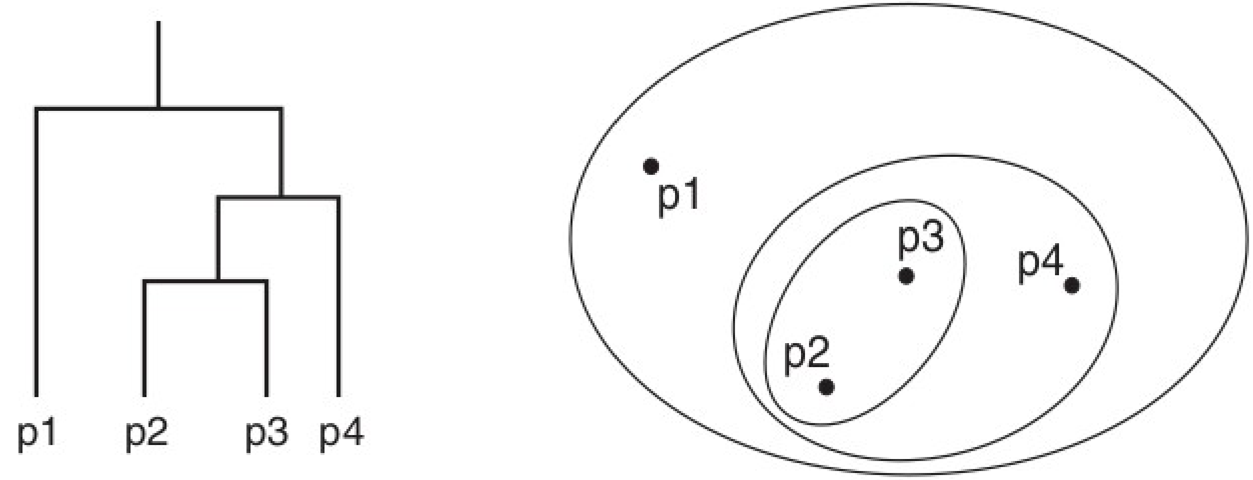
IIC 2433 Minería de Datos

https://github.com/marcelomendoza/IIC2433

- HAC -

Clustering Jerárquico

Idea:

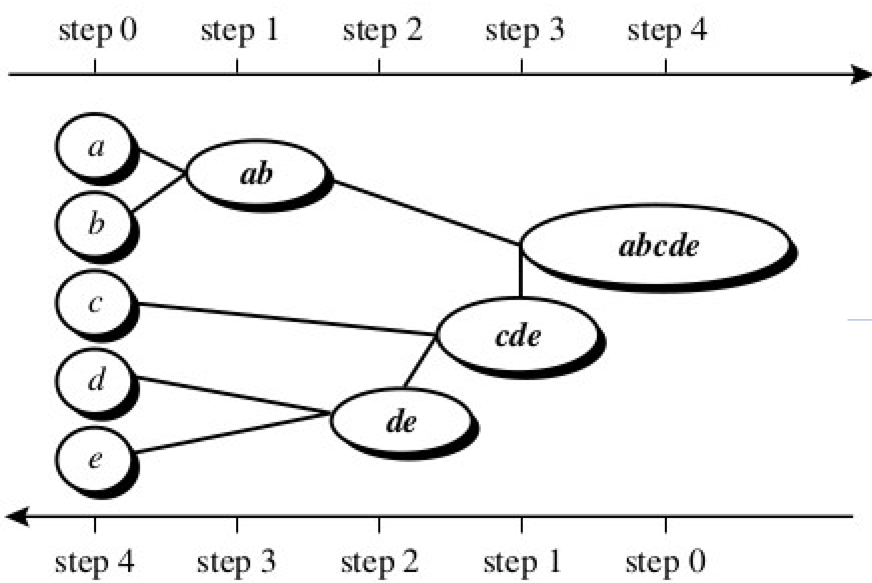


Algorithm Basic agglomerative hierarchical clustering algorithm.

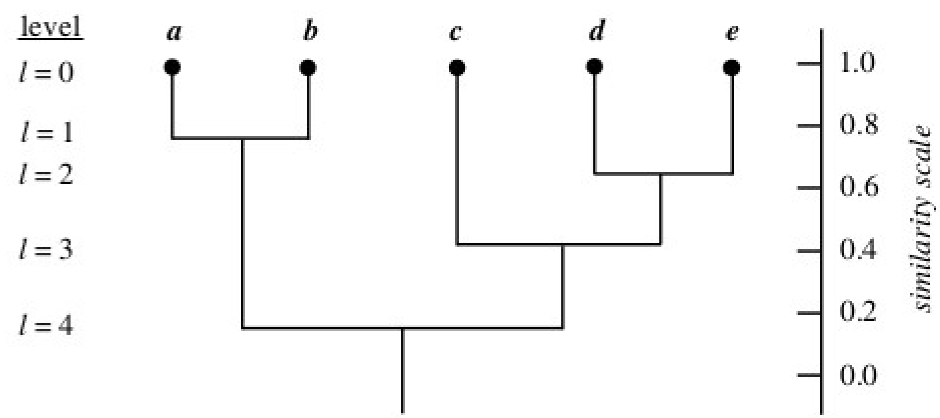
```
Compute the proximity matrix, if necessary.
repeat
    Merge the closest two clusters.
    Update the proximity matrix to reflect the proximity between the new
    cluster and the original clusters.
until Only one cluster remains.
```

Clustering Jerárquico

aglomerativo

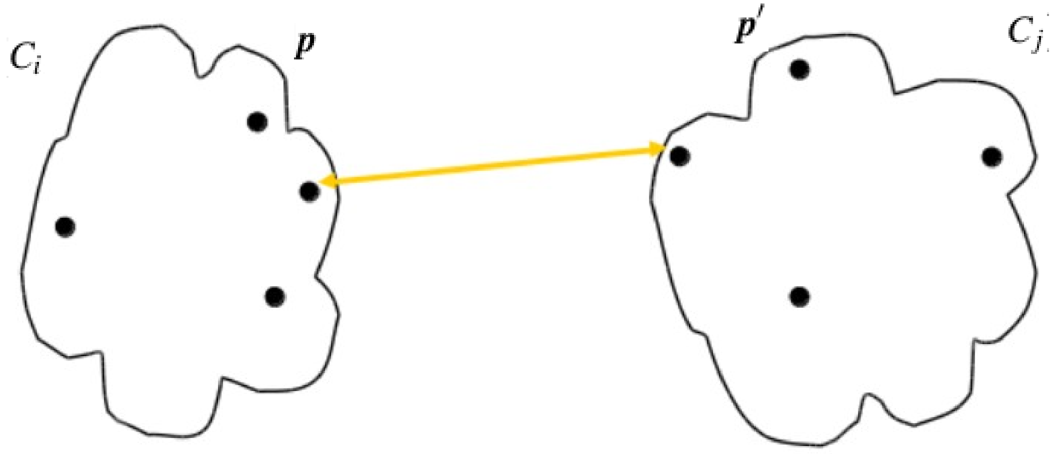


divisivo



Clustering Jerárquico Aglomerativo

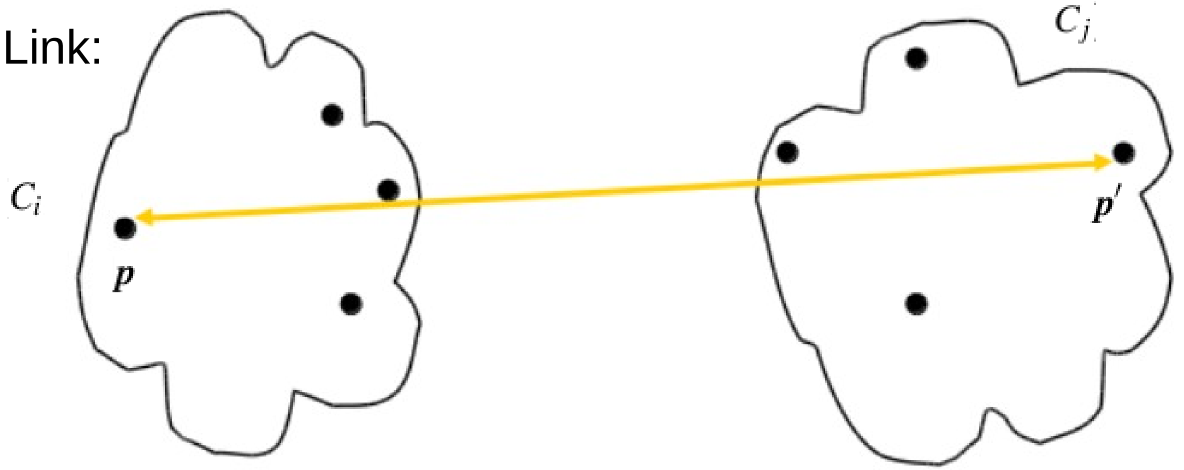
Single Link:



$$d_{min}(C_i, C_j) = \quad min_{\mathbf{p} \in C_i, \mathbf{p}' \in C_j} |\mathbf{p} - \mathbf{p}'|$$

Clustering Jerárquico Aglomerativo

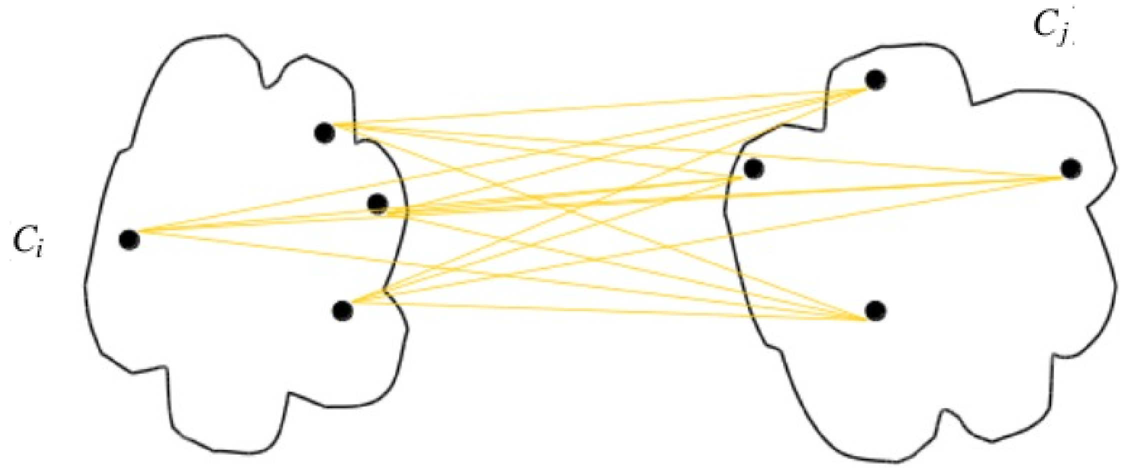
Complete Link:



$$d_{max}(C_i, C_j) = \quad max_{\boldsymbol{p}\in C_i,\, \boldsymbol{p}'\in C_j} |\boldsymbol{p} - \boldsymbol{p}'|$$

Clustering Jerárquico Aglomerativo

Average Link:



$$d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{\boldsymbol{p} \in C_i} \sum_{\boldsymbol{p}' \in C_j} |\boldsymbol{p} - \boldsymbol{p}'|$$

Clustering Jerárquico Aglomerativo

#datos de cada cluster

Método de Ward:

$$\Delta(A,B) = \sum_{i \in A \bigcup B} ||\overrightarrow{x_i} - \overrightarrow{m}_{A \bigcup B}||^2 - \sum_{i \in A} ||\overrightarrow{x_i} - \overrightarrow{m}_A||^2 - \sum_{i \in B} ||\overrightarrow{x_i} - \overrightarrow{m}_B||^2 = \frac{n_A n_B}{n_A + n_B} ||\overrightarrow{m}_A - \overrightarrow{m}_B||^2$$

Centroide del nuevo cluster

Clustering Jerárquico Aglomerativo

Método de Ward:

#datos de cada cluster

$$\Delta(A,B) = \sum_{i \in A \bigcup B} ||\overrightarrow{x_i} - \overrightarrow{m}_{A \bigcup B}||^2 - \sum_{i \in A} ||\overrightarrow{x_i} - \overrightarrow{m}_A||^2 - \sum_{i \in B} ||\overrightarrow{x_i} - \overrightarrow{m}_B||^2 = \frac{n_A n_B}{n_A + n_B} ||\overrightarrow{m}_A - \overrightarrow{m}_B||^2$$

Centroide del nuevo cluster

$$\sim \quad d_{mean}(C_i, C_j) = \left| \boldsymbol{m_i} - \boldsymbol{m_j} \right|$$

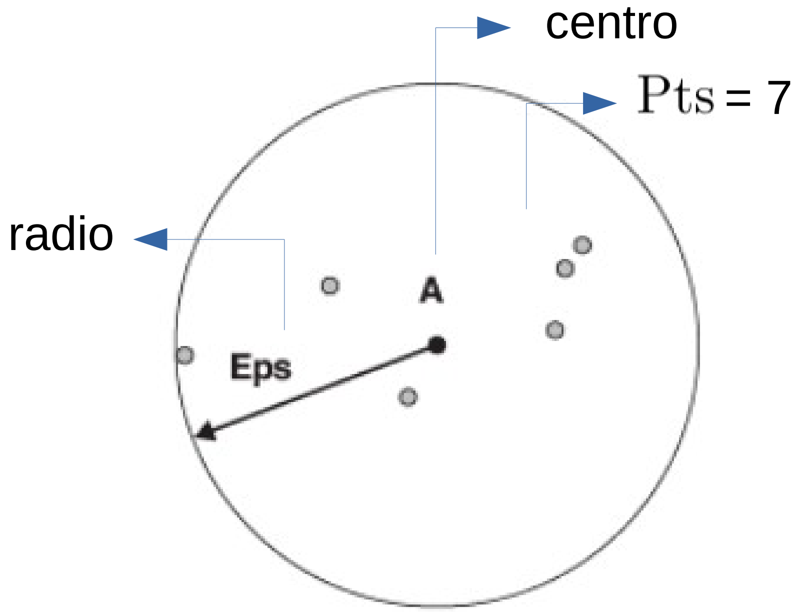
- DBSCAN Y OPTICS -

DBSCAN

Density-based clustering

Idea: Interpretar regiones de alta densidad como clusters.

Enfoque: Center-based density



Noción de densidad: Circunferencia centrada en **A** de radio mínimo tal que contiene al menos **MinPts** vecinos.

DBSCAN

La noción de densidad centrada en puntos nos permite clasificar los datos en tres categorías:

Dado MinPts y Eps :

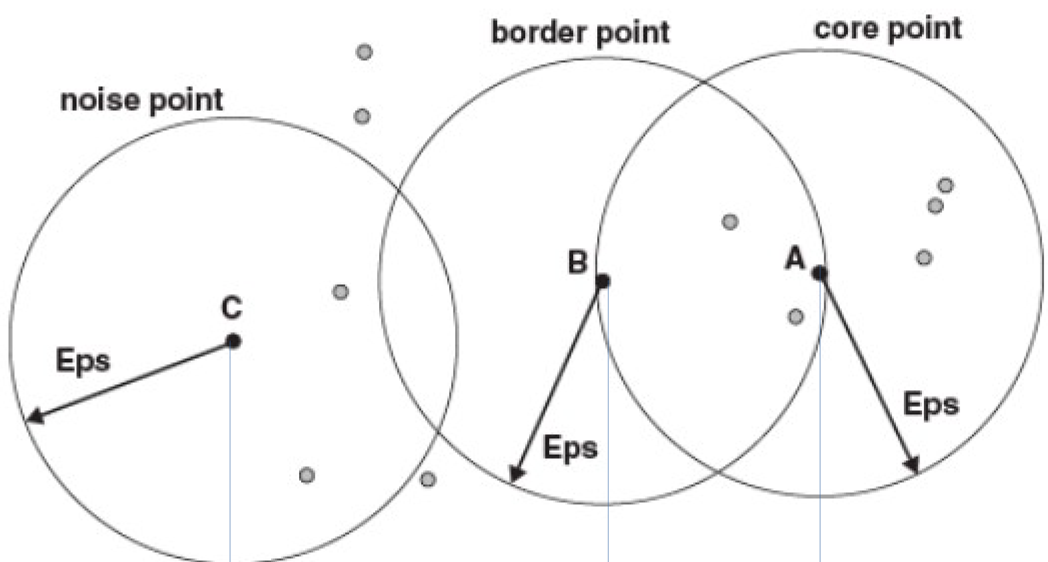
- **Core point**: un dato es un **core point** si la circunferencia de radio Eps centrada en torno del dato cumple que $\mathrm{MinPts} \leq \mathrm{Pts}$

- **Border point**: un dato es un **border point** si no es un core point pero pertenece al vecindario de un core point.

- **Noise point**: Un dato es un **noise point** si no cumple con ninguna de las definiciones anteriores.

DBSCAN

$\mathrm{MinPts} = 7$



$\mathrm{MinPts} \nleq \mathrm{Pts}$

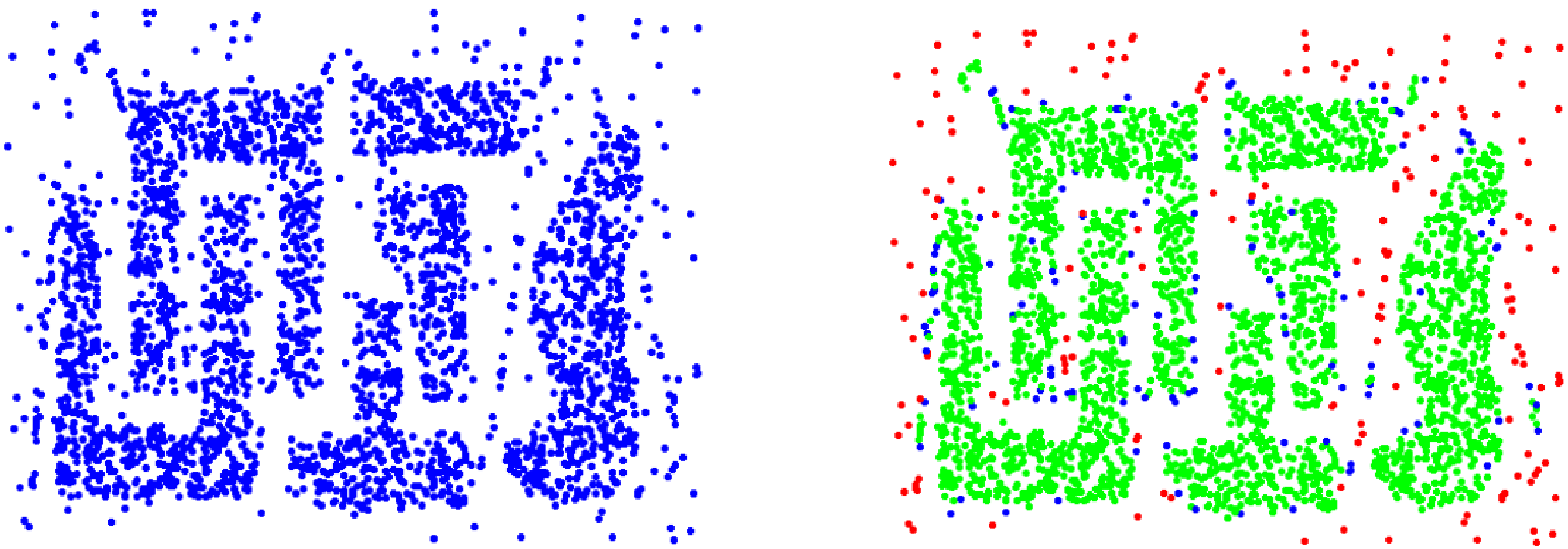
$\mathrm{MinPts} \nleq \mathrm{Pts}$

pero $B \in \odot(A, \mathrm{Eps})$

$\mathrm{MinPts} \leq \mathrm{Pts}$

DBSCAN

Ejemplo:



Core, border y noise points (verde, azul y rojo, resp.)

DBSCAN

Algoritmo:

Algorithm DBSCAN algorithm.
1: Label al. points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

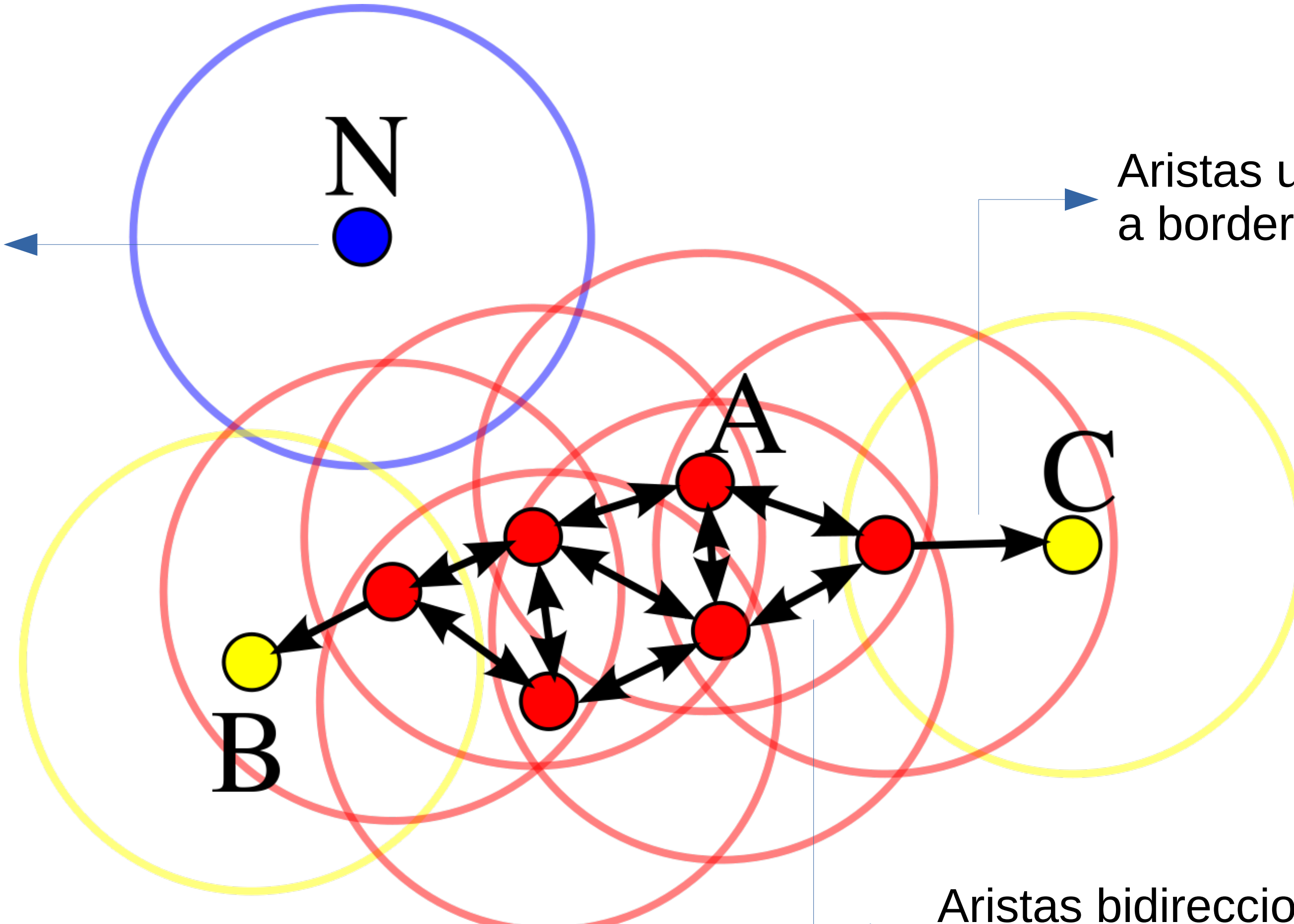
DBSCAN construye un grafo de vecinos cercanos y lo colorea usando componentes conexas

DBSCAN

Grafo dirigido construido conectando core points y border points:

Noise points
disconexos

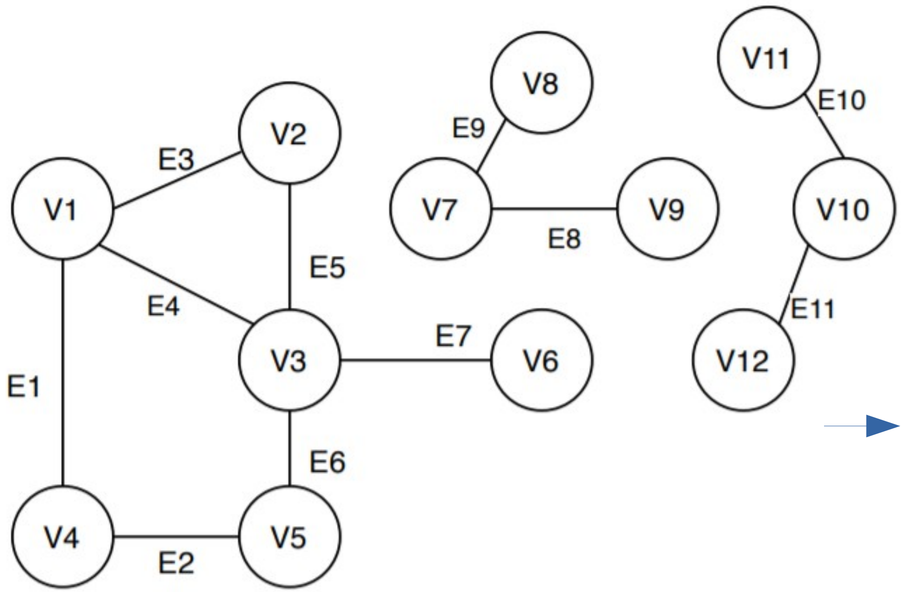
Aristas unidireccionales
a border points

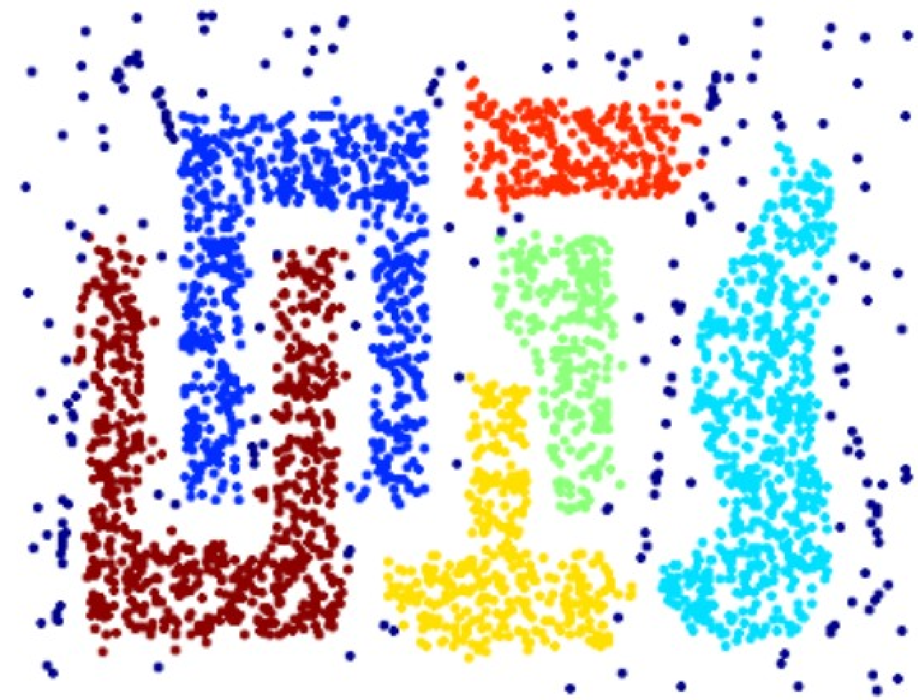


Aristas bidireccionales
entre core points (CC)

DBSCAN

Componentes conexas en DBSCAN:

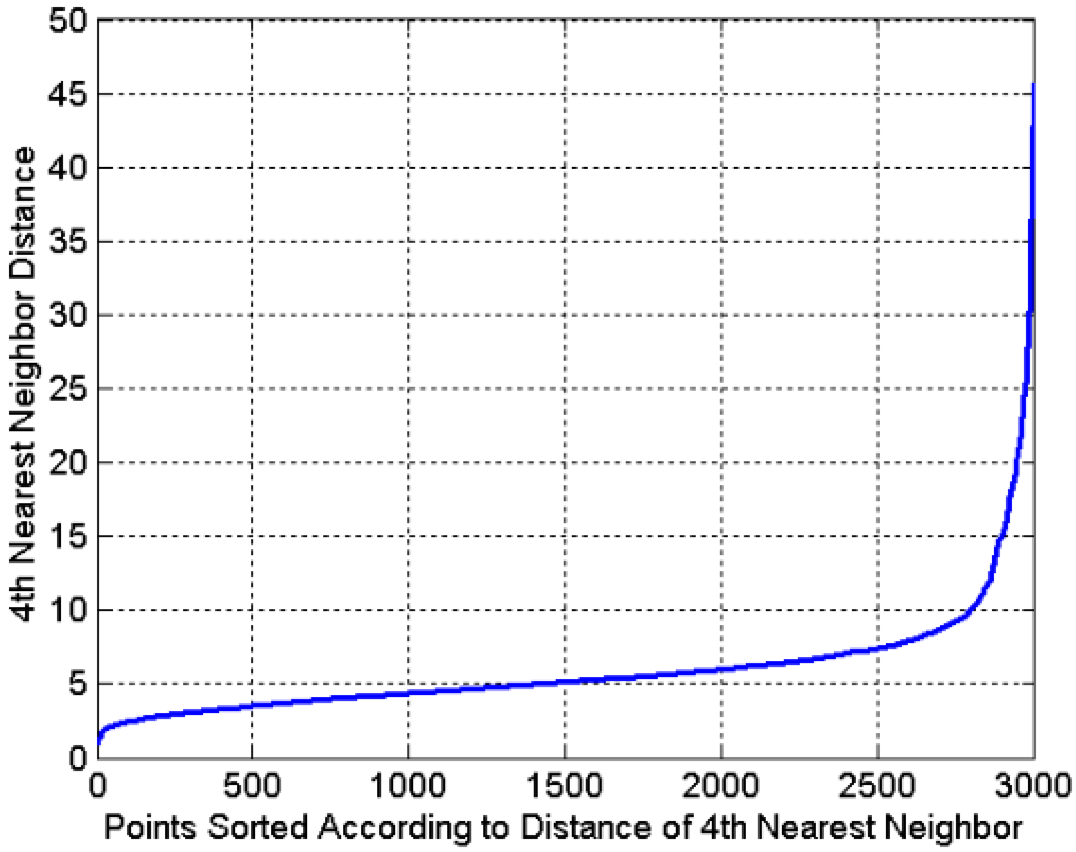




DBSCAN

Sintonización del algoritmo

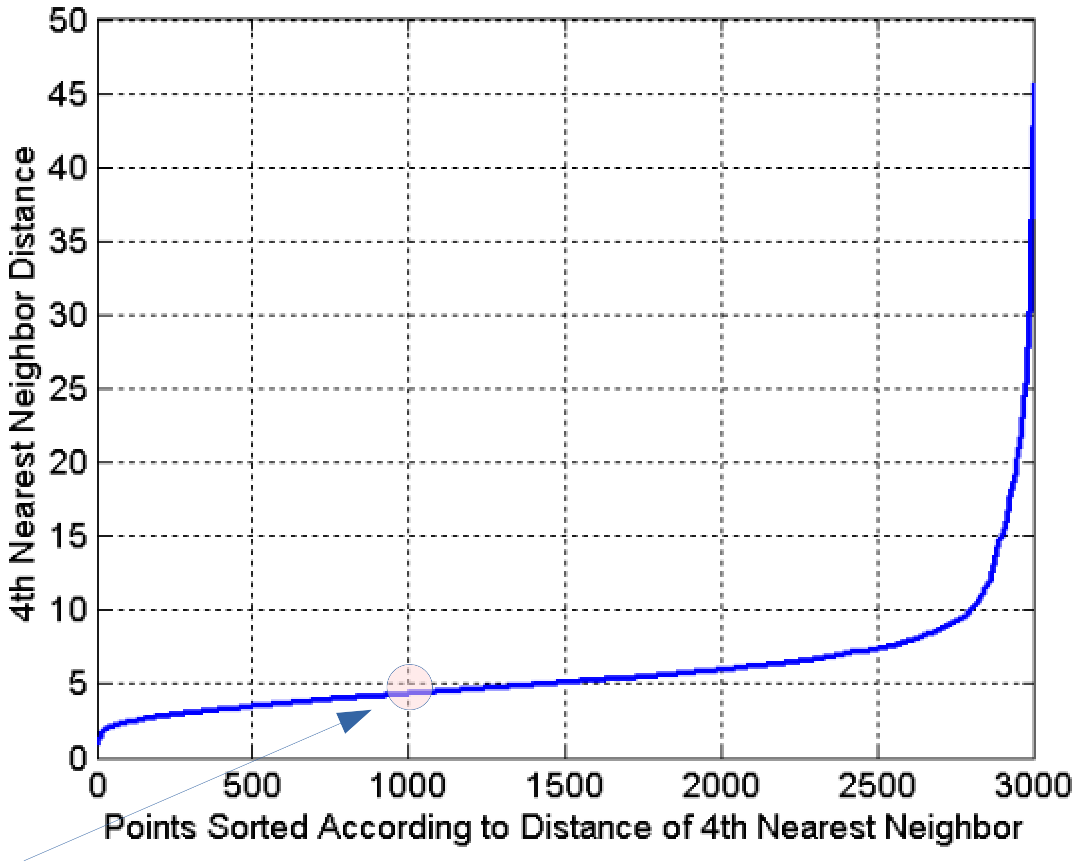
k-dist plot (*k=4*) ← MinPts (candidato)



DBSCAN

Sintonización del algoritmo

k-dist plot (*k=4*) ◄── MinPts (candidato)



1000 puntos tienen a lo más
distancia = 5 a su 4° vecino

Si EPS = 5 y MinPts = 5 → 1000 core points

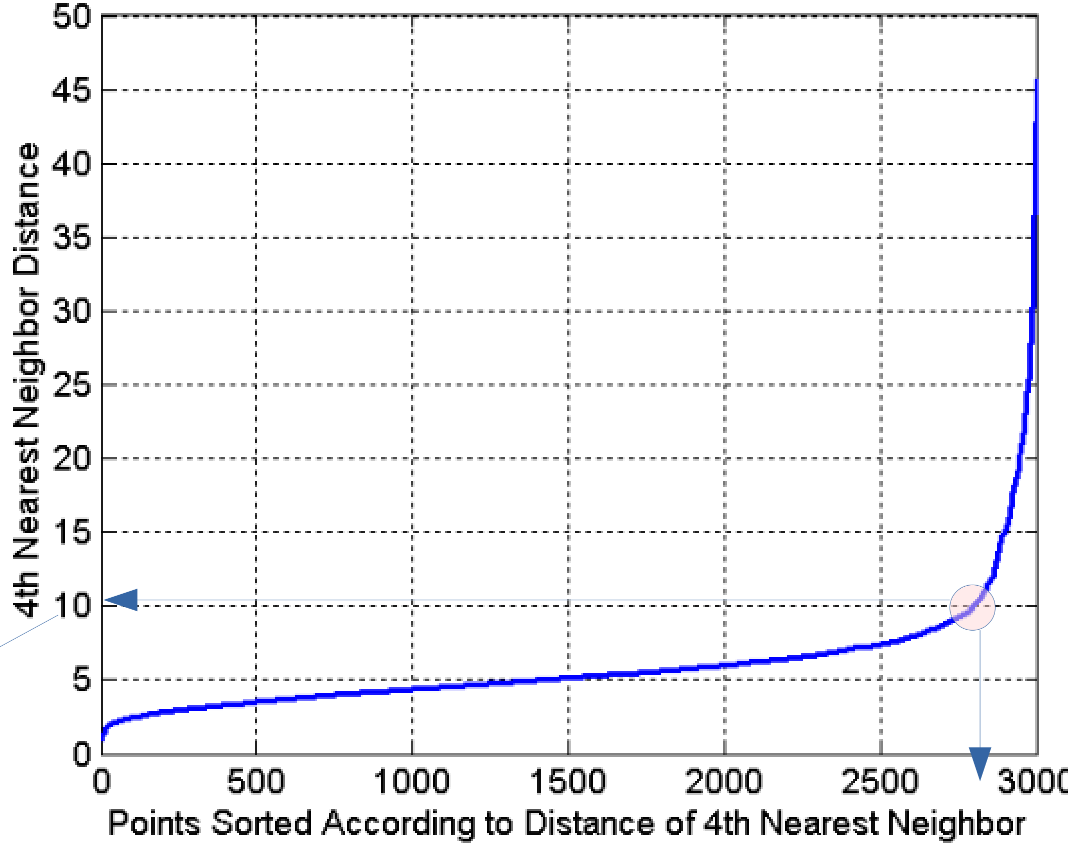
Notar que si aumento EPS, los clusters son menos densos

DBSCAN

Sintonización del algoritmo

MinPts

k-dist plot (*k=4*) MinPts (candidato)



Use un valor de EPS mayor que 10

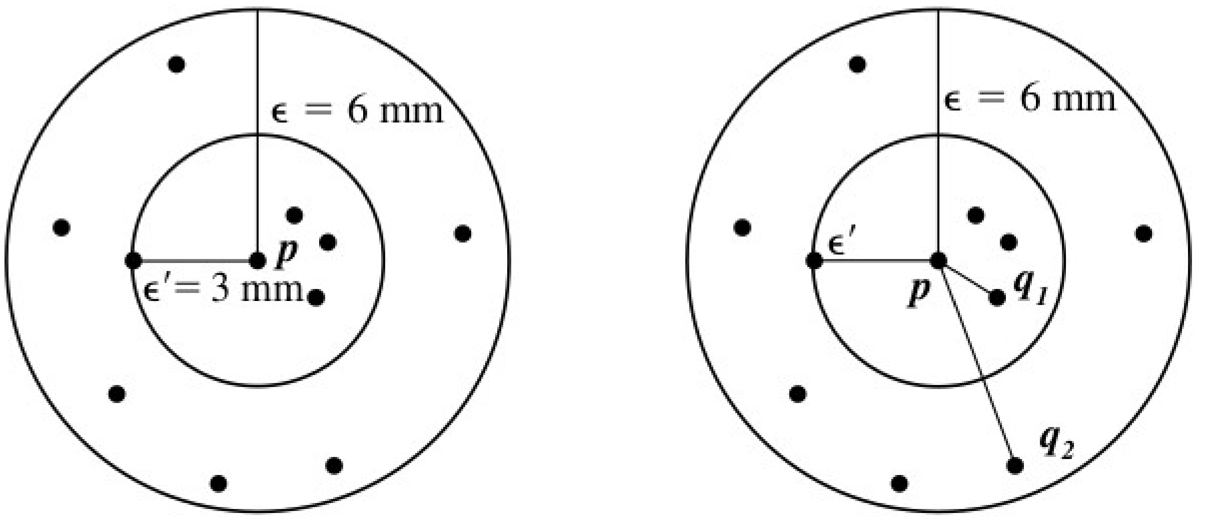
Eps

Noise points (< 10%)

OPTICS

Sintonizar EPS no es fácil, por lo que OPTICS elimina este parámetro.

Si aumento el valor de EPS, los clusters densos quedarán contenidos en los nuevos clusters.

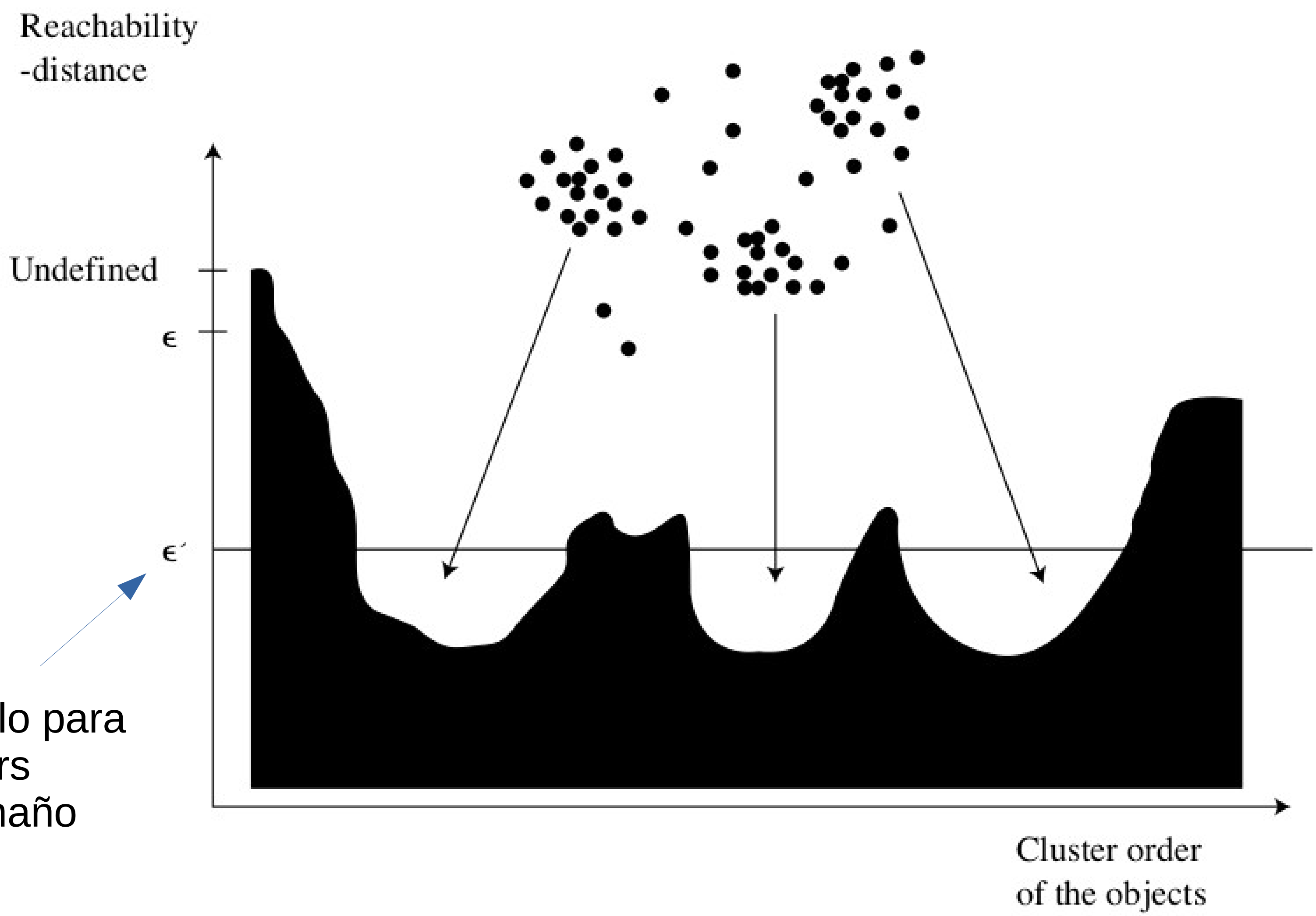


Se define la **core-distance** de *p* como el menor EPS para el cual *p* es CORE.

La **reachability-distance** entre *q* y *p* es el mayor valor entre la **core-distance** de *p* y la distancia Euclideana entre *p* y *q*.

OPTICS

OPTICS ordena los objetos según su **reachability-distance** a los CORE point.



Puedo ajustarlo para que los clusters tengan un tamaño mínimo