UTEC Posgrado

UTEC Posgrado

UTEC Posgrado

Una escuela para
Early Adopters

MACHINE LEARNING

AGRUPAMIENTO
DBSCAN

Motivación

Contexto y necesidad **del agrupamiento basado en** n**idad**

Repaso: Análisis de conglomerados

Objetivo: Agrupar N objetos en subconjuntos (clústeres) tales que:

- Objetos **dentro** de un clúster: muy **similares**

Repaso: Análisis de conglomerados

Objetivo: Agrupar N objetos en subconjuntos (clústeres) tales que:

- Objetos **dentro** de un clúster: muy **similares**
- Objetos en clústeres **distintos**: muy **diferentes**

Repaso: Análisis de conglomerados

Objetivo: Agrupar N objetos en subconjuntos (clústeres) tales que:

- Objetos **dentro** de un clúster: muy **similares**
- Objetos en clústeres **distintos**: muy **diferentes**

Métodos clásicos: K-means (particionamiento) y jerárquico (dendrograma).

Pero... ¿funcionan **siempre**?

Limitaciones de los métodos clásicos

K-means

- Asume clústeres **esféricos**

Jerárquico

- Dificultad con formas no convexas

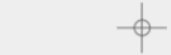
Limitaciones de los métodos clásicos

K-means

- Asume clústeres **esféricos**
- Requiere fijar K de antemano

Jerárquico

- Dificultad con formas no convexas
- Complejidad $O(N^2)$ en espacio



Limitaciones de los métodos clásicos

K-means

- Asume clústeres **esféricos**
- Requiere fijar K de antemano
- Sensible a **outliers**

Jerárquico

- Dificultad con formas no convexas
- Complejidad $O(N^2)$ en espacio
- Fusiones irreversibles

Limitaciones de los métodos clásicos

K-means

- Asume clústeres **esféricos**
- Requiere fijar K de antemano
- Sensible a **outliers**
- No detecta formas arbitrarias

Jerárquico

- Dificultad con formas no convexas
- Complejidad $O(N^2)$ en espacio
- Fusiones irreversibles
- Encadenamiento (enlace simple)

Limitaciones de los métodos clásicos

K-means

- Asume clústeres **esféricos**
- Requiere fijar K de antemano
- Sensible a **outliers**
- No detecta formas arbitrarias

Jerárquico

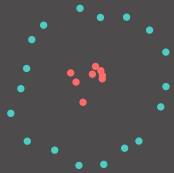
- Dificultad con formas no convexas
- Complejidad $O(N^2)$ en espacio
- Fusiones irreversibles
- Encadenamiento (enlace simple)

¿Existe un método que detecte clústeres de forma arbitraria y maneje ruido?
$\Longrightarrow$ **DBSCAN**

¿Por qué formas arbitrarias?

K-means: OK

K-means: FALLA



Datos reales pueden tener clústeres con forma de "S", anillos, espirales... Los métodos de particionamiento y jerárquicos identifican incorrectamente regiones **convexas**.

La idea clave: Densidad

Intuición: Los clústeres son **regiones densas** separadas por regiones **dispersas**.

La idea clave: Densidad

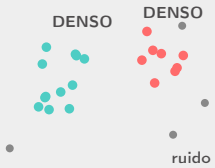
Intuición: Los clústeres son **regiones densas** separadas por regiones **dispersas**.

- No asumimos forma específica

La idea clave: Densidad

Intuición: Los clústeres son **regiones densas** separadas por regiones **dispersas**.

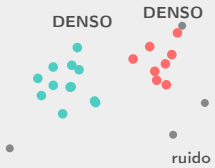
- No asumimos forma específica
- La **densidad** = número de objetos cercanos



La idea clave: Densidad

Intuición: Los clústeres son **regiones densas** separadas por regiones **dispersas**.

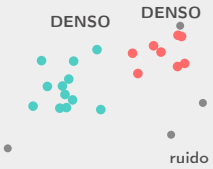
- No asumimos forma específica
- La **densidad** = número de objetos cercanos
- Puntos aislados = **ruido**



La idea clave: Densidad

Intuición: Los clústeres son **regiones densas** separadas por regiones **dispersas**.

- No asumimos forma específica
- La **densidad** = número de objetos cercanos
- Puntos aislados = **ruido**
- Funciona con cualquier forma de clúster



Aplicaciones de DBSCAN

Área	Aplicación
Geoespacial	Detección de zonas urbanas, puntos calientes de crimen
Astronomía	Identificación de galaxias y cúmulos estelares
Imágenes	Segmentación de regiones con textura similar
Anomalías	Detección de fraude, intrusiones en redes
Biología	Agrupamiento de secuencias genéticas

¿Qué es DBSCAN?

DBSCAN = *Density-Based Spatial Clustering of Applications with Noise*

- Propuesto por Ester, Kriegel, Sander y Xu (1996)

¿Qué es DBSCAN?

DBSCAN = *Density-Based Spatial Clustering of Applications with Noise*

- Propuesto por Ester, Kriegel, Sander y Xu (1996)
- Encuentra **objetos núcleo** (regiones densas)

¿Qué es DBSCAN?

DBSCAN = *Density-Based Spatial Clustering of Applications with Noise*

- Propuesto por Ester, Kriegel, Sander y Xu (1996)
- Encuentra **objetos núcleo** (regiones densas)
- Conecta objetos núcleo y sus vecindarios $\rightarrow$ clústeres

¿Qué es DBSCAN?

DBSCAN = *Density-Based Spatial Clustering of Applications with Noise*

- Propuesto por Ester, Kriegel, Sander y Xu (1996)
- Encuentra **objetos núcleo** (regiones densas)
- Conecta objetos núcleo y sus vecindarios $\rightarrow$ clústeres
- Puntos que no pertenecen a ningún clúster $\rightarrow$ **ruido**

¿Qué es DBSCAN?

DBSCAN = *Density-Based Spatial Clustering of Applications with Noise*

- Propuesto por Ester, Kriegel, Sander y Xu (1996)
- Encuentra **objetos núcleo** (regiones densas)
- Conecta objetos núcleo y sus vecindarios $\rightarrow$ clústeres
- Puntos que no pertenecen a ningún clúster $\rightarrow$ **ruido**

Dos parámetros:

- $\epsilon > 0$: radio del vecindario
- *MinPts*: número mínimo de puntos para ser denso

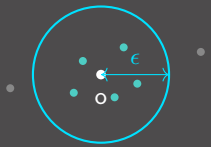
Parámetros de DBSCAN: ϵ y *MinPts*

ϵ (epsilon):

- Radio del vecindario
- Define el "alcance" de cada punto
- ϵ pequeño $\rightarrow$ muchos puntos de ruido
- ϵ grande $\rightarrow$ clústeres se fusionan

***MinPts*:**

- Umbral de densidad
- ¿Cuántos vecinos necesita un punto para ser "núcleo"?
- *MinPts* alto $\rightarrow$ criterio más estricto
- Regla práctica: $MinPts \geq d + 1$ donde d = dimensiones



ϵ-vecindario de o: 5 puntos dentro

Tres tipos de puntos en DBSCAN

- **Objeto núcleo (core):** tiene $\geq$ *MinPts* vecinos en su ϵ-vecindario

Tres tipos de puntos en DBSCAN

- **Objeto núcleo (core):** tiene $\geq$ *MinPts* vecinos en su ϵ-vecindario
- **Punto frontera (border):** está en el ϵ-vecindario de un núcleo, pero no es núcleo él mismo

Tres tipos de puntos en DBSCAN

- **Objeto núcleo (core):** tiene $\geq$ *MinPts* vecinos en su ϵ-vecindario
- **Punto frontera (border):** está en el ϵ-vecindario de un núcleo, pero no es núcleo él mismo
- **Ruido (noise):** no es núcleo ni frontera

Tres tipos de puntos en DBSCAN

- **Objeto núcleo (core):** tiene $\geq$ *MinPts* vecinos en su ϵ-vecindario
- **Punto frontera (border):** está en el ϵ-vecindario de un núcleo, pero no es núcleo él mismo
- **Ruido (noise):** no es núcleo ni frontera

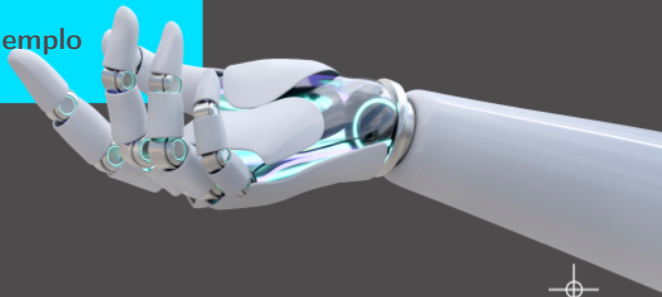
Tipo	$\geq$ *MinPts* vecinos?	Vecino de un núcleo?
Núcleo	Sí	—
Frontera	No	Sí
Ruido	No	No

UTEC Posgrado

Una escuela para
Early Adopters

Desarrollo

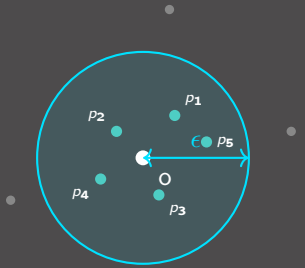
DBSCAN **Conceptos, algoritmo y ejemplo**



ϵ-vecindario

El ϵ-**vecindario** de un objeto o es el espacio dentro de un radio ϵ centrado en o:

$$N_\epsilon(o) = \{p \in D \mid dist(o, p) \leq \epsilon\}$$



$|N_\epsilon(o)| = 5$ puntos (sin contar o)

Objeto núcleo

Un objeto es un **objeto núcleo** si su ϵ-vecindario contiene al menos *MinPts* objetos:

$$|\{p \in D \mid \text{dist}(o, p) \leq \epsilon\}| \geq \textit{MinPts}$$

Directamente alcanzable por densidad

Un objeto p es **directamente alcanzable por densidad** desde q si:

1. q es un **objeto núcleo**

Directamente alcanzable por densidad

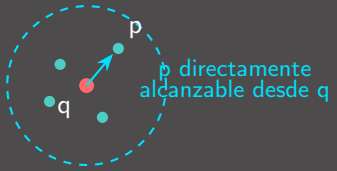
Un objeto p es **directamente alcanzable por densidad** desde q si:

1. q es un **objeto núcleo**
2. p está en el ϵ-vecindario de q

Directamente alcanzable por densidad

Un objeto p es **directamente alcanzable por densidad** desde q si:

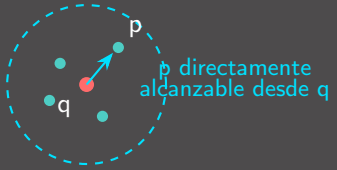
1. q es un **objeto núcleo**
2. p está en el ϵ-vecindario de q



Directamente alcanzable por densidad

Un objeto p es **directamente alcanzable por densidad** desde q si:

1. q es un **objeto núcleo**
2. p está en el ϵ-vecindario de q



Nota: La relación **no** es simétrica si p no es núcleo.

Alcanzable por densidad (cadena)

p es **alcanzable por densidad** desde q si existe una cadena $p_1, \ldots, p_n$ donde:

- $p_1 = q$, $p_n = p$
- p_{i+1} es directamente alcanzable desde p_i

Cadena de objetos núcleo conecta q con p

Conectado por densidad

Dos objetos p_1, p_2 están **conectados por densidad** si existe un objeto q tal que:

- p_1 es alcanzable por densidad desde q
- p_2 es alcanzable por densidad desde q

Propiedad: La conectividad por densidad es una **relación de equivalencia** (reflexiva, simétrica, transitiva).

Clúster basado en densidad: Definición

Un subconjunto $C \subseteq D$ es un **clúster basado en densidad** si:

1. **Conectividad:** Para cualquier $o_1, o_2 \in C$, o_1 y o_2 están conectados por densidad

Clúster basado en densidad: Definición

Un subconjunto $C \subseteq D$ es un **clúster basado en densidad** si:

1. **Conectividad:** Para cualquier $o_1, o_2 \in C$, o_1 y o_2 están conectados por densidad
2. **Maximalidad:** No existe $o \in C$ y $o' \in (D - C)$ tal que o y o' estén conectados por densidad

Clúster basado en densidad: Definición

Un subconjunto $C \subseteq D$ es un **clúster basado en densidad** si:

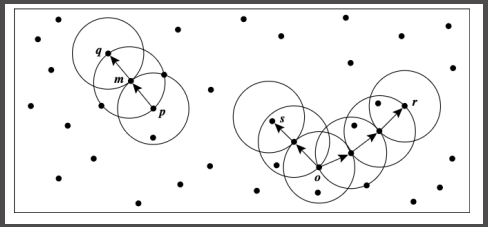
1. **Conectividad:** Para cualquier $o_1, o_2 \in C$, o_1 y o_2 están conectados por densidad
2. **Maximalidad:** No existe $o \in C$ y $o' \in (D - C)$ tal que o y o' estén conectados por densidad

En otras palabras:

- Todos los puntos dentro están conectados entre sí (vía densidad)
- No se puede agregar ningún punto más sin romper la definición
- Los puntos que no pertenecen a ningún clúster son **ruido**

Ejemplo visual: Alcanzabilidad y conectividad

Sea $MinPts = 3$ y ϵ representado por el radio de los círculos:



- m, p, o, r: objetos núcleo q: directamente alcanzable desde m
- o, r, s: conectados por densidad p no alcanzable desde q (q no es núcleo)

Algoritmo DBSCAN

Entrada: D: conjunto de datos con n objetos, ϵ: radio, *MinPts*: umbral de densidad
Salida : Un conjunto de clústeres basados en densidad

```
1  Marcar todos los objetos como no visitados
2  repeat
3  |  Seleccionar aleatoriamente un objeto no visitado p
4  |  Marcar p como visitado
5  |  if el ϵ-vecindario de p tiene al menos MinPts objetos then
6  |  |  Crear nuevo clúster C y agregar p a C
7  |  |  N ← objetos en el ϵ-vecindario de p
8  |  |  foreach punto p′ en N do
9  |  |  |  if p′ no ha sido visitado then
10 |  |  |  |  Marcar p′ como visitado
11 |  |  |  |  if ϵ-vecindario de p′ tiene ≥ MinPts puntos then
12 |  |  |  |  |  Agregar esos puntos a N
13 |  |  |  |  end
14 |  |  |  end
15 |  |  |  if p′ no pertenece a ningún clúster then
16 |  |  |  |  Agregar p′ a C
17 |  |  |  end
18 |  |  end
19 |  |  Producir C como salida
20 |  else
21 |  |  Marcar p como ruido
22 |  end
23 until ningún objeto esté sin visitar
```

¿Cómo funciona DBSCAN?

1. **Inicio:** Todos los puntos están “no visitados”

¿Cómo funciona DBSCAN?

1. **Inicio:** Todos los puntos están “no visitados”
2. **Seleccionar** un punto p al azar, marcarlo como “visitado”

¿Cómo funciona DBSCAN?

1. **Inicio:** Todos los puntos están "no visitados"
2. **Seleccionar** un punto p al azar, marcarlo como "visitado"
3. **Verificar:** ¿El ϵ-vecindario de p tiene $\geq$ *MinPts* puntos?

¿Cómo funciona DBSCAN?

1. **Inicio:** Todos los puntos están "no visitados"
2. **Seleccionar** un punto p al azar, marcarlo como "visitado"
3. **Verificar:** ¿El ϵ-vecindario de p tiene $\geq$ *MinPts* puntos?
   - **No:** marcar p como **ruido** (puede cambiar después)

¿Cómo funciona DBSCAN?

1. **Inicio:** Todos los puntos están "no visitados"
2. **Seleccionar** un punto p al azar, marcarlo como "visitado"
3. **Verificar:** ¿El ϵ-vecindario de p tiene $\geq$ *MinPts* puntos?
   - **No:** marcar p como **ruido** (puede cambiar después)
   - **Sí:** crear un nuevo clúster C, agregar p y sus vecinos

¿Cómo funciona DBSCAN?

1. **Inicio:** Todos los puntos están "no visitados"
2. **Seleccionar** un punto p al azar, marcarlo como "visitado"
3. **Verificar:** ¿El ϵ-vecindario de p tiene $\geq$ *MinPts* puntos?
   - **No:** marcar p como **ruido** (puede cambiar después)
   - **Sí:** crear un nuevo clúster C, agregar p y sus vecinos
4. **Expandir:** Para cada vecino no visitado, repetir la verificación

¿Cómo funciona DBSCAN?

1. **Inicio:** Todos los puntos están "no visitados"
2. **Seleccionar** un punto p al azar, marcarlo como "visitado"
3. **Verificar:** ¿El ϵ-vecindario de p tiene $\geq$ *MinPts* puntos?
   - **No:** marcar p como **ruido** (puede cambiar después)
   - **Sí:** crear un nuevo clúster C, agregar p y sus vecinos
4. **Expandir:** Para cada vecino no visitado, repetir la verificación
5. **Cuando** C no puede crecer más $\rightarrow$ clúster completo

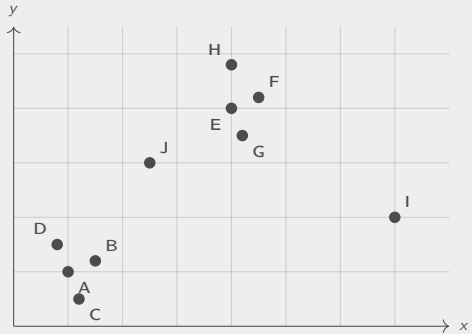
¿Cómo funciona DBSCAN?

1. **Inicio:** Todos los puntos están "no visitados"
2. **Seleccionar** un punto p al azar, marcarlo como "visitado"
3. **Verificar:** ¿El ϵ-vecindario de p tiene $\geq$ *MinPts* puntos?
   - **No:** marcar p como **ruido** (puede cambiar después)
   - **Sí:** crear un nuevo clúster C, agregar p y sus vecinos
4. **Expandir:** Para cada vecino no visitado, repetir la verificación
5. **Cuando** C no puede crecer más $\rightarrow$ clúster completo
6. **Repetir** con el siguiente punto no visitado hasta terminar

Ejemplo paso a paso: Dataset

Consideremos 10 puntos en 2D con $\epsilon = 1.5$ y $MinPts = 3$:

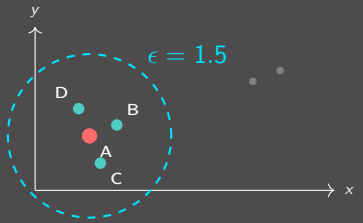
Punto	x	y
A	1.0	1.0
B	1.5	1.2
C	1.2	0.5
D	0.8	1.5
E	4.0	4.0
F	4.5	4.2
G	4.2	3.5
H	4.0	4.8
I	7.0	2.0
J	2.5	3.0



Paso 1: Visitar punto A

Seleccionamos A (1.0, 1.0). Buscamos vecinos con $\epsilon = 1.5$:

Par	dist
$d(A, B)$	0.54 ✓
$d(A, C)$	0.54 ✓
$d(A, D)$	0.54 ✓
$d(A, E)$	4.24
$d(A, J)$	2.50

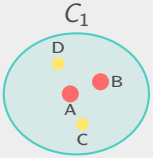


$|N_\epsilon(A)| = 3 \geq MinPts$
$\Rightarrow$ **A es núcleo**
Crear clúster $C_1 = \{A, B, C, D\}$

Paso 2: Expandir clúster C_1

Verificar vecinos de A: B, C, D. ¿Son núcleo?

Punto	Vecinos	¿Núcleo?
B	A, C, D	Sí ($3 \geq 3$)
C	A, B	No ($2 < 3$)
D	A, B	No ($2 < 3$)

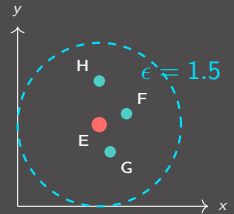


B es núcleo $\rightarrow$ expandir con sus vecinos.
Vecinos de B ya están en C_1.
$C_1 = \{A, B, C, D\}$ (completo)

Paso 3: Visitar punto E

Siguiente no visitado: **E (4.0, 4.0)**. Vecinos con $\epsilon = 1.5$:

Par	dist
$d(E, F)$	0.54 ✓
$d(E, G)$	0.54 ✓
$d(E, H)$	0.80 ✓
$d(E, J)$	1.80



$|N_\epsilon(E)| = 3 \geq MinPts$

$\Rightarrow$ **E es núcleo**

Crear clúster $C_2 = \{E, F, G, H\}$

Paso 4: Expandir clúster C_2 y visitar restantes

Expandir C_2:

Punto	Vecinos	¿Núcleo?
F	E, G, H	Sí
G	E, F	No
H	E, F	No

F es núcleo $\rightarrow$ sus vecinos ya están en C_2.
$C_2 = \{E, F, G, H\}$ (completo)

Puntos restantes:

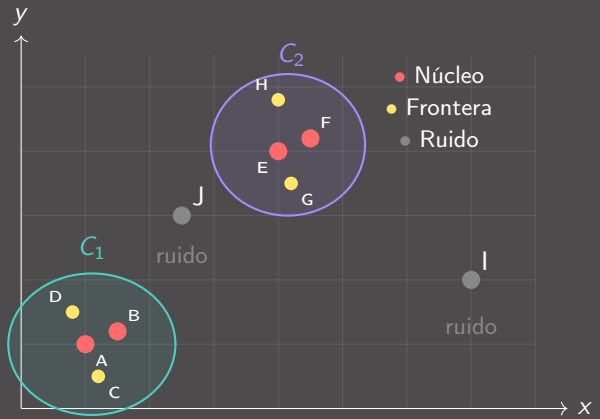
I (7.0, 2.0):

- Sin vecinos en $\epsilon = 1.5$
- $\rightarrow$ RUIDO

J (2.5, 3.0):

- Sin vecinos en $\epsilon = 1.5$
- $\rightarrow$ RUIDO

Resultado final del ejemplo



Resultado: 2 clústeres + 2 puntos de ruido. Parámetros: $\epsilon = 1.5$, $MinPts = 3$.

Resumen del ejemplo paso a paso

Paso	Acción	Resultado	Estado
1	Visitar A	$\|N_\epsilon\| = 3 \geq 3 \rightarrow$ núcleo	Crear C_1
2	Expandir A	Agregar B, C, D	$C_1 = \{A, B, C, D\}$
3	Verificar B	$\|N_\epsilon\| = 3 \geq 3 \rightarrow$ núcleo	Expandir (sin nuevos)
4	Verificar C,D	$\|N_\epsilon\| < 3 \rightarrow$ frontera	C_1 completo
5	Visitar E	$\|N_\epsilon\| = 3 \geq 3 \rightarrow$ núcleo	Crear C_2
6	Expandir E	Agregar F, G, H	$C_2 = \{E, F, G, H\}$
7	Verificar F	$\|N_\epsilon\| = 3 \rightarrow$ núcleo	Expandir (sin nuevos)
8	Verificar G,H	$\|N_\epsilon\| < 3 \rightarrow$ frontera	C_2 completo
9	Visitar I	$\|N_\epsilon\| = 0$	**Ruido**
10	Visitar J	$\|N_\epsilon\| = 0$	**Ruido**

Efecto de ϵ: Muy pequeño vs. muy grande

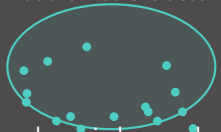
ϵ pequeño

Casi todo es ruido

ϵ demasiado pequeño

ϵ grande

Todo un solo clúster



ϵ demasiado grande

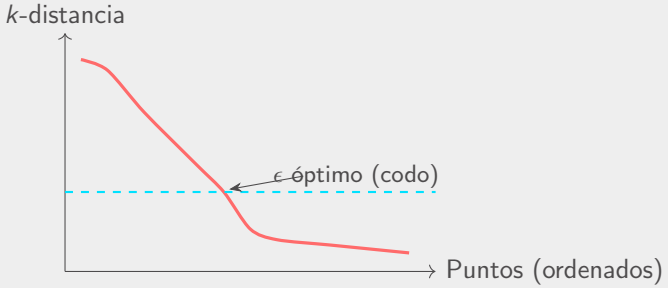
Regla práctica: Usar el gráfico de k-distancias ordenadas para elegir ϵ.

Elección de ϵ: Gráfico de k-distancias

Método: Para cada punto, calcular la distancia a su k-ésimo vecino más cercano ($k = MinPts$). Ordenar de mayor a menor y graficar.

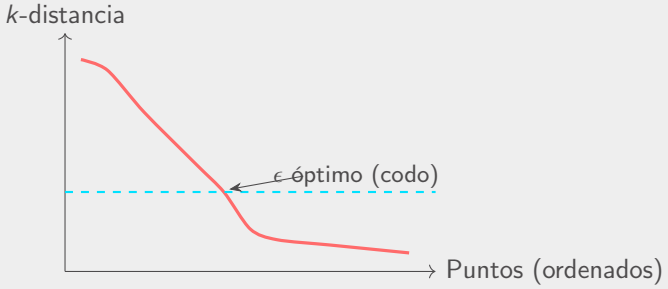
Elección de ϵ: Gráfico de k-distancias

Método: Para cada punto, calcular la distancia a su k-ésimo vecino más cercano ($k = MinPts$). Ordenar de mayor a menor y graficar.



Elección de ϵ: Gráfico de k-distancias

Método: Para cada punto, calcular la distancia a su k-ésimo vecino más cercano ($k = MinPts$). Ordenar de mayor a menor y graficar.



El **"codo"** de la curva indica un buen valor de ϵ: separa puntos densos (abajo) de ruido (arriba).

Efecto de *MinPts*

	MinPts bajo	*MinPts* alto
Sensibilidad	Más sensible al ruido	Más robusto al ruido
Clústeres	Más y más pequeños	Menos y más grandes
Ruido	Menos puntos de ruido	Más puntos de ruido
Regla	$MinPts \geq d + 1$	$MinPts = 2 \times d$

donde d es la dimensionalidad de los datos.

Recomendación: Empezar con $MinPts = 2 \times d$ y ajustar según resultados.

Complejidad computacional

Escenario	Tiempo	Espacio
Con índice espacial	$O(n \log n)$	$O(n)$
Sin índice espacial	$O(n^2)$	$O(n)$

Complejidad computacional

Escenario	Tiempo	Espacio
Con índice espacial	$O(n \log n)$	$O(n)$
Sin índice espacial	$O(n^2)$	$O(n)$

- **Índice espacial** (R-tree, kd-tree): acelera la búsqueda de vecinos

Complejidad computacional

Escenario	Tiempo	Espacio
Con índice espacial	$O(n \log n)$	$O(n)$
Sin índice espacial	$O(n^2)$	$O(n)$

- **Índice espacial** (R-tree, kd-tree): acelera la búsqueda de vecinos
- Sin índice: cada consulta de vecindario es $O(n) \rightarrow O(n^2)$ total

Complejidad computacional

Escenario	Tiempo	Espacio
Con índice espacial	$O(n \log n)$	$O(n)$
Sin índice espacial	$O(n^2)$	$O(n)$

- **Índice espacial** (R-tree, kd-tree): acelera la búsqueda de vecinos
- Sin índice: cada consulta de vecindario es $O(n) \rightarrow O(n^2)$ total
- **Ventaja sobre jerárquico:** no necesita la matriz $n \times n$ de distancias

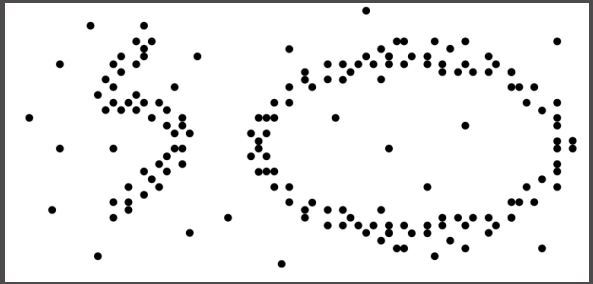
Complejidad computacional

Escenario	Tiempo	Espacio
Con índice espacial	$O(n \log n)$	$O(n)$
Sin índice espacial	$O(n^2)$	$O(n)$

- **Índice espacial** (R-tree, kd-tree): acelera la búsqueda de vecinos
- Sin índice: cada consulta de vecindario es $O(n) \rightarrow O(n^2)$ total
- **Ventaja sobre jerárquico:** no necesita la matriz $n \times n$ de distancias

Método	Tiempo	Espacio
K-means	$O(nKpI)$	$O(np)$
Jerárquico	$O(n^2 \log n)$	$O(n^2)$
DBSCAN (con índice)	$O(n \log n)$	$O(n)$

Clústeres de forma arbitraria



DBSCAN puede detectar clústeres con forma de “S”, anillos, espirales y cualquier forma arbitraria, algo imposible para K-means o métodos jerárquicos clásicos.

DBSCAN vs. K-means vs. Jerárquico

Aspecto	DBSCAN	K-means	Jerárquico
Parámetros	ϵ, *MinPts*	K	Enlace
Forma clusters	Arbitraria	Esférica	Depende enlace
Manejo ruido	Sí (nativo)	No	No
Núm. clusters	Automático	Fijo (K)	Dendrograma
Complejidad	$O(n \log n)$	$O(nKpI)$	$O(n^2 \log n)$
Densidad variable	No	No	Parcial

Ventajas y desventajas de DBSCAN

Ventajas

- Detecta clústeres de **forma arbitraria**

Desventajas

- No maneja bien **densidades variables**

Ventajas y desventajas de DBSCAN

Ventajas

- Detecta clústeres de **forma arbitraria**
- No requiere especificar K

Desventajas

- No maneja bien **densidades variables**
- Sensible a la elección de ϵ y *MinPts*

Ventajas y desventajas de DBSCAN

Ventajas

- Detecta clústeres de **forma arbitraria**
- No requiere especificar K
- **Robusto al ruido**: lo identifica explícitamente

Desventajas

- No maneja bien **densidades variables**
- Sensible a la elección de ϵ y *MinPts*
- Dificultad en **alta dimensionalidad** ("maldición")

Ventajas y desventajas de DBSCAN

Ventajas

- Detecta clústeres de **forma arbitraria**
- No requiere especificar K
- **Robusto al ruido**: lo identifica explícitamente
- Un solo recorrido de los datos

Desventajas

- No maneja bien **densidades variables**
- Sensible a la elección de ϵ y *MinPts*
- Dificultad en **alta dimensionalidad** ("maldición")
- Puntos frontera pueden asignarse a clústeres distintos según el orden

Ventajas y desventajas de DBSCAN

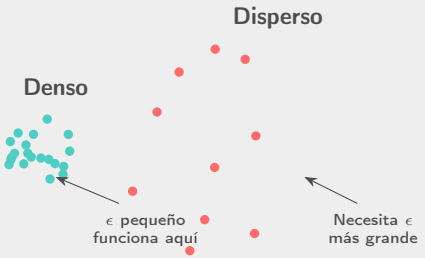
Ventajas

- Detecta clústeres de **forma arbitraria**
- No requiere especificar K
- **Robusto al ruido**: lo identifica explícitamente
- Un solo recorrido de los datos
- Eficiente con índices espaciales: $O(n \log n)$

Desventajas

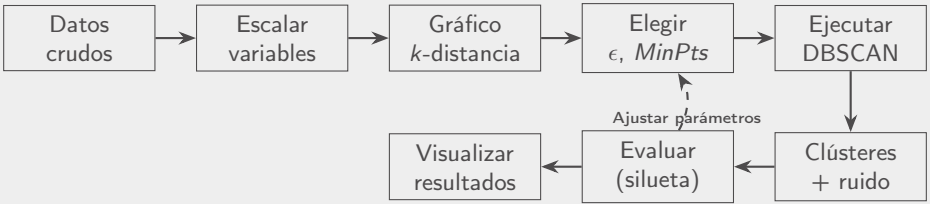
- No maneja bien **densidades variables**
- Sensible a la elección de ϵ y *MinPts*
- Dificultad en **alta dimensionalidad** ("maldición")
- Puntos frontera pueden asignarse a clústeres distintos según el orden
- $O(n^2)$ sin índice espacial

Limitación: Densidades variables



Un solo ϵ global no puede capturar ambos clústeres simultáneamente.
Solución: Usar diferentes valores de ϵ por región o métodos de densidad adaptativa.

Pipeline de DBSCAN



Conclusiones

Resumen y reflexiones **sobre DBSCAN**

Resumen: DBSCAN

1. **Basado en densidad:** clústeres = regiones densas separadas por regiones dispersas

Resumen: DBSCAN

1. **Basado en densidad:** clústeres = regiones densas separadas por regiones dispersas
2. **Dos parámetros:** ϵ (radio) y *MinPts* (umbral de densidad)

Resumen: DBSCAN

1. **Basado en densidad:** clústeres = regiones densas separadas por regiones dispersas
2. **Dos parámetros:** ϵ (radio) y *MinPts* (umbral de densidad)
3. **Tres tipos de puntos:** núcleo, frontera, ruido

Resumen: DBSCAN

1. **Basado en densidad:** clústeres = regiones densas separadas por regiones dispersas
2. **Dos parámetros:** ϵ (radio) y *MinPts* (umbral de densidad)
3. **Tres tipos de puntos:** núcleo, frontera, ruido
4. **Formas arbitrarias:** no asume clústeres esféricos

Resumen: DBSCAN

1. **Basado en densidad:** clústeres = regiones densas separadas por regiones dispersas
2. **Dos parámetros:** ϵ (radio) y *MinPts* (umbral de densidad)
3. **Tres tipos de puntos:** núcleo, frontera, ruido
4. **Formas arbitrarias:** no asume clústeres esféricos
5. **Número automático:** no requiere fijar K

Resumen: DBSCAN

1. **Basado en densidad:** clústeres = regiones densas separadas por regiones dispersas
2. **Dos parámetros:** ϵ (radio) y *MinPts* (umbral de densidad)
3. **Tres tipos de puntos:** núcleo, frontera, ruido
4. **Formas arbitrarias:** no asume clústeres esféricos
5. **Número automático:** no requiere fijar K
6. **Eficiente:** $O(n \log n)$ con índice espacial

Resumen: DBSCAN

1. **Basado en densidad:** clústeres = regiones densas separadas por regiones dispersas
2. **Dos parámetros:** ϵ (radio) y *MinPts* (umbral de densidad)
3. **Tres tipos de puntos:** núcleo, frontera, ruido
4. **Formas arbitrarias:** no asume clústeres esféricos
5. **Número automático:** no requiere fijar K
6. **Eficiente:** $O(n \log n)$ con índice espacial
7. **Limitación:** un solo ϵ global (densidad uniforme)

¿Cuándo usar DBSCAN?

Escenario	Recomendación
Clústeres no convexos	**DBSCAN**
Datos con ruido/outliers	**DBSCAN**
No se conoce K	**DBSCAN** o jerárquico
Clústeres esféricos, N grande	K-means
Se necesita jerarquía	Jerárquico
Densidades muy variables	Ajustar ϵ por región
Alta dimensionalidad	Reducir dimensiones primero

Ideas clave para recordar

- DBSCAN conecta **objetos núcleo** y sus vecindarios para formar clústeres

Ideas clave para recordar

- DBSCAN conecta **objetos núcleo** y sus vecindarios para formar clústeres
- Los puntos que no están en ninguna región densa son **ruido**

Ideas clave para recordar

- DBSCAN conecta **objetos núcleo** y sus vecindarios para formar clústeres
- Los puntos que no están en ninguna región densa son **ruido**
- La elección de ϵ se puede guiar con el **gráfico de k-distancias**

Ideas clave para recordar

- DBSCAN conecta **objetos núcleo** y sus vecindarios para formar clústeres
- Los puntos que no están en ninguna región densa son **ruido**
- La elección de ϵ se puede guiar con el **gráfico de k-distancias**
- $MinPts \geq d + 1$ es una buena regla inicial

Ideas clave para recordar

- DBSCAN conecta **objetos núcleo** y sus vecindarios para formar clústeres
- Los puntos que no están en ninguna región densa son **ruido**
- La elección de ϵ se puede guiar con el **gráfico de k-distancias**
- $MinPts \geq d + 1$ es una buena regla inicial
- Para **densidades variables**, considerar extensiones con densidad adaptativa

Ideas clave para recordar

- DBSCAN conecta **objetos núcleo** y sus vecindarios para formar clústeres
- Los puntos que no están en ninguna región densa son **ruido**
- La elección de ϵ se puede guiar con el **gráfico de k-distancias**
- $MinPts \geq d + 1$ es una buena regla inicial
- Para **densidades variables**, considerar extensiones con densidad adaptativa
- DBSCAN es **determinístico** (salvo puntos frontera en el borde entre clústeres)

Referencias

- Ester, Kriegel, Sander & Xu (1996). *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. KDD.
- Han, Kamber & Pei (2012). *Data Mining: Concepts and Techniques*, Cap. 10.4.

UTEC Posgrado

UTEC Posgrado