OPTICS

OPTICS (Ordering Points To Identify the Clustering Structure), es otro algoritmo de clustering utilizado en mineria de datos y análisis de datos para descubrir patrones y estructuras en conjuntos de datos; siendo su objetivo principal descubrir grupos de puntos que están densamente agrupados en el espacio de características. Fue propuesto como una mejora del algoritmo DBSCAN, dado que este tiene problemas con las fronteras.

El algoritmo OPTICS comienza identificando los puntos centrales (core points) en el conjunto de datos (llamado `minPts`) dentro de un radio específico (llamado `eps`). Dado que una de sus limitaciones es la elección adecuada de estos parámetros, ya que son cruciales para obtener resultados óptimos, a continuación se optimiza su búsqueda:

Búsqueda de los parámetros óptimos

Optimización de la búsqueda de parámetros para épsilon y minPts en Optics:

Primeramente, definimos los valores que se van a probar para `eps` y `minPts`, creando una cuadrícula de parámetros y, seguidamente, se establece el número de núcleos (cores) a utilizar para la optimización en paralelo, que se calcula automáticamente.

Función para ejecutar OPTICS con una combinación de parámetros y calcular el coeficiente de silueta:

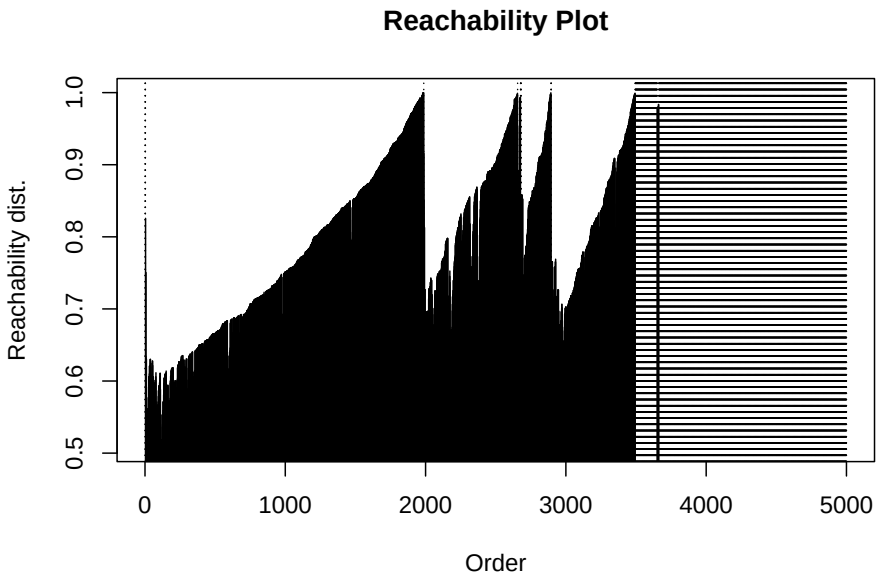
Cuadro 24: Obtención de los Parámetros Óptimos

	eps	minPts
20	1	10

Como vemos, después del proceso iterativo, la combinación de resultados más óptima ha sido un radio (`eps`) de 1 con un mínimo de 10 puntos (`minPts`).

Así mismo, a continuación creamos el modelo OPTICS con los parámetros óptimos encontrados y observamos su reachability plot:

Figura 63: Reachability Plot



El gráfico de reachability (alcance) que acabamos de generar, es una herramienta para visualizar la estructura de clústeres identificados.

En los gráficos de reachability, cada punto representa un objeto de datos y la altura de la curva indica la distancia a la que se encuentra el objeto más cercano dentro del mismo clúster. Los valles en la curva indican la presencia de clústeres, ya que los puntos dentro de un mismo clúster tienden a estar más cerca entre sí que con puntos de otros clústers.

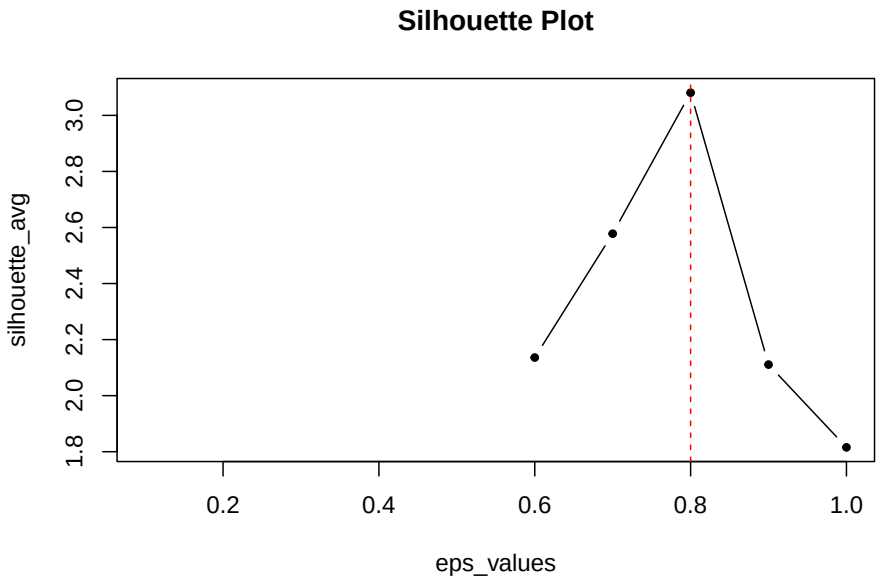
Así pues, como se puede observar, a primera vista vemos como apriori podriamos clasificar nuestra base de datos en tres clústeres. Aun así, hay una gran parte de nuestros datos que no está bien representada (la parte derecha del gráfico).

Método de la silueta

Otra forma de encontrar los valores óptimos de los parámetros necesarios es a través del método de la silueta.

En esta sección, se ejecutará OPTICS con diferentes valores de `eps`y se calculará la medida de silueta para cada valor. Luego, se graficará esta medida en función de épsilon y se identificará su valor óptimo.

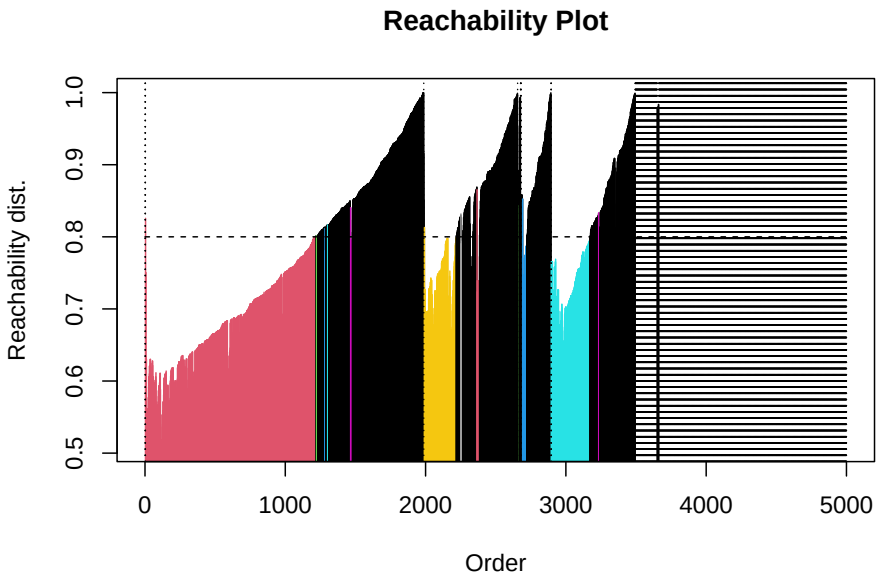
Figura 64: Gráfico del método de la silueta



Como se puede apreciar, al agregar una línea vertical en el valor óptimo de épsilon, vemos que se aconseja cortar en 0.8, valor que maximiza la silueta.

Por último, entramos en la etapa posterior al cálculo de la estructura de clústeres utilizando OPTICS. Esta última etapa, consiste en extraer y visualizar los resultados del clustering, donde a partir de la variable `opt_eps`, se determinará como se corta la curva de alcance para identificar los clústeres (diferenciados por colores).

Figura 65: Reachability Plot

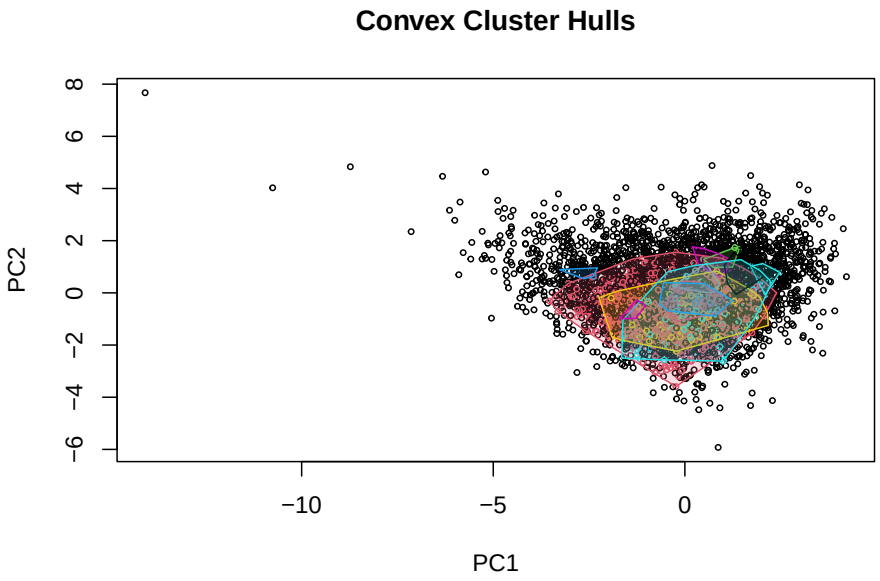


Con `plot(res)` se genera un gráfico que visualiza los clústeres obtenidos. Los puntos de datos se colorean de acuerdo con los clústeres a los que pertenecen, y los puntos que se consideran ruido se muestran en negro.

De igual manera que pasaba en el reachability plot anterior, hay una gran parte de nuestros datos que no sale bien representada. Además, al colorear los diferentes clústeres por colores, vemos que hay una gran parte de nuestros datos que se consideran ruido. Por otro lado, contrariamente a los resultados del primer reachability plot, en este se puede apreciar como nuestros datos podrían estar clasificados entre más grupos. No obstante, la presencia de tanto ruido y la parte no explicada nos podría estar informando de que nuestra base de datos no es adecuada para técnicas de clústring basadas en densidades.

Finalmente, visualizamos el gráfico con los grupos creados en forma de polígonos convexos. Estos polígonos nos ayudan a delimitar visualmente la extensión de cada clúster.

Figura 66: Polígonos de Convexidad para los Clústers Identificados



Por un lado, el gráfico obtenido no nos permite extraer buenos resultados, dado que es una gran nuve de puntos en donde la mayoría de los polígonos convexos se superponen entre sí.

Cuadro 25: Resumen del número de puntos en cada clúster

Clúster	Frecuencia de puntos
0	3153
1	1215
2	9
3	7
4	8
5	13
6	227
7	14
8	21
9	18
10	1
11	30
12	276
13	8

Por otro lado, la tabla obtenida nos resume el número de puntos en cada clúster. Así pues, observamos como aunque nos divide los datos en 13 clústers, siendo el grupo 0 el dominante, indicando la presencia de 3153 outliers. En los 12 grupos siguientes, la mayoria de las observaciones se agrupan mayoritariamente en el primero, con 1215, seguidos por el clúster 6 con 227 y el 12 con 276. Los grupos restantes tienen muy pocas observaciones en cada uno de ellos.

Estos resultados nos indican que para nuestra base de datos, este tipo de clustering no es el más adecuado.

Conclusión

En conclusión, aunque las técnicas utilizadas nos hayan ayudado a encontrar unos buenos parámetros para poder agrupar nuestros datos de la manera más óptima, vemos como estos resultados nos ayudan a respaldar aún más el hecho de que nuestra base de datos no es válida para técnicas de clustering basados en densidad, posiblemente por no tener una distribución de densidad variable.

Las técnicas de clustering basadas en densidad asumen que los clústers se forman en regiones de alta densidad de datos. Por lo tanto, si nuestros datos no tienen una distribución de densidad variable (puntos uniformemente distribuidos o clústeres sin una densidad significativamente mayor que el fondo), las téncicas de clustering basadas en densidad pueden no ser efectivas.

Así pues, ni DBSCAN ni OPTICS nos permiten extraer un buen análisis de nuestra base de datos, hay que recurrir, por ejemplo, a clustering jerárquico.