
  (decide metrics and aggregation criterion)

  Cut the tree and find classes
- Select a set of disperse points/class

Clustering very large data sets

FIB UPC UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



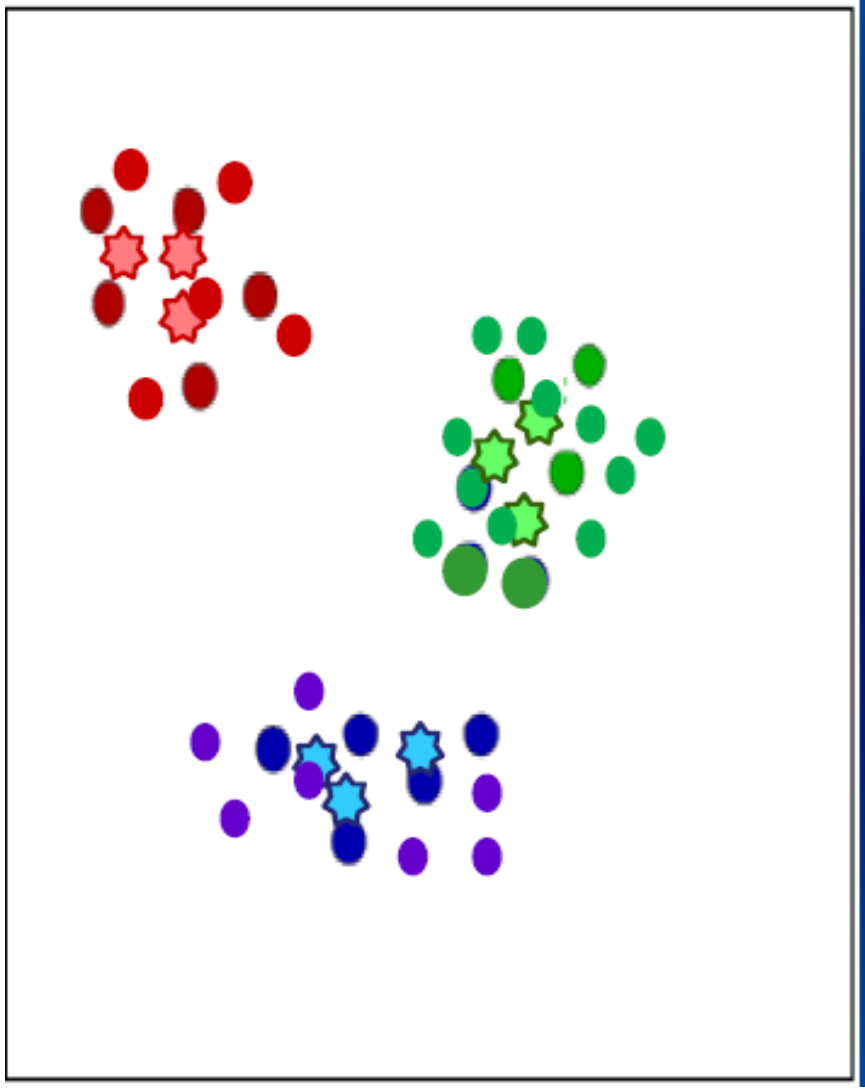
Accelerates hierarchical clustering

- Pick a reduced random sample
- Hierarchical clustering of the sample
  (decide metrics and aggregation criterion)

  Cut the tree and find classes
- Select a set of disperse points/class
- Move representative h% vers centroid (h% = 20%,.....)

UPC

CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



Accelerates hierarchical clustering

- Pick a reduced random sample
- Hierarchical clustering of the sample
  (decide metrics and aggregation criterion)

  Cut the tree and find classes
- Select a set of disperse points/class
- Move representative h% vers centroid (h% = 20%,.....)
- Assign remaining objects to closest representative