



# IIC 2433 Minería de Datos

<https://github.com/marcelomendoza/IIC2433>

## Cierre de la clase 4 – Isolation Forest y HAC



### **Isolation Forest:**

¿Por qué se dice que Isolation Forest requiere de un uso más bajo de memoria que LOF?

¿Por qué se dice que Isolation Forest escala mejor en alta dimensionalidad que LOF?

¿Qué es lo que mejora en Isolation Forest si uso más iTrees?

¿Por qué se pueden producir falsos negativos en Isolation Forest?

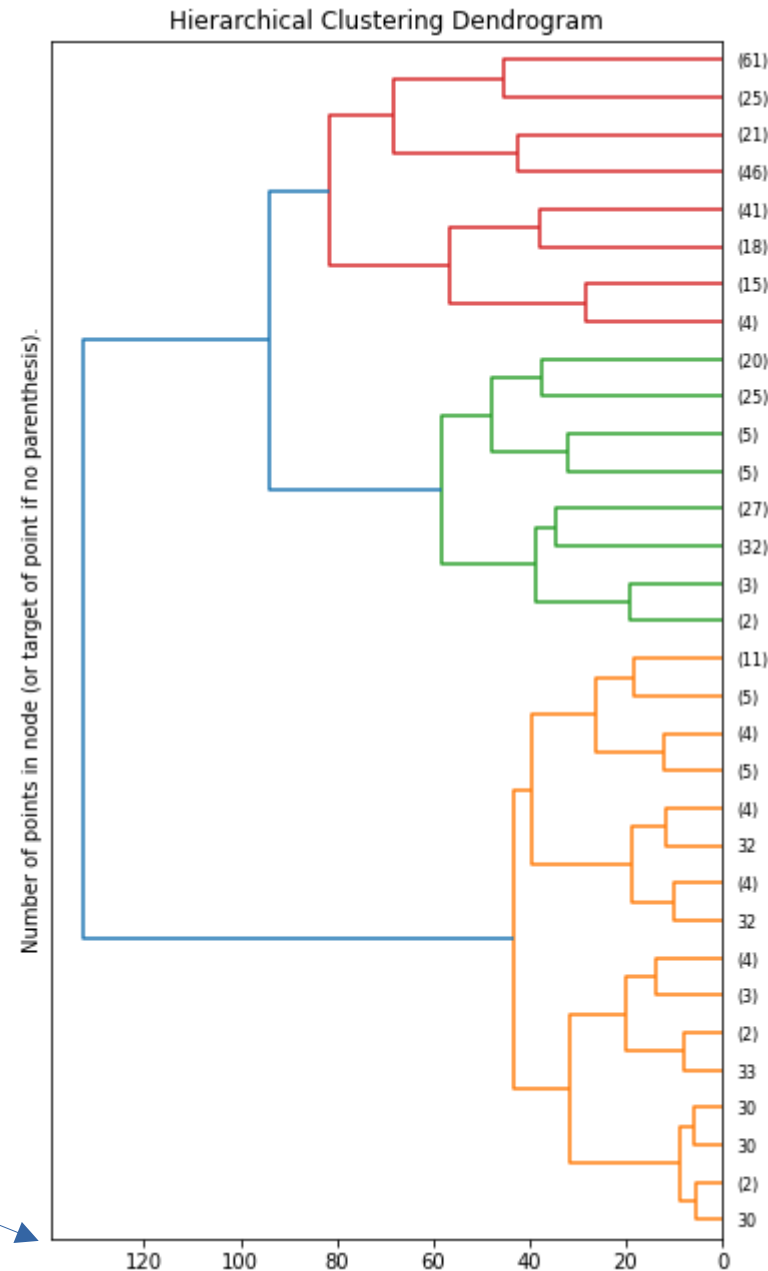
### **HAC:**

¿Qué le hace al dendrograma el parámetro `n_clusters`?

¿Por qué Ward o average link funciona mejor que single y complete link?

## Cierre de la clase 4 – Actividad formativa

En la actividad construyeron un dendrograma con Ward o average, que eran los dos mejores métodos. Un dendrograma a  $p=4$  se ve así:

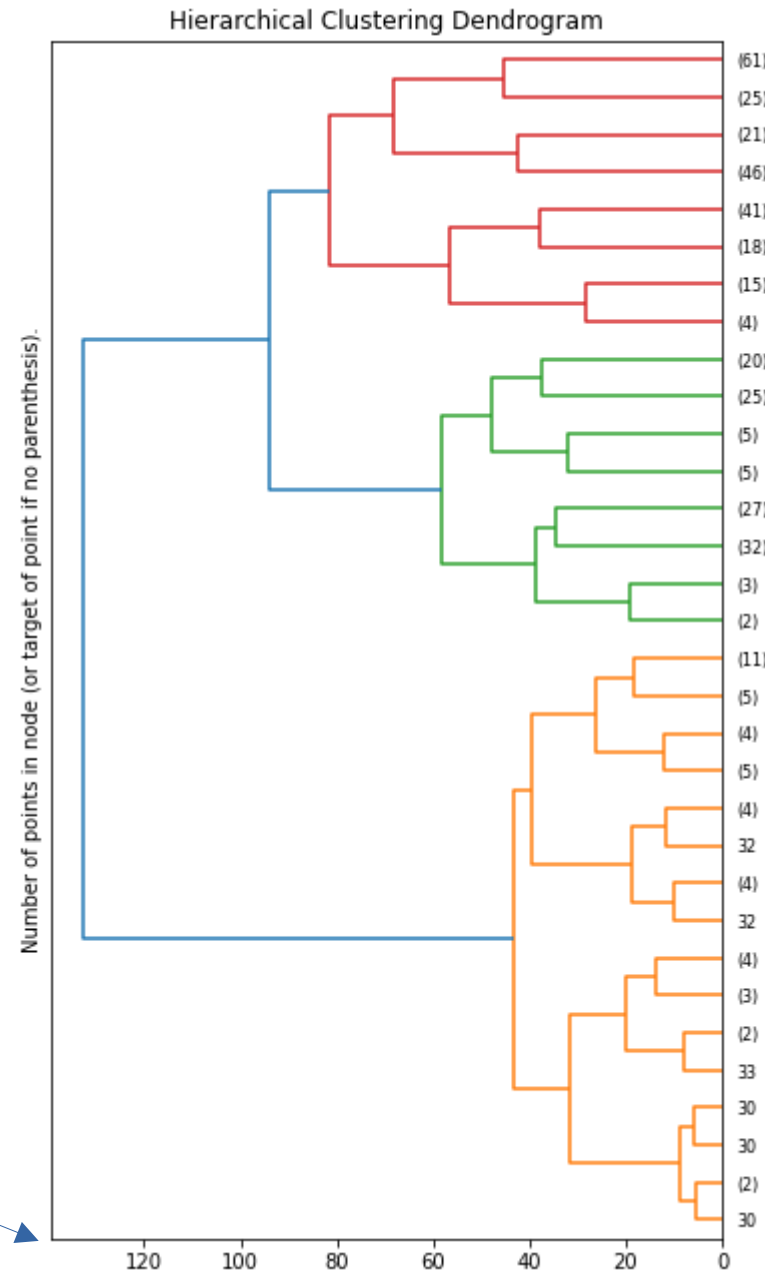


¿Qué representa este eje?



## Cierre de la clase 4 – Actividad formativa

En la actividad construyeron un dendrograma con Ward o average, que eran los dos mejores métodos. Un dendrograma a  $p=4$  se ve así:



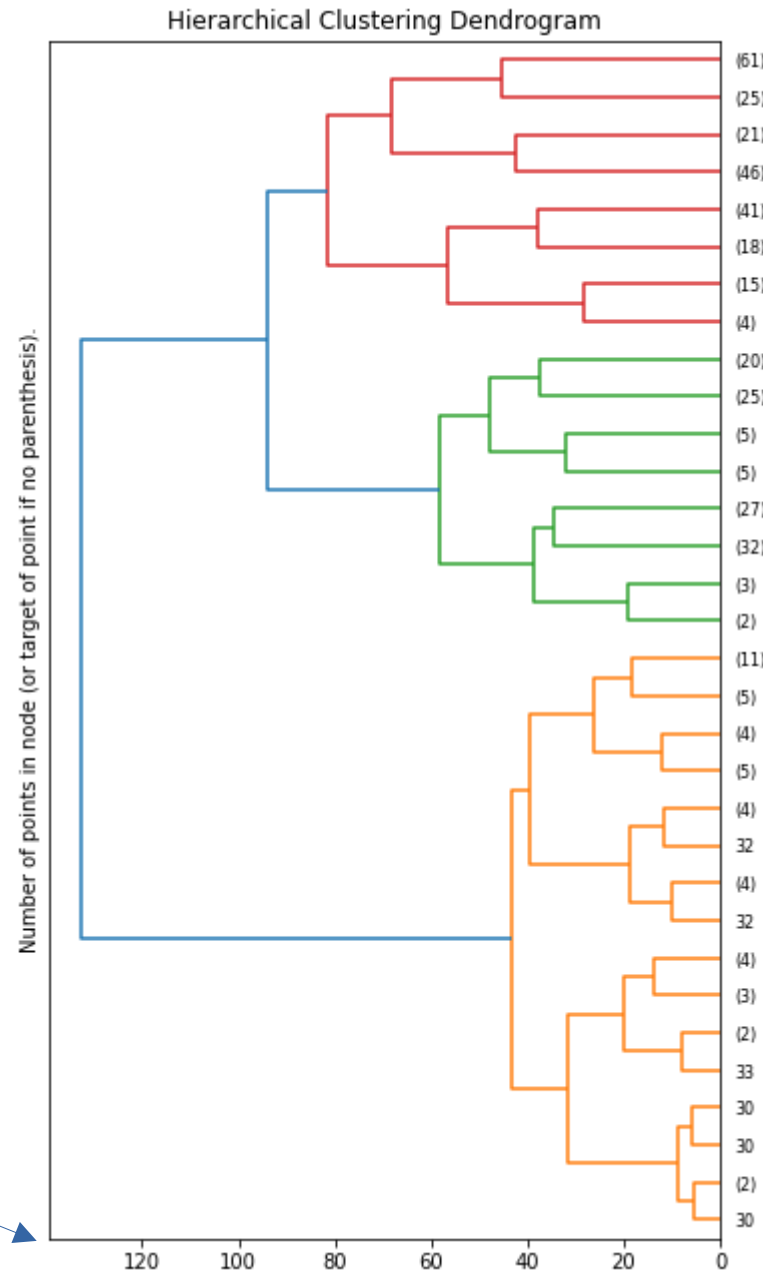
¿Qué representa este eje?



El árbol, ¿Está balanceado por la altura?

## Cierre de la clase 4 – Actividad formativa

En la actividad construyeron un dendrograma con Ward o average, que eran los dos mejores métodos. Un dendrograma a  $p=4$  se ve así:



¿Qué representa este eje?



El árbol, ¿Está balanceado por la altura?

Altura de un árbol binario balanceado:

$$h = O(\log_2 n)$$

Si  $n = 400$ ,  $h \sim 8 - 9$

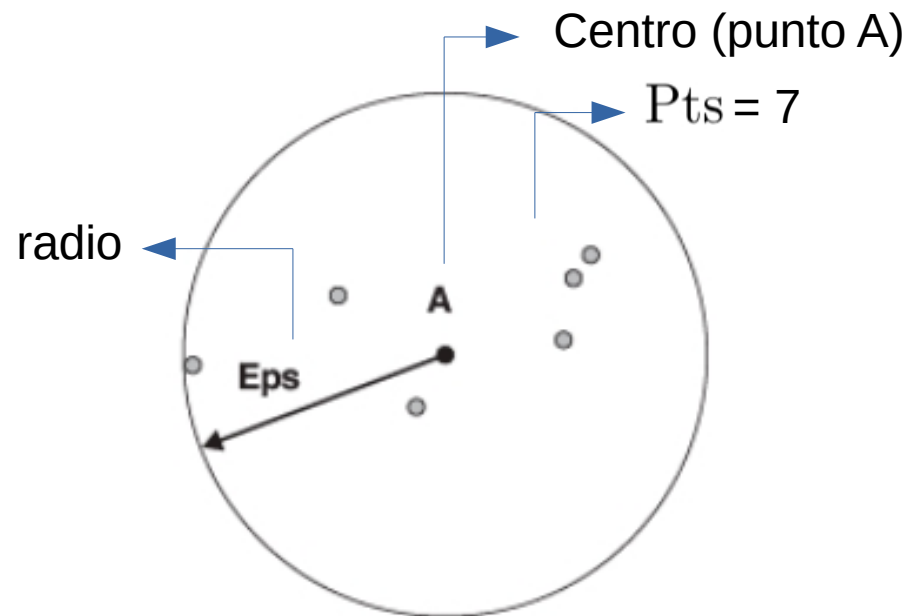
- DBSCAN -

# 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 **Eps** tal que contiene al menos **MinPts** vecinos.

# DBSCAN

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

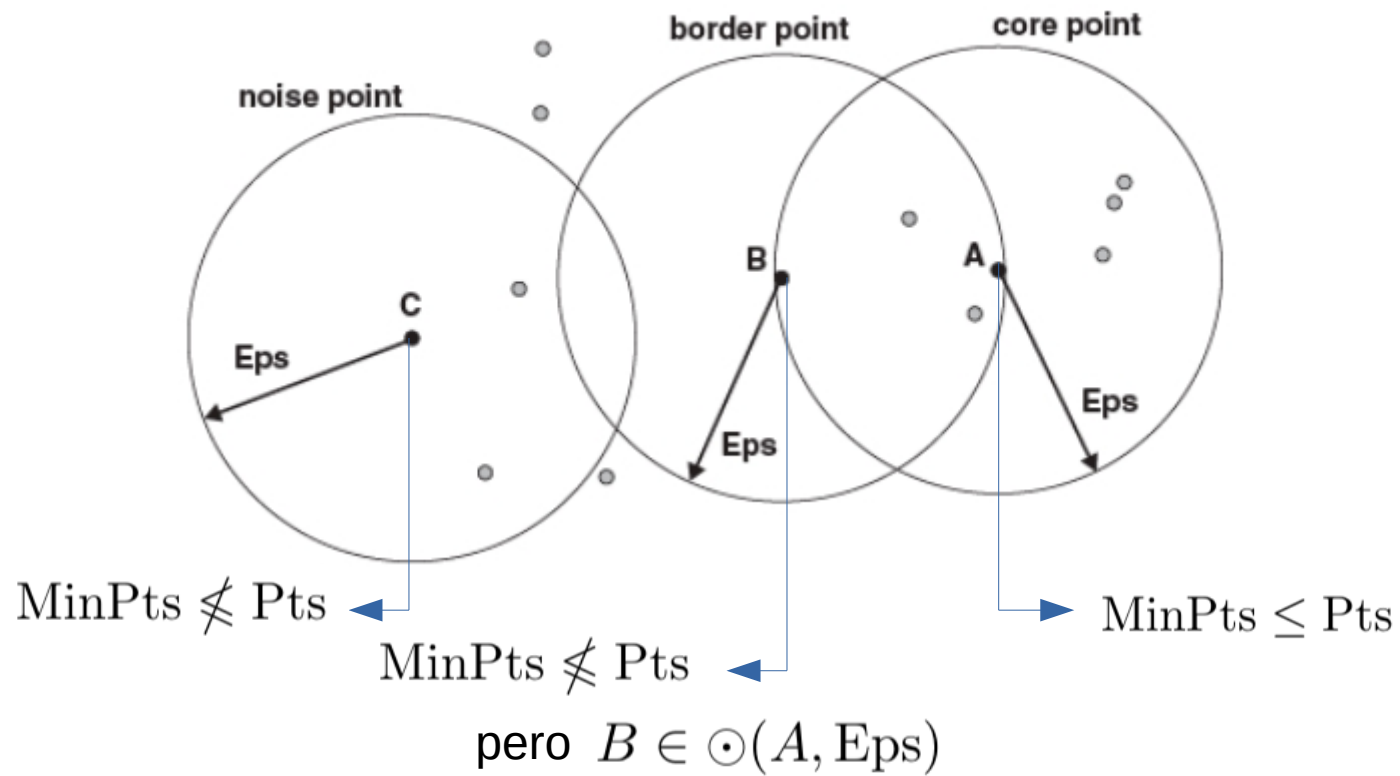
Dado  $\text{MinPts}$  y  $\text{Eps}$  :  hiperparámetros

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



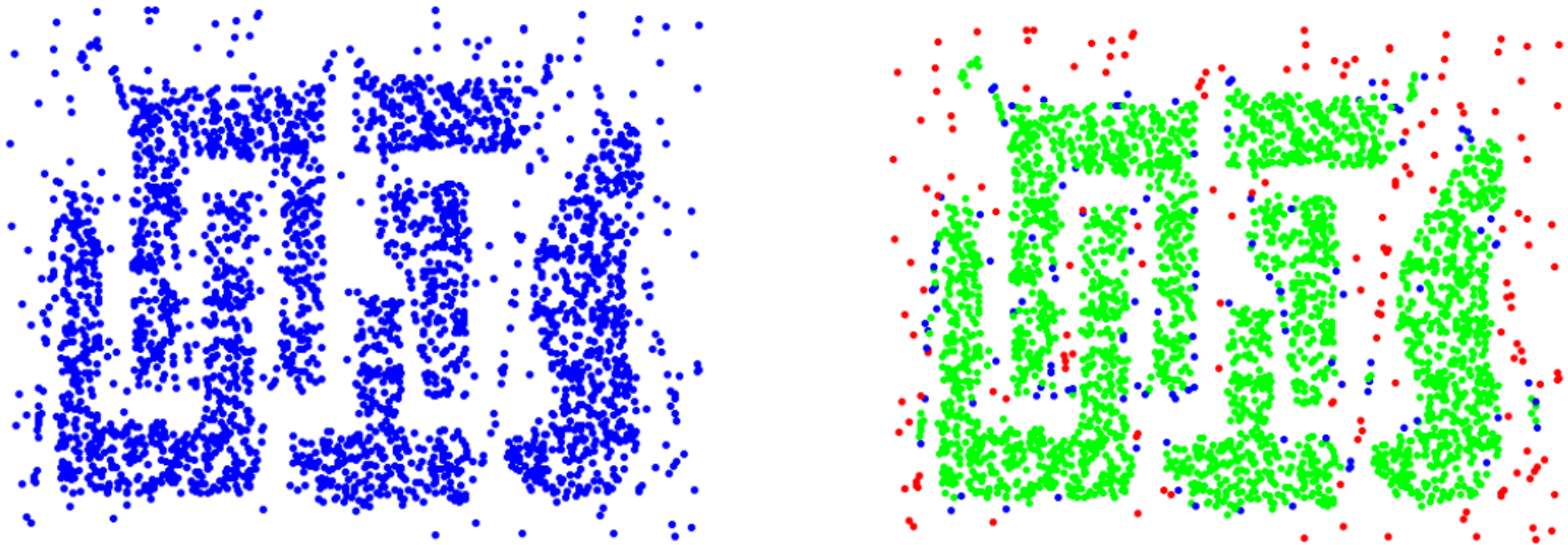
# DBSCAN

MinPts = 7



# DBSCAN

Ejemplo:



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


# DBSCAN


Algoritmo:

---

**Algorithm** DBSCAN algorithm.

---

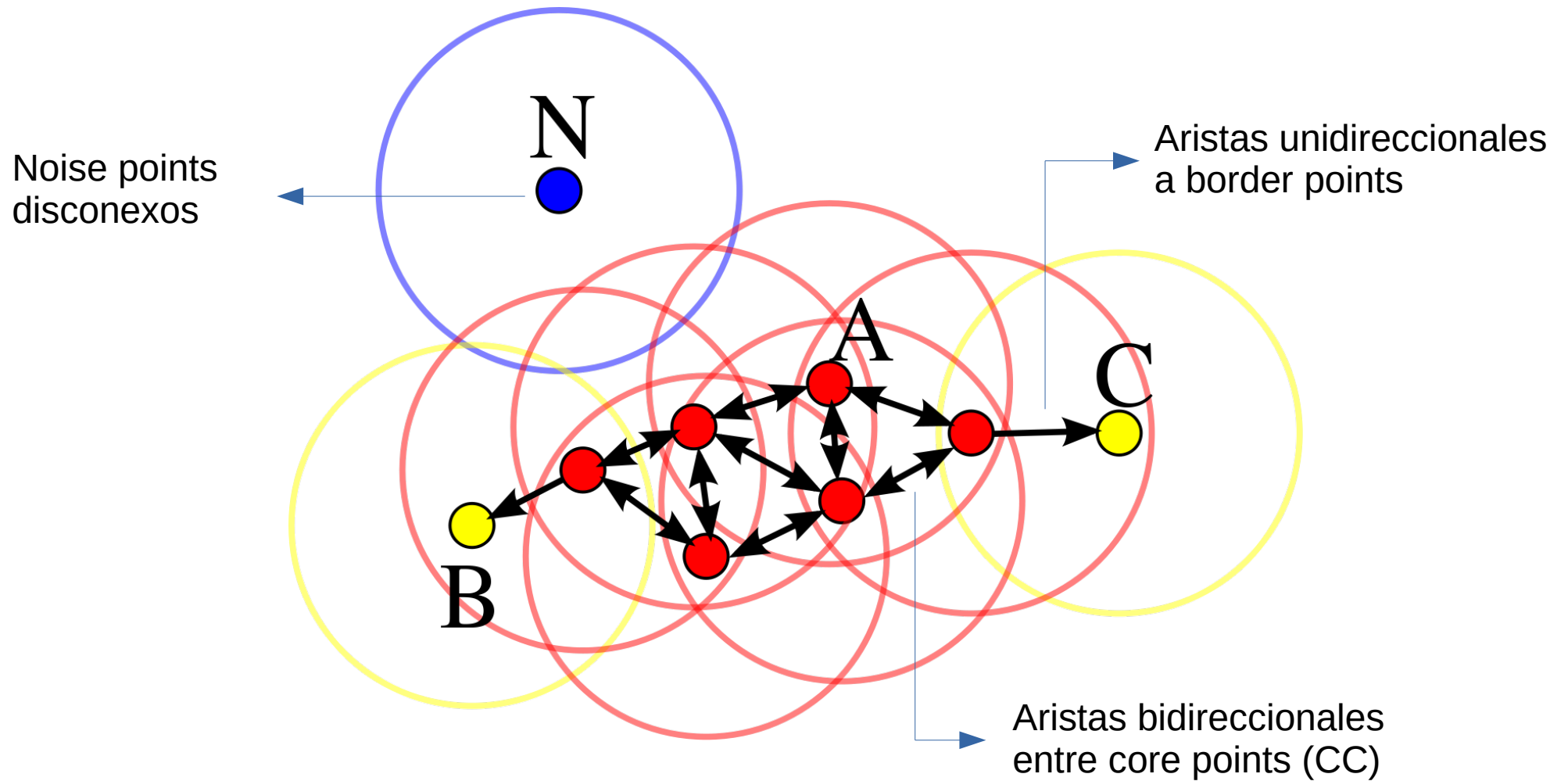
- 1: Label all points as core, border, or noise points.
  - 2: Eliminate noise points.
  - 3: Put an edge between all core points that are within  $Eps$  of each other. 
  - 4: Make each group of connected core points into a separate cluster.
  - 5: Assign each border point to one of the clusters of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



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

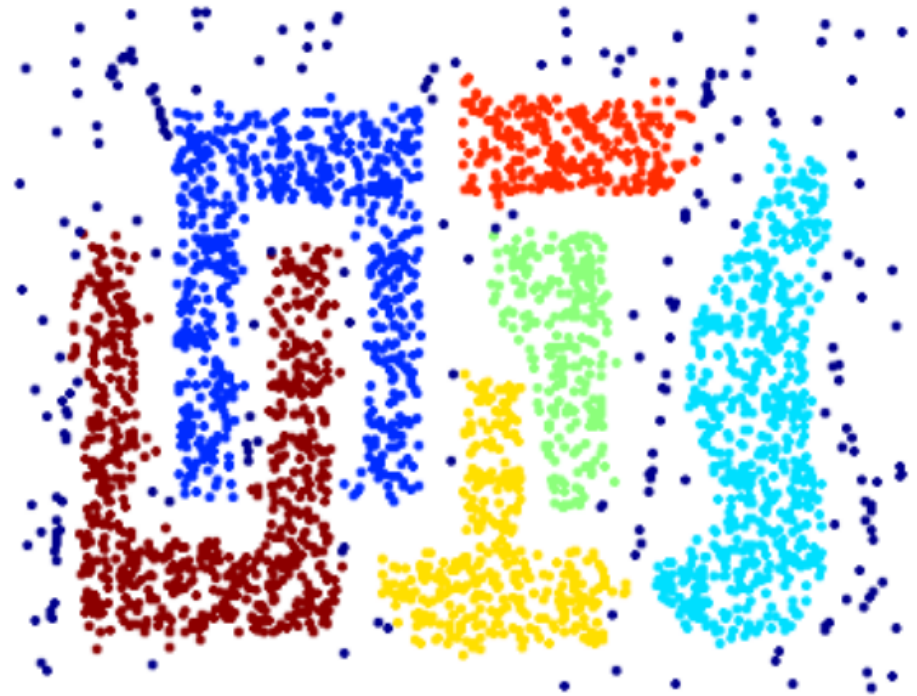
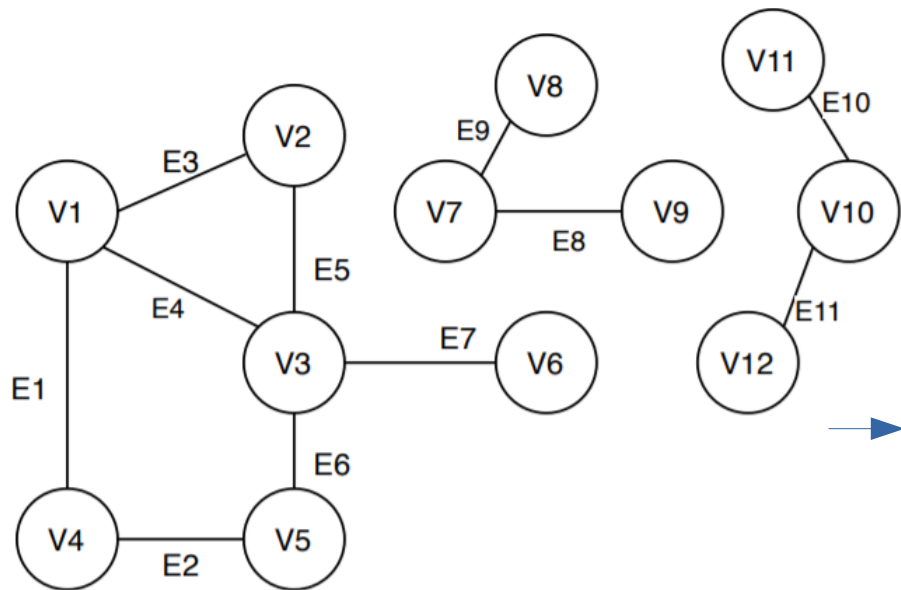
# DBSCAN

Grafo dirigido construido conectando core points y border points:



# DBSCAN

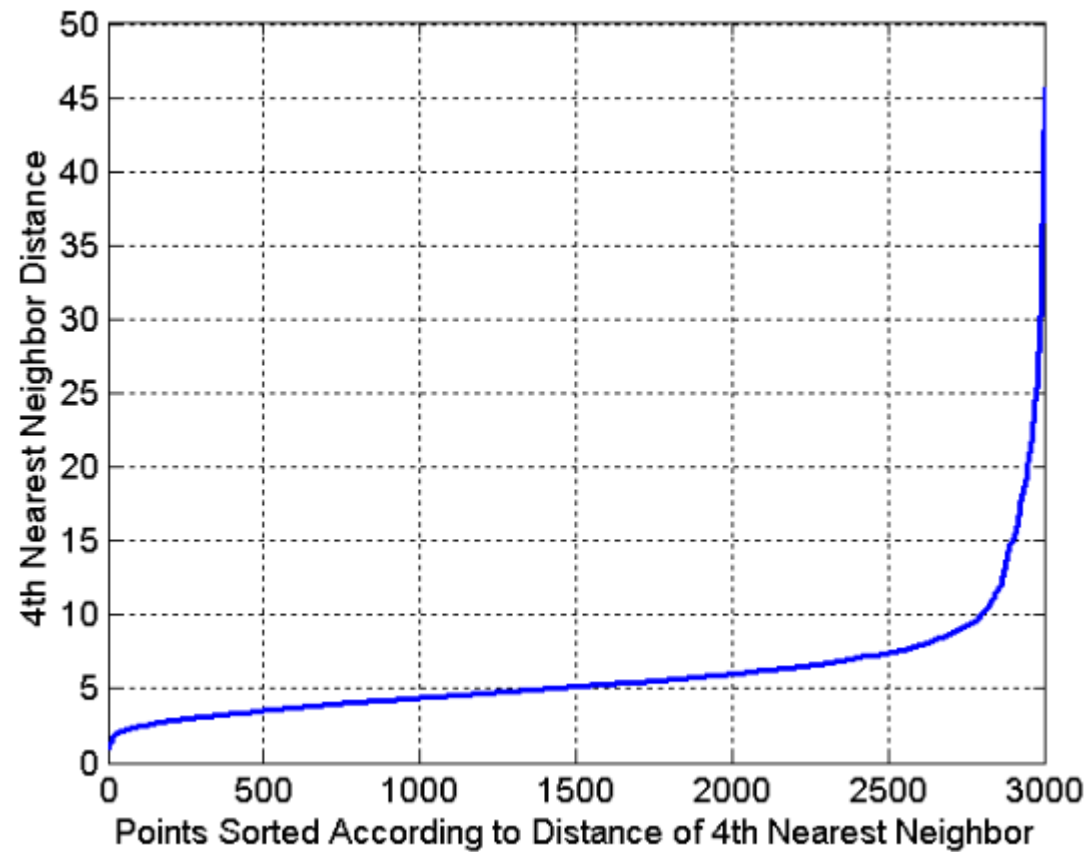
Componentes conexas en DBSCAN:



# DBSCAN

Sintonización del algoritmo

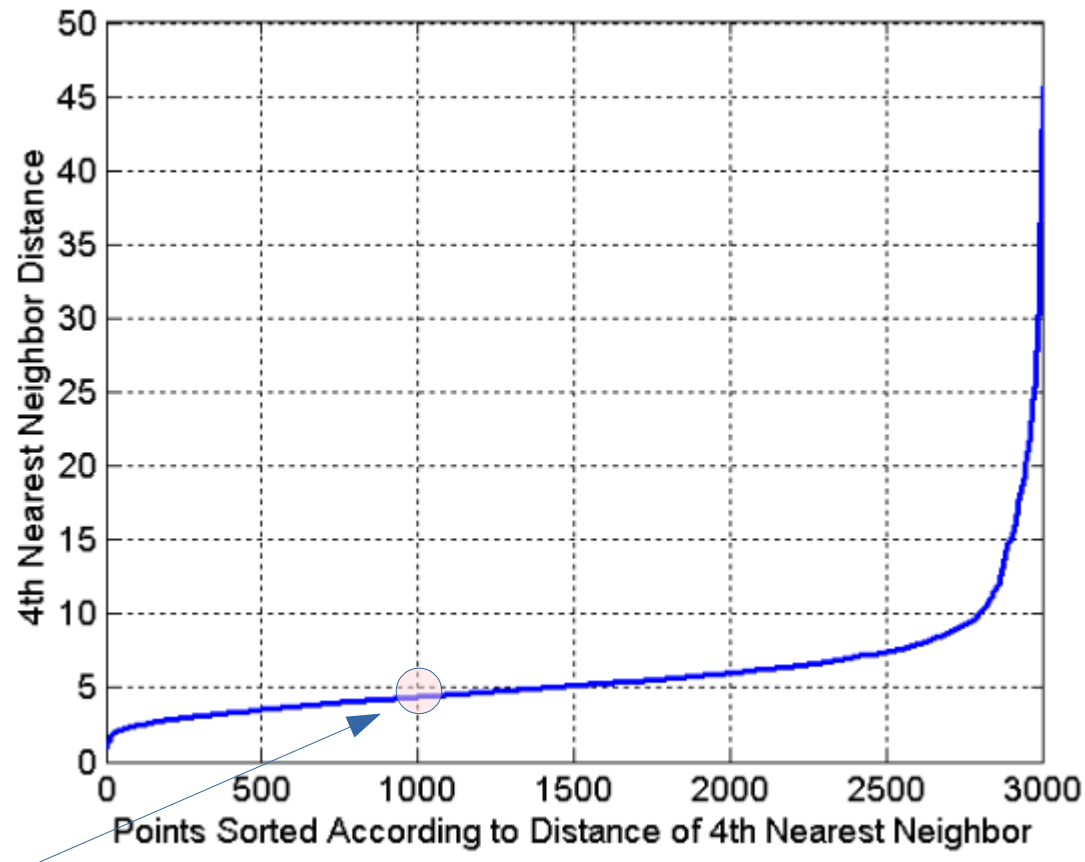
$k$ -dist plot ( $k=4$ ) ← MinPts (candidato)



# DBSCAN

## Sintonización del algoritmo

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



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

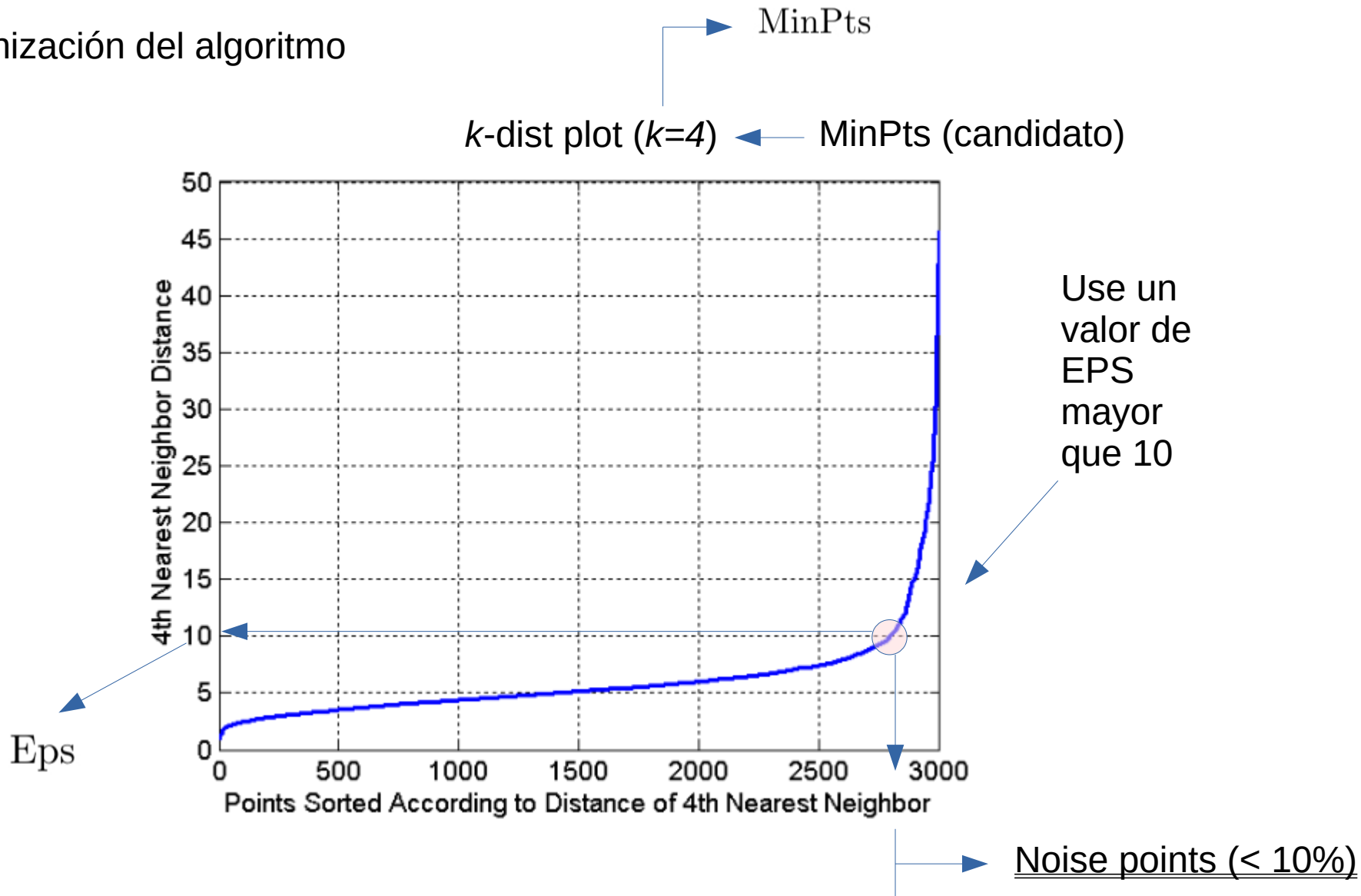
- UC - M. Mendoza -

→ Si  $EPS = 5$  y  $MinPts = 5 \rightarrow 1000$  core points

Notar que si aumento EPS, los clusters son menos densos

## DBSCAN

Sintonización del algoritmo





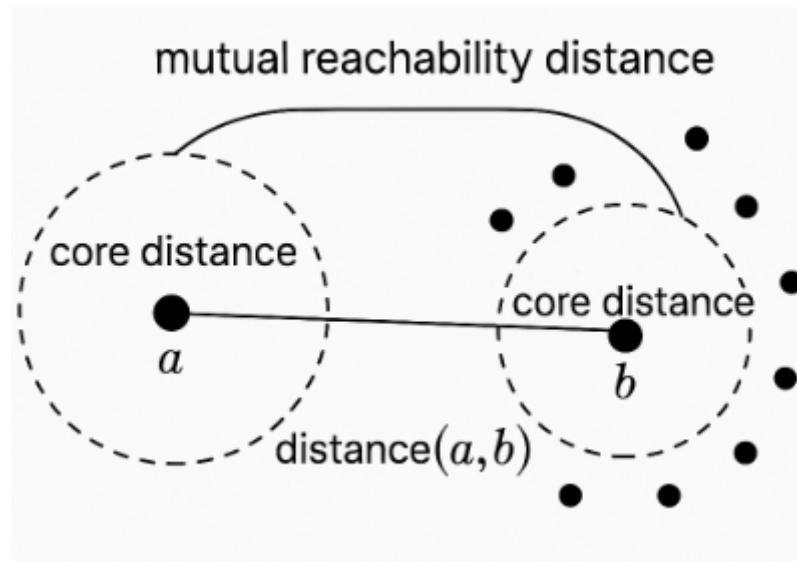
- HDBSCAN -

# HDBSCAN

Intuición: separar las regiones de baja y alta densidad en base a comparaciones de  $k$ -distances. Le denominan core distance (distancia al  $k$ -ésimo vecino más cercano).

Luego definen la mutual reachability distance entre dos puntos como:

$$d_{\text{mreach-}k}(a, b) = \max\{\text{core}_k(a), \text{core}_k(b), d(a, b)\}$$

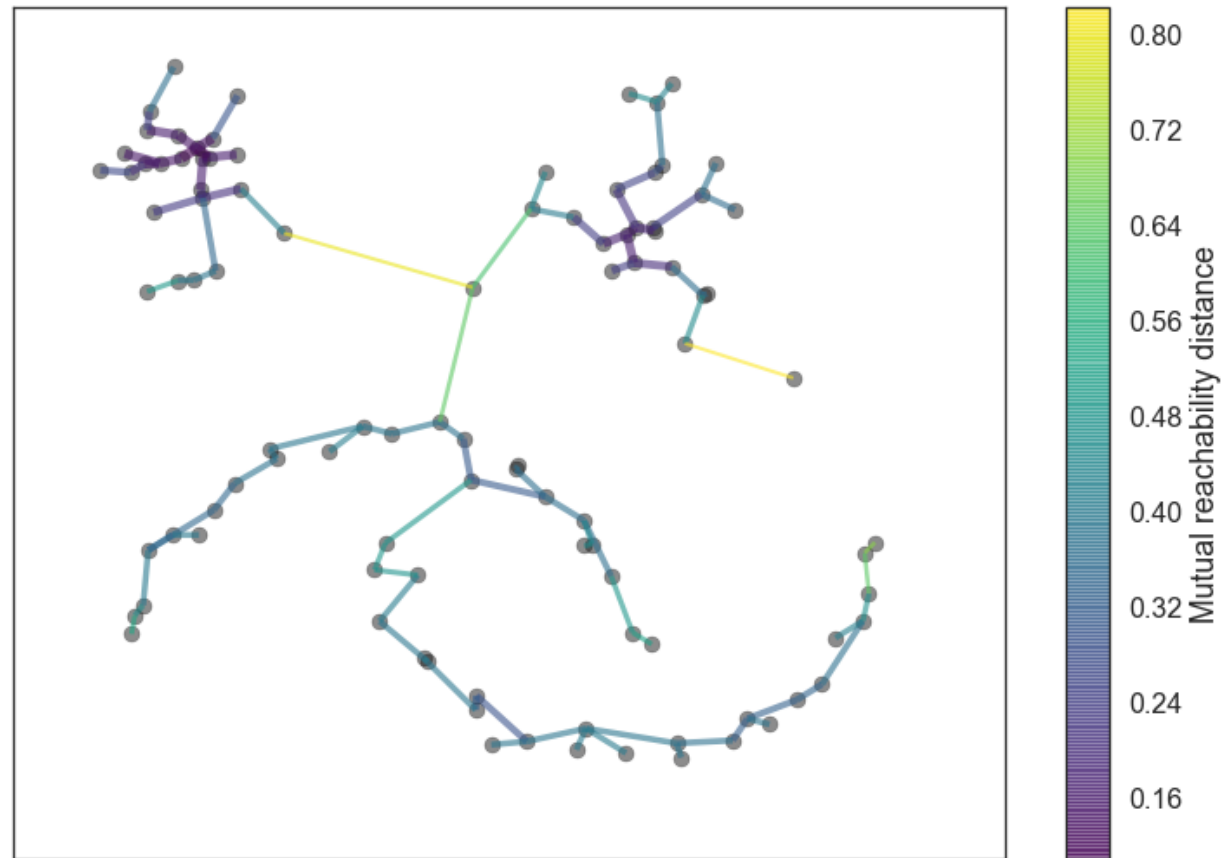


Si mido la distancia entre un outlier y un inlier, la distancia va a ser  $d(a, b)$ , más alta que las core distances.

Si mido la distancia entre dos inliers, la distancia va a estar dominada por las core distances, que son menores a la distancia directa.

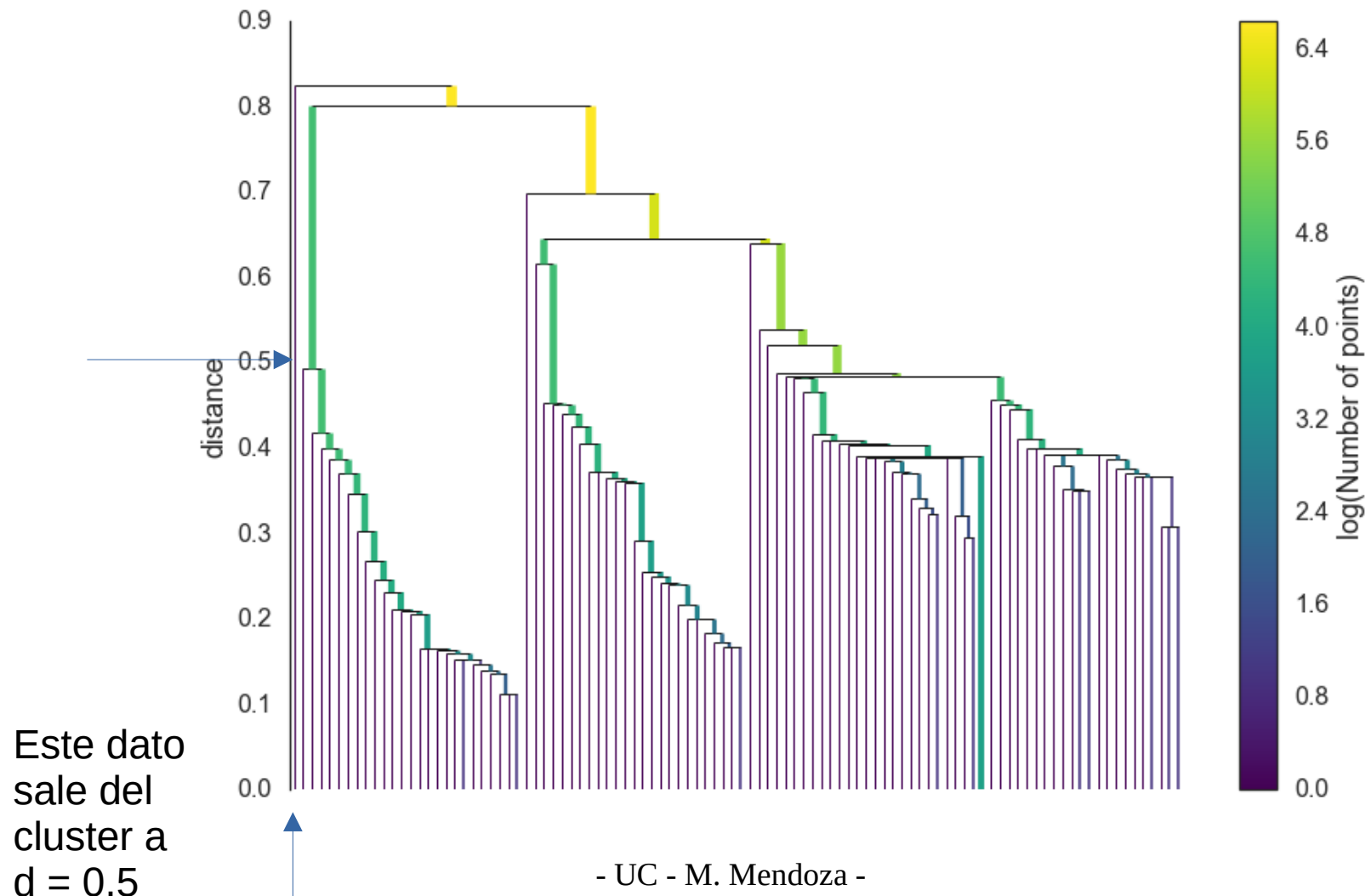
# HDBSCAN

- Vamos a construir un grafo de conectividad sobre la base de la mutual reachability distance.
- Para esto usamos el *minimum spanning tree* usando el algoritmo de Prim. El invariante del algoritmo de Prim es agregar la arista de menor peso que conecta al árbol actual un nodo que no está conectado.



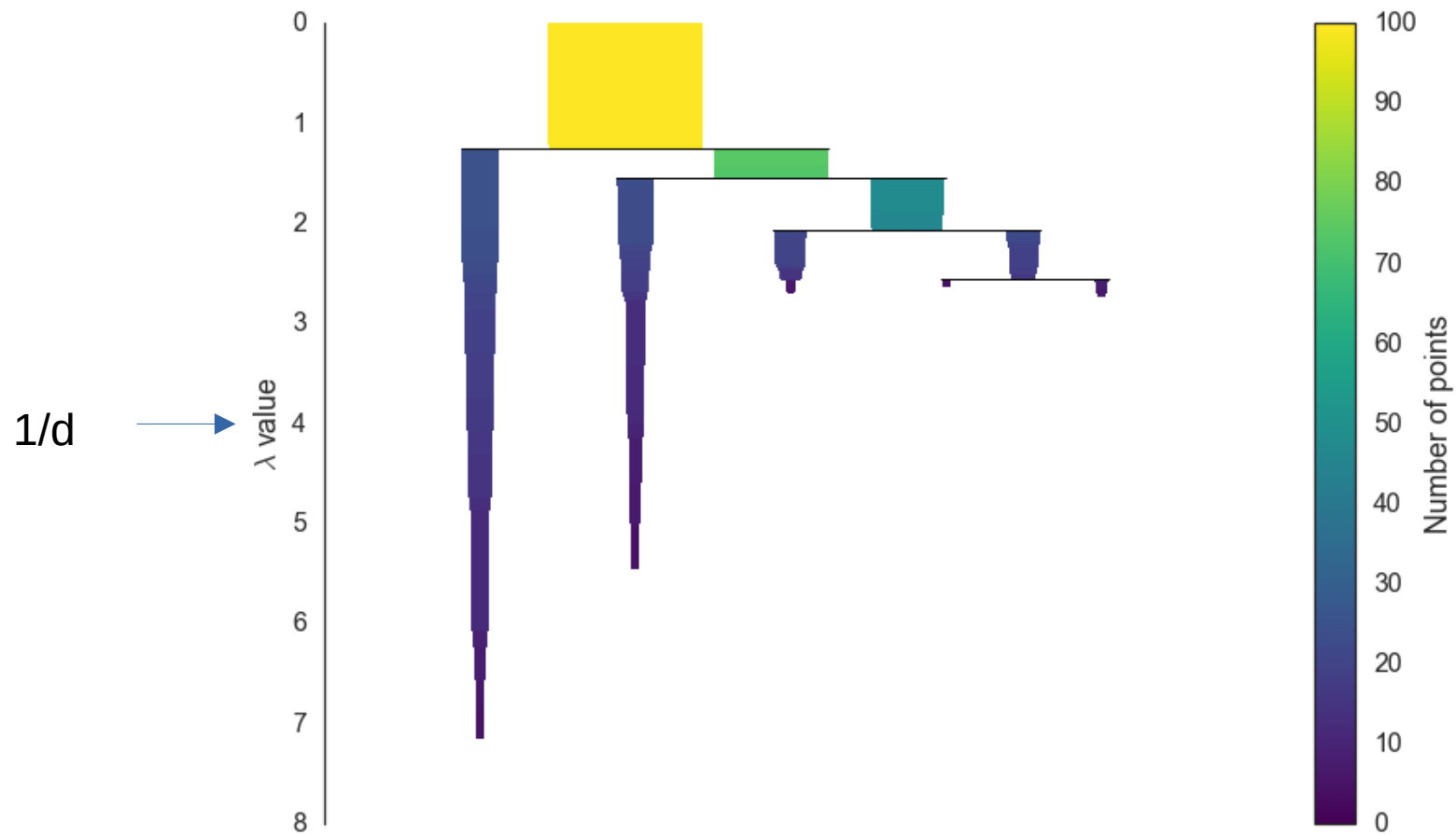
# HDBSCAN

- Construimos un dendrograma (bottom-up, según la mutual reachability distance).



# HDBSCAN

- Condensaremos bottom-up el dendrograma con una condición de **minimum cluster size**. Si un cluster tiene menos datos, se mezcla con su cluster padre (merge-up). Si un split produce dos clusters que cumplen con la condición de tamaño, el split se mantiene.



# HDBSCAN

Se usa el concepto de **cluster estable** para indicar cuando un cluster sobrevive a los splits definidos por un umbral de distancia en el dendrograma. Es inverso a la distancia, por lo que se define la variable lambda como  $1/d$ .

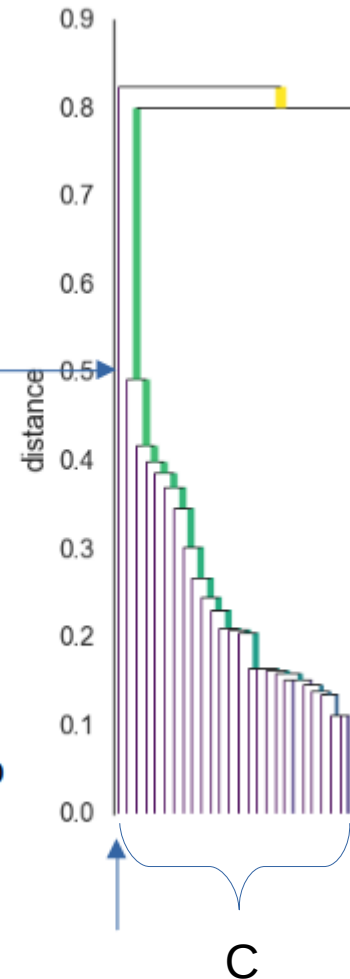
Hacemos un recorrido top-down del dendrograma. Vamos a tener un lambda de nacimiento del cluster, y en la medida que  $d$  disminuya (cortamos más abajo) en algún momento el cluster va a morir.

Para cada dato del cluster vamos a medir cuando sale del cluster (lambda del dato), y calcularemos la **estabilidad** del cluster según:

$$\text{Stability}(C) = \sum_{p \in C} (\lambda_{\text{death}}(p) - \lambda_{\text{birth}})$$

Aquí nace el cluster

Este dato sale del cluster a  $d = 0.5$



# HDBSCAN

**Los clusters estables son clusters reales:** Recorremos el dendrograma bottom-up. Si la suma de las estabilidades de los hijos es mayor que la de un cluster, definimos la estabilidad del cluster como la suma de las estabilidades de sus hijos. En otro caso, si el cluster es más estable que la suma de sus hijos, sacamos a los hijos del cluster.

Un cluster estable es robusto a  $\delta$

