Minería de Datos

Tema 5 - Clústering

Nombre: Alejandro pérez Belando

---

1. Introducción

En el **aprendizaje no supervisado**, no tenemos información sobre la organización de los elementos en grupos o clases. Es por ello que el objetivo del clustering es, a partir de la relación entre los elementos, identificar distintos grupos de esos elementos:

- Dentro de un grupo, los elementos deben ser similares entre sí.
- Los elementos de un grupo deben ser distintos de los elementos de otros grupos.

Hay dos tipos de agrupamiento:

- **Agrupamiento jerárquico**: se basan en una jerarquía de agrupamientos particionales anidados (los clusters de un nivel superior se dividen en clusters más pequeños a medida que se profundiza en el análisis). Este agrupamiento jerárquico se representa mediante **dendogramas**.
- **Agrupamiento particional**: cuando los grupos creados son disjuntos[1] y cubren todo el conjunto de elementos.

2. Distancias y similaridad

Sean n elementos, recogidos en un conjunto $\Omega = \{e_1, \ldots, e_n\}$

Una **distancia** es una función d tal que:

$$d : \Omega \times \Omega \to \mathbb{R} d(e_i, e_j) \to d_{ij}$$

Con propiedades:

1. $d(e_i, e_j) \geq 0$
2. $d(e_i, e_i) = 0$

[1] Disjuntos: no tienen elementos en común

3. $d(e_i, e_j) = d(e_j, e_i)$
4. Desigualdad triangular: $d(e_i, e_j) \leq d(e_i, e_k) + d)e_k, e_j) \ \ \forall\ e_i, e_j, e_k \in \Omega$
   Cuando se cumple esta última propiedad, la distancia es una métrica y (Ω, d) forman un espacio métrico.

Una **similaridad** es una función s tal que:

$$s : \Omega \times \Omega \to \mathbb{R} s(e_i, e_j) \to s_{ij}$$

Con propiedades:

1. $s(e_i, e_j) \in [0, 1]$
2. $1 = s(e_i, e_i) \geq s(e_i, e_j)$
3. $s(e_i, e_j) = s(e_j, e_i)$

Ejemplo de similaridad
similaridad del coseno:

$$s(e_i, e_j) = cos(\theta) = \frac{e_i^T e_j}{\|e_i\|_2 \ \ \|e_j\|_2}$$

Distancia y similaridad están relacionados:

- $d_{ij} = 1 - s_{ij}$
- $d_{ij} = \sqrt{1 - s_{ij}^2}$

2.1. Distancia para variables continuas

Sean $x = \{x_1, \ldots, x_m\}$ e $y = \{y_1, \ldots, y_m\}$ dos elementos del conjunto Ω en el que todas las variables son continuas, algunas **medidas de distancia**:

Función	**Fórmula**
Euclídea	$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
Manhattan	$d(x, y) = \sum_{i=1}^{n} \lvert x_i - y_i \rvert$
Norma del supremo	$d(x, y) = \sup_{i \in \{1,2,\ldots,n\}} \lvert x_i - y_i \rvert$
Minkowski	$d(x, y) = (\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p)^{\frac{1}{p}}, \quad p > 0$
Mahalanobis	$d(x, y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)}, \quad \Sigma$ es la matriz de covarianzas

Normalización: para evitar que unas variables continuas dominen sobre otras. se hace empleando el Z-score:

$$Z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \begin{cases} \mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \\ \sigma_j = \sqrt{\frac{\sum_{i=1}^{n}(x_{ij} - \mu_j)^2}{n-1}} \\ x_{ij} \equiv \text{valor de la variable j en la instancia i} \end{cases}$$

2.2. Similitud para dos variables binarias

En este caso es más sencillo calcular primero la similitud y luego transformarla en distancia.

Sean $x = \{x_1, \ldots, x_m\}$ e $y = \{y_1, \ldots, y_m\}$ dos elementos del conjunto Ω en el que todas las variables son binarias.

Primero se calcula la **matriz de confusión** para calcular las coincidencias de las m variables:

		x_i		
		1	0	
y_i	1	a	b	a+b
	0	c	d	c+d
		a+c	b+d	m

Luego, se pueden calcular las **distancias** según las siguientes funciones:

Función	**Fórmula**
Índice de Acuerdo	$s(x,y) = \frac{a+d}{m}$
Jaccard	$s(x,y) = \frac{a}{a+b+c}$
Russel-Roo	$s(x,y) = \frac{a}{m}$
Czekanowski	$s(x,y) = \frac{2a}{2a+b+c}$

2.3. Distancia y similaridad para variables Cualitativas o nominales

Tenemos dos posibilidades:

- Calcular la distancia contando las coincidencias: sea m el número total de variables y p el número de coincidencias, la distancia es $d(x,y) = \frac{m-p}{m}$
- Crear un atributo binario para cada uno de los posibles valores calcular la similaridad como en el apartado anterior.

2.4. Distancia y similaridad para variables ordinales

Hay que calcular su correspondencia al intervalo $[0, 1]$. Sea $z_{ij} \equiv$ valor transformado para el objeto i de la variable j, $x'_{ij} \equiv$ el valor entero, mayor que 1, que indica el orden que ocupa el valor x_{ij} entre los valores ordinales en el dominio de la variable j y $M_k \equiv$ el límite superior del dominio de la variable j (asumiendo que el límite inferior es 1):

$$z_{ij} = \frac{x'_{ij} - 1}{M_k - 1}$$

2.5. Similaridad para variables mixtas

Para el caso de que se tiene variables de distintos tipos, se puede emplear cualquier medida de agregación de las distancias o similaridades de las variables independientes:

$$d(x, y) = \sum_{l=1}^{n} \omega_l \; d(x_l, y_l) \; ; \qquad \sum_{l=1}^{n} \omega_l = 1$$

Donde:

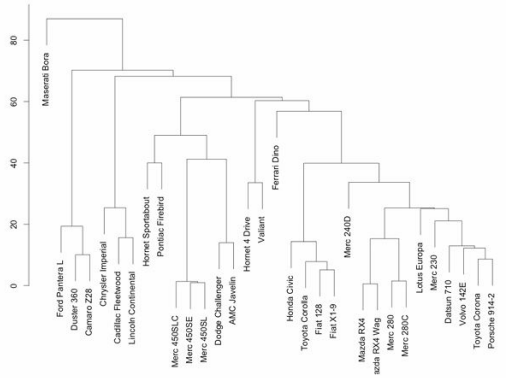
- $d(x_l, y_l) \equiv$ distancias de cada una de las variables.
- $\omega_l \equiv$ peso asociado a cada una de las variables.

3. Agrupamiento Jerárquico

Se basan en una jerarquía de agrupamientos particionales anidados. Esto quiere decir que los clústeres de un nivel superior se dividen en clústeres más pequeños. Esta estructura se representa mediante dendogramas, que es similar a un árbol binario en el que:

- Nodos hojas: elementos individuales.
- Nodos intermedios: agrupaciones de elementos.

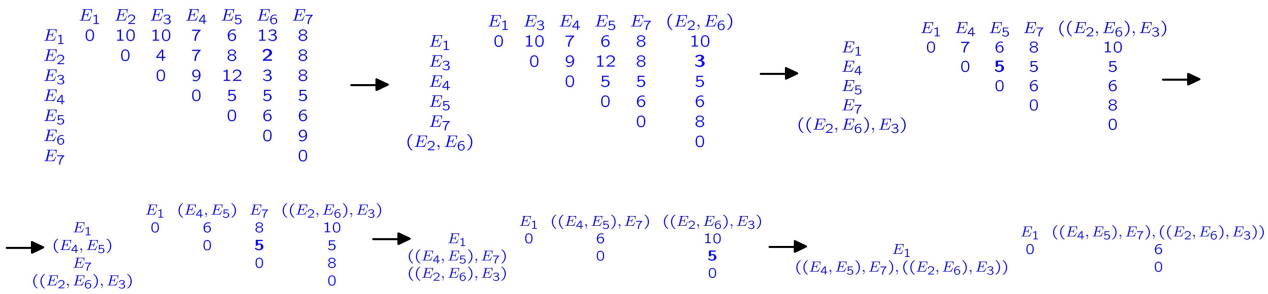
Un ejemplo de un dendograma:



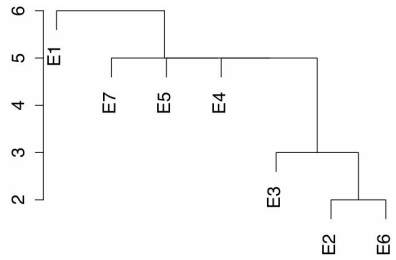
Dos técnicas para construir agrupamientos jerárquicos:

- **Técnicas divisivas**: los nuevos clústers se generan dividiendo clústeres más grandes. Tienen la ventaja de que parten de la información global que hay en los datos, aunque son más lentas que las aglomerativas.
- **Técnicas aglomerativas**: generan nuevos clústeres uniendo clústeres similares. Son las técnicas más usadas y son más eficientes que las divisivas.

Ejemplo de agrupamiento jerárquico mediante técnicas aglomerativas



Y el dendograma resultante de este ejemplo es el siguiente:



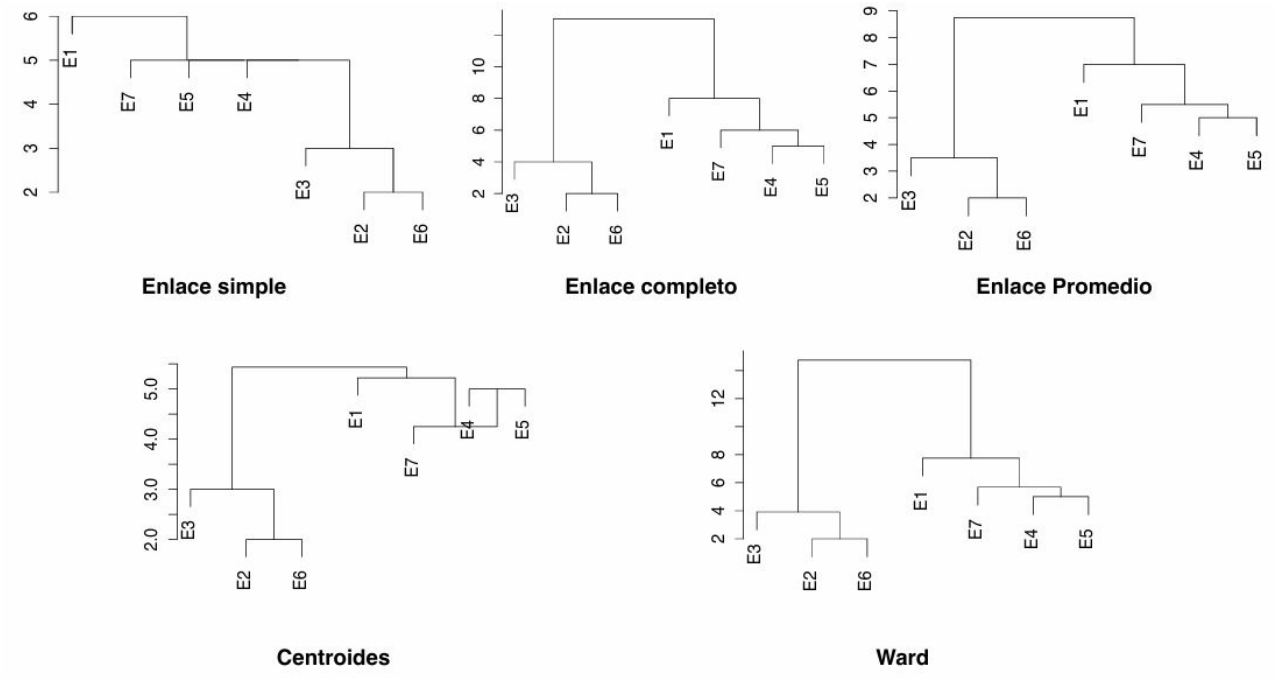
3.1. Distancia entre clústeres

Sean A y B dos clústeres, la **distancia entre ambos** se puede definir de distintas formas:

- **Método de enlace simple**: la distancia es la distancia mínima entre los elementos de ambos clústeres: $d(A,B) = \min_{x\in A, y\in B} d(x,y)$
- **Método de enlace completo**: la distancia es la distancia máxima entre los elementos de ambos clústeres: $d(A,B) = \max_{x\in A, y\in B} d(x,y)$
- **Método de enlace promedio**: la distancia es el promedio de la distancia entre cada par de elementos de ambos clústeres: $d(A,B) = \frac{1}{|A|\ |B|} \sum_{x\in A, y\in B} d(x,y)$

- **Método del centroide**: la distancia es la distancia entre los centroides de cada clúster. El centroide de (por ejemplo) el clúster A se calcula: $\bar{x} = \frac{1}{|A|}\sum_{x \in A} x$
- **Método de Ward**: se basa en fusionar aquellos clústeres de forma que en el nuevo clúster la suma de las distancias de los elementos al centroide sea menor.
- **Versiones ponderadas de los métodos promedio y centroide**: intentan compensar el hecho de fusionar clústeres de tamaños muy dispares.

Ejemplos del agrupamiento jerárquico según el método empleado:



4. Agrupamiento particional

Como se ha mencionado en la introducción, el agrupamiento particional se da cuando los grupos que se crean no tienen elementos en común.

Sea un conjunto de n elementos representados en un espacio de dimensión d, el **objetivo** del agrupamiento particional es encontrar una partición del mismo en k subconjuntos de forma que los elementos dentro de un grupo deben parecerse más entre sí que a los elementos de otros grupos.

En la mayoría de técnicas, el número k de subgrupos es un parámetro.

Necesitamos un **criterio para medir la coherencia de cada grupo y entre grupos**. Hay dos tipos de criterios:

Criterios globales: representan cada grupo mediante un prototipo, asignando cada elemento al grupo del prototipo más cercano. Dos técnicas pueden ser **K-medias** y **K-medoides**.

Criterios locales: forman grupos empleando la estructura local de los datos. Una técnica es **DBSCAN**.

4.1. Agrupamiento en base a encoders

Dado un conjunto de n elementos ($x = \{x_1, \ldots, x_n\}$, el resultado de aplicar una técnica de agrupamiento para encontrar k clústeres ($k < n$) se puede definir mediante un **encoder**:

$$C(i) = t \Leftrightarrow x_i \in C_i$$

Es decir, el encoder (C) nos dice a qué clúster pertenece cada elemento.

4.2. Distancia intraclúster (W) e interclúster (B)

Sea C un encoder de k clústeres sobre el conjunto $x = \{x_1, \ldots, x_n\}$, la **separación total entre los elementos de** x siempre es constante:

$$T = \frac{1}{2}\sum_{j=1}^{k}\sum_{i:C(i)=j}\sum_{i':C(i')=j} d_{ii'} + \frac{1}{2}\sum_{j=1}^{k}\sum_{i:C(i)=j}\sum_{i':C(i')\neq j} d_{ii'} = W(C) + B(C)$$

Donde W(C) $\equiv$ distancia intraclúster (distancia entre los puntos del mismo clúster) y B(C) $\equiv$ distancia entre los puntos de distintos clústeres.

Entonces, para tener un buen agrupamiento (encoder), podemos:

- Maximizar la distancia interclúster B(c) (buscar clústeres lo más separados entre sí)
- Minimizar la distancia intraclúster W(c) (buscar clústeres lo más compactos posibles).

4.3. Método K-medias

El más popular para variables numéricas. Como la distancia euclídea es:

$$d_{ij} = d(x_i, x_j) = \sum_{t=1}^{m} (x_{it} - x_{jt})^2 = \|x_i - x_j\|^2$$

Entonces, la distancia intraclúster es:

$$W(C) = \frac{1}{2}\sum_{l=1}^{k}\sum_{i:C(i)=l}\sum_{j:C(j)=l} \|x_i - x_j\|^2$$

Minimizar la distancia intraclúster, W(C), es equivalente a minimizar a distancia a los centroides:

$$\sum_{l=1}^{k} \sum_{i:C(i)=l} \|x_i - \mu_l\|^2$$

Y la distancia de cada elemento a su respectivo centroide μ_i:

$$\mu_i = \frac{1}{|C_i|} \sum_{i:C(i)=l} x_i$$

Por lo tanto, nuestro problema de optimización:

$$C^* = \min_{C} \sum_{l=1}^{k} \sum_{i:C(i)=l} \|x_i - \mu_i\|^2$$

Algoritmo de K-medias:

1. Hay dos posibles configuraciones iniciales:
   - Si se inicializan aleatoriamente los centroides (μ_l) de cada uno de los k clústeres, ir al **paso 2**.
   - Si se parte de una partición aleatoria del conjunto de entrada en k clústeres, ir al **paso 3**.
2. Calcular el encoder (distribur los elementos entre los k clústeres de acuerdo a los centroides μ_i): $C(i) = arg \min_{1 \leq l \leq k} \|x_i - \mu_i\|^2$
3. Calcular los nuevos centroides μ_i para $l = 1, \ldots, k$
4. Si los nuevos centroides se han estabilizaso, fin. Si no, repetir el **paso 2**.

Consideraciones del método K-medias:

- Este método se aplica cuando todos los atributos son reales.
- La distancia euclídea amplifica el efecto de los outliers. Se 'pueden considerar otras medidas de distancia e incluso medidas de similitud (pero en lugar de minimizar habrá que maximizar).

Problemas del método K-medias:

- Es muy sensible a la elección de los centroides. Se pueden hacer varias ejecuciones:
  - Escoger de entre las n instancias del conjunto de observaciones, k instancias para inicializar los μ_l.
  - Usar el vector de medias μ de todo el conjunto de datos.

    - Hacer análisis PCA o usar clústering jerárquico.
- Es muy sensible al número de clústeres k, por lo que hay que elegirlos a priori. Esto se puede hacer mediante un método jerárquico para estimar k o simplemente aplicar K-medias (probar) para varios valores de k.
- Al usar la media para calcular los centroides, el método es sensible a los outliers.
- Para trabajar con datos no numéricos, se trabajaría con la moda y no la media.
- K-medias no funciona bien cuando los clústeres son de distinto tamaño, densidad y no convexos.

4.4. Método K-medoides

Se emplea para evitar la sensibilidad del método K-medias a los outliers, y elige como representante de cada clúster usando la mediana[2]. Como no se han escogido como representantes los valores medios sino elementos del conjunto de datos, el método es más robusto e interpretable. Sin embargo, es más costoso computacionalmente y hay que conocer el número de grupos.

El proceso es idéntico al método K-medias, pero en este caso se calculan los medoides[3] y no los centroides.

Algoritmo de K-medoides:

1. Hay dos posibles configuraciones iniciales:
    - Si se escogen aleatoriamente k elementos como representantes m_i de los k clústeres, ir al **paso 2**.
    - Si se parte de una partición aleatoria del conjunto de entrada en k clústeres, ir al **paso 3**.
2. Calcular el encoder (distribur los elementos entre los k clústeres de acuerdo a los representantes m_i): $C(i) = arg \min_{1 \leq l \leq k} d(x_i, m_i)$
3. Calcular los nuevos medoides m_i para $l = 1, \ldots, k$ de cada clúster: $m_i = arg \min_{i:C(i)=l} \sum_{j:C(j)=l} d(x_i, x_j)$
4. Si los nuevos medoides se han estabilizaso, fin. Si no, repetir el **paso 2**.

Implementaciones del método K-medoides

- **PAM (Partition Arround Medoids)**: implementación de las ideas anteriormente expuestas.

[2]Mediana: valor central de un conjunto de datos ordenados en una dimensión

[3]Medoide: punto representativo de un grupo en múltiples dimensiones, minimizando la distancia total a los demás puntos.

- **CLARA (Clustering LARge Applications)**: reduce la carga computacional del PAM seleccionando los medoides de una muestra aleatoria y significativa de los datos.
- **CLARANS**: se diferencia del CLARA en que la búsqueda de los medoides se aproxima como unproceso de búsqueda, realizando muestreo cada vez que se aplica el medoide.
- **DBSCAN (Density-Based Spatial Clustering Applications with Noise)**.

4.5. Método DBSCAN

Basado en criterios locales que se apoya en el concepto de densidad de los puntos: son **clústeres** aquellas regiones con alta densidad de puntos, y **puntos que no están asociados** aquellas regiones con baja densidad de puntos.

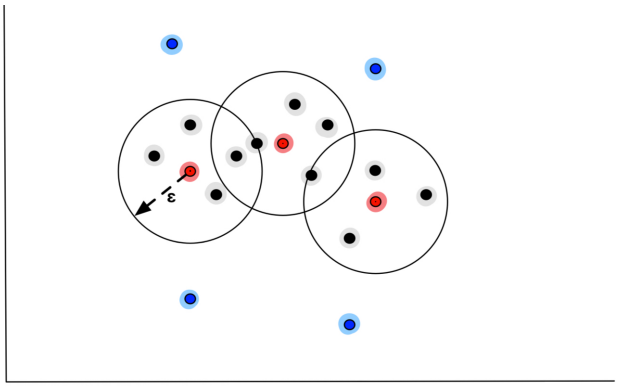
4.5.1. Conceptos previos

ϵ-vecindad: sea $x_k \in \Omega = \{x_1, \ldots, x_n\}$ y $\epsilon > 0$; $\epsilon \in \mathbb{R}$, se define $N_\epsilon(x_k)$ (la ϵ-vecindad del punto x_k) como:

$$N_\epsilon(x_k) = \{x \in \Omega | d(x, x_k) \leq \epsilon\}$$

es decir, la ϵ-vecindad de un punto incluye todos aquellos puntos que están a una distancia menor o igual que ϵ.

- La geometría de la ϵ-vecindad viene determinada por la media de la distancia utilizada.
- Para determinar cuándo una ϵ-vecindad tiene una densidad alta, se usa el parámetro $MinPts$: si se cumple que $|N_\epsilon(x_k)| \geq MinPts$, entonces la densidad entorno al punto x_k es alta y x_k es un **punto núcleo**.
- Todo punto dentro de la ϵ-vecindad de un punto núcleo es un **punto frontera**. Un punto frontera puede pertenecer a más de una ϵ-vecindades distintas.
- El resto de puntos son **puntos ruido**.



Concepto de $\epsilon-$ vecindad y puntos núcleo (rojo), puntos frontera (negro) y puntos ruido (azul) (MinPts = 4).

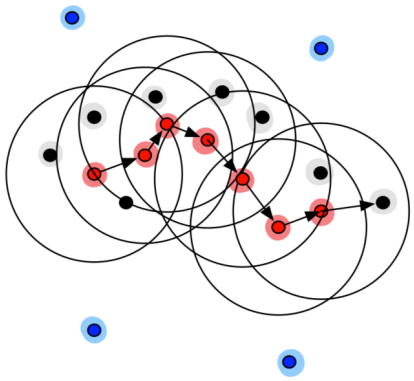
Densidad alcanzable directa: sean p y q dos elementos del conjunto Ω, q es directamente densidad alcanzable desde p, si y solo si:

1. $q \in N_\epsilon(p)$
2. p es un punto núcleo.

Densidad alcanzable: sean p y q dos elementos del conjunto Ω, q es densidad alcanzable desde p, si existe una cadena de puntos $p_1 \ldots, p_l$ tal que:

1. $p_1 = p$ y $p_l = q$
2. $\forall i \in \{1, \ldots, l\}_{p_{i+1}}$ es directamente densidad alcanzable desde p_i.

Esta densidad es transitiva pero no simétrica.



Concepto de densidad alcanzable

Puntos densamente conectados: sean p y q dos elementos del conjunto Ω, p y q están densamente conectados si son directamente alcanzables desde un mismo punto o. La conectividad densa es simétrica.

Clúster: sea un conjunto Ω y los parámetros ϵ y $MinPts$, un clúster C_l es un subconjunto de Ω que satisface:

1. Maximalidad: $\forall p, q$, si $p \in C_l$ y q es densidad alcanzable desde p, entonces $q \in C_l$
2. Conectividad: $\forall p, q \in C_l$, p y q están densamente conectados.

Un clúster contiene puntos núcleo y puntos frontera.

4.5.2. Algoritmo DBSCAN

La idea básica es crear clústeres con todos los puntos que son densidad alcanzable.

1. Especificar los parámetros ϵ y $MinPts$.
2. Seleccionar arbitrariamente un punto $x_k \in \Omega$.

3. Encontrar aquellos puntos densidad alcanzables desde x_k.
4. Si x_k es un:
   - punto núcleo, se forma un clúster y se incluyen también los puntos en su ϵ-vecindad.
   - punto frontera, se procede con el siguiente punto.
   - En otro caso, el punto se etiqueta como ruido y se desecha.
5. Los pasos 2-4 se repiten hsata que todos los puntos hayan sido visitados o añadidos a algún clúster.

Parámetros que influyen notablemente en le método

- La función de distancia elegida: define la geometría de la ϵ-vecindad. Algunas consideraciones:
  - Valores altos para ϵ requieren valores altos para $MinPts$.
  - Un valor bajo para ϵ dará lugar a un número alto de clústeres pequeños; conforme se aumenta el valor, habrá menos clústeres pero más puntos ruido.
- Parámetro MinPts: suele ser $MinPts = d + 1$, aunque algunos autores usan $MinPts = 2d - 1$. Algunas consideraciones:
  - $MinPts = 1$ no tiene sentido.
  - $MinPts = 2$ equivale a un agrupamiento jerárquico de enlace simple cortado a la altura ϵ
  - Valores grandes de $MinPts$ suelen ser mejores para datos con ruido.
  - Cuanto mayor sea el conjunto de datos, mayor debe ser $MinPts$
- Valor ϵ: se puede escoger por medio de un gráfico de k-distancias con $k = MinPts$.
  - Se fija el valor de ϵ a la distancia en la que se muestre una fuerte curvatura
  - A medida que ϵ se hace más grande, el tamaño de los clústeres obtenidos aumentará.

Ventajas del método DBSCAN

- Los clústeres pueden tener formas y tamaños arbitrarios.
- El número de clústeres se determina automáticamente
- Se puede detectar y aislar el ruido.

Desventajas del método DBSCAN

- No es enteramente determinista, por ejemplo un punto frontera puede pertenecer a dos clústeres.
- Muy sensible a los valores de los parámetros.

4.6. Método OPTICS

Es una generalización de DBSCAN en la que solo se fija el parámetro $MinPts$.

Este método no genera un conjunto de clústeres; ordena los elementos de forma que aquellos puntos cercanos son vecinos en dicho orden. Se almacena la distancia que se necesita para que dichos puntos pertenezcan al mismo clúster.

5. Evaluación de agrupamientos

La mayoría de las técnicas se ven influenciadas por la medida de distancia utilizada y el número de clústeres que debe buscar.

Por ello, para escoger una configuración entre varias posibles, se hace una **medida de la calidad del agrupamiento**. Una técnica de agrupamiento debe satisfacer:

- **Compactación**: indica cómo de cerca están entre sí los elementos de un clúster (distancia intraclúster). A mayor varianza, menos compacto.
- **Separabilidad**: indica cuán distintos son los clústeres entre sí (distancia interclúster).

5.1. Índice Silueta

Sea $x_i \in C_i$, el índice silueta $s(x_i)$:

$$s(x_i) = \frac{b(x_i) - a(x_i)}{max(a(x_i), b(x_i))}$$

Donde:

- $a(x_i) = \frac{1}{|C_l|} \sum_{\forall x_j \in C_l \bigwedge j \neq i} d(x_i, x_j) \equiv$ distnacia media entre x_i y todos los elementos de su mismo clúster, C_l
- $b(x_i) = min_{k \neq l} \left(d(x_i, C_k) \right); \quad d(x_i, C_k) = \frac{1}{|C_k|} \sum_{\forall x_j \in C_k} d(x_i, x_j)$ distnacia media entre x_i y todos los elementos de su mismo clúster, C_l

Propiedades del índice silueta:

- Si el clúster C_l tiene un solo elemento, entonces $s(x_i) = 0$
- El índice silueta varía entre $[-1, 1]$
- Si $s(x_i)$ es cercano a 1, la observación está muy bien agrupada.
- Si $s(x_i)$ es cercano a 0, la observación está entre dos clústeres.
- Si $s(x_i)$ es negativo, la observación está mal agrupada.

- El índice silueta para todo el agrupamiento sería la media de todos los índices siluetas.

5.2. Índice GAP

Compara la varianza total intraclúster observada para distintos valores de k.

Supongamos que nuestros datos han sido agrupados en k clústeres $\{C_1, \ldots, C_k\}$; $\quad n_r = |C_r|$

Sean:

- $D_r = \sum_{x_i, x_j \in C_r} d(x_i, x_j) \equiv$ suma de la distancia entre todos los elementos del clúster r
- $W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \equiv$ distancia intraclúster para k número de clústeres.

Algoritmo del índice GAP:

1. Calcular W_k para distintos valores de k.
2. Generar B conjuntos de referencia usando un muestreo uniforme.
3. Calcular la suma de la distancia intraclúster W^*_{bk} en cada uno de los B conjuntos y para distintos valores de k
4. Calcular el índice GAP:
$$Gap(k) = \frac{1}{B} \sum_b \log(W^*_{kb}) - \log(W_k)$$
5. Calcular las desviaciones estándares:
$$s_k = \sqrt{\frac{1}{B} \sum_b \left(\log(W^*_{kb}) - \bar{I})^2\right)} \; ; \quad \bar{I} = \frac{1}{B} \sum_b \log(W^*_{kb})$$
6. Determinar el valor óptimo de k:
$$k_{opt} = \min_k (gap(k) \geq gap(k+1) - s_k)$$

5.3. Índice Davies-Bouldin

Sea un agrupamiento de k clústeres $\{C_1, \ldots, Ck\}$

la distancia intraclúster se puede definir: $w_i = \frac{1}{|C_i|} \sum_{x_i \in C_l} d(x_i, \mu_l)$, donde $\{\mu_1, \ldots, \mu_k\}$ son los centroides de cada clúster.

La distancia entre centroides: $d_{ij} = d(\mu_i, \mu_1)$. Si para cada par de clústeres calculamos el siguiente ratio:
$$r_i = \max_{j, j \neq i} \frac{w_i + w_j}{d_{ij}}$$

El índice de Davies-Bouldin: $r = \frac{1}{c} \sum_{i=1}^{c} r_i$

- El valor óptimo para el número de clústeres es aquel que hace mínimo el índice.
- Como el valor de r mímimo se consigue con valores pequeños del numerador (r_i) y grandes del denominador (c), se favorece la creación de grupos compactos y separados.