



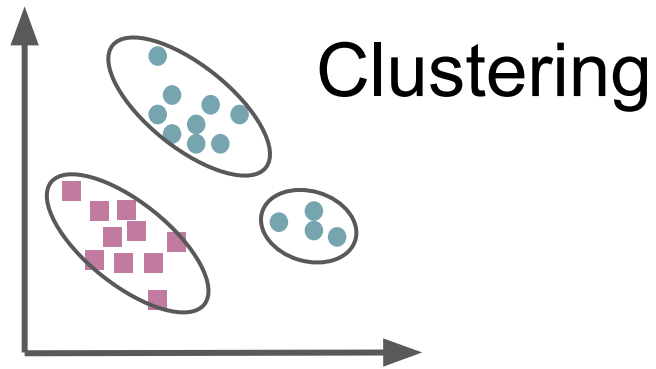
Curso DM Clustering

Parte 2: Algoritmos e Implementación

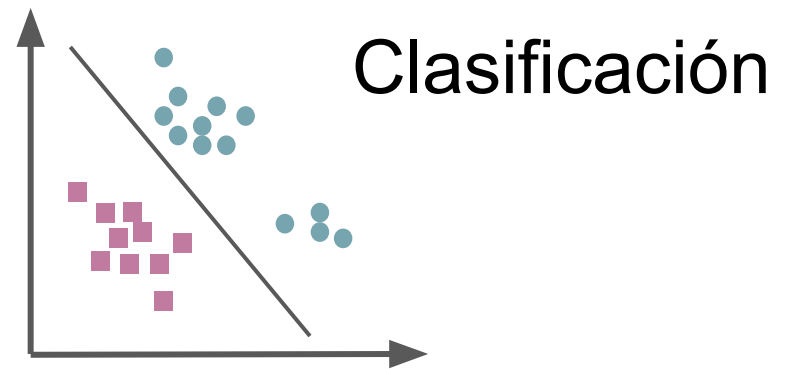
Primavera 2023

Profesoras: Jazmine Maldonado y Cinthia Sánchez

Basada en versiones previas de Felipe Bravo y Bárbara Poblete



Aprendizaje
No-Supervisado



Aprendizaje
Supervisado

Diferentes Métodos de Clustering

K-Means

Fuzzy C-means

DBSCAN

Mezcla de Gaussianas
y algoritmo EM

Método
jerárquico
aglomerativo

Mapas
auto-organizados de
Kohonen (SOM)

Diferentes Métodos de Clustering

K-Means

Fuzzy C-means

DBSCAN

Mezcla de Gaussianas
y algoritmo EM

**Método
jerárquico
aglomerativo**

Mapas
auto-organizados de
Kohonen (SOM)

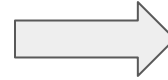
Dataset de
atributos
numéricos

Número de
clusters K



K-Means

Algoritmo de
clustering
particional

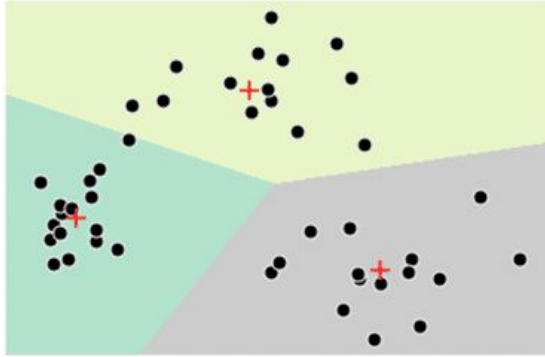


K clusters y todos
los puntos fueron
asignados a
alguno de los
clusters.

¿Cómo funciona?

1. Se asignan K centroides iniciales.
2. Itera hasta converger:
 - a. Asignar puntos a su centroide más cercano
 - b. Recalcular centroides promediando puntos.

¿Qué es un centroide?



Sean 3 vectores de 3 dimensiones:

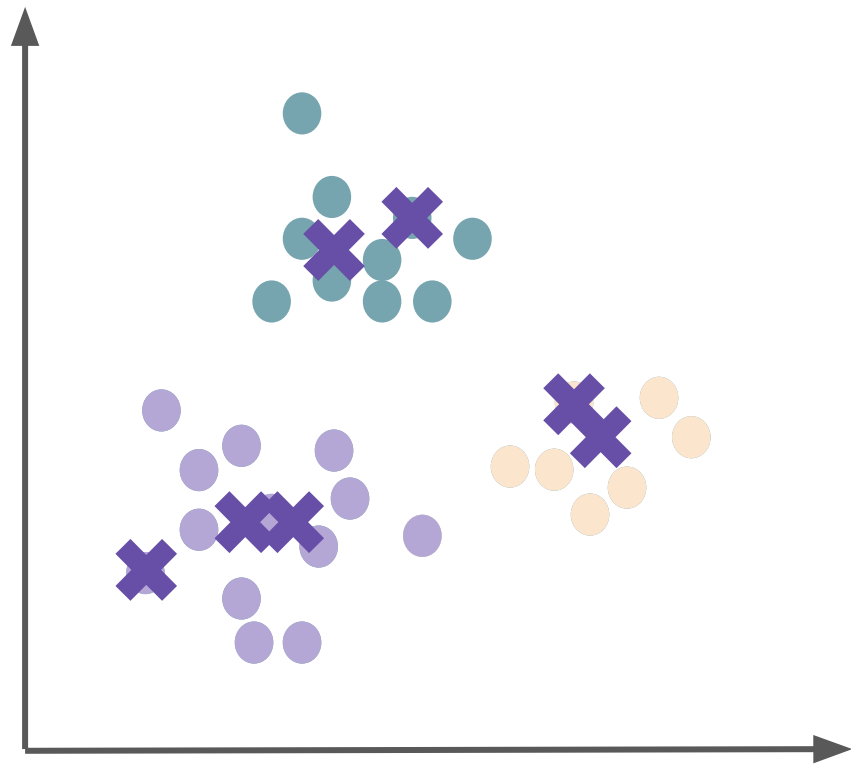
$$X1 = [6,4,3]$$

$$X2 = [4,5,1]$$

$$X3 = [2,-3,5]$$

El centroide de estos vectores es:

$$C(X1, X2, X3) = [(6+4+2)/3, (4+5-3)/3, (3+1+5)/3] = [4,2,3]$$



Otras variables a tener en cuenta

- ¿Cómo se escogen centroides iniciales?

El enfoque tradicional es escogerlos aleatoriamente

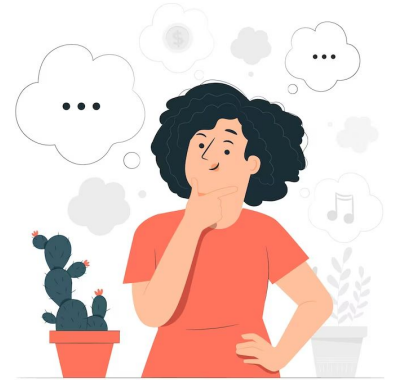
- ¿Qué medida de distancia usar?

Generalmente se usa distancia Euclideana

- ¿Cómo estimar la cantidad de clusters?

Podemos usar una técnica visual: Método del codo.

¿Qué consecuencias tiene el que los clusters iniciales se asignen aleatoriamente?



SSE: Sum Square Error

Mide la varianza de un cluster

Suma de las distancias cuadradas de cada punto al centroide de su cluster asignado.

$$SSE = \sum_{i=1}^K \sum_{\mathbf{x} \in C_i} dist(\mathbf{c}_i, \mathbf{x})^2$$



SSE: Sum Square Error

Mide la varianza de un cluster

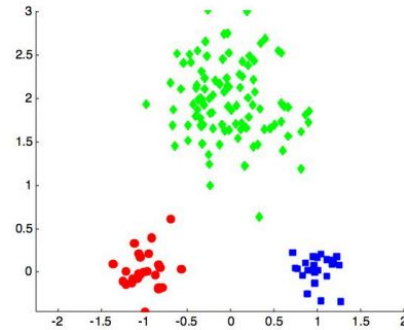
Permite calcular el aporte individual de cada cluster al SSE total.

Permite juzgar si un cluster es bueno o no.

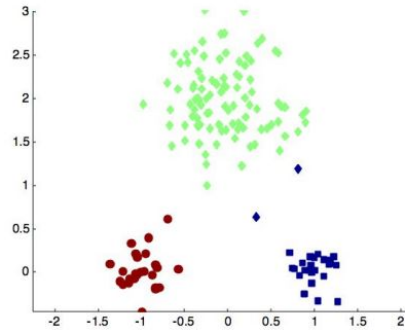


Limitantes

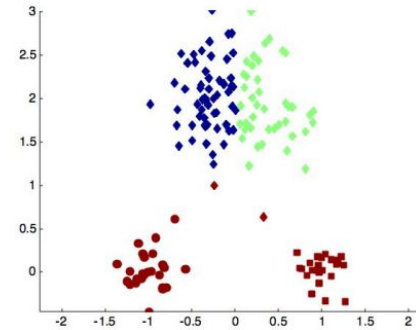
K-means no
asegura encontrar
clusters óptimos



Puntos originales



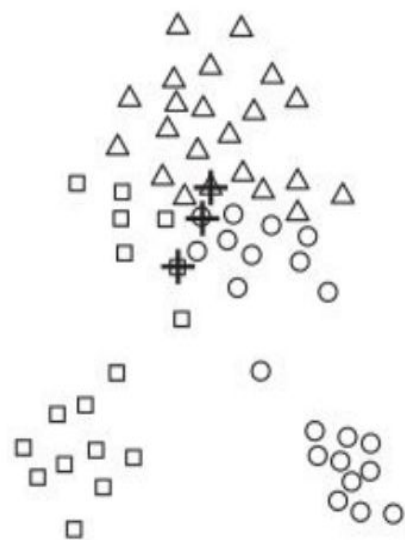
Clustering óptimo



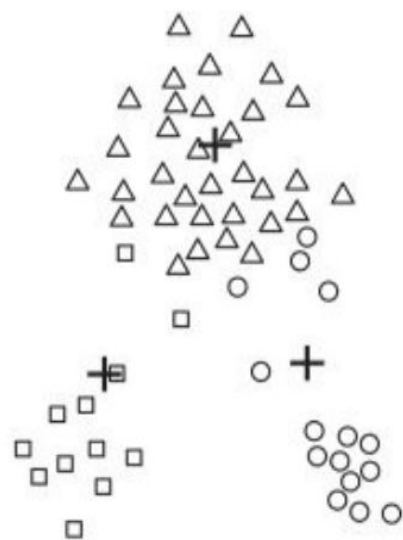
Clustering sub-optimal

Comparación de inicializaciones

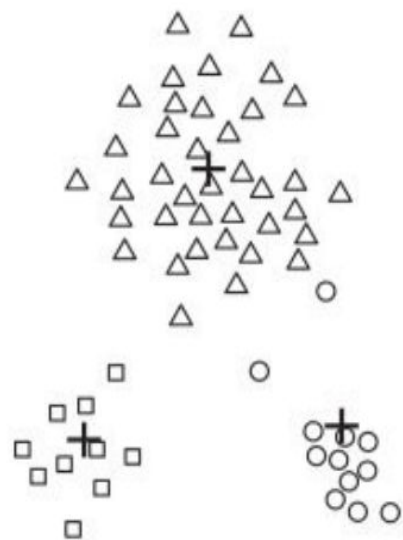
Caso 1:



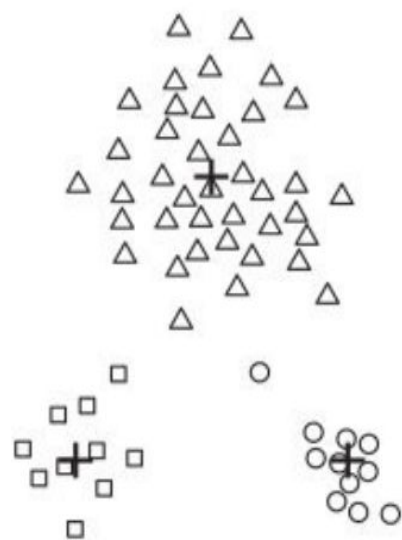
(a) Iteration 1.



(b) Iteration 2.

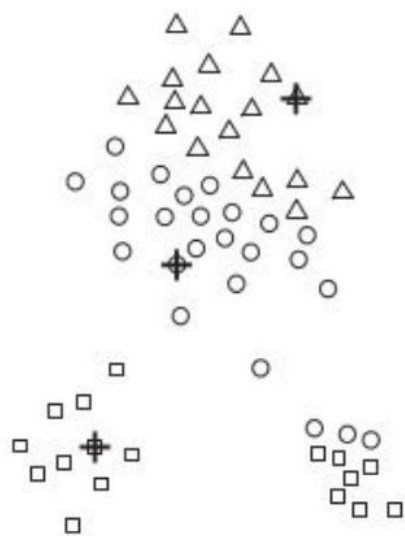


(c) Iteration 3.

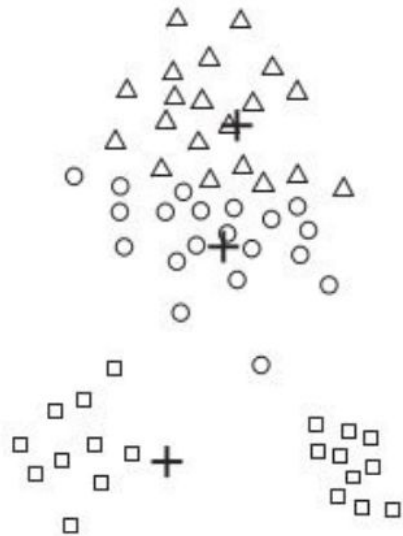


(d) Iteration 4.

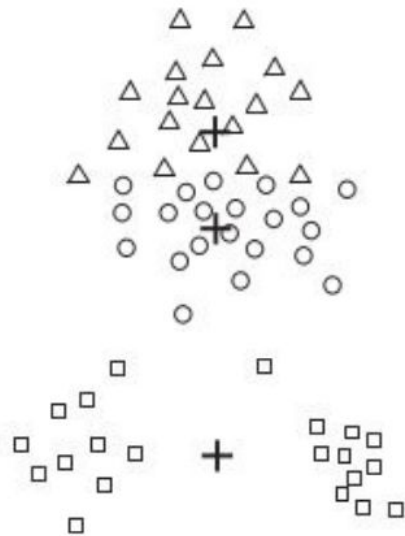
Caso 2:



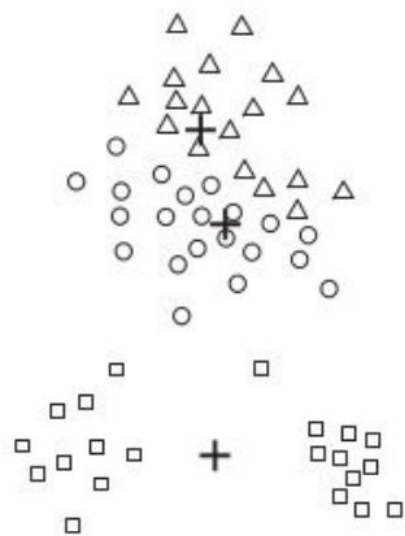
(a) Iteration 1.



(b) Iteration 2.



(c) Iteration 3.



(d) Iteration 4.

La probabilidad de escoger un centroide por cada cluster óptimo es baja.

$$P = \frac{\text{\# formas de escoger un centroide de cada cluster}}{\text{\# formas de escoger K centroides}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

Ej.: si $K = 10$, entonces $P = 10!/10^{10} = 0.00036$

También puede pasar que queden clusters vacíos

Esto pasa si es que no se le asignan puntos al cluster en el paso de asignación.



Algunas soluciones a estos problemas

Ejecutar K-means varias veces variando la semilla aleatoria y quedarse con el modelo de menor SSE.

Reemplazar centroides de clusters vacíos escogiendo el punto más lejano a todos los centroides como nuevo centroide u otro punto aleatorio con mayor SSE.

K-means: ¿Cómo estimar la cantidad de clusters?

K-Means necesita este valor al momento de correr el algoritmo, no podemos dejarlo al azar.

Técnica visual: **Método del codo:**

1. Calculamos el SSE para varios números de clusters*.
2. Graficamos cómo varía el SSE.
3. Elegimos el "mejor". La idea es elegir el último cluster antes de encontrarnos con el punto de *diminishing returns*, que sería cuando aumentar a más clusters nos da una mejora muy pequeña.

***Sum of Squared Error (SSE):** *scikit-learn* este dato se llama `inertia_`

```
sse = []

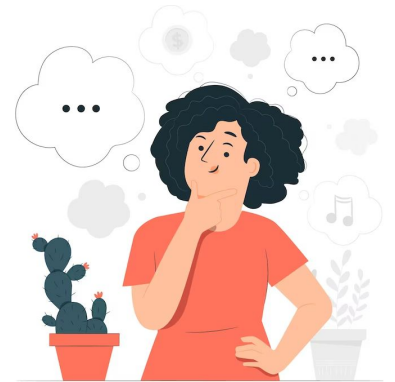
clusters = list(range(1, 16))
for k in clusters:
    kmeans = KMeans(n_clusters=k).fit(X)
    sse.append(kmeans.inertia_)

plt.plot(clusters, sse, marker="o")
plt.title("Metodo del codo de 1 a 15 clusters")
plt.grid(True)
plt.show()
```

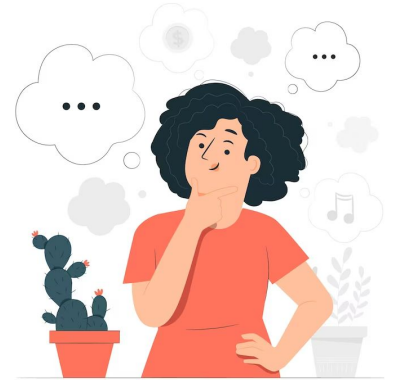


Ojo que este método es una heurística y no siempre el codo es claramente visible.

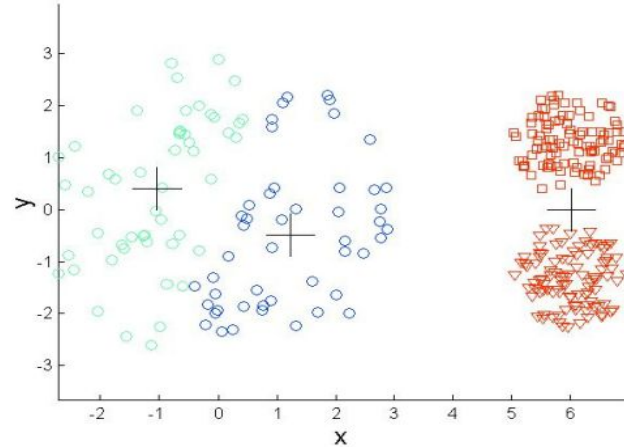
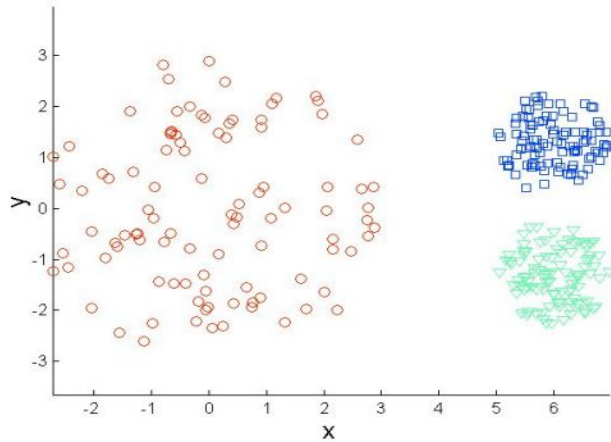
¿Cómo afectan los outliers a K-Means?



¿Qué pasa si los datos no están normalizados?

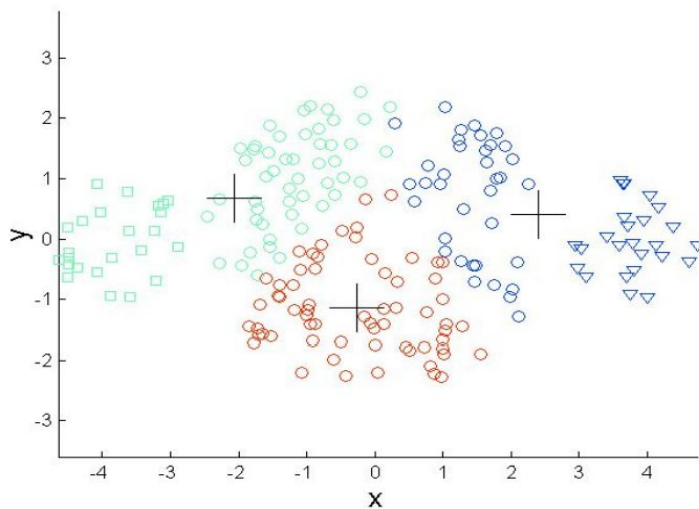
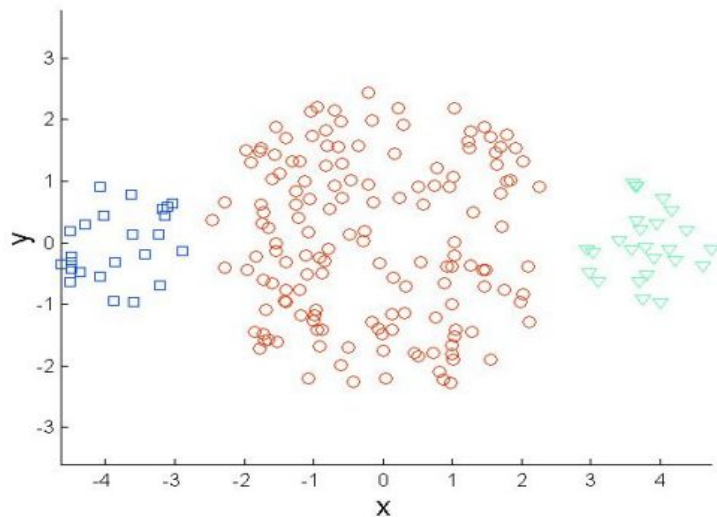


Otras limitantes a considerar



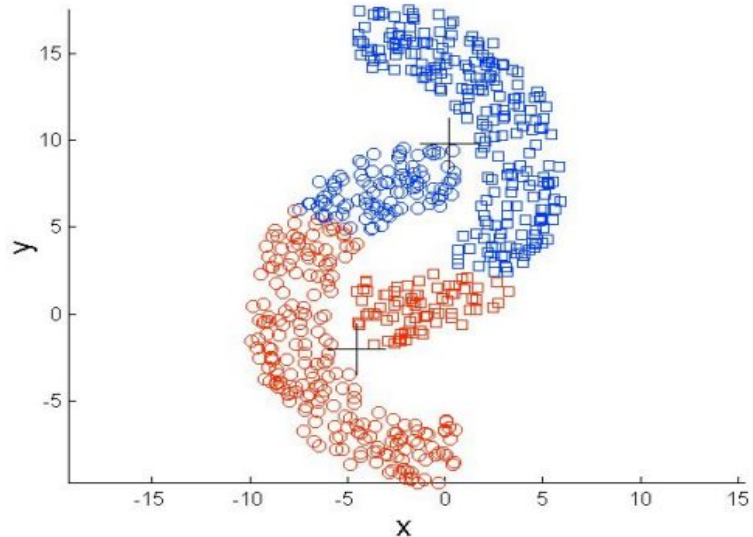
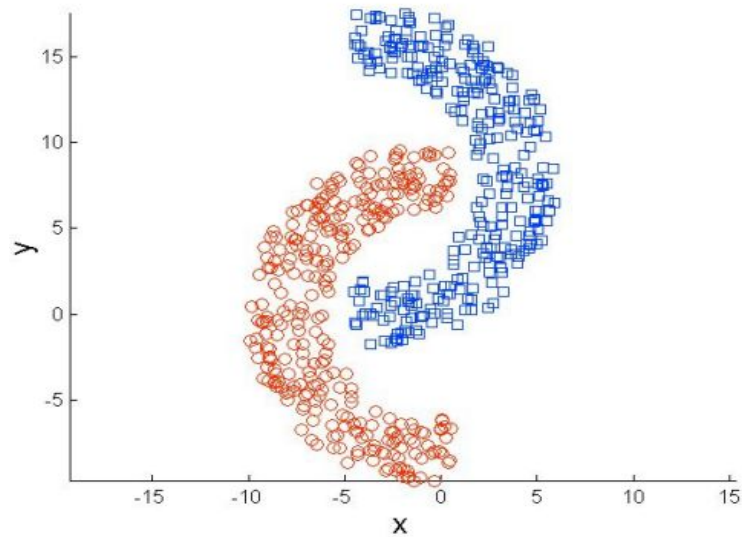
K-means tiene limitaciones cuando los clusters tienen diferentes densidades.

Otras limitantes a considerar



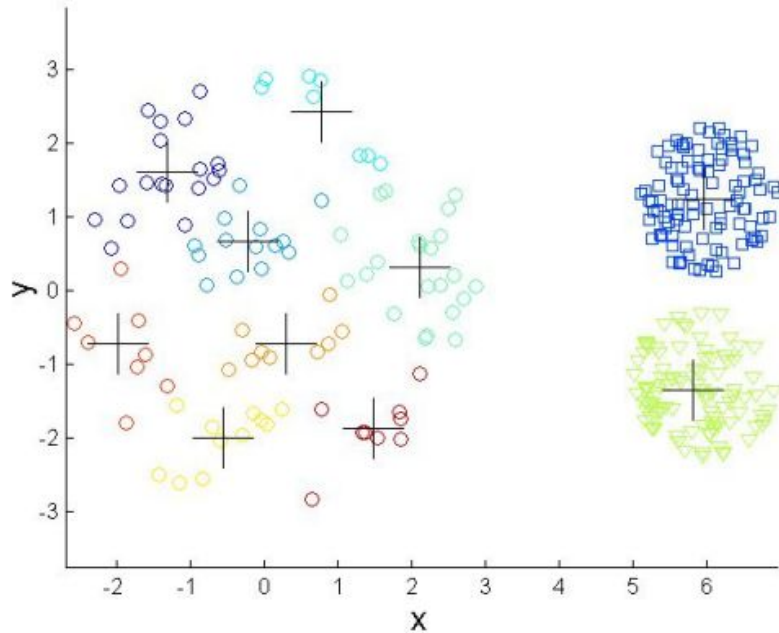
O tamaños diferentes.

Otras limitantes a considerar



O formas no esféricas

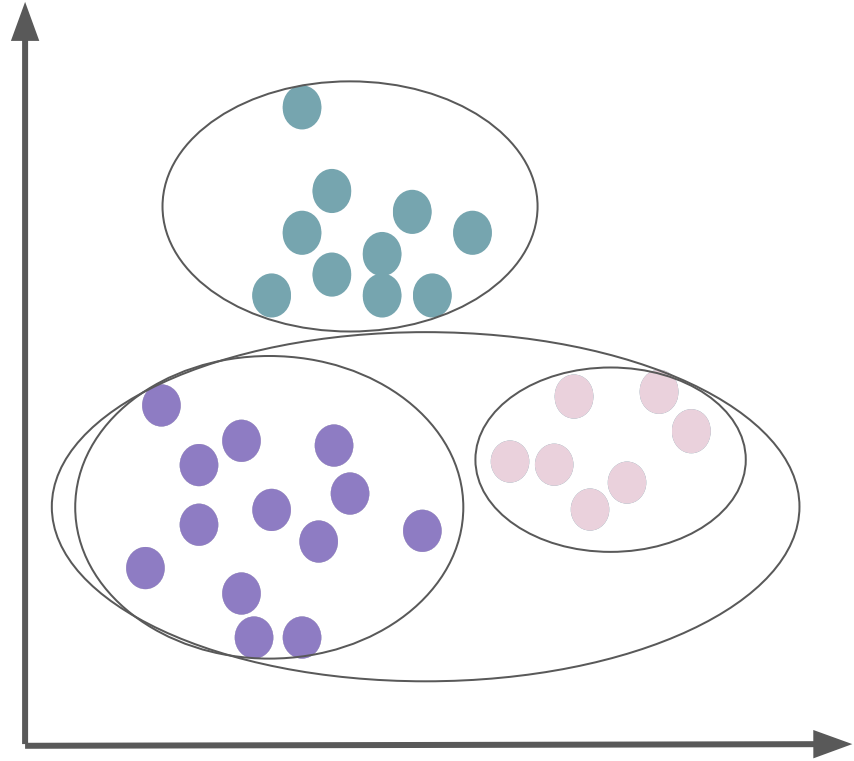
Técnica de post-procesamiento para estos casos



Usar un K más alto y luego mezclar los clusters durante una fase de post-procesamiento.

Bisecting K-means

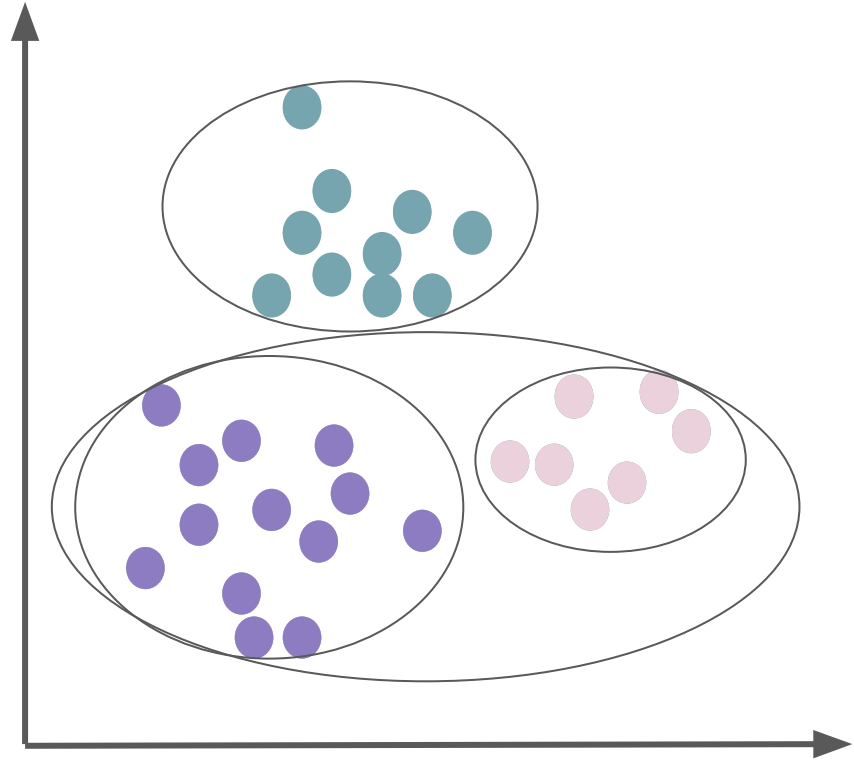
Extensión de K-means que se basa en dividir el conjunto siempre en 2 clusters, escoger uno de ellos y continuar dividiendo en 2, iterando hasta llegar a K clusters.



Bisecting K-means

Cada división se obtiene ejecutando K-means ($k=2$)

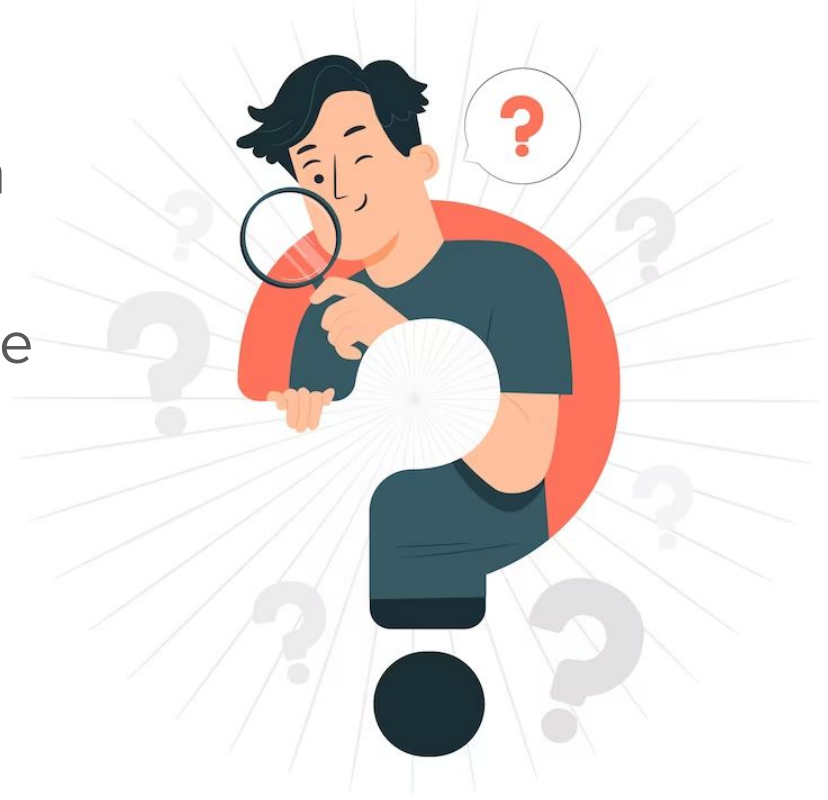
El siguiente cluster a dividir se puede escoger considerando su tamaño, su SSE o una estrategia híbrida.



Bisecting K-means

Menos problemas de inicialización que K-means ya que realiza varios intentos de bisección y toma la que minimiza el SSE.

Si se registra la secuencia de clusters bisectados podemos producir un clustering jerárquico.



Dataset de
atributos
numéricos

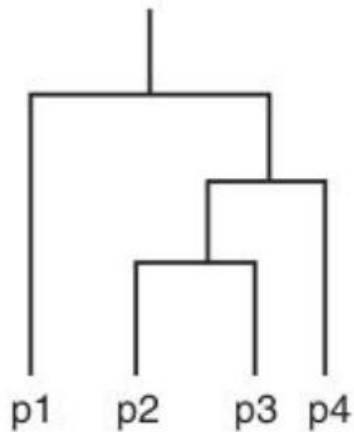


**Clustering
Jerárquico
Aglomerativo**

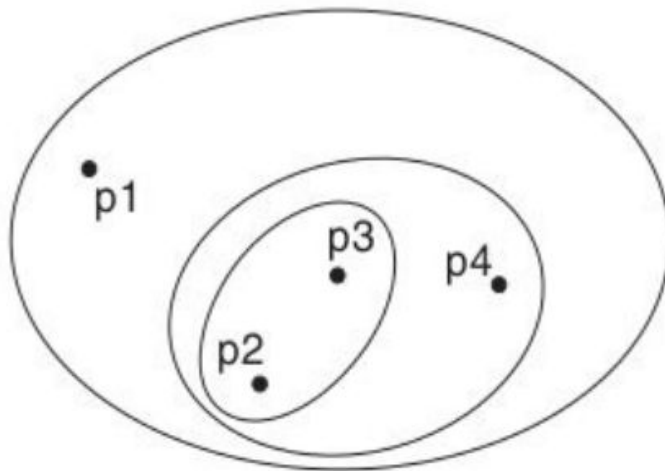


Clusters anidados
organizados en un
árbol jerárquico.

Visualizaciones



(a) Dendrogram.



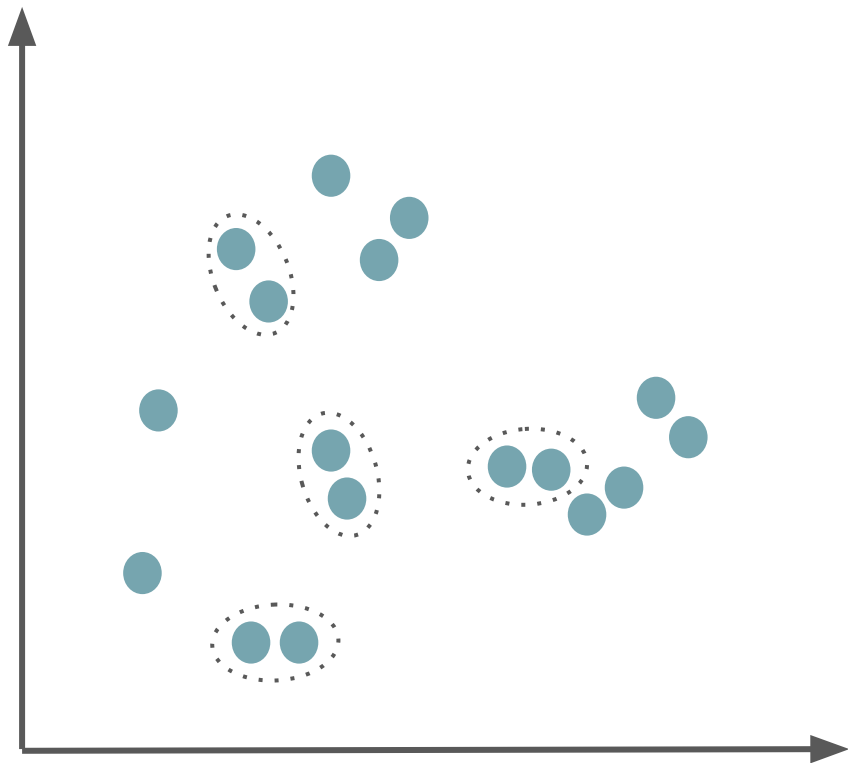
(b) Nested cluster diagram.

Tipos de Clustering Jerárquico

Aglomerativo	Divisivo
<p data-bbox="156 426 925 532">Empieza con cada punto como un cluster individual.</p> <p data-bbox="156 609 925 784">En cada paso mezcla el par de clusters más cercanos hasta que queda un solo cluster o k clusters.</p>	<p data-bbox="966 426 1734 536">Empieza con un cluster que contiene todos los puntos.</p> <p data-bbox="966 609 1734 841">En cada paso divide un cluster en dos hasta que todo cluster contenga un solo punto (o haya k clusters)</p>







¿Cómo funciona el algoritmo aglomerativo?



1. Calcular matriz de distancias
2. Sea cada punto un cluster
3. Iterar hasta que quede un cluster
 - a. Mezclar par de clusters más cercanos
 - b. Actualizar matriz de distancias



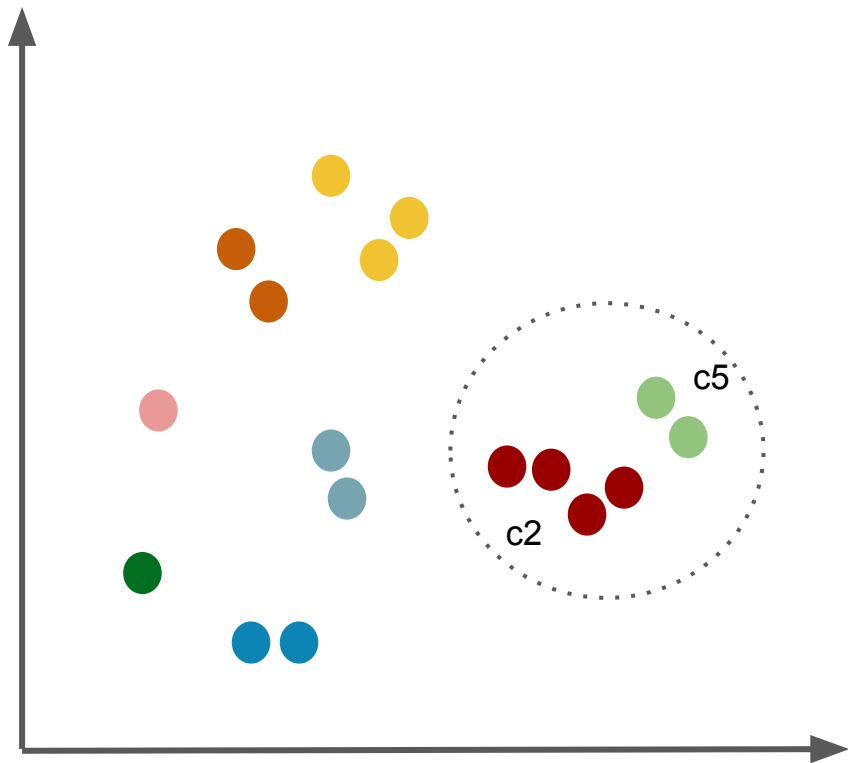
	<i>p1</i>	<i>p2</i>	<i>p3</i>	<i>p4</i>	<i>p5</i>	...
<i>p1</i>						
<i>p2</i>						
<i>p3</i>						
<i>p4</i>						
<i>p5</i>						

Matriz de distancias

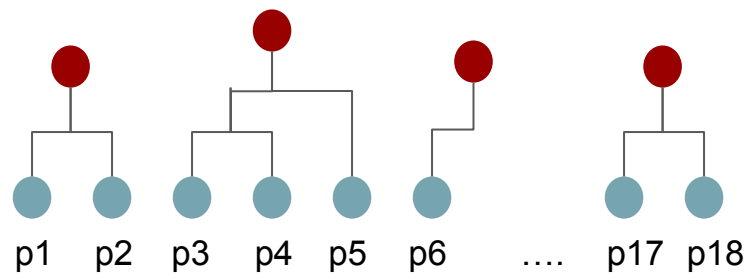
 


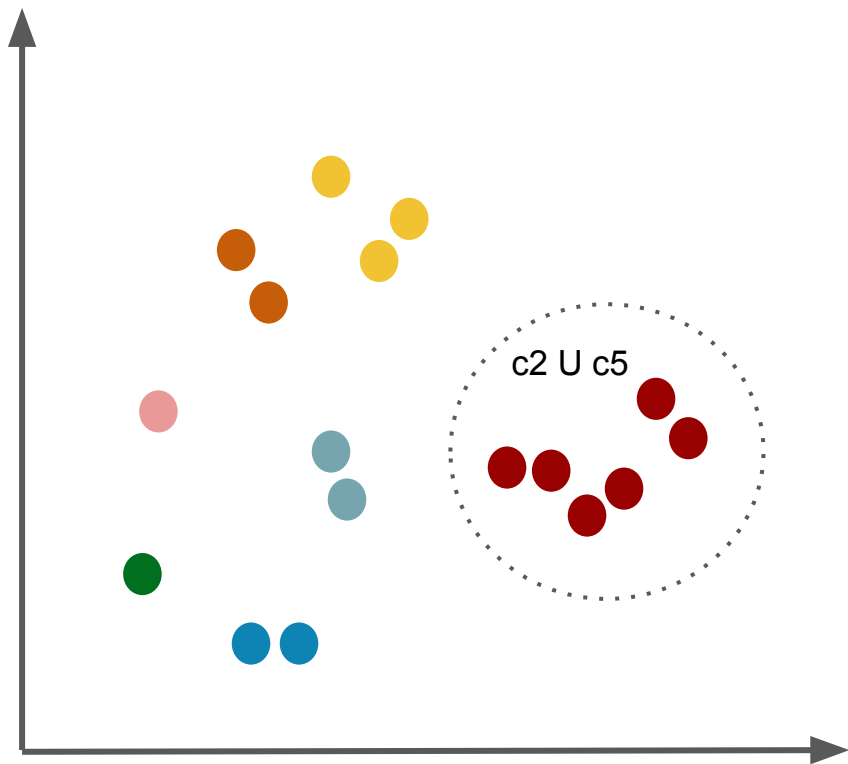
p1 *p2* *p3* *p4* *p5* *p6* *p17* *p18*



	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

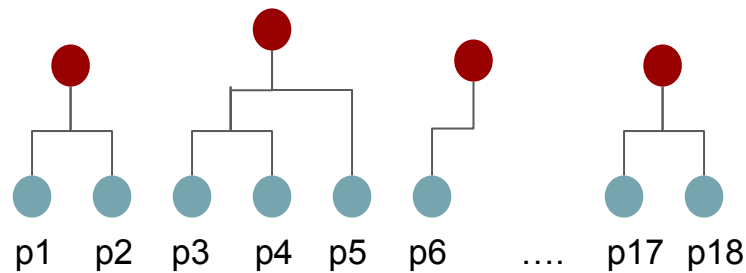
Matriz de distancias



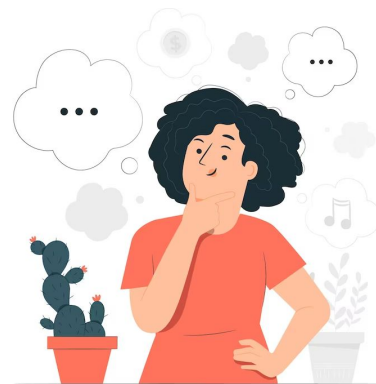


	C2 U			
	C1	C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Matriz de distancias



¿Cómo calcular cuáles son los clusters más cercanos?



Existen varias métricas de distancia entre clusters

MIN (single link)

MAX (complete link)

Promedio del grupo

Método de Ward

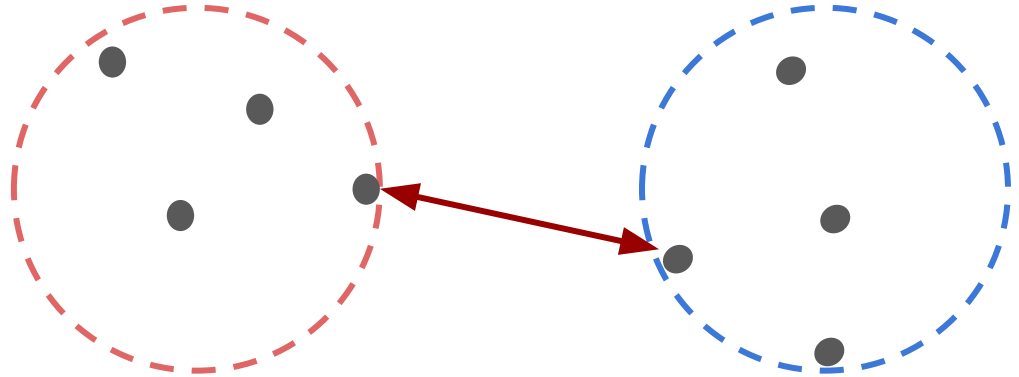


Método MIN (single link)

Considero los dos puntos más cercanos entre sí (cada uno de un cluster distinto)

Fortaleza: Puede manejar formas no elípticas.

Limitante: Sensible al ruido y outliers

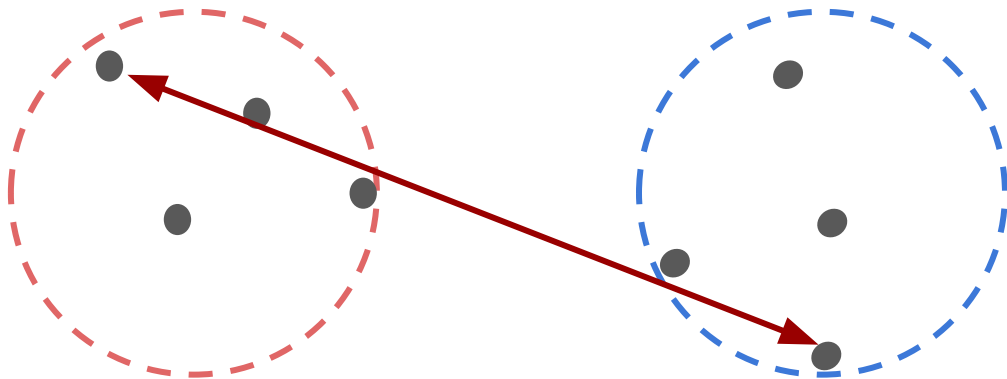


Método MAX (complete link)

Considero los dos puntos más lejanos entre sí (cada uno de un cluster distinto)

Fortaleza: Menos susceptible al ruido y outliers.

Limitante: Tiende a quebrar clusters grandes y es sesgado a clusters esféricos.

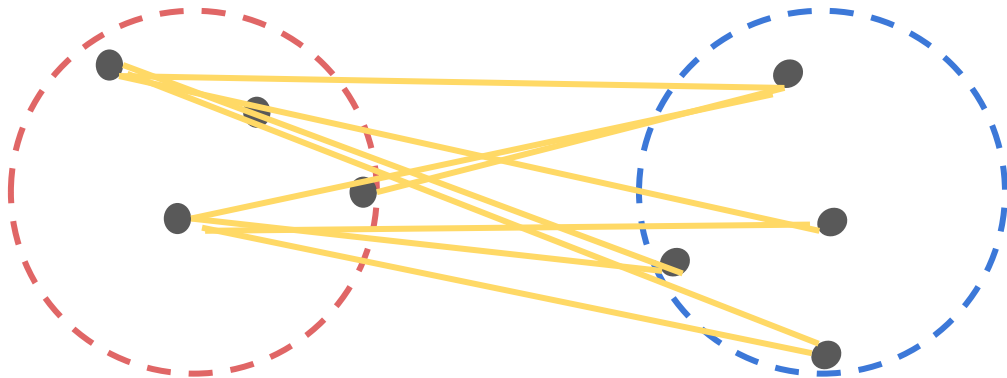


Método Promedio del grupo

Distancia promedio de todos los pares de puntos (cada par tiene un punto por cluster)

Fortaleza: Menos susceptible al ruido y outliers.

Limitante: sesgado a clusters esféricos.

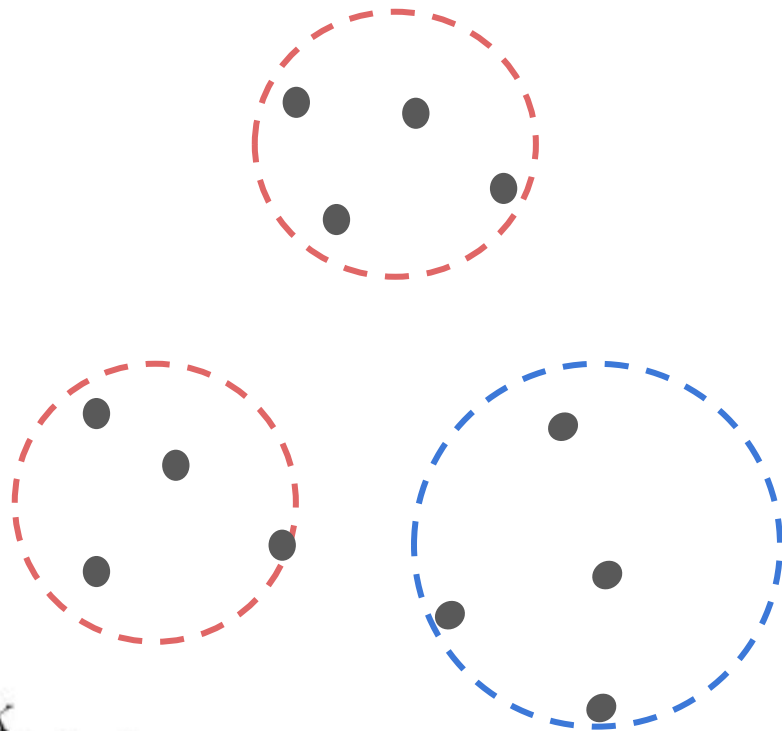


Método Ward

Se basa en el incremento de SSE cuando se mezclan dos clusters.

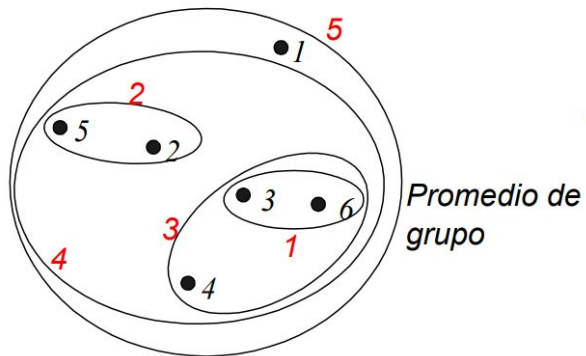
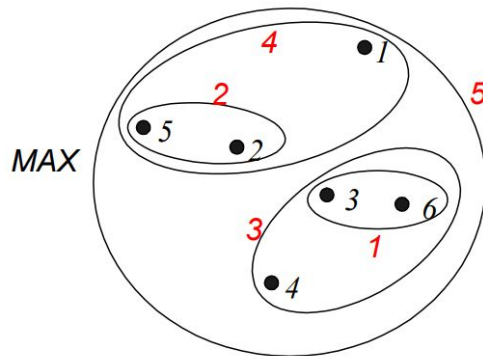
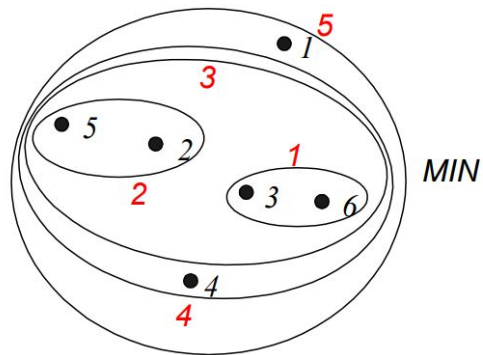
Fortaleza: Menos susceptible al ruido y outliers.

Limitante: sesgado a clusters esféricos.

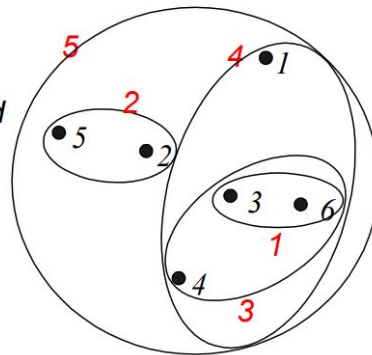


$$SSE = \sum_{i=1}^K \sum_{\mathbf{x} \in C_i} dist(\mathbf{c}_i, \mathbf{x})^2$$

Distintos clusters dependiendo del método utilizado



Método de Ward



Problemas y limitaciones

- No hay una función objetivo que sea directamente minimizada.
- Sensibles a ruido y outliers.
- Dificultad para manejar clusters de distinto tamaño.
- Pueden romper clusters grandes.
- No escala muy bien.



Dataset de
atributos
numéricos

Eps: Radio
especificado

MinPts:
Número min
de puntos en
una región



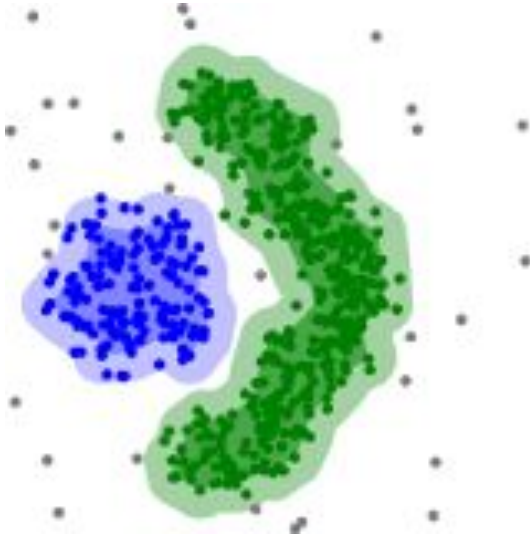
DBSCAN

Algoritmo de
clustering
basado en
densidad



Conjunto de
clusters densos y
puntos que no
quedaron en
ningún cluster
(outliers)

¿Qué es la densidad?



La densidad de un punto es el número de puntos que tiene dentro de un radio dado.

¿Cómo funciona?

1. Etiqueta todos los puntos en **core**, **border** o **noise**
2. Elimina los puntos **noise**
3. Asigna un arco entre los puntos **core** que se encuentran a una distancia menor o igual a Eps
4. Asigna cada grupo de puntos **core** conectados a un cluster separado
5. Asigna los puntos **border** a uno de los clusters de los puntos **core** asociados

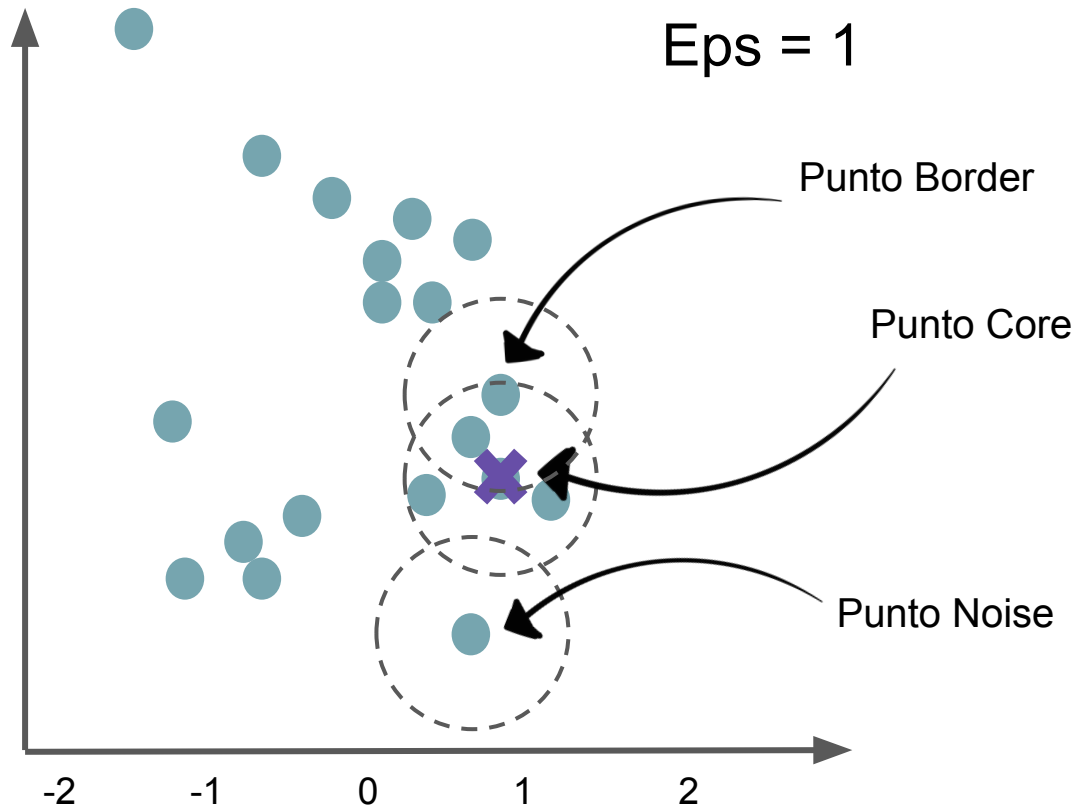
Tipos de puntos

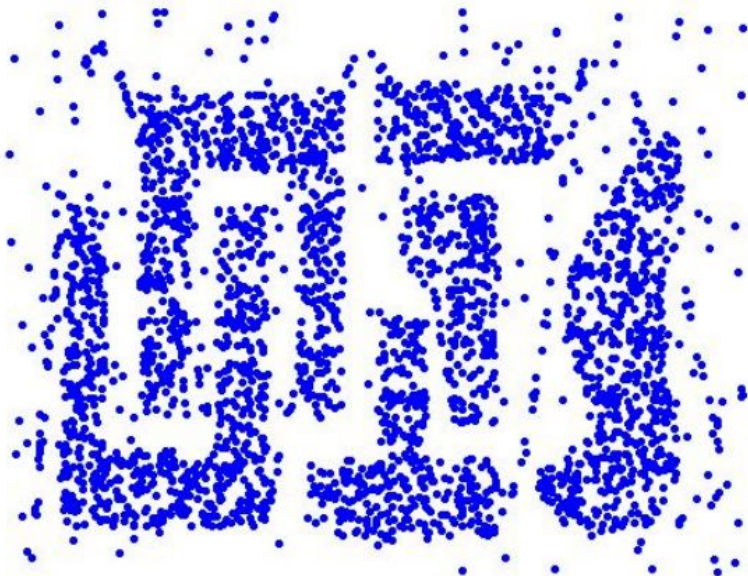
Considerando los parámetros del modelo *Eps* y *MinPts*, existen tres tipos de puntos:

- Punto “**core**”: punto con más de *MinPts* puntos a distancia *Eps*
- Punto “**border**”: punto con menos de *MinPts* puntos en el radio *Eps*, pero está en la vecindad de un punto core.
- Punto “**noise**”: cualquier punto que no es core ni border.

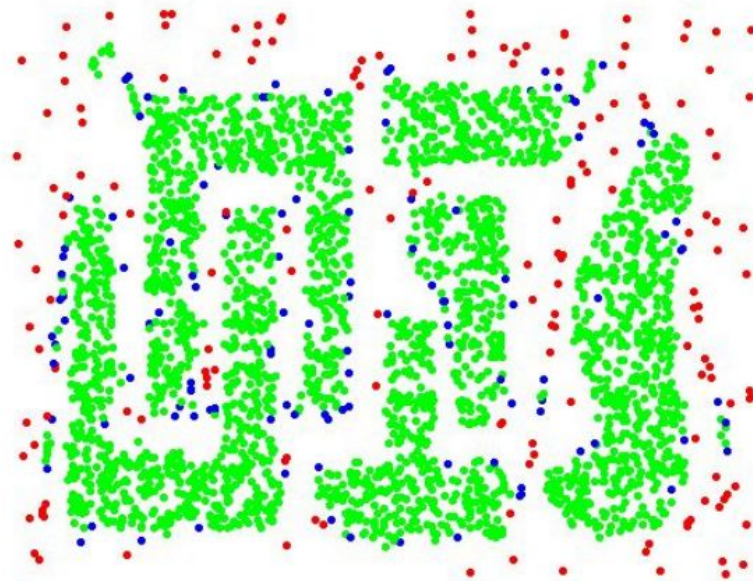


MinPts = 4
Eps = 1



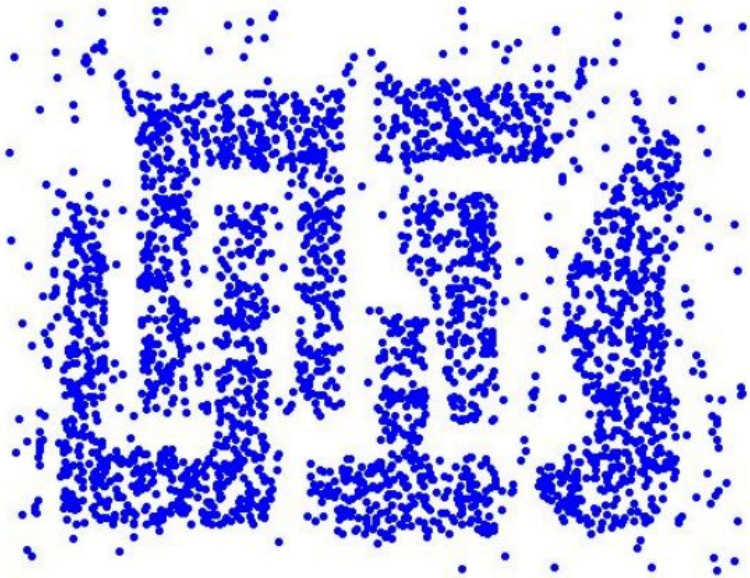


Puntos originales

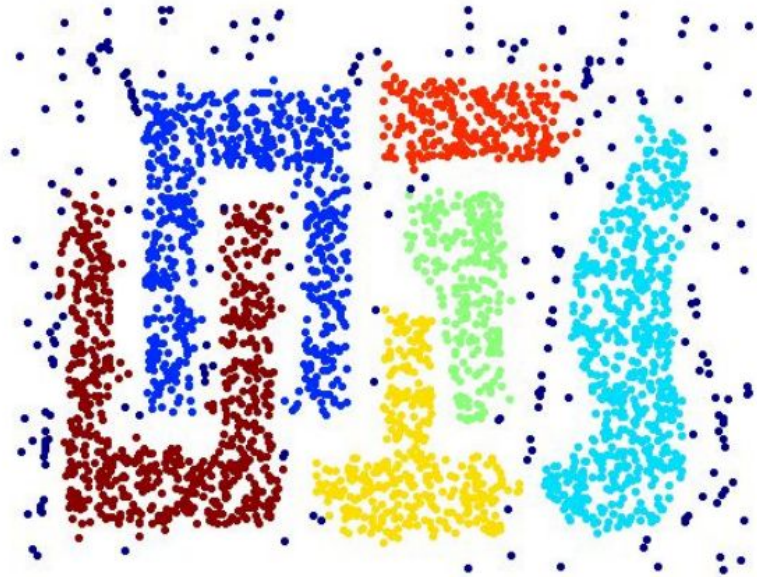


*Tipos de punto: core,
border y noise*

Eps = 10, MinPts = 4



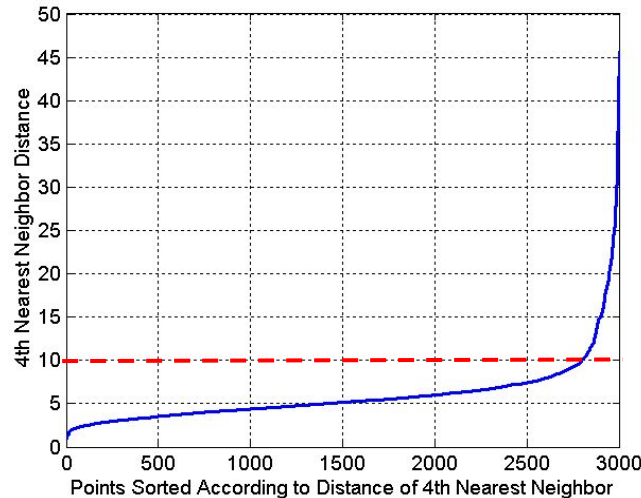
Puntos originales



Clusters

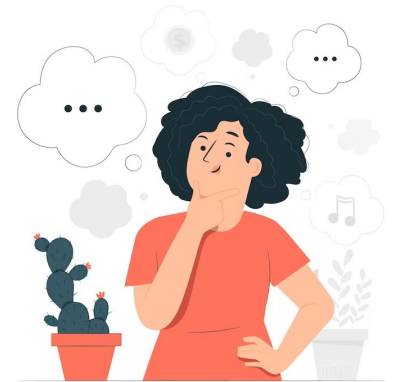
DBSCAN: Determinando Eps y MinPts

Calculamos el valor de **k-dist** para todos los puntos con un valor de k fijo y los ordenamos de manera creciente. Graficamos k-dist vs la cantidad de puntos con ese valor.



Eps seleccionado: valor de k-dist cuando ocurre el salto MinPts: el valor de k.

¿Cuáles serían las fortalezas de este algoritmo de clustering?



Problemas y limitaciones

- No funciona bien con densidades variables. Clusters de baja densidad se confunden con ruido.
- No funciona bien con datos de alta dimensionalidad.



Implementación



<https://colab.research.google.com/drive/12gYiYvM65wvqjjqbnzsoIxebDYSCTy2b?usp=sharing>