KIBERNUM
IT Academy

Módulo 6
Clase 5

Aprendizaje de Máquina No Supervisado

Objetivos

- Utilizar los conceptos básicos de aprendizaje de máquinas no supervisado.
- Conocer los distintos tipos de algoritmos para agrupamiento jerárquico.
- Conocer sobre Dendogramas.
- Implementación en Python.
- Ventajas y desventajas de la clusterización jerárquica.

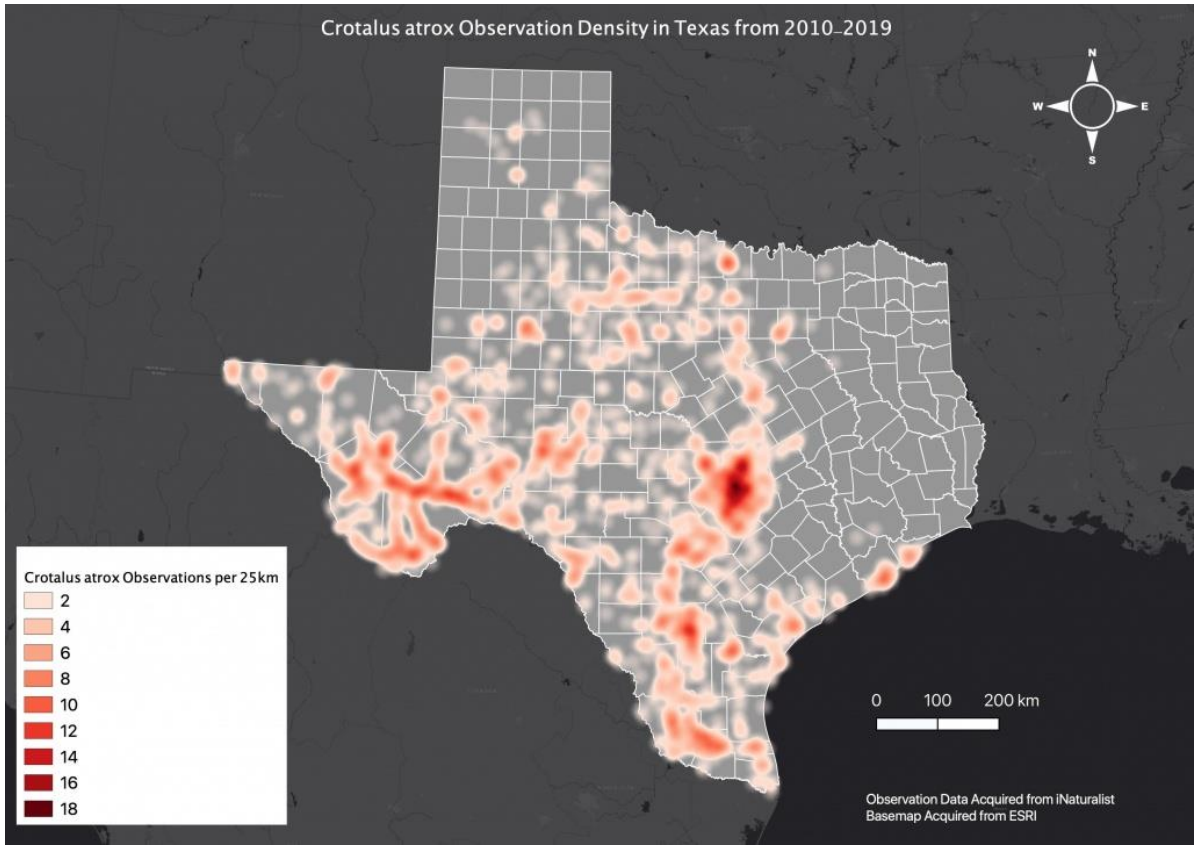
DBSCAN

Algoritmos de Clustering basados en Densidad

- El agrupamiento basado en la densidad se refiere a métodos de aprendizaje no supervisados que identifican grupos/conglomerados distintivos en los datos, basados en la idea de que **un conglomerado en el espacio de datos es una región contigua de alta densidad de puntos, separada de otros conglomerados por regiones contiguas de baja densidad de puntos**.

- **DBSCAN** (Clustering de aplicaciones con ruido basadas en densidad espacial) es un algoritmo base para el agrupamiento basado en la densidad. Puede descubrir grupos de diferentes formas y tamaños a partir de una gran cantidad de datos, que contienen ruido y valores atípicos.

Intuición



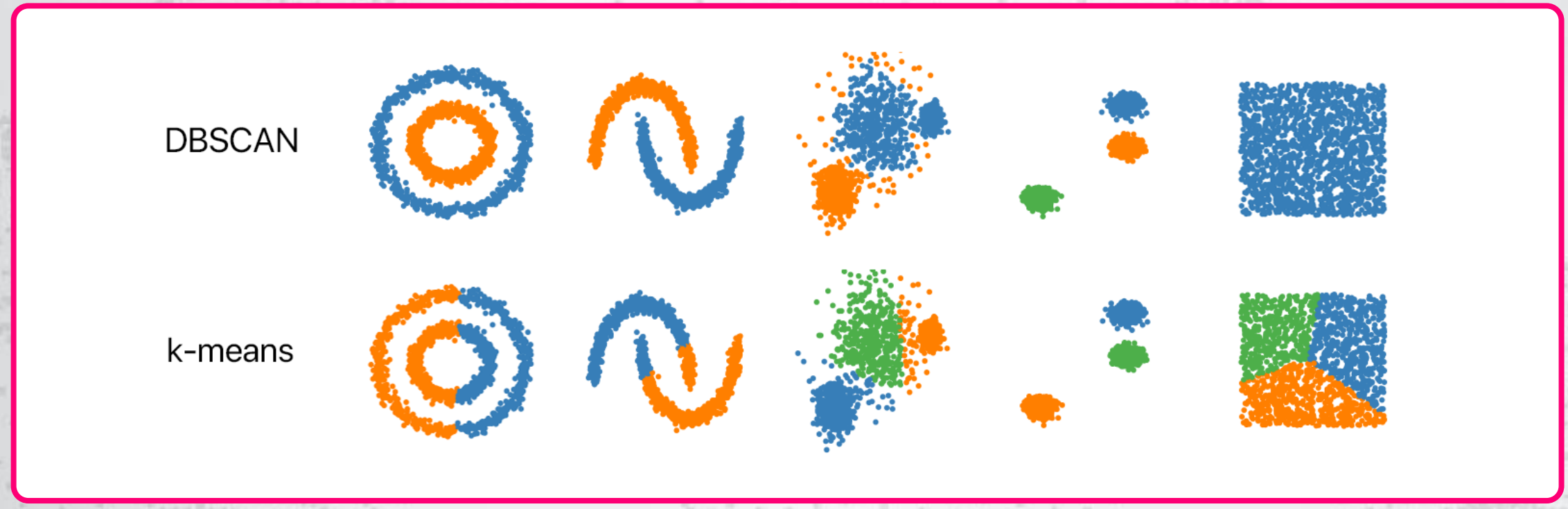
El siguiente diagrama de calor, muestra la densidad poblacional del estado de Texas.

Es razonable pensar que aquellas áreas de mayor densidad corresponden a ciudades más grandes, con edificios, con muchos residentes y turistas. Por otra parte, las zonas menos densas pueden corresponder a ciudades más pequeñas, pequeñas villas o poblados, que se encuentran fuera de las ciudades y que son menos densas.

Entonces, definiendo ciertos criterios de densidad, se podrían identificar dónde se encuentran ciudades y pueblos.

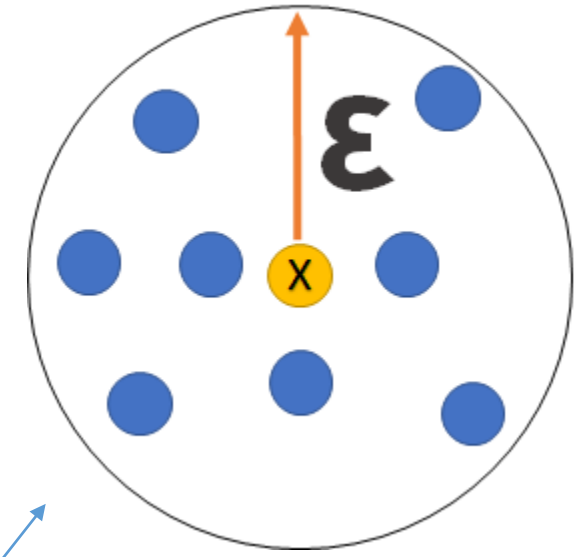
DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) es un algoritmo de clustering que se utiliza para agrupar datos en grupos (clústeres) basados en su densidad. A diferencia de otros algoritmos de clustering, DBSCAN puede identificar clústeres de forma irregular y también, puede detectar puntos de datos que no pertenecen a ningún cluster (ruido).

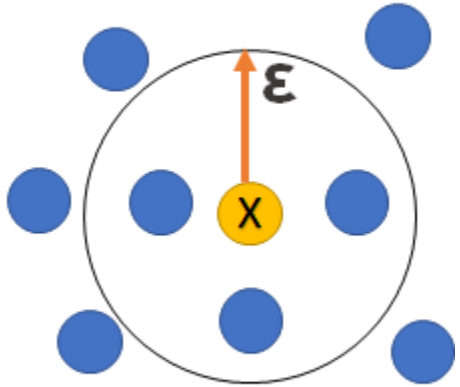


Conceptos Básicos de DBSCAN

eps (*ε*): una medida de distancia que define el tamaño de un vecindario, en donde épsilon corresponde al radio.



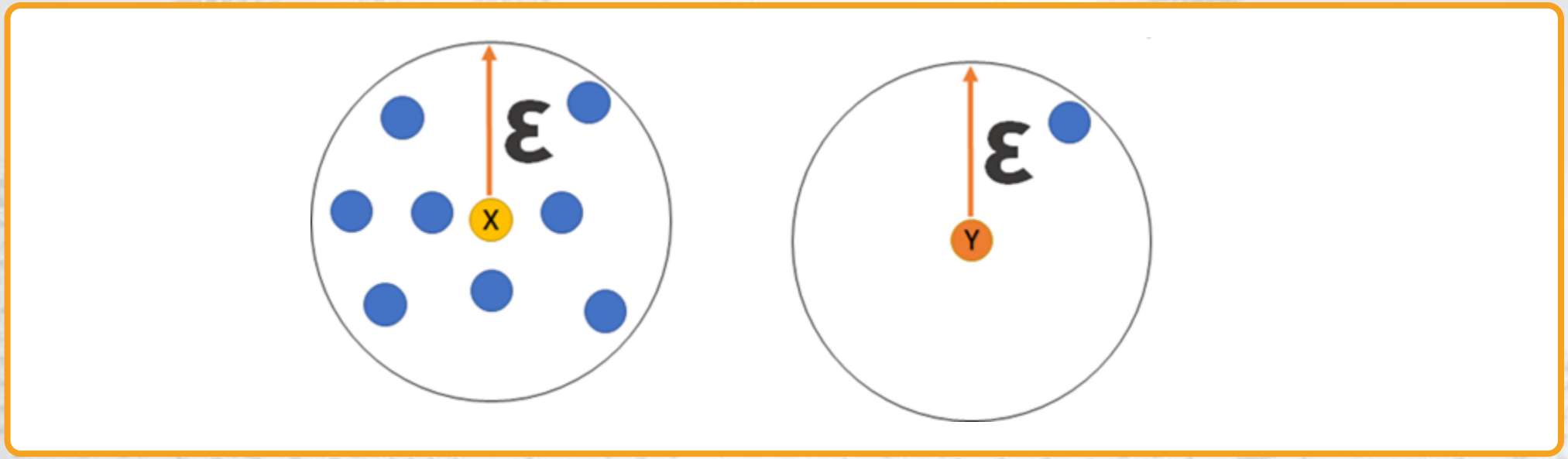
En este ejemplo, X tiene un vecindario compuesto por 9 vecinos para un valor dado de épsilon.



En este ejemplo, X tiene un vecindario compuesto por 4 vecinos para un valor dado de épsilon más pequeño.

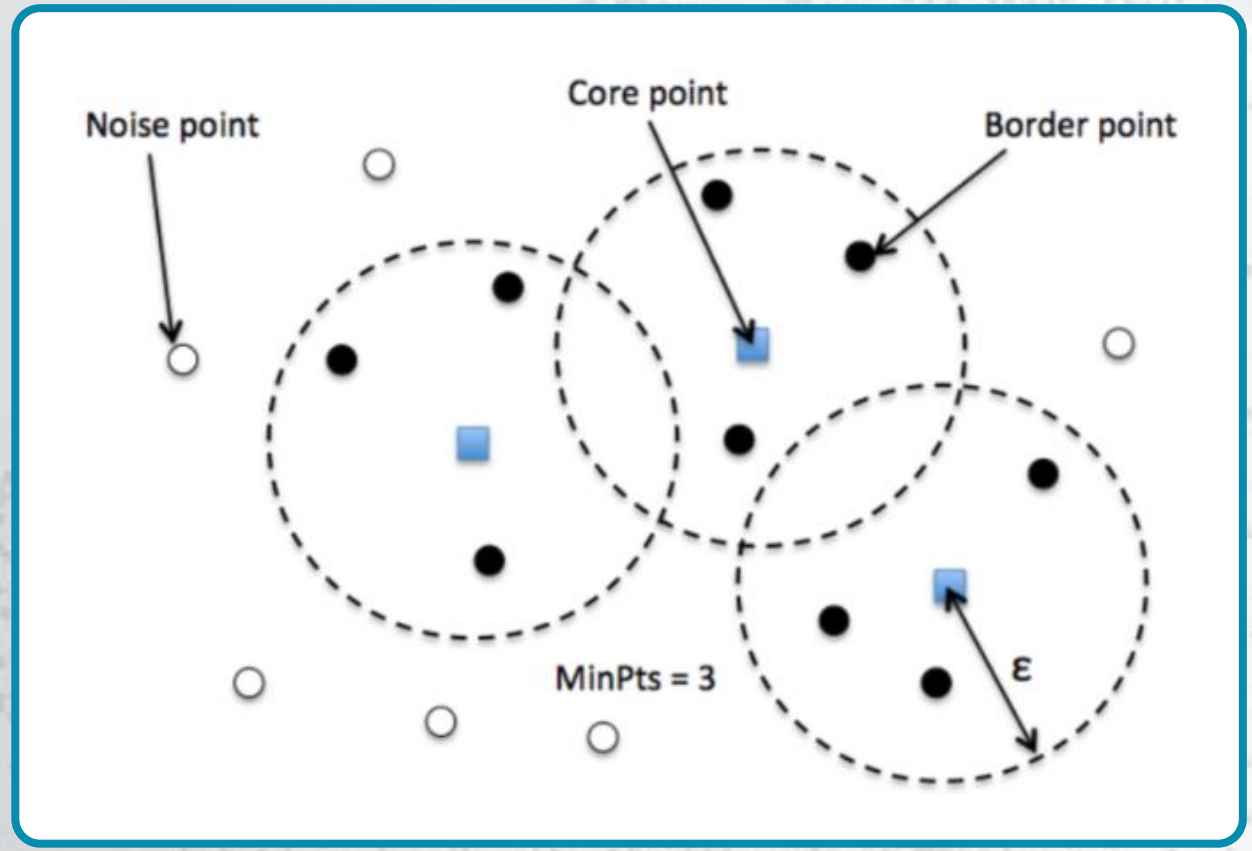
Conceptos Básicos de DBSCAN

Para poder decidir si un vecindario es considerado denso, se define el parámetro **minPts**. Este parámetro, corresponde a la cantidad mínima de puntos (umbral) que un vecindario debe tener para ser considerado como **denso**.



Supongamos que definimos épsilon y minPts = 5. En este caso, el vecindario del punto X sería considerado como un cluster y el punto Y junto a sus vecinos serían considerados como outlaiers o ruido. Esto, debido a que el vecindario de Y no es lo suficientemente denso.

Conceptos Básicos de DBSCAN

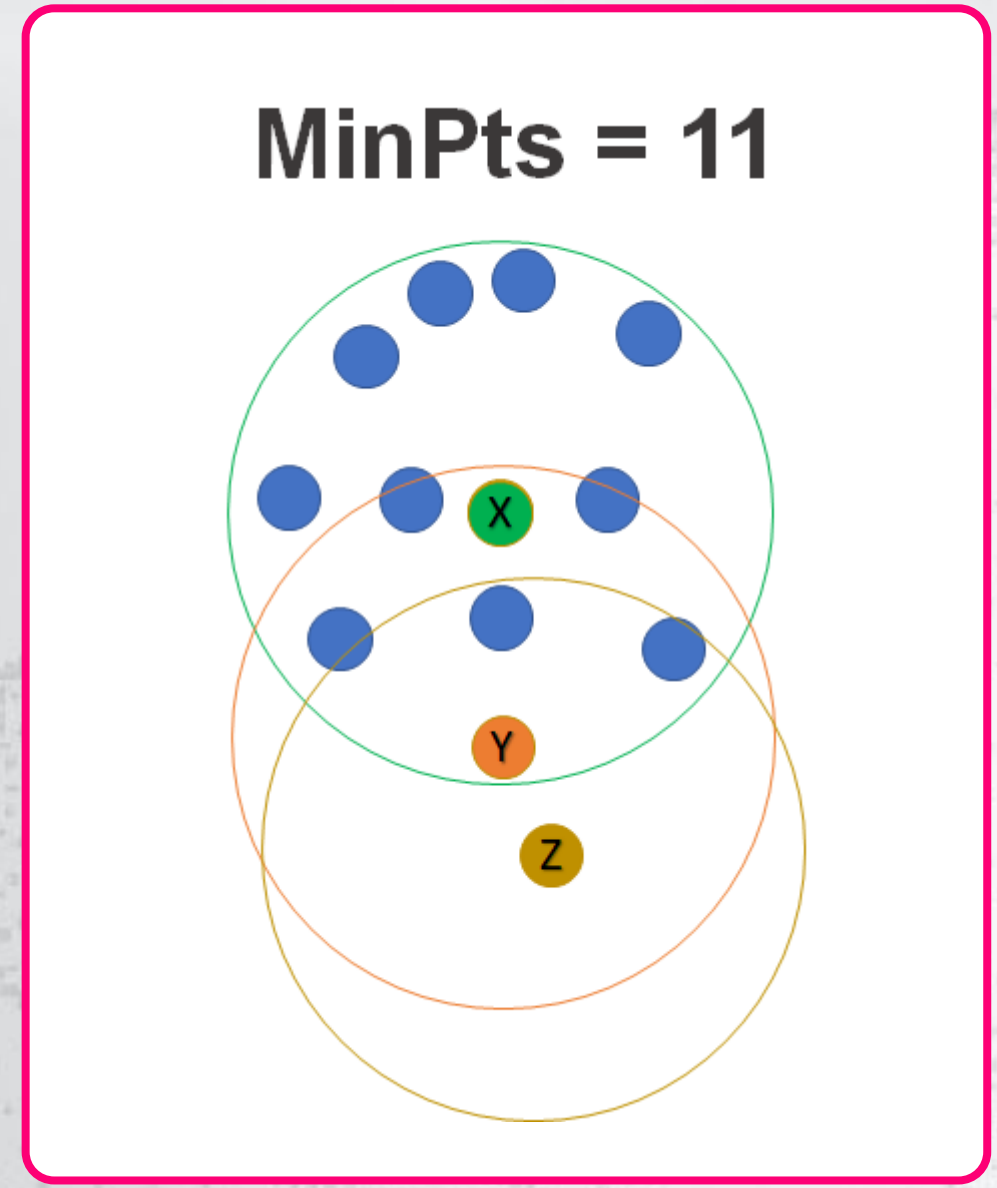


Core point: es un punto que en su vecindario de radio ε tiene al menos minPts puntos vecinos.

Border point: es un punto que en su vecindario de radio ε tiene no alcanza a minPts puntos, pero este punto pertenece al vecindario de un punto core.

Noise point: es un punto que no es Core ni Border.

Conceptos Básicos de DBSCAN



Ahora que sabemos que los puntos **core** y los puntos **border** estarán en un cluster, ¿cómo hace DBSCAN para saber qué punto va en qué clúster? Para eso, incorporaremos dos nuevos conceptos:

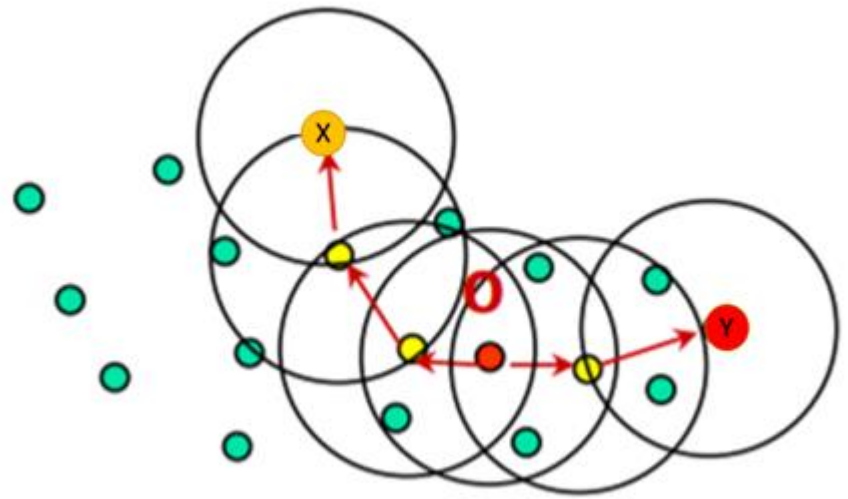
Densidad directamente alcanzable
Un punto Y es directamente alcanzable desde el punto X si se cumple lo siguiente:

- El punto Y pertenece al ε -vecindario del punto X
- Punto X es un punto core

En este ejemplo, el punto Y es directamente alcanzable por X. Pero note que el punto X no es directamente alcanzable por Y, puesto que el punto Y no es un punto core.

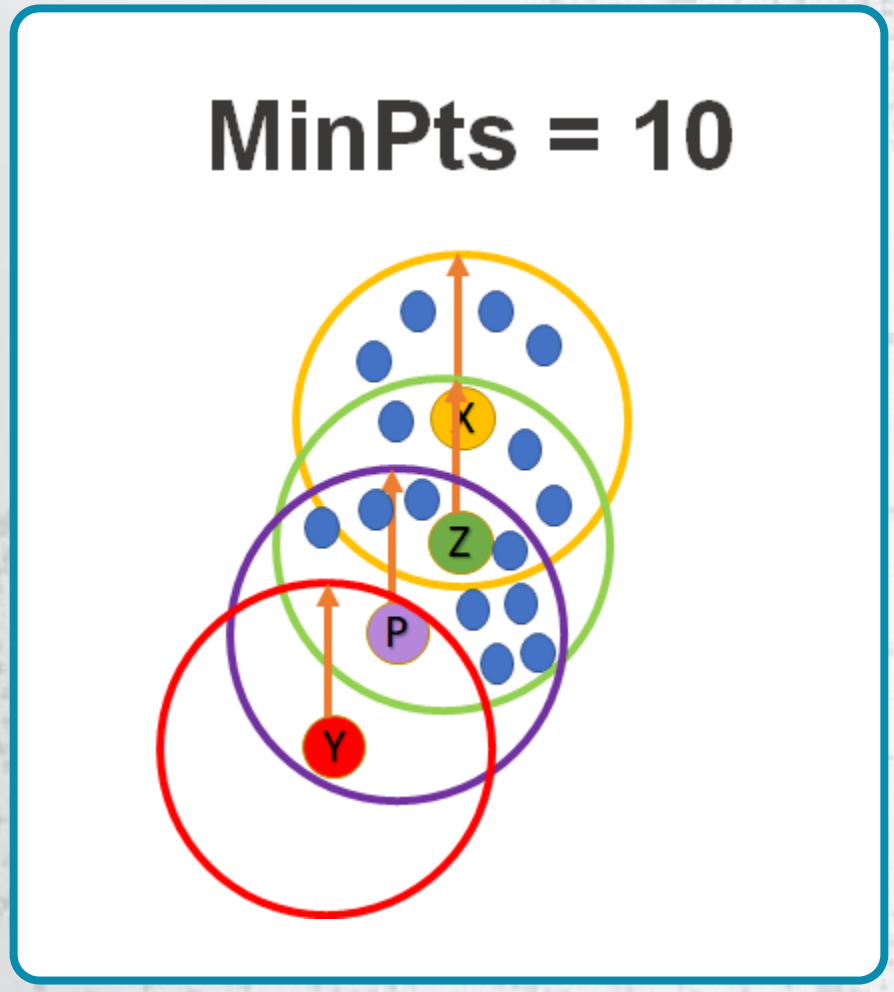
Conceptos Básicos de DBSCAN

Conexión densa
Un punto X es conectado densamente a un punto Y, si hay un punto O en el cual tanto X como Y son densamente alcanzables.



En esta figura, el punto X e Y son densamente alcanzables desde el punto O. Por lo tanto, el **punto X está conectado densamente con el punto Y**.

Conceptos Básicos de DBSCAN



Cluster basado en densidad
Un cluster C, es un grupo de puntos no vacíos dado que:

El punto X está en C y el punto Y es densamente alcanzable desde X, en cuyo caso, Y estaría en C
Todos los puntos en C están densamente conectados unos a otros.

En esta figura, los puntos X, Y, Z, P e Y están en el mismo clúster.

Algoritmo DBSCAN

```
DBSCAN(dataset, eps, MinPts){
# cluster index
C = 1
for each unvisited point p in dataset {
          mark p as visited
          # find neighbors
          Neighbors N = find the neighboring points of p

          if |N|>=MinPts:
              N = N U N'
              if p' is not a member of any cluster:
                  add p' to cluster C
}
```

1. Encuentre todos los puntos vecinos dentro de eps e identifique los puntos centrales o visitados con más de MinPts vecinos.

2. Para cada punto central, si aún no está asignado a un clúster, cree un nuevo clúster.

3. Encuentre recursivamente todos sus puntos conectados por densidad y asígnelos al mismo grupo que el punto central. Se dice que un punto a y b están conectados por densidad si existe un punto c que tiene un número suficiente de puntos en sus vecinos y ambos puntos a y b están dentro de la distancia eps. Este es un proceso de encadenamiento. Entonces, si b es vecino de c, c es vecino de d, d es vecino de e, que a su vez es vecino de a implica que b es vecino de a.

4. Iterar a través de los puntos no visitados restantes en el conjunto de datos. Aquellos puntos que no pertenecen a ningún clúster son ruido.

Elección de ε y minPts

La elección de los valores de eps y minPts en el algoritmo DBSCAN depende en gran medida del conjunto de datos específico que se está utilizando y del problema que se está tratando de resolver. En general, se pueden seguir los siguientes pasos para determinar los valores óptimos de estos parámetros:

- **Analizar los datos:** Es importante tener un buen conocimiento del conjunto de datos antes de aplicar DBSCAN. Esto incluye visualizar los datos y comprender su distribución, densidad y características.

- **Selección de minPts:** La elección de minPts depende de la densidad de los datos. Si los datos son muy densos, es recomendable utilizar un valor alto de minPts. Por otro lado, si los datos son muy dispersos, se recomienda un valor bajo de minPts.

Elección de ε y minPts

- **Selección de eps**: La elección de eps es más complicada y generalmente requiere un enfoque de prueba y error. Una buena estrategia es probar con diferentes valores de eps y observar cómo cambia el número de grupos y la calidad de los mismos. Se puede utilizar la función k-distance para ayudar a determinar el valor adecuado de eps. Esta función devuelve la distancia de cada punto a su k-ésimo vecino más cercano. Al trazar los puntos en orden ascendente de esta distancia, se puede identificar un punto en el que la distancia aumenta significativamente, lo que sugiere un cambio en la densidad de los datos. Este punto se puede utilizar como valor de eps.

- **Evaluación de los resultados**: Es importante evaluar los resultados de DBSCAN para determinar si los valores elegidos para eps y minPts son apropiados. Si los grupos resultantes no son coherentes o si hay un número inesperadamente alto o bajo de grupos, puede ser necesario ajustar los valores de eps y minPts y volver a ejecutar el algoritmo.

En resumen, la selección de los valores de eps y minPts en DBSCAN es una tarea que requiere cierta experimentación y ajuste fino para obtener los mejores resultados posibles.

Ventajas y Desventajas

Ventajas

- No requiere especificar el número de clústeres de antemano: DBSCAN determina automáticamente el número de clústeres.
- Es robusto ante ruido: DBSCAN puede identificar y separar puntos de datos que no pertenecen a ningún clúster. Esto permite que el algoritmo sea más robusto en presencia de datos ruidosos o atípicos.
- Puede encontrar clústeres de formas irregulares, a diferencia de otros algoritmos.

Desventajas

- Set de datos con densidades variables podrían ser problemáticos.
- La elección de los parámetros de entrada (ε y MinPts) puede ser dificultosa y altamente sensible.
- Computacionalmente complejo cuando la dimensionalidad es alta, pues toma $O(n^2)$.

Dudas y consultas

Gracias

KIBERNUM