DBSCAN

Nicolás Kossacoff

Noviembre 2024

1. Introducción

La idea detrás de **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) es que los clusters están definidos por regiones de alta concentración (o densidad de puntos) en el espacio, separados por regiones de baja concentración. Esto nos permite utilizar DBSCAN para encontrar clusters de diferentes formas.

Este algoritmo surge como una alternativa a los tipos de algoritmos más utilizados:

- **Algoritmos de partición.** Estos algoritmos definen una partición del conjunto de datos, $\mathcal{C} = \{C_1, \ldots, C_K\}$, donde cada sub-conjunto es un cluster. Una desventaja de estos algoritmos es que hay que definir de antemano el número K de clusters, lo cual no siempre es una tarea trivial.

  Otra desventaja de estos algoritmos es que devuelven clusters convexos, lo cual es muy restrictivo.

- **Algoritmos jerárquicos.** Estos algoritmos construyen una descomposición jerárquica de nuestros datos, donde cada nodo es un cluster. No es necesario definir la cantidad de clusters de antemano, pero si hay que definir una condición de finalización, lo cual no siempre es una tarea sencilla.

2. Definiciones

Antes de analizar como funciona el algoritmo, debemos definir algunos conceptos importantes.

Definition (ε-vecindad)**.** *La ε-vecindad de un punto p se define como:*

$$N_\varepsilon(p) = \{q \in D : d(p, q) < \varepsilon\}$$

Entonces, la ε-vecindad de un punto p son todos los puntos que se encuentran a una distancia menor a ε.

Hay que tener en cuenta que la forma de la vecindad depende de la distancia utilizada (e.g., si d es la distancia Manhattan, la forma de la vecindad es rectangular).

Definition ($MinPts$)**.** *Es un número natural que funciona como umbral.*

Dados ε y $MinPts$, definimos tres clases de puntos en nuestros datos.

- **Punto núcleo.** Decimos que el punto p es un punto **núcleo** si en su entorno $N_\varepsilon(p)$ contiene, al menos, $MinPts$ puntos.
- **Punto borde.** Decimos que el punto q es un punto **borde** si en su entorno $N_\varepsilon(q)$ contiene menos de $MinPts$ puntos y existe un punto núcleo p tal que $q \in N_\varepsilon(p)$. Entonces, q no es un punto núcleo, pero sí está contenido en el entorno de uno.
- **Ruido.** Un punto o es considerado ruido si no cumple con ninguna de las condiciones anteriores.

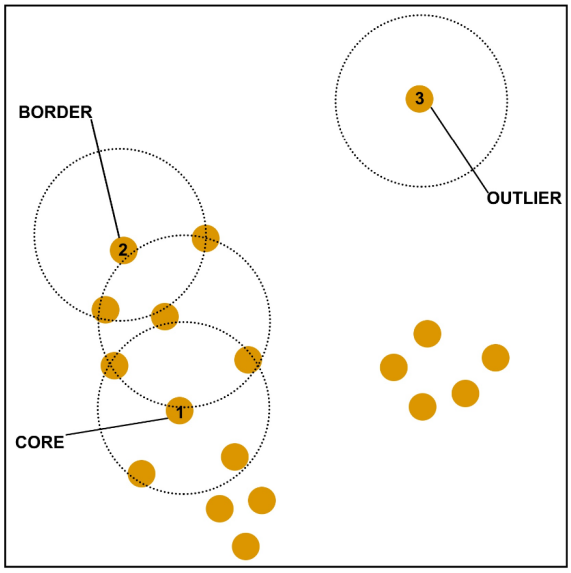

Figura 1: $MinPts = 5$

Entonces, no todos los puntos dentro de un cluster tienen la misma concentración de puntos en sus entornos. Como vimos antes, los puntos bordes tienen una menor concentración que los puntos núcleo. Esto quiere decir que, si agrupamos los puntos mirando únicamente la cantidad de puntos que tienen en sus entornos, para poder incluir a todos los puntos bordes, tendríamos que fijar un $MinPts$ muy chico. Esto podría traer problemas cuando hay mucho ruido.

Dicho esto, para que un punto q pertenezca al cluster C, se necesita un punto $p \in C$ cuyo entorno contenga al menos $MinPts$ (p es un punto núcleo) y que $q \in N_\varepsilon(p)$. En ese caso decimos que q es **directamente alcanzable mediante densidad** (directamente alcanzable a partir de ahora) desde el punto p.

Esta propiedad no es simétrica. Si q no es un punto núcleo, entonces no es cierto que el punto p sea directamente alcanzable desde el punto q.

De igual manera podemos decir que un punto p es **alcanzable mediante densidad** desde el punto q si existe una sucesión de puntos, $\{p_1, \ldots, p_n\}$ tales que $p_1 = p$, $p_n = q$ y p_{i+1} es directamente alcanzable desde p_i. En la Figura 2 podemos observar como la sucesión de puntos nos permiten alcanzar p desde q.

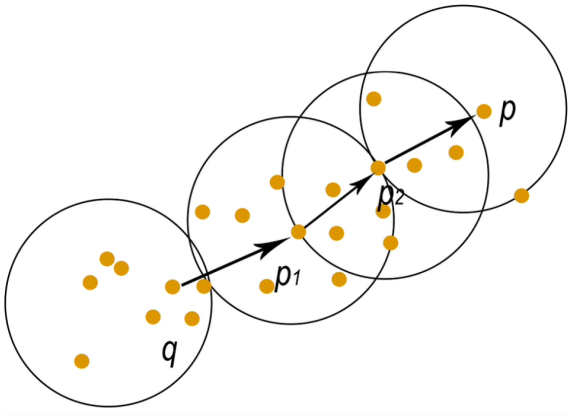

Figura 2: $MinPts = 7$

Puede ocurrir que dos puntos bordes no sean alcanzables mediante densidad. En esos casos decimos que p está **conectado mediante densidad** a un punto q si existe un punto o tal que p y q son alcanzables desde o. La Figura 3 nos muestra como dos puntos bordes están conectados mediante densidad.

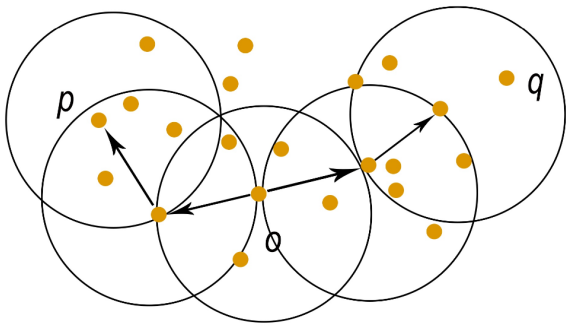

Figura 3: $MinPts = 7$

Finalmente, definimos un **cluster** C como un sub-conjunto de nuestros datos D que satisface dos condiciones:

- **Maximalidad.** Para cualquier par de puntos p y q se cumple que, si $p \in C$ y q es alcanzable mediante densidad desde p, entonces $q \in C$.
- **Conectividad.** Para cualquier par de puntos $p, q \in C$, se cumple que p está conectado mediante densidad a q.

Definimos como **ruido** al conjunto de puntos que no pertenecen a ningún cluster, es decir, aquellos puntos que no cumplen con las condiciones de maximalidad y conectividad.

3. Algoritmo

Dados ε y $MinPts$, la implementación del algoritmo se puede resumir en los siguientes pasos:

1. Asignar a cada punto su correspondiente categoría: núcleo, borde o ruido.
2. Eliminar los puntos que son categorizados como ruido.
3. Juntar los puntos núcleo que son alcanzables en un cluster.
4. Juntar los puntos bordes y asignarlos a su correspondiente cluster.

3.1. Selección de parámetros

Existe un método heurístico que nos permite seleccionar el valor de ε y $MinPts$. Para cada observación $i \in \{1, \ldots, n\}$ calculamos la distancia entre la observación y el k-ésimo vecino más cercano. Luego, ordenamos las observaciones según su distancia respecto a su vecino más cercano, tal como podemos observar en la Figura 4.

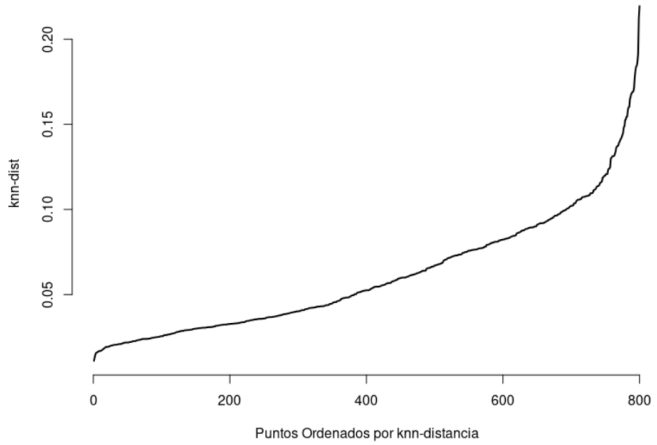

Figura 4: Distancia entre cada observación y su k vecino más cercano.

El corte lo hacemos donde cambia la pendiente. Los puntos que tienen una distancia mayor a la del corte son considerados ruido, el resto son observaciones que pertenecen a algún cluster.

Finalmente, elegimos $\varepsilon = k - dist^*$, donde $k - dist^*$ es la distancia donde cambia la pendiente, y $MinPts = k$.