

Módulo: Aprendizaje No Supervisado

Clústering

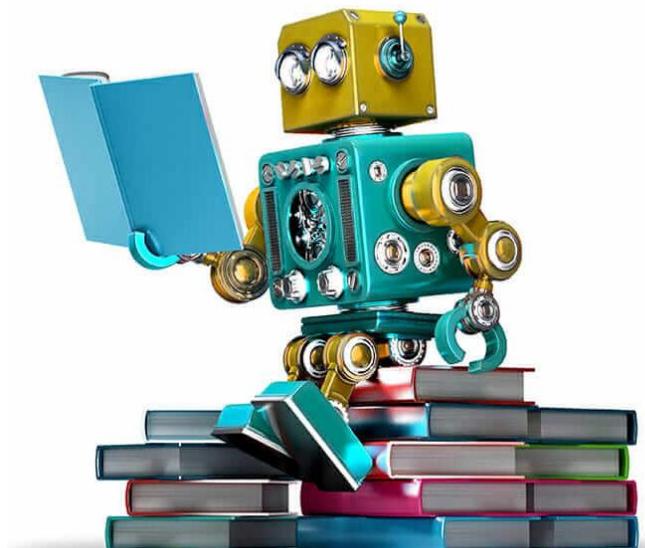
9-10 enero, 2026

Juan José Garcés Iniesta
jjgarcesiniesta@gmail.com

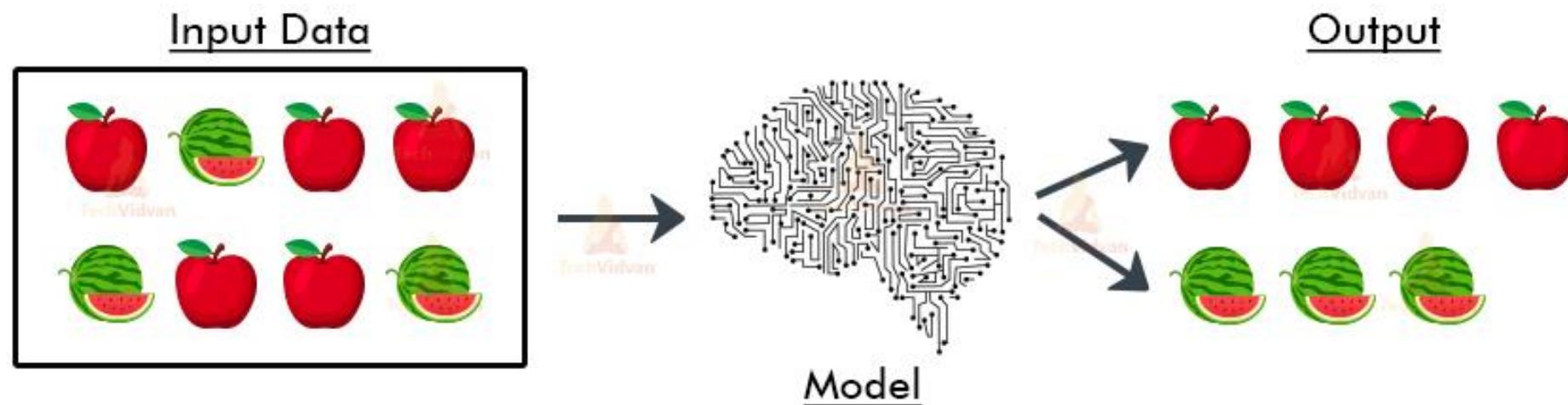
Índice

1. Aprendizaje no supervisado: conceptos generales
2. Clustering
 - Métodos particionales (K-means, k-medoides, etc)
 - Métodos jerárquicos
 - Métodos basados en Densidad (DBScan, etc.)
 - Métodos basados en modelos
3. Métricas para clustering

Resumen de algoritmos



1. Aprendizaje no supervisado



Tipos de aprendizaje

Tipo	Características	Aplicaciones
Supervisado	Se conoce la salida que debe proporcionar el modelo.	Clasificación, modelización, predicción.
No supervisado	No se conoce la salida que debe proporcionar el modelo.	Clustering, Reducción de la dimensionalidad, Estimación de densidades
Semisupervisado	Una mezcla de los dos anteriores, se conocen algunos valores.	Las mismas que en supervisado.
Reforzado	No se conoce el valor exacto de la salida deseada sino que se tiene una señal de refuerzo.	Optimización compleja

Aprendizaje no supervisado

DEFINICIÓN

Paradigma del machine learning donde el algoritmo aprende patrones a partir de datos sin etiquetas predefinidas.

Características principales

- No requiere variable objetivo (y)
- Descubre estructura inherente en los datos
- Resultados requieren interpretación experta
- Validación más compleja que supervisado

Técnicas principales

Clustering

Agrupamiento de observaciones similares

Reducción de dimensionalidad

PCA, t-SNE, UMAP, Autoencoders

Reglas de asociación

Descubrimiento de patrones frecuentes

Detección de anomalías

Identificación de outliers

Aplicaciones para aprendizaje no supervisado

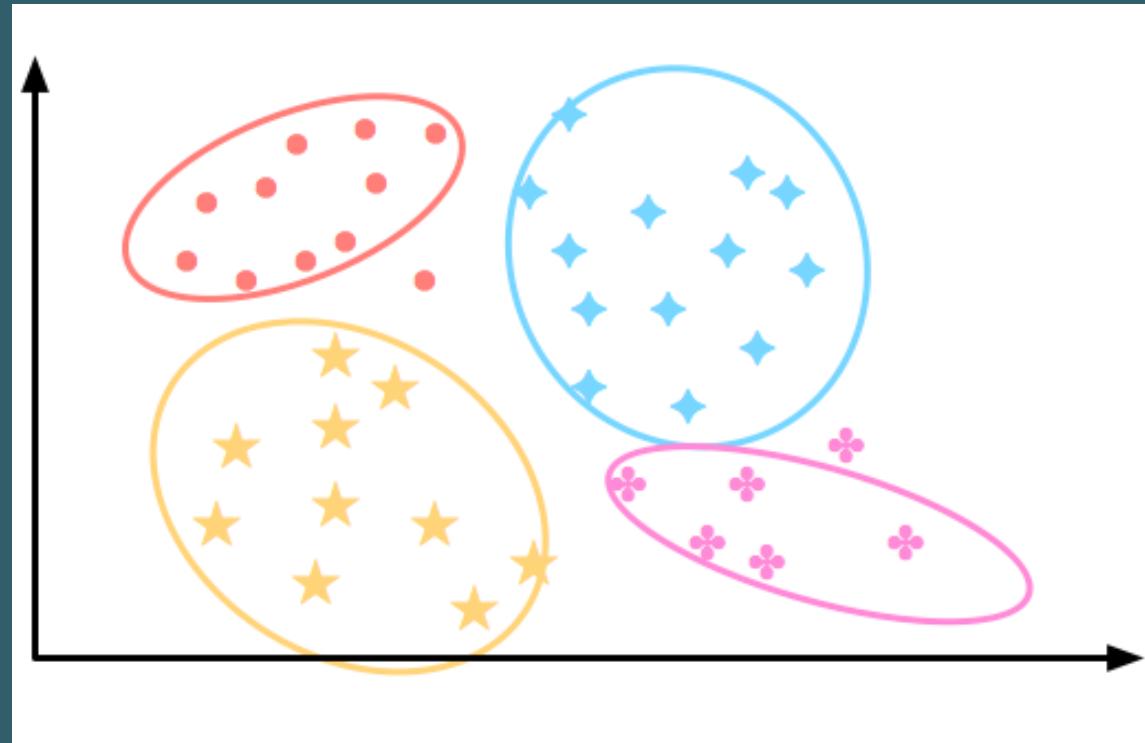
- **Clasificación**
 - ¿A que grupo se parece más este elemento?
- **Compresión** y comunicación
 - Envío el string “abababababababab” o envío “ab*7”
- **Reducción**/selección de características
- Detección de **anomalías**/outliers
 - ¿es esto normal?



En todos los casos nos basamos en buscar **patrones** que puedan describir los datos que estudiamos

Aplicaciones para aprendizaje no supervisado

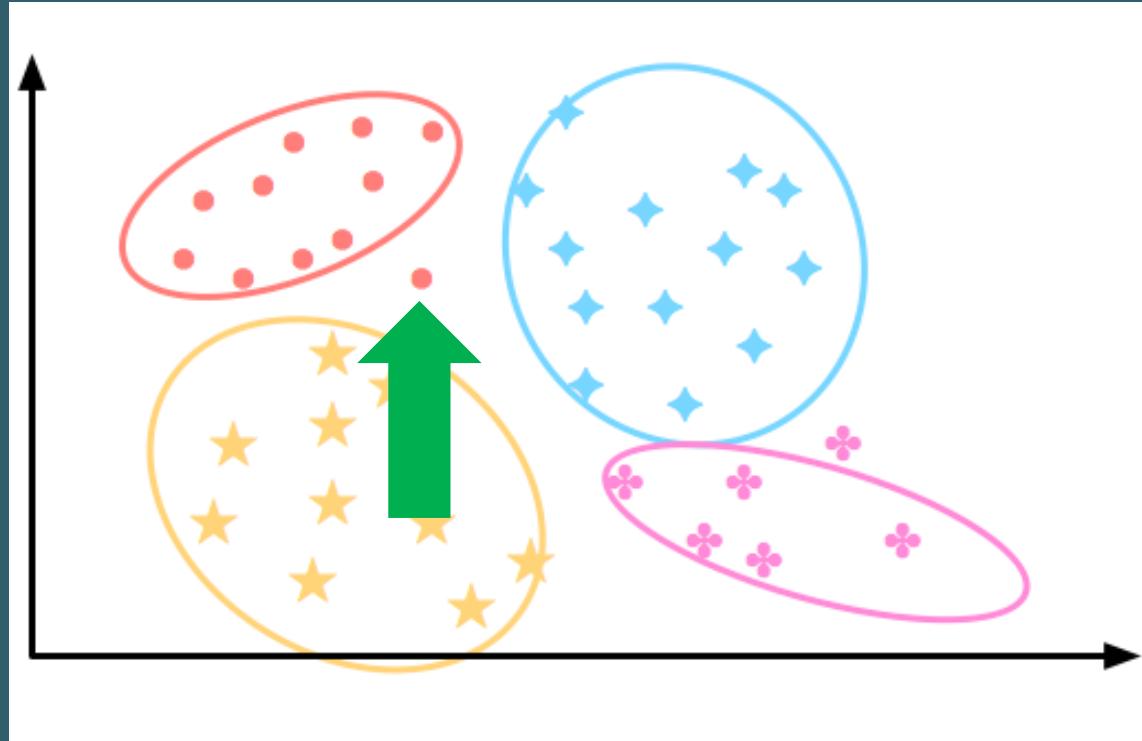
Clustering



El objetivo es agrupar los datos

Aplicaciones para aprendizaje no supervisado

Detección de outliers

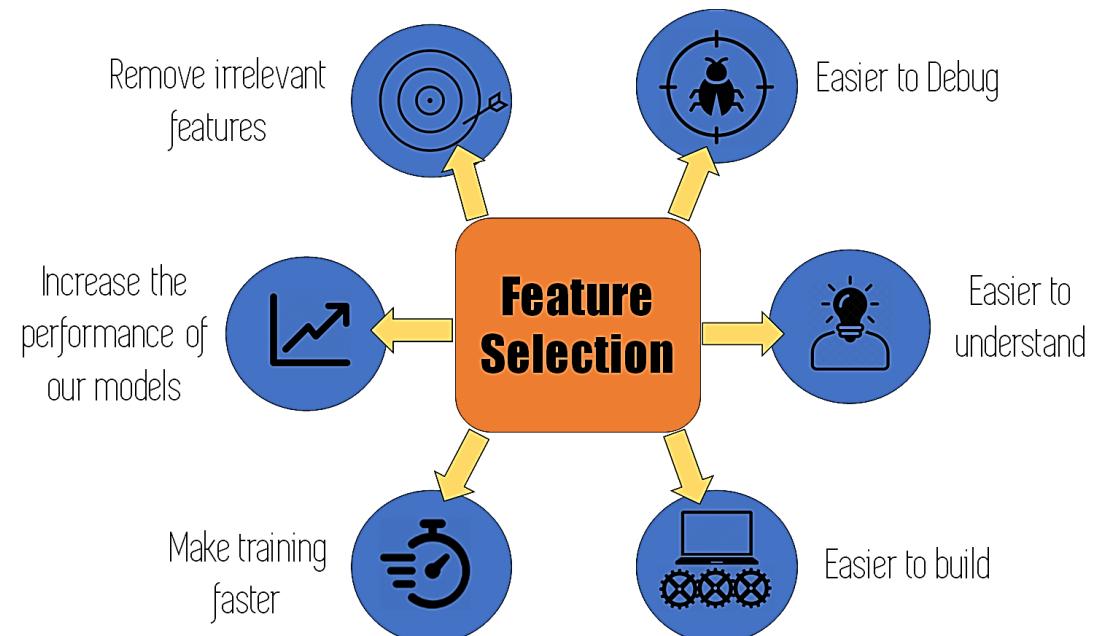


Determinación de casos **atípicos** dentro del conjunto de datos

Aplicaciones para aprendizaje no supervisado

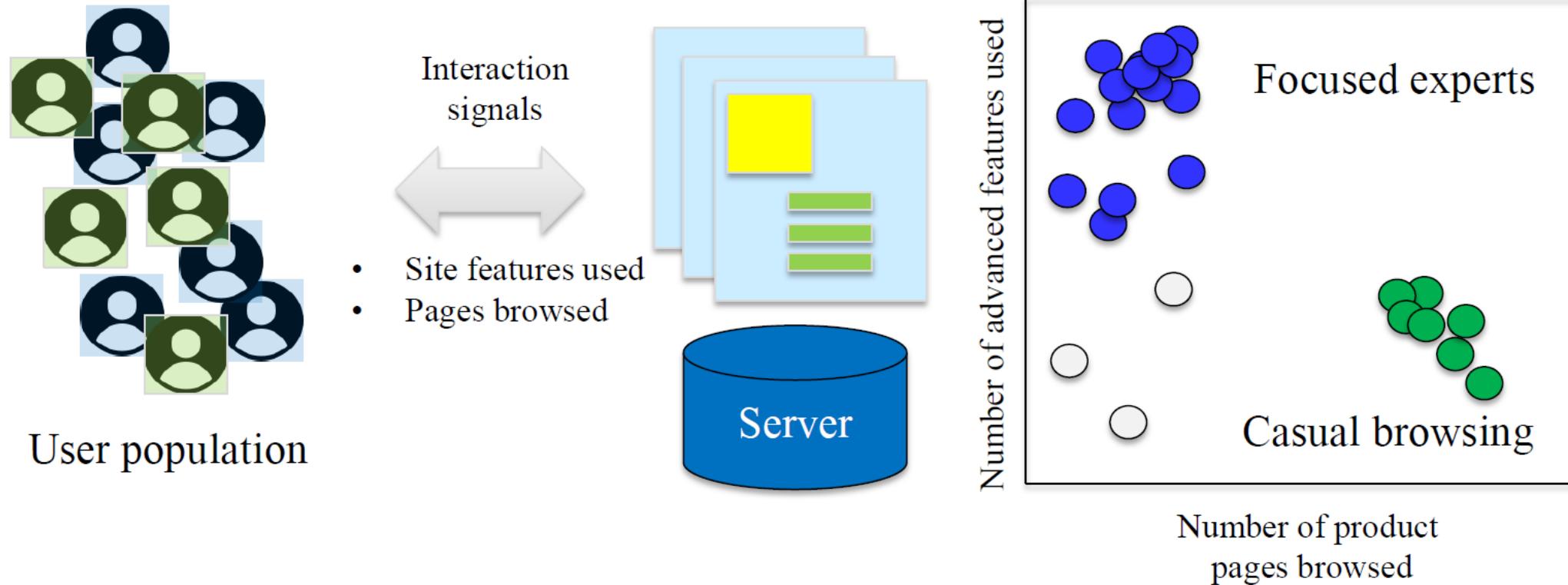
Selección de características

Obtención de las variables más importantes a la hora de definir un problema: encontrar las claves de nuestro problema



Aplicaciones para aprendizaje no supervisado

Ejemplo: clasificación de usuarios en la web



Preparación de los datos

La etapa de exploración y preparación de los datos es común independientemente del problema que tratemos.

Depende del algoritmo o técnica pueden ser necesarios ciertas acciones concretas

TAREA	OBJETIVOS
Caracterización estadística de las variables	Identificación de las relaciones elementales entre datos, etc.
Representación visual de los datos	Obtención de posibles relaciones de forma gráfica
Detección de datos anómalos e incompletos	Obtener un conjunto de datos completo sin incoherencias
Recodificación y transformación de variables	Preparar los datos en formatos adecuados
Generación de nuevas variables	Añadir información al conjunto
Selección y reducción de características	Reducción del nº de variables de un modelo

2. Clustering



¿Qué es el Clustering?

DEFINICIÓN FORMAL

El clustering es el proceso de particionar un conjunto de datos $X = \{x_1, x_2, \dots, x_n\}$ en k subconjuntos (clusters) $C = \{C_1, C_2, \dots, C_k\}$ de manera que los elementos dentro de cada cluster sean más similares entre sí que con elementos de otros clusters.

Objetivo matemático

Optimizar una función objetivo que mide la calidad del agrupamiento:

$$\min \sum_i \sum_{x \in C_i} d(x, \mu_i)^2$$

Minimizar varianza intra-cluster (WCSS)

Propiedades deseables

- Cohesión: alta similitud intra-cluster
- Separación: baja similitud inter-cluster
- Interpretabilidad: clusters significativos
- Estabilidad: robustez ante perturbaciones



Taxonomía de Algoritmos de Clustering

Métodos Particionales

Dividen los datos en k grupos disjuntos optimizando una función objetivo.

K-Means, K-Medoids, K-Modes

Métodos Jerárquicos

Construyen una jerarquía de clusters representada mediante dendrogramas.

Aglomerativo, Divisivo, AGNES

Métodos Basados en Densidad

Identifican regiones densas separadas por regiones de baja densidad.

DBSCAN, OPTICS, HDBSCAN

Métodos Basados en Modelos

Asumen que los datos provienen de una mezcla de distribuciones probabilísticas.

GMM, Bayesian Clustering



Métodos Particionales

Principio fundamental

Particionar n observaciones en k clusters donde cada observación pertenece al cluster con el centroide más cercano.

Función objetivo (WCSS)

$$J = \sum_{i=1}^k \sum_{x \in C_i} ||x - \mu_i||^2$$

Ventajas

Eficiente $O(nkt)$, escalable

Limitaciones

Requiere k , sensible a inicio

Algoritmos representativos

K-Means

Centroide = media. Óptimo para clusters esféricos. Sensible a outliers.

K-Medoids (PAM)

Centroide = punto real. Más robusto a outliers. Mayor coste computacional.

K-Modes / K-Prototypes

Extensión para datos categóricos y mixtos.

Métodos Jerárquicos

Principio fundamental

Construyen una jerarquía de clusters representada mediante un dendrograma. No requieren especificar k a priori.

Aglomerativo

Bottom-up: cada punto inicia como cluster individual, se fusionan iterativamente.

Divisivo

Top-down: todos los puntos en un cluster, se dividen recursivamente.

Complejidad computacional

$O(n^3)$ tiempo, $O(n^2)$ espacio. Limitado a datasets pequeños/medianos.

Criterios de enlace (Linkage)

Single Linkage

Distancia mínima entre clusters. Sensible al efecto cadena.

Complete Linkage

Distancia máxima. Produce clusters compactos y esféricos.

Average Linkage (UPGMA)

Promedio de distancias. Balance entre single y complete.

Ward's Method

Minimiza incremento de varianza. Más usado en la práctica.



Métodos Basados en Densidad

DBSCAN: Conceptos clave

ϵ -vecindad: puntos a distancia $\leq \epsilon$ de un punto dado. Punto núcleo: tiene $\geq \text{minPts}$ vecinos. Punto borde: en vecindad de núcleo. Punto ruido: outlier.

Parámetros principales

`eps (ϵ)`: radio de vecindad. `min_samples`: mínimo de puntos para ser núcleo.

OPTICS

Ordena puntos según alcanzabilidad. No requiere ϵ fijo, genera diagrama de alcanzabilidad.

HDBSCAN

Combina DBSCAN con clustering jerárquico. Selección automática de clusters.

Ventajas

Formas arbitrarias, detecta outliers

Limitaciones

Sensible a parámetros



Métodos Basados en Modelos

Gaussian Mixture Models (GMM)

Modelo probabilístico que asume que los datos son generados por una mezcla de k distribuciones gaussianas con parámetros desconocidos.

Modelo de mezcla

$$p(x) = \sum_k \pi_k N(x | \mu_k, \Sigma_k)$$

π_k : peso del componente k, μ_k : media, Σ_k : covarianza

Soft vs Hard Clustering

GMM proporciona probabilidades de pertenencia a cada cluster (soft assignment), a diferencia de K-Means (hard assignment).

Algoritmo EM

1. E-step: Calcular responsabilidades (probabilidad de cada punto por cluster)
2. M-step: Actualizar parámetros (π_k , μ_k , Σ_k) maximizando verosimilitud
3. Repetir hasta convergencia

Selección de componentes

BIC (Bayesian Information Criterion) y AIC (Akaike Information Criterion): penalizan complejidad del modelo.

Ventajas

Clusters elípticos, probabilidades, flexible

Limitaciones

Requiere k, sensible a inicialización

Métricas de Distancia y Similitud

Distancia Euclídea

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

Distancia en línea recta. Sensible a escala.

Similitud del Coseno

$$\cos(\theta) = (x \cdot y) / (\|x\| \|y\|)$$

Mide ángulo entre vectores. Ideal para texto (TF-IDF).

Distancia Manhattan

$$d(x,y) = \sum_i |x_i - y_i|$$

Suma de diferencias absolutas. Más robusta a outliers.

Índice de Jaccard

$$J(A,B) = |A \cap B| / |A \cup B|$$

Para conjuntos y datos binarios. Rango [0,1].

Consideración clave: La elección de la métrica depende del tipo de datos y el contexto del problema.



Espacios Métricos

Definición formal

Un espacio métrico es un par (X, d) donde X es un conjunto y $d: X \times X \rightarrow \mathbb{R}$ es una función de distancia que satisface ciertos axiomas.

Axiomas de una métrica

1. No negatividad: $d(x,y) \geq 0$
2. Identidad: $d(x,y) = 0 \Leftrightarrow x = y$
3. Simetría: $d(x,y) = d(y,x)$
4. Desigualdad triangular: $d(x,z) \leq d(x,y) + d(y,z)$

Importancia en clustering

Garantiza comportamiento consistente de algoritmos. La desigualdad triangular permite optimizaciones computacionales. Fundamenta la noción matemática de "cercanía" entre puntos.

Distancia de Minkowski

$$d(x,y) = (\sum_i |x_i - y_i|^p)^{(1/p)}$$

$p=1$: Manhattan | $p=2$: Euclidiana | $p \rightarrow \infty$: Chebyshev



La Maldición de la Dimensionalidad

Definición

Conjunto de fenómenos que emergen al analizar datos en espacios de alta dimensionalidad, donde la intuición geométrica del espacio 3D deja de aplicar.

Fenómenos principales

Concentración de distancias (todas tienden a ser similares). Escasez de datos (volumen crece exponencialmente). Datos concentrados en esquinas. Pérdida de significado de "vecino cercano".

Impacto en clustering

Métricas de distancia pierden capacidad discriminativa. Clusters más difusos y difíciles de separar. Mayor coste computacional.

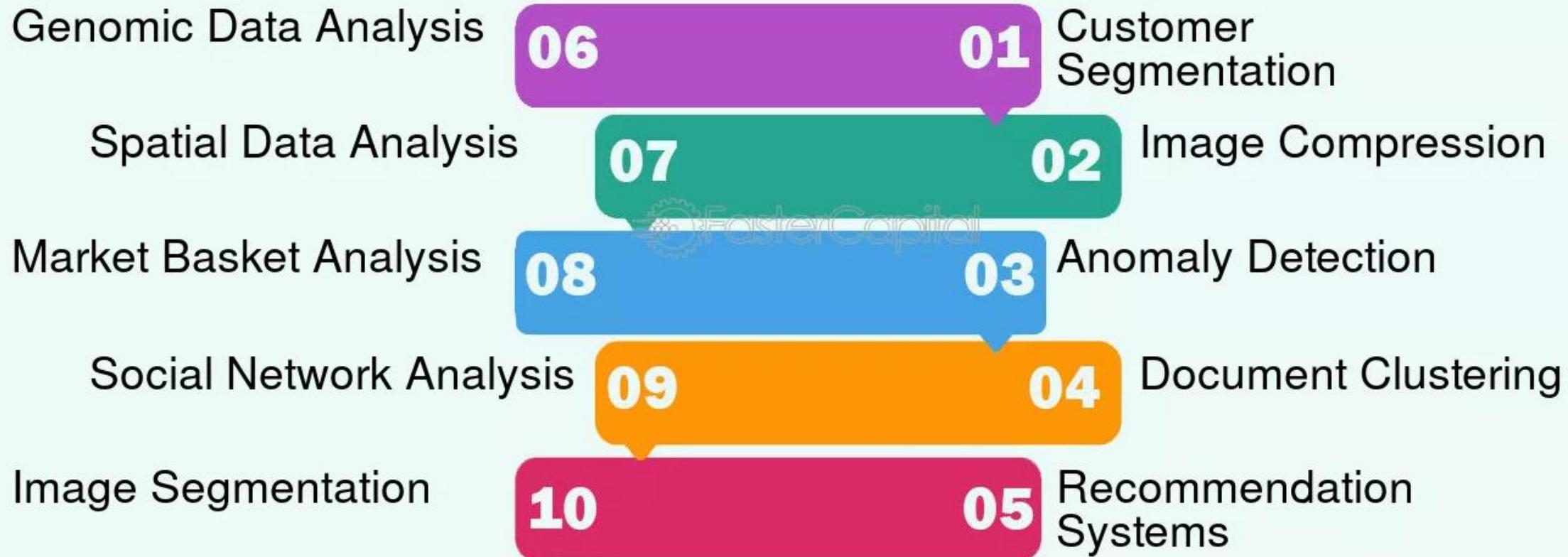
Estrategias de mitigación

Reducción de dimensionalidad (PCA, t-SNE, UMAP). Selección de características relevantes. Métricas adaptadas. Clustering en subespacios.

Regla práctica: necesitamos exponencialmente más datos conforme aumentan las dimensiones.



Aplicaciones y casos de uso del clustering



MÓDULO 2.1

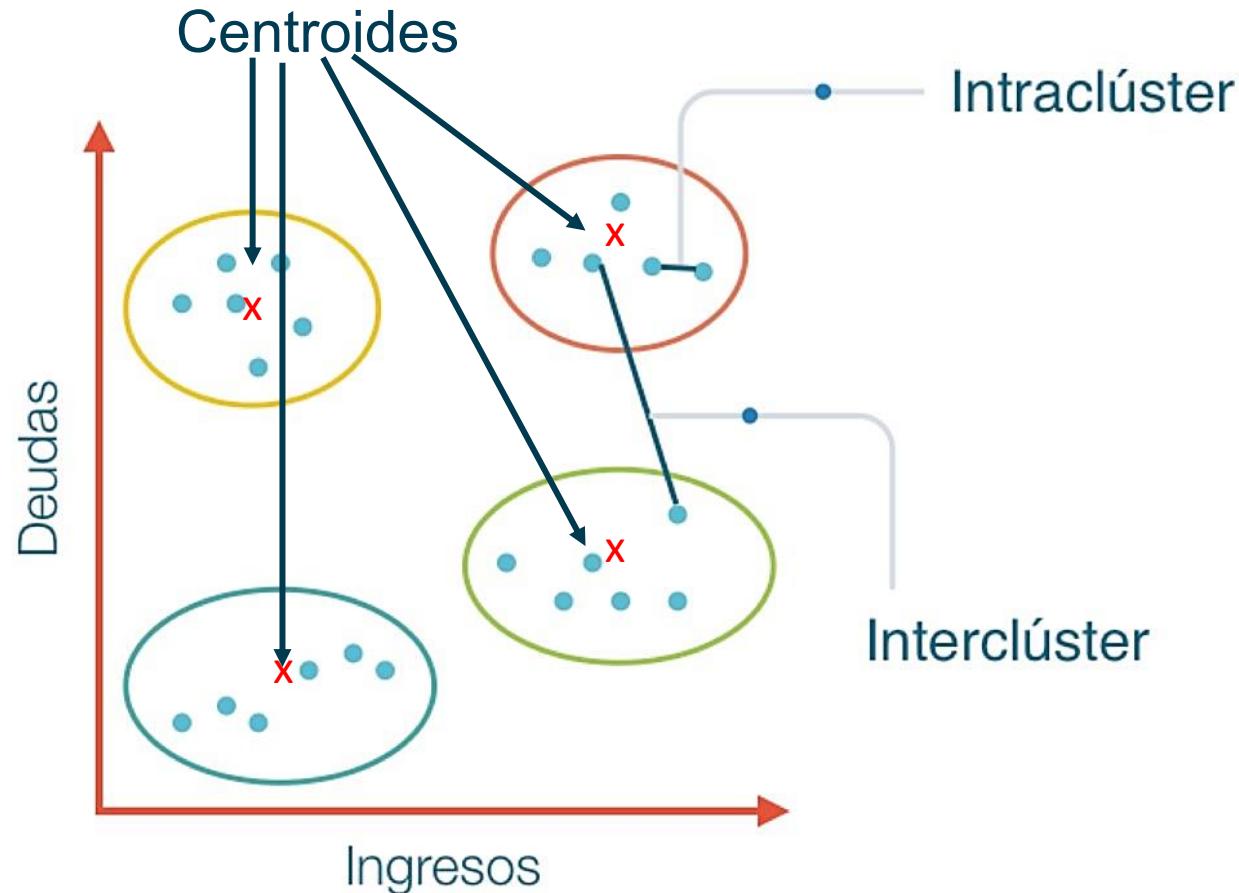
K-Means y Variantes

Algoritmos de Particionamiento para Clustering



Métodos Particionales

Basado en las distancias entre elementos del mismo grupo, entre grupos y centroides



Cluster = grupo
Centroide = centro de un cluster

No hay una definición precisa de grupo: depende de los criterios / medidas de similaridad

Introducción a K-Means

Definición

Algoritmo de particionamiento que divide n observaciones en k clusters, donde cada observación pertenece al cluster cuyo centroide (media) es más cercano.

Historia

Propuesto por Stuart Lloyd en 1957 (Bell Labs). Publicado formalmente en 1982. Independientemente desarrollado por MacQueen (1967) quien acuñó el nombre "K-Means".

Características principales

Algoritmo iterativo de optimización local. Requiere especificar k a priori. Asigna cada punto a exactamente un cluster (hard clustering). Complejidad $O(n \cdot k \cdot d \cdot i)$ donde i son iteraciones.

Popularidad

Uno de los algoritmos más utilizados en minería de datos. Simple, eficiente y escalable. Base para muchas variantes y extensiones modernas.



Función objetivo: WCSS

Within-Cluster Sum of Squares

$$J = \sum_k \sum_{x \in C_k} \|x - \mu_k\|^2$$

También llamada inercia o distorsión

Componentes de la fórmula

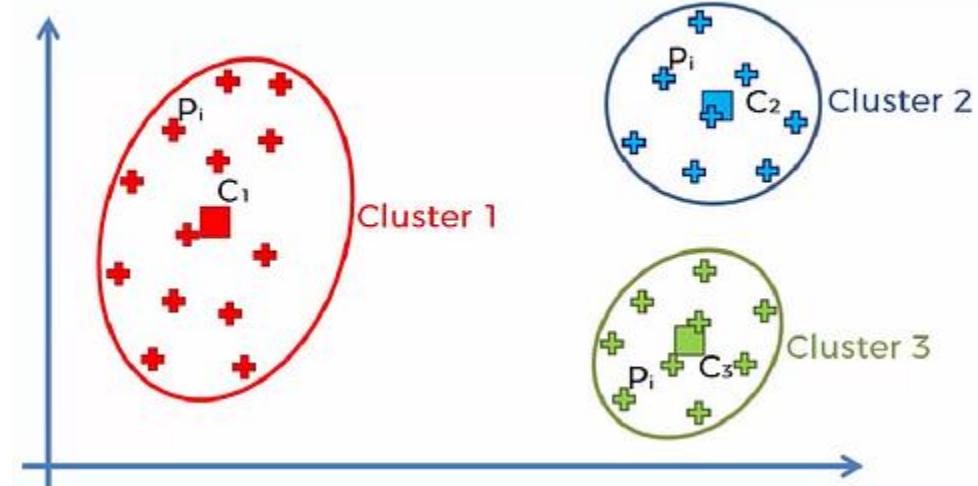
C_k : Conjunto de puntos asignados al cluster k

μ_k : Centroide del cluster k (media de sus puntos)

$\|x - \mu_k\|^2$: Distancia euclídea al cuadrado

Interpretación

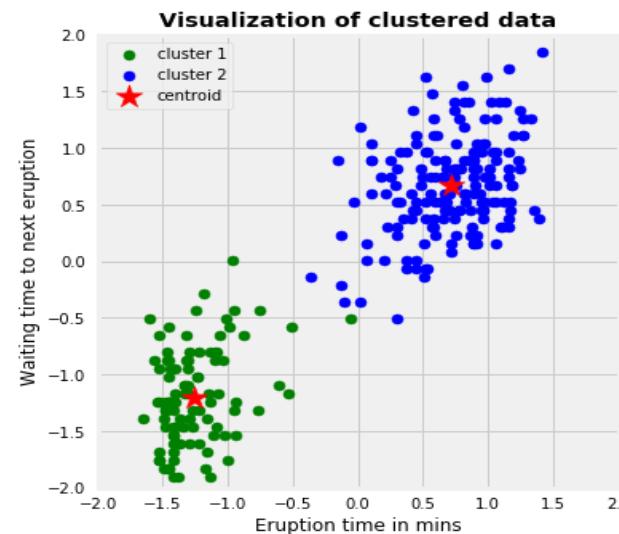
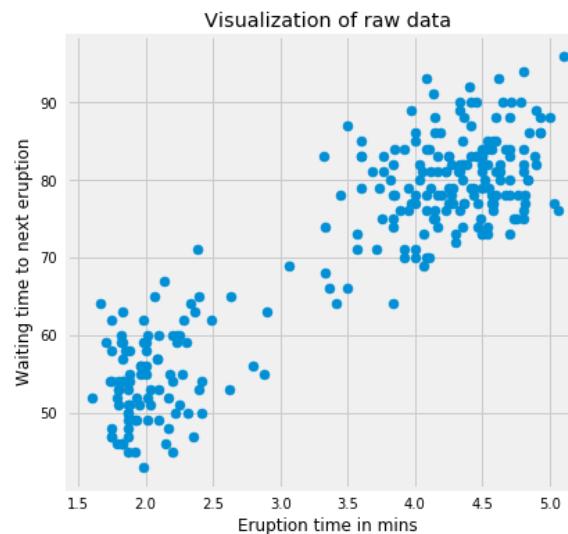
Suma de las distancias al cuadrado de cada punto a su centroide asignado. Mide la compactidad total de los clusters. Objetivo: minimizar J.



Algoritmo K-Means

Uno de los algoritmos más sencillo y más popular

- Se inicia posicionando **k** puntos en el conjunto a estudiar: centroides
- Establece la pertenencia a cada grupo según la distancia de cada punto al centroide más cercano
- Actualiza los centroides generando unos nuevos por promediado (**means**) de los puntos que se han asignado a cada grupo



Demo:
[https://www.naftaliharris.com
 /blog/visualizing-k-means-
 clustering/](https://www.naftaliharris.com/blog/visualizing-k-means-clustering/)

Algoritmo K-Means: proceso iterativo

Paso 1: Inicialización

Seleccionar k centroides iniciales $\mu_1, \mu_2, \dots, \mu_k$ (aleatoriamente o con estrategia específica)

Paso 2: Asignación

Asignar cada punto x_i al cluster cuyo centroide sea más cercano: $C_k = \operatorname{argmin} \|x_i - \mu_k\|^2$

Paso 3: Actualización

Recalcular cada centroide como la media de los puntos asignados: $\mu_k = (1/|C_k|) \sum_{x \in C_k} x$

Paso 4: Iteración

Repetir pasos 2 y 3 hasta convergencia (sin cambios en asignaciones o máximo de iteraciones)

Garantías del algoritmo

Cada iteración reduce o mantiene el valor de J (monotonía decreciente). Convergencia garantizada en número finito de iteraciones. El resultado depende de la inicialización.

Convergencia y criterios de parada

Criterios de convergencia

Estabilidad: No hay cambios en las asignaciones de puntos entre iteraciones consecutivas.

Tolerancia: El cambio en J es menor que un umbral ϵ predefinido.

Centroides: El desplazamiento de centroides es menor que δ .

Propiedades matemáticas

El algoritmo siempre converge porque: hay un número finito de particiones posibles, J decrece monótonamente, y existe un límite inferior ($J \geq 0$).

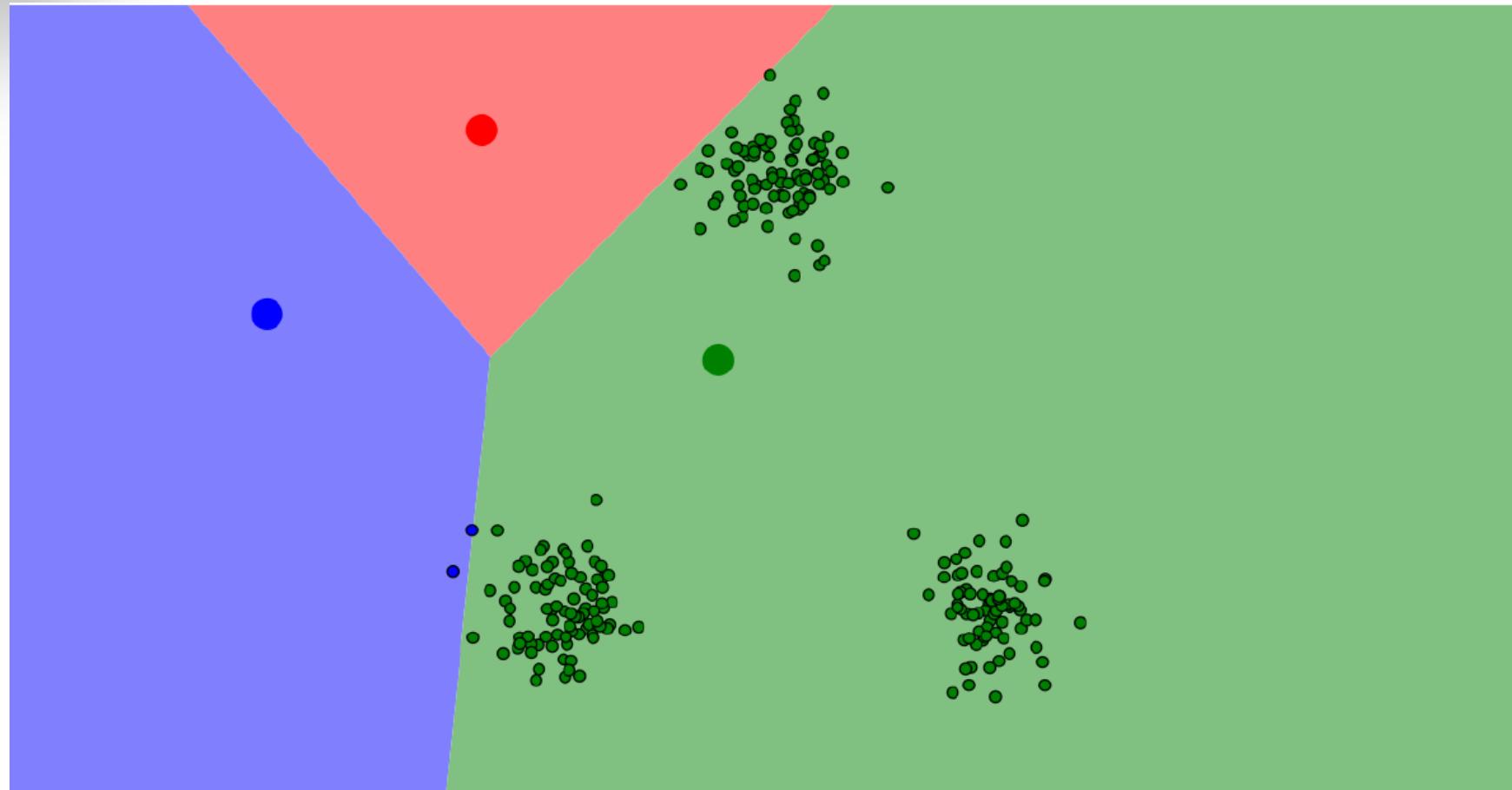
Máximo de iteraciones

Límite práctico para evitar ejecuciones excesivamente largas.
Típicamente entre 100-300 iteraciones. Si se alcanza, puede indicar problemas de convergencia.

Mínimos locales

La convergencia no garantiza el óptimo global. El resultado final depende fuertemente de la inicialización. Solución práctica: múltiples ejecuciones con diferentes semillas.

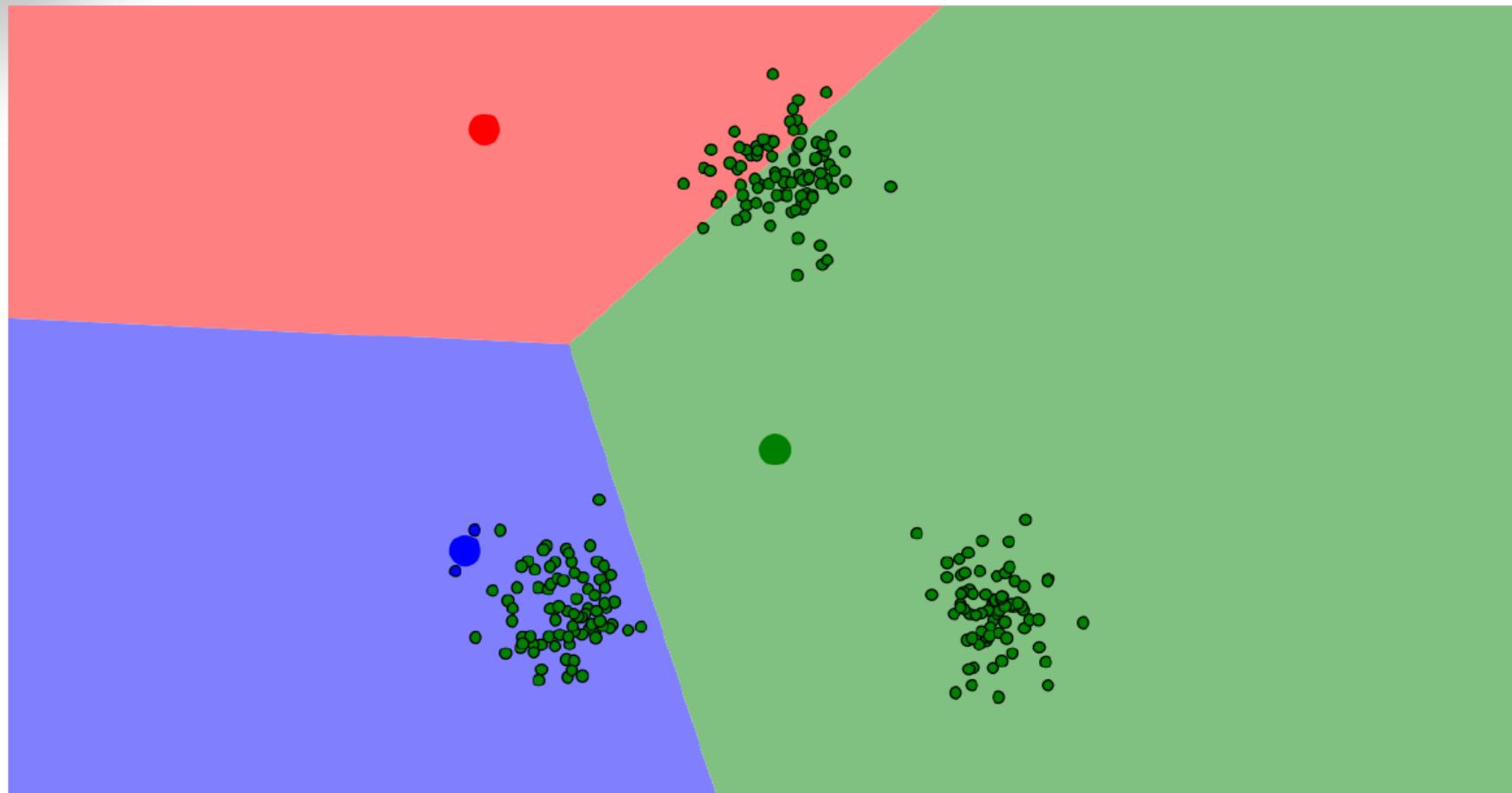
K-Means: Ejemplo 1A



1 - Posicionamiento
de centroides

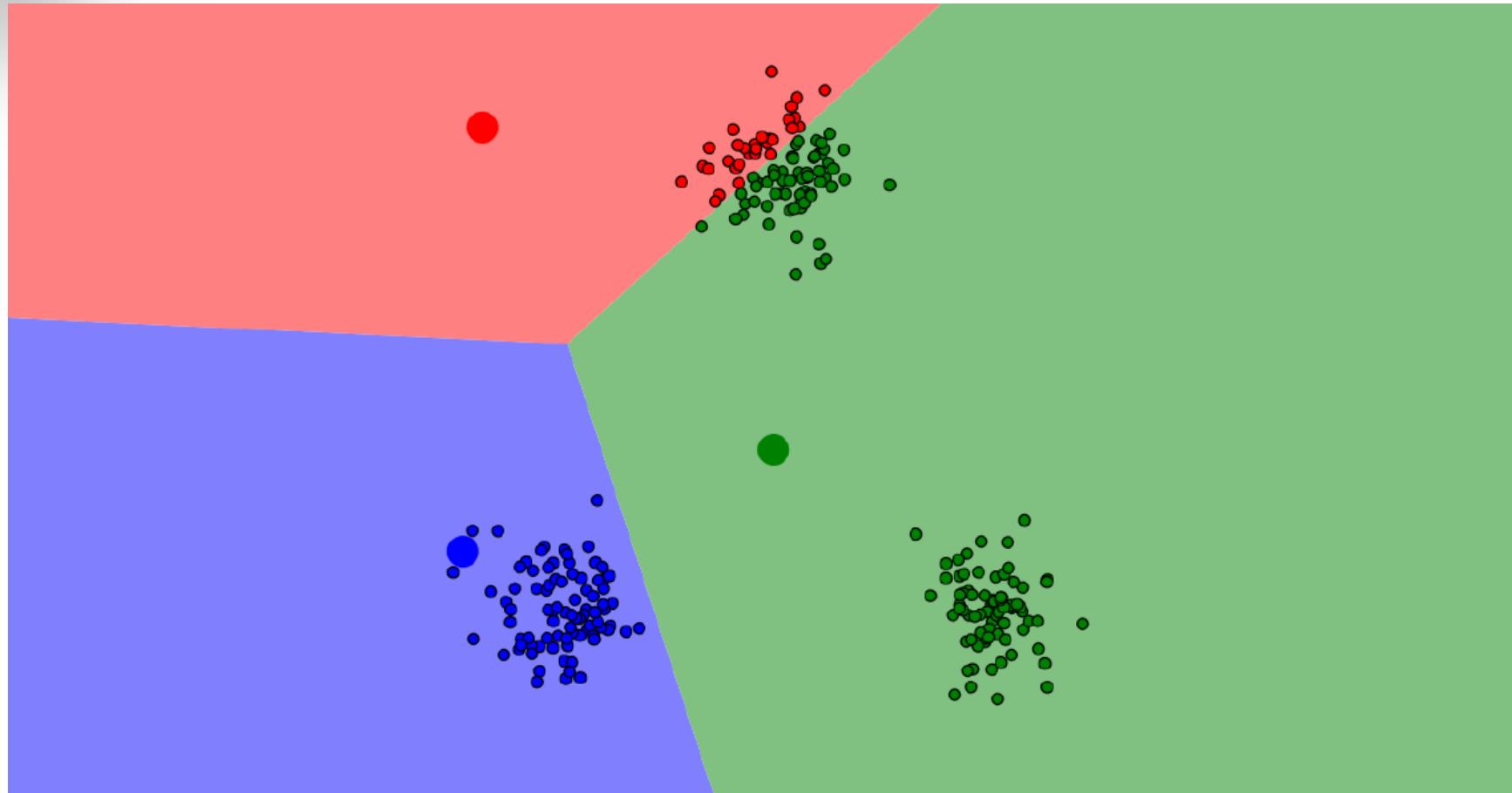
1A.- Asignación de
puntos a cada
centroide

K-Means: Ejemplo 1B



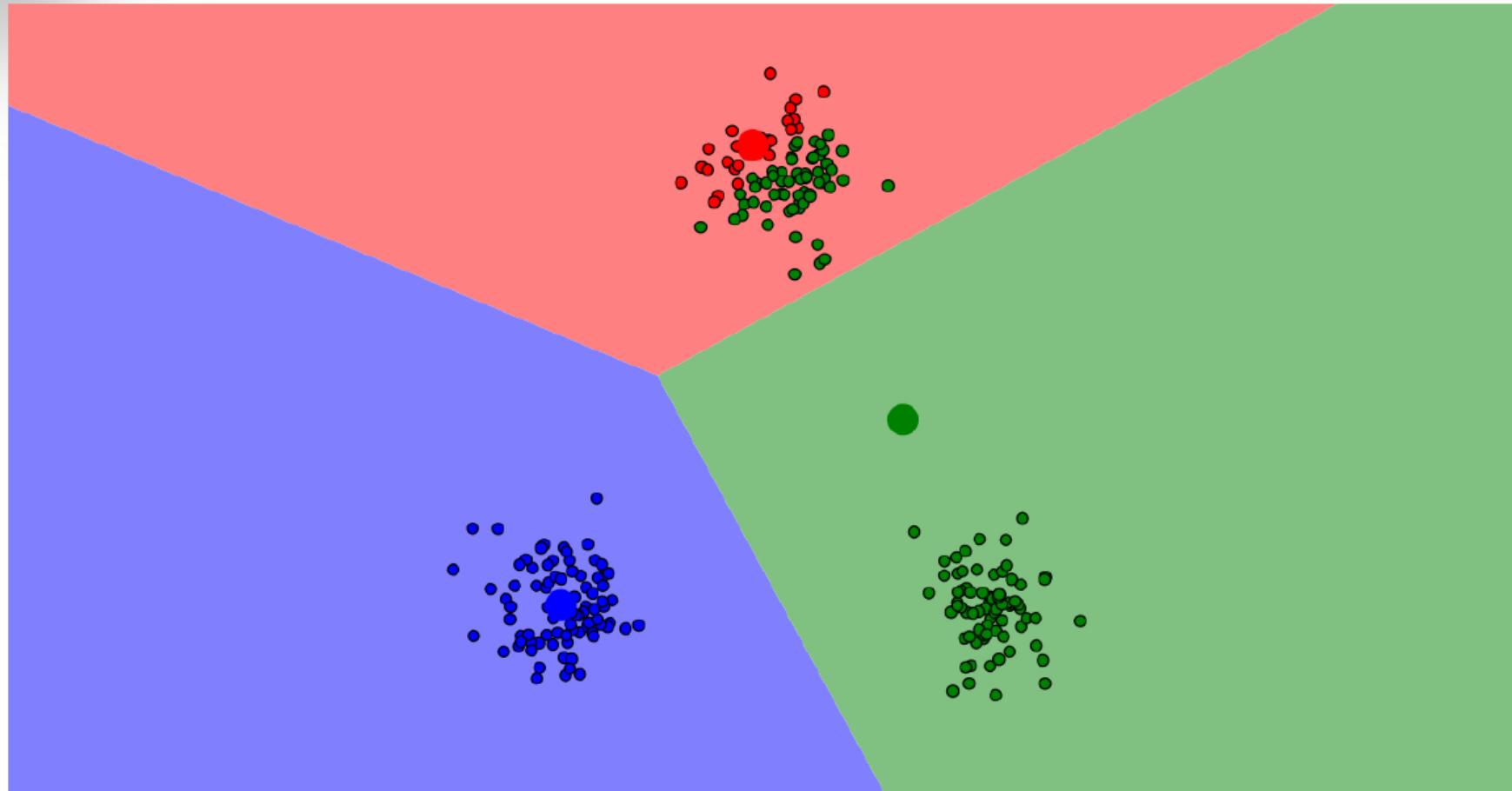
1B.- Actualización
de centroides:
nueva área de
influencia

K-Means: Ejemplo 2A



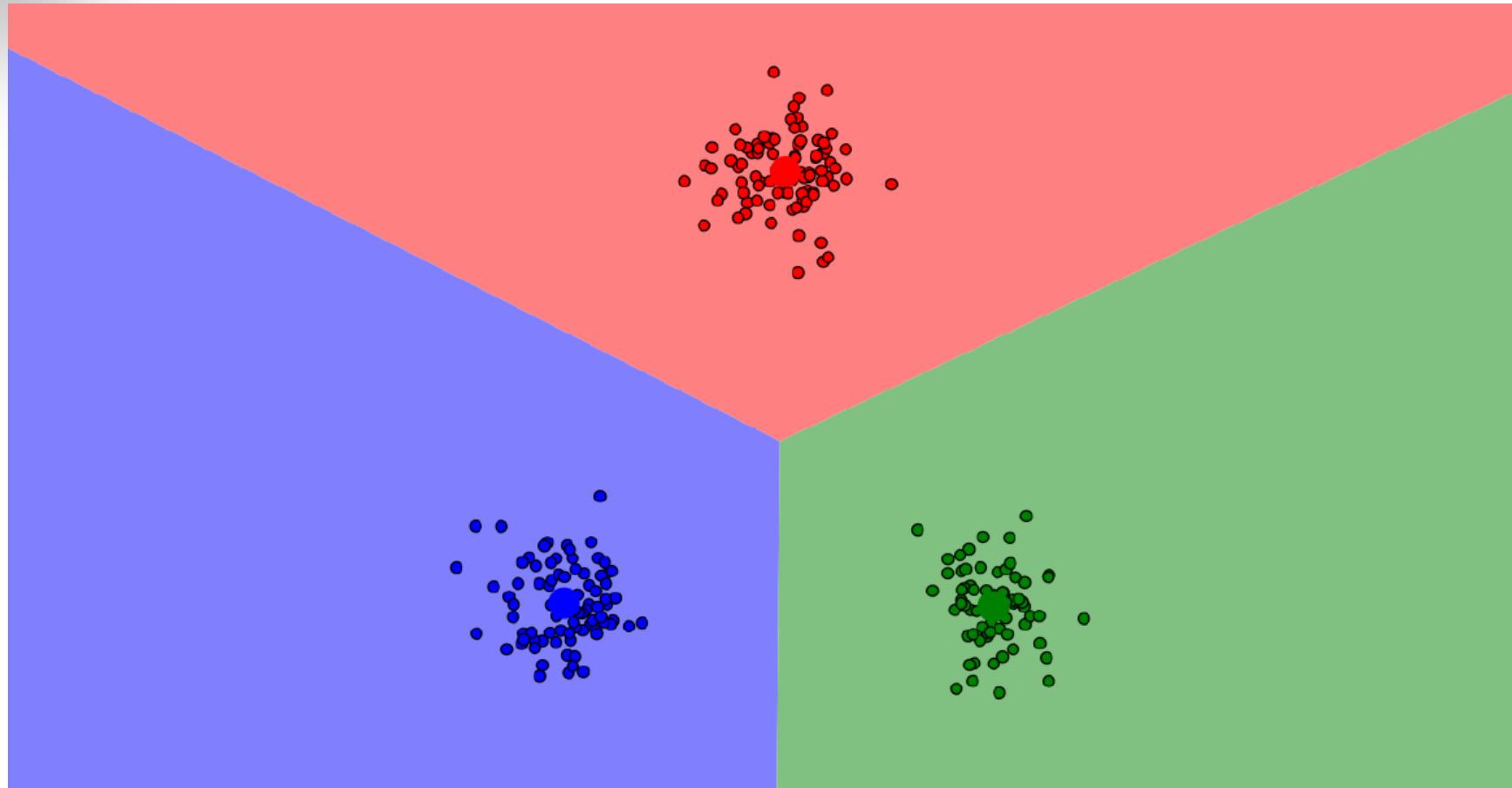
2A.- Asignación de puntos a cada centroide

K-Means: Ejemplo 2B



2B.- Actualización de centroides:
nueva área de influencia

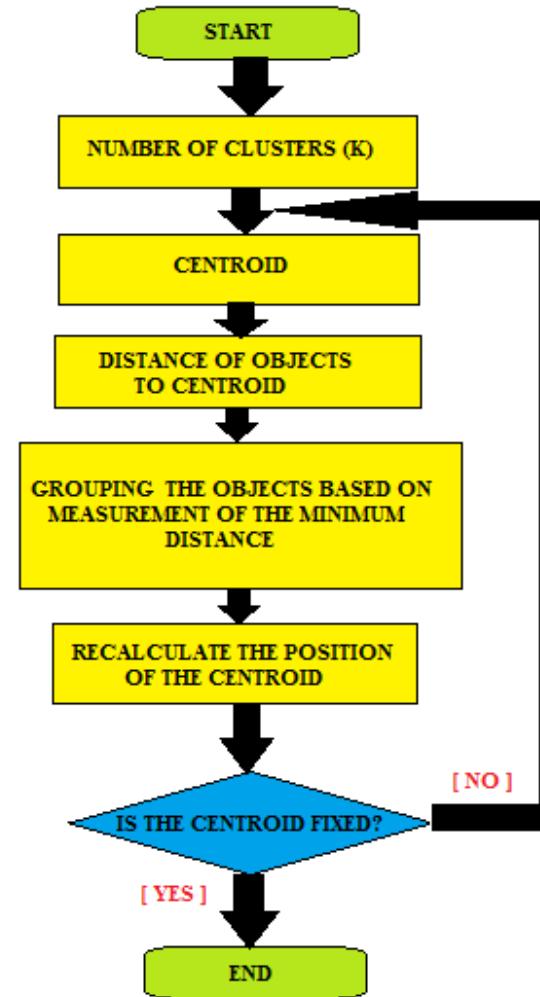
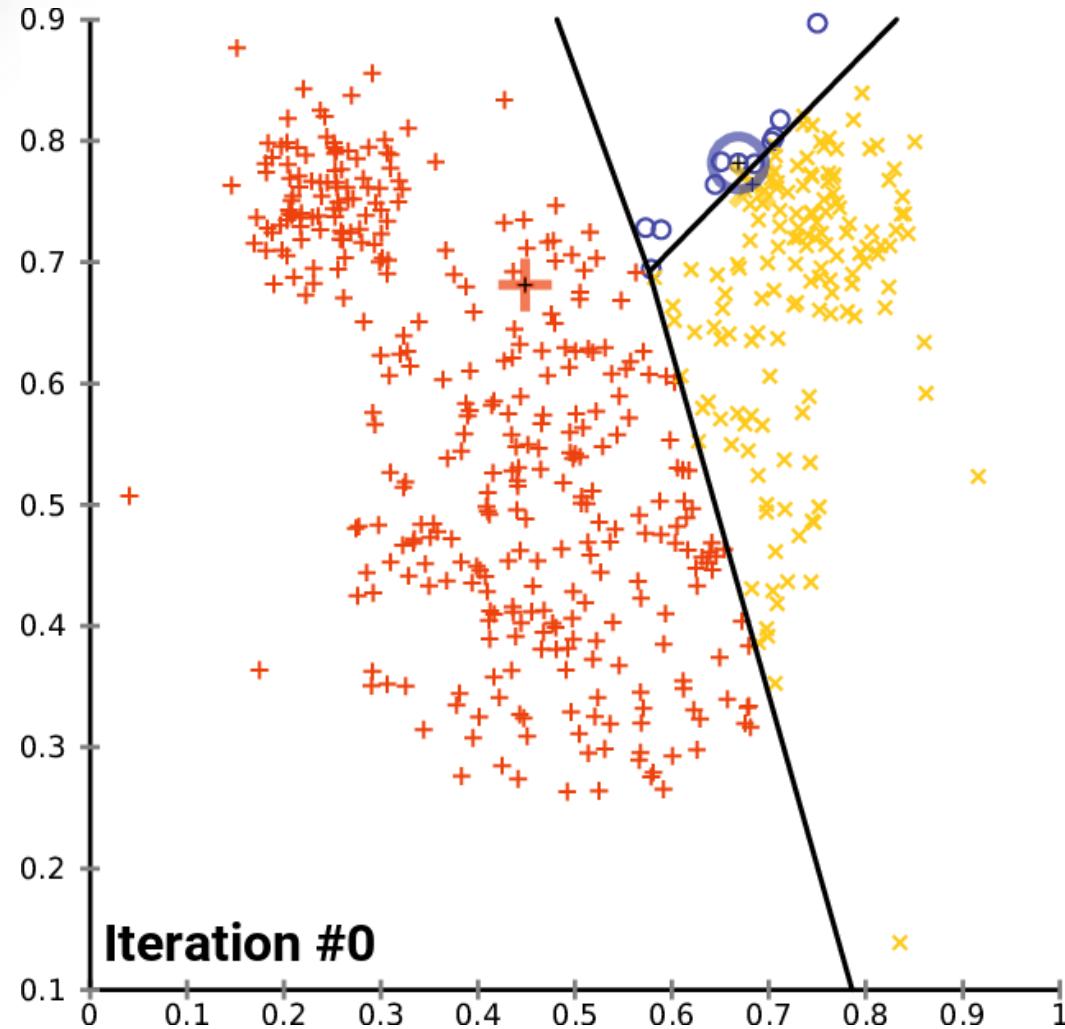
K-Means: Ejemplo 3^a. Convergencia



3A.- Asignación de puntos a cada centroide

Convergencia: cuando se repite el proceso y no hay variación

Algoritmo K-Means: proceso iterativo



Inicialización de centroides

Inicialización aleatoria

Seleccionar k puntos del dataset al azar como centroides iniciales. Simple pero puede llevar a resultados subóptimos o convergencia lenta.

Problemas de inicialización aleatoria

Centroides iniciales muy cercanos entre sí. Centroides en regiones de baja densidad. Alta variabilidad en resultados entre ejecuciones. Posible convergencia a mínimos locales pobres.

Estrategias alternativas

Forgy: Selección aleatoria uniforme de k puntos.

Random Partition: Asignación aleatoria de puntos a clusters, luego calcular centroides.

K-Means++: Selección probabilística basada en distancias (recomendada).

Práctica recomendada

Ejecutar K-Means múltiples veces (`n_init`) con diferentes inicializaciones y quedarse con el resultado de menor inercia J .



Limitaciones de K-Means

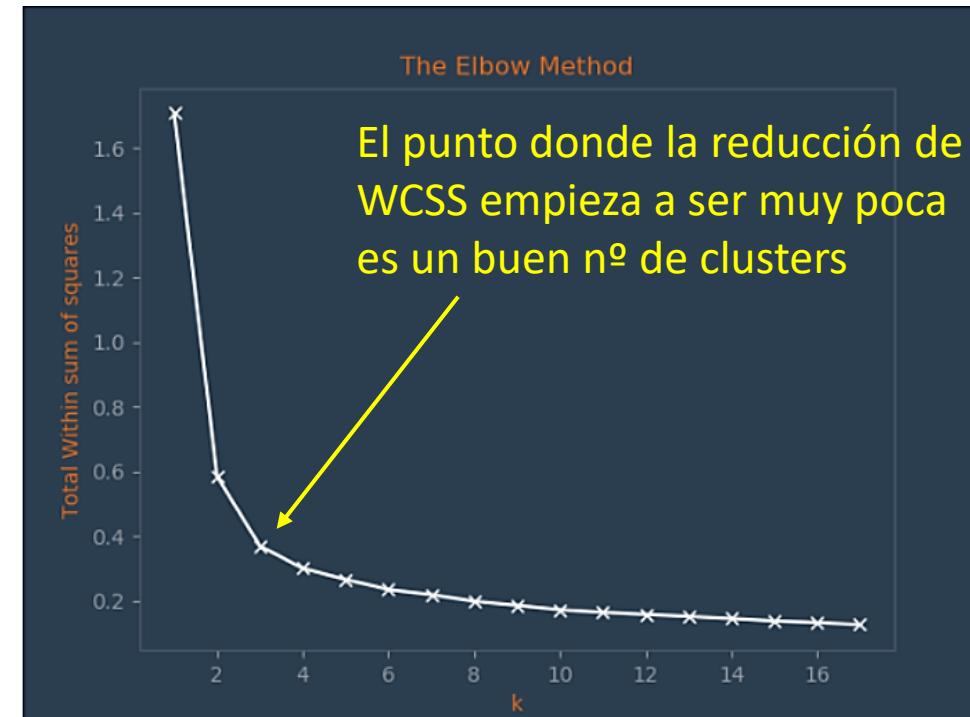
- Necesidad de establecer el valor de K (desconocimiento del número adecuado de grupos)
- Una aproximación se puede hacer por el método del codo “*elbow method*”

WCSS (within-cluster sum of squares): suma de distancias al cuadrado de cada elemento respecto a su centroide

$$\text{WCSS}(k) = \sum_{j=1}^k \sum_{x_i \in \text{cluster } j} \|x_i - \bar{x}_j\|^2,$$

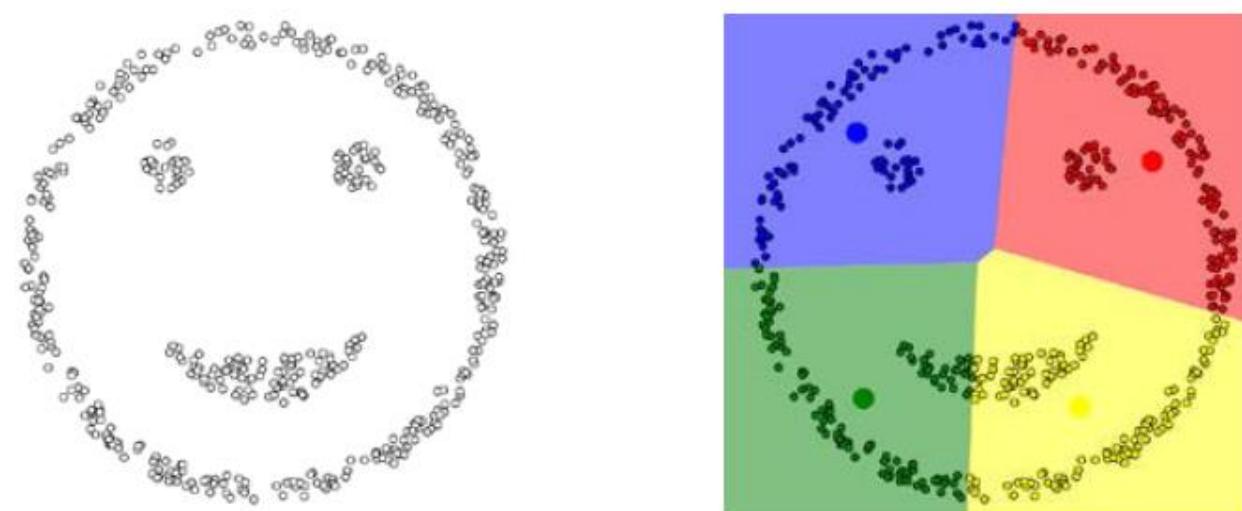
where \bar{x}_j is the sample mean in cluster j

Calculamos la WCSS para diferentes valores de k ($k=1, 2, 3, \dots$) y lo graficamos



Limitaciones de K-Means

- Funciona mal en clústeres irregulares
- Depende de la elección inicial de los centroides: diferentes ejecuciones dan lugar a diferentes grupos



K-means typically performs poorly with data having complex, irregular clusters.

Limitaciones de K-Means: resumen

Requiere especificar k

El número de clusters debe conocerse a priori. No existe un método universalmente óptimo para determinarlo.

Asume tamaños similares

Tiende a producir clusters de tamaño comparable. Puede dividir clusters grandes o fusionar pequeños incorrectamente.

Asume clusters esféricos

La distancia euclídea genera fronteras de Voronoi lineales. No detecta clusters con formas elongadas, anulares o irregulares.

Solo variables numéricas

El cálculo de medias requiere datos continuos. No aplicable directamente a datos categóricos o mixtos.

Sensibilidad a outliers

Los valores atípicos afectan el cálculo de centroides (media). Un solo outlier puede desplazar significativamente un centroide.

Solución: usar variantes

K-Medoids para outliers, K-Modes para categóricos, GMM para formas no esféricas, DBSCAN para clusters irregulares.

Selección de K: método del codo

Principio

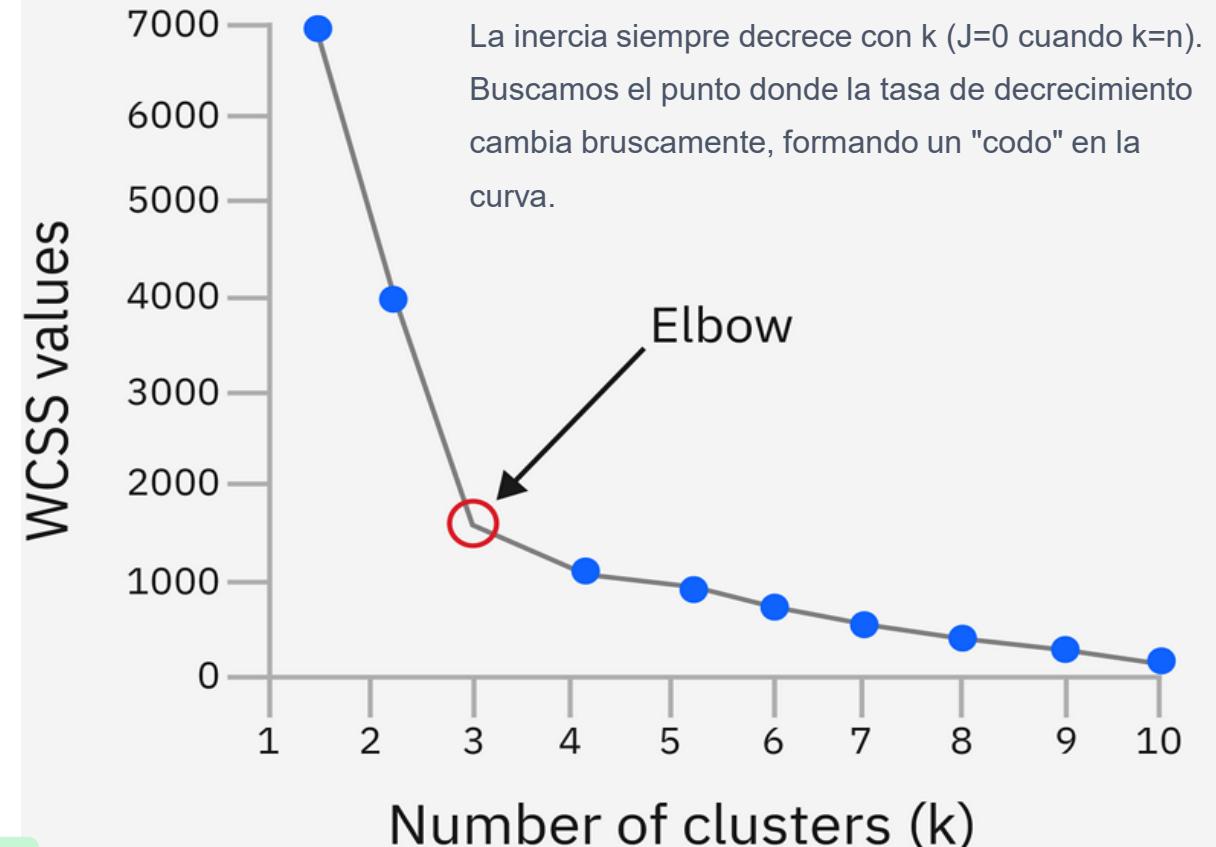
Ejecutar K-Means para diferentes valores de k y graficar la inercia (WCSS) en función de k. Buscar el punto de inflexión donde añadir más clusters no reduce significativamente la inercia.

Procedimiento

1. Ejecutar K-Means para $k = 1, 2, 3, \dots, k_{\max}$.
2. Calcular la inercia J para cada k .
3. Graficar J vs k .
4. Identificar el "codo" en la curva.

Ventajas

Simple e intuitivo. Visual y fácil de comunicar. No requiere ground truth.



Limitaciones

Subjetivo. El codo puede ser difuso. No siempre hay un codo claro.

Selección de K: método de la silueta

Coeficiente de silueta

$$s(i) = (b(i) - a(i)) / \max(a(i), b(i))$$

Rango: [-1, 1]. Valores altos indican buena asignación.

Componentes

a(i): Distancia media al resto de puntos del mismo cluster (cohesión).

b(i): Distancia media mínima a puntos del cluster vecino más cercano (separación).

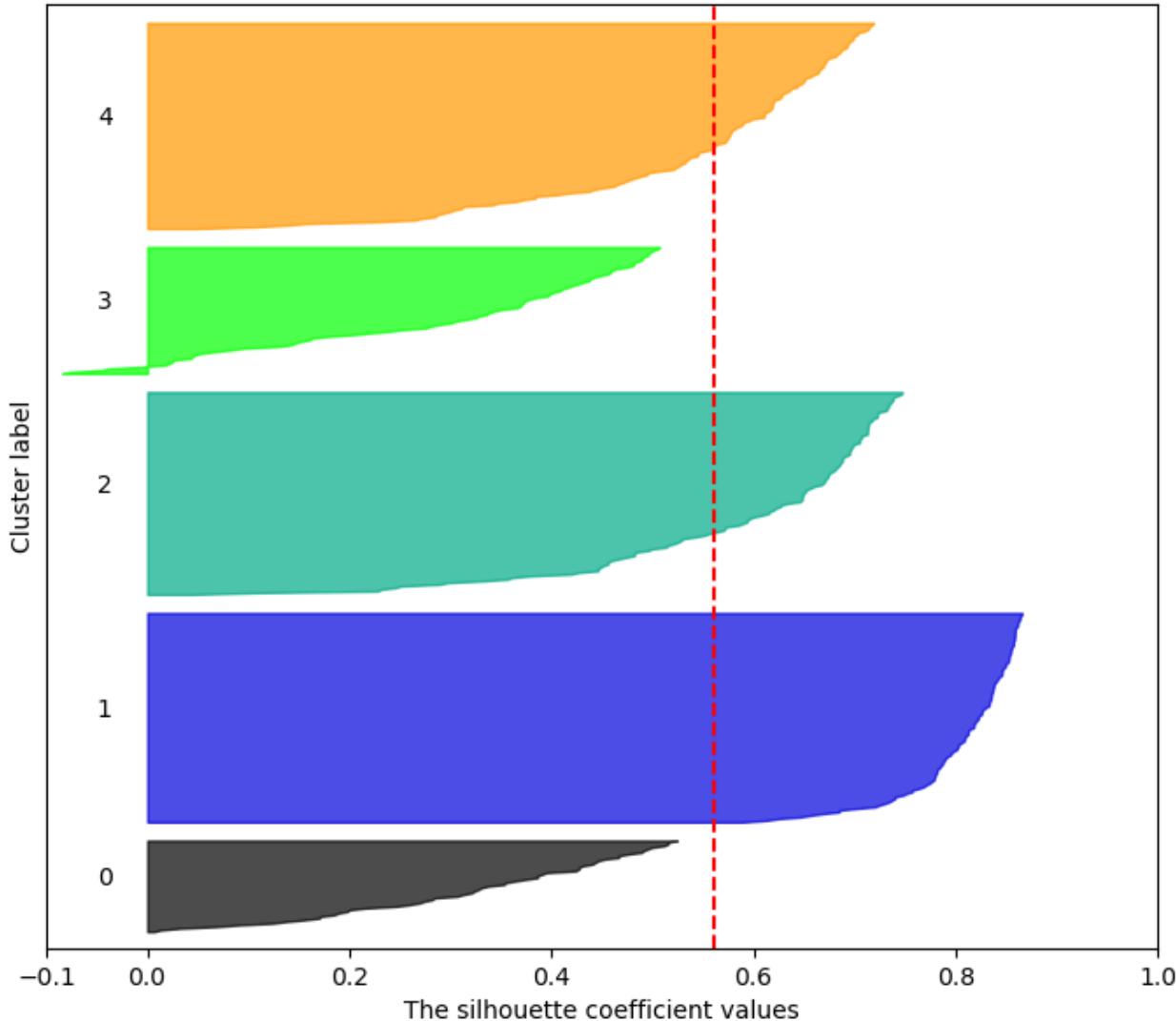
Interpretación del valor

s ≈ 1: Punto bien asignado, lejos de clusters vecinos.

s ≈ 0: Punto en frontera entre clusters.

s < 0: Punto probablemente mal asignado.

The silhouette plot for the various clusters.



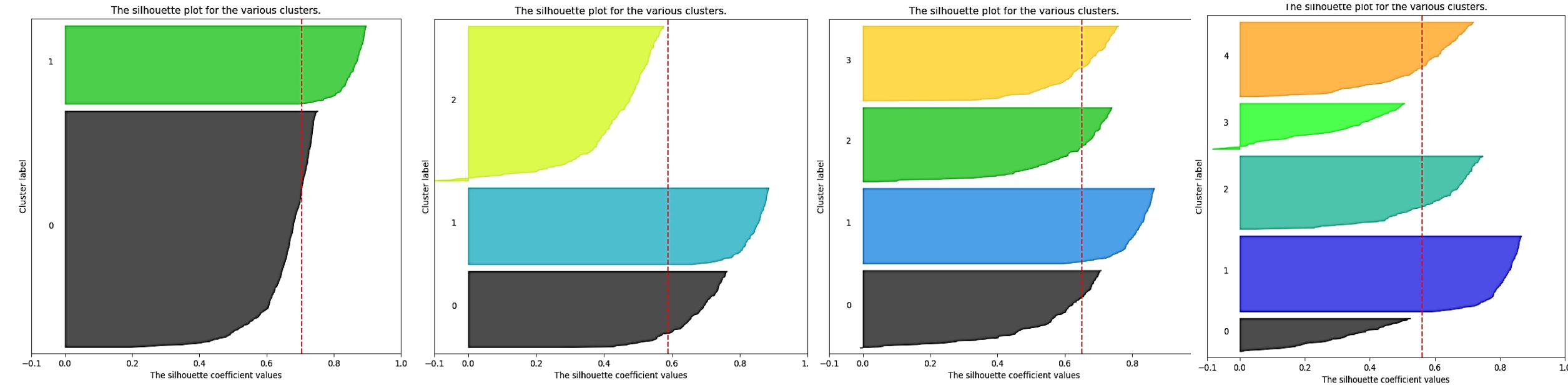
Selección de K: método de la silueta

Selección de k óptimo

Calcular silueta promedio para cada k. Elegir k que maximice el promedio global. Visualizar diagrama de siluetas por cluster para validar homogeneidad.

Ventaja clave

Proporciona información por punto y por cluster, no solo global. Permite detectar clusters problemáticos.



Selección de K: Gap statistic

Idea fundamental

Comparar la inercia observada con la inercia esperada bajo una distribución de referencia sin estructura de clusters (hipótesis nula). El gap mide cuánto mejor es el clustering real que el aleatorio.

Fórmula

$$\text{Gap}(k) = E[\log(W_k^*)] - \log(W_k)$$

W_k : inercia observada. W_k^* : inercia de datos uniformes.

Ventajas

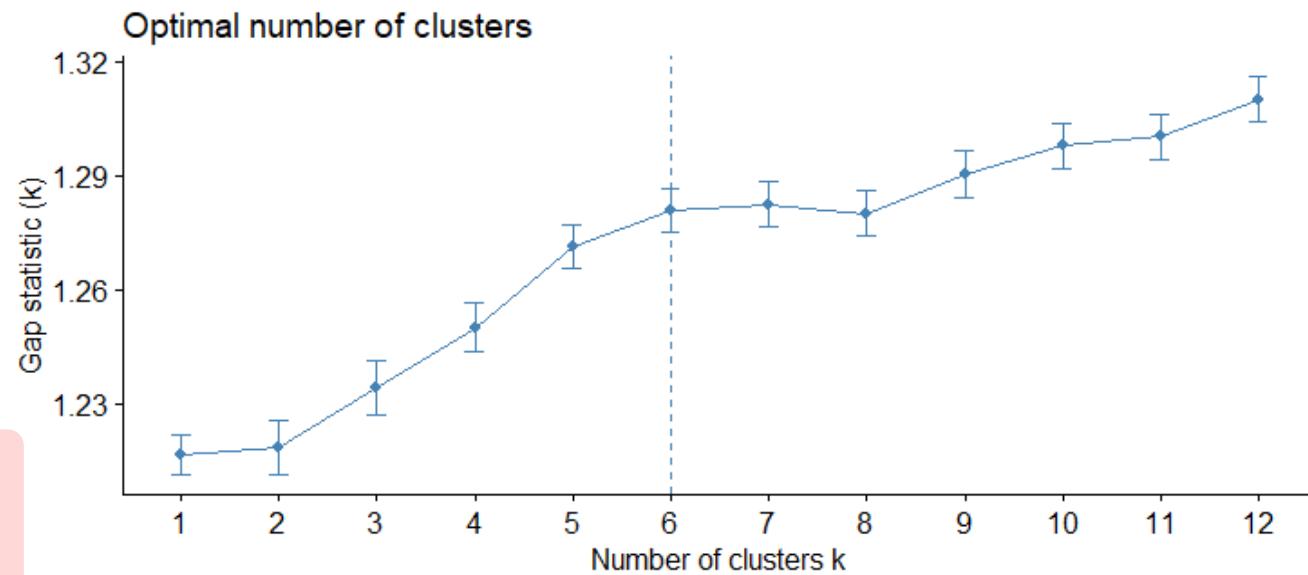
Fundamento estadístico sólido.
 Detecta $k=1$ (sin clusters).
 Criterio objetivo.

Limitaciones

Computacionalmente costoso (B simulaciones). Sensible a la distribución de referencia.

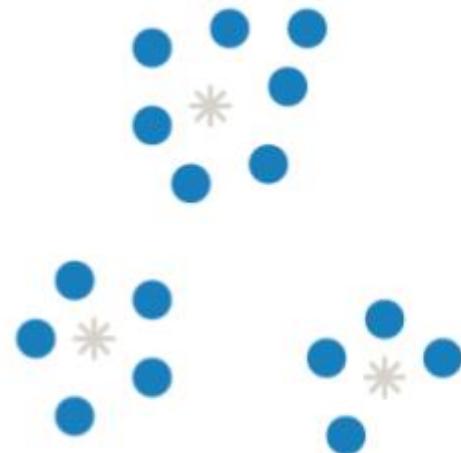
Procedimiento

1. Para cada k , calcular $\log(W_k)$ con datos reales.
2. Generar B datasets uniformes de referencia.
3. Calcular $E[\log(W_k^*)]$ y desviación s_k .
4. Elegir menor k donde $\text{Gap}(k) \geq \text{Gap}(k+1) - s_{k+1}$.



Variantes de K-Means

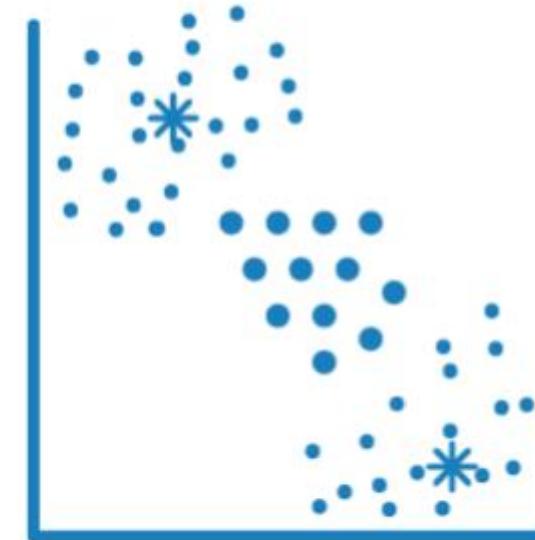
k-Means



k-Medoids



Fuzzy c-Means



Gaussian Mixture Model



K-Means ++

Idea central

Seleccionar centroides iniciales que estén lo más separados posible entre sí, utilizando muestreo probabilístico ponderado por distancia.

Algoritmo

1. Elegir el primer centroide uniformemente al azar.
2. Para cada punto x , calcular $D(x)$ = distancia al centroide más cercano ya elegido.
3. Elegir nuevo centroide con probabilidad proporcional a $D(x)^2$.
4. Repetir pasos 2-3 hasta tener k centroides.

Probabilidad de selección

$$P(x) = D(x)^2 / \sum_i D(x_i)^2$$

Puntos más alejados tienen mayor probabilidad

Ventajas

Mejor solución inicial.
 Convergencia más rápida.
 Garantía teórica: $O(\log k)$ competitivo.

Coste

$O(n \cdot k)$ adicional. Justificado por mejor calidad. Default en scikit-learn.



K-Medoids (PAM)

Concepto

Similar a K-Means pero usa medoides en lugar de centroides. Un medoide es el punto del cluster que minimiza la suma de distancias a todos los demás puntos del cluster.

PAM: Partitioning Around Medoids

BUILD: Selección inicial de medoides de forma greedy.

SWAP: Intercambiar medoides con no-medoides si reduce el coste total.

Ventajas

Robusto a outliers. Medoides son **puntos reales** del dataset. Funciona con cualquier métrica de distancia.

Desventajas

Complejidad $O(k(n-k)^2)$ por iteración. No escalable a datasets grandes.

Cuándo usar K-Medoids

Datasets pequeños-medianos con outliers. Cuando se necesita un representante real por cluster. Datos no numéricos con distancias definidas.

Variante escalable: CLARA

Aplica PAM sobre muestras del dataset. Mejor escalabilidad a costa de precisión.



Mini-Batch K-Means

Motivación

K-Means estándar requiere procesar todos los puntos en cada iteración. Para datasets muy grandes esto es prohibitivo computacionalmente.

Algoritmo

1. Muestrear un mini-batch aleatorio de b puntos.
2. Asignar puntos del batch al centroide más cercano.
3. Actualizar centroides con media ponderada (streaming).

Actualización de centroides

$$\mu_k \leftarrow (1 - \eta)\mu_k + \eta \cdot x$$

$\eta = 1/\text{count}(k)$ decrece con las asignaciones

Ventajas

Mucho más rápido. Escalable a millones de puntos. Uso de memoria constante.

Trade-off

Resultados ligeramente peores. Mayor varianza. Parámetro adicional: `batch_size`.

Cuándo usar

Datasets con más de 10,000 puntos. Aplicaciones en tiempo real o streaming.



K-Modes y K-Prototypes

K-Modes

Extensión de K-Means para datos categóricos.

Distancia: Número de atributos diferentes (Hamming).

Centroide: Moda de cada atributo en el cluster.

Distancia de matching

$$d(x,y) = \sum_j \delta(x_j \neq y_j)$$

Cuenta el número de atributos que difieren

K-Prototypes

Combina K-Means y K-Modes para datos mixtos (numéricos y categóricos).

$$d = d_{\text{euclídea}} + \gamma \cdot d_{\text{matching}}$$

γ balancea la contribución de cada tipo

Aplicaciones

Segmentación de clientes.
Análisis de encuestas. Datos transaccionales.

Consideraciones

Selección de γ es crucial.
Sensible a la escala de variables numéricas.



MÓDULO 2.2

Clustering jerárquico

Algoritmos de Particionamiento para Clustering



Introducción al clustering jerárquico

Definición

Familia de algoritmos que construyen una jerarquía anidada de clusters, representable mediante una estructura de árbol llamada dendrograma.

Características principales

No requiere especificar k a priori. Proporciona múltiples niveles de granularidad. Permite visualizar la estructura de agrupamiento a diferentes escalas. Resultado determinístico (sin inicialización aleatoria).

Diferencias con K-Means

K-Means: Partición plana, requiere k , resultado depende de inicialización.

Jerárquico: Estructura anidada, k flexible post-hoc, determinístico.

Aplicaciones típicas

Análisis filogenético en biología. Taxonomías de documentos o productos. Segmentación exploratoria cuando k es desconocido. Análisis de estructuras sociales.



Enfoques: Aglomerativo vs Divisivo

Aglomerativo (Bottom-Up)

Inicio: Cada punto es un cluster individual (n clusters).

Proceso: En cada paso, fusionar los dos clusters más similares.

Fin: Todos los puntos en un único cluster.

Divisivo (Top-Down)

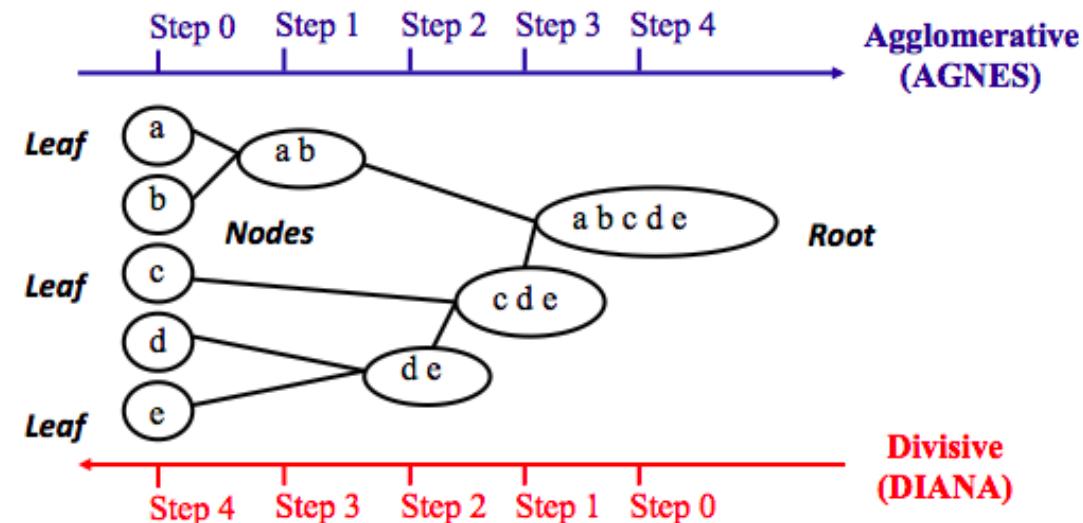
Inicio: Todos los puntos en un único cluster.

Proceso: En cada paso, dividir un cluster en dos.

Fin: Cada punto es un cluster individual.

Ventajas

- Más utilizado en la práctica.
- Implementación eficiente disponible.
- Múltiples criterios de enlace.

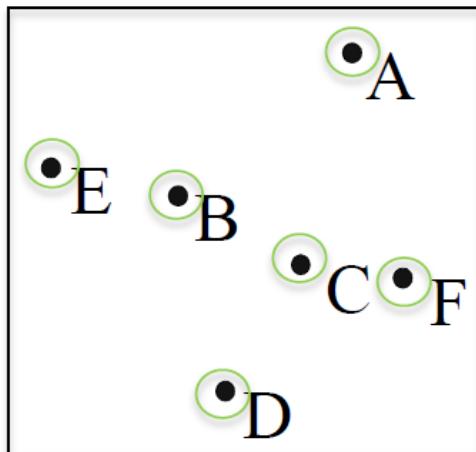


Desafíos

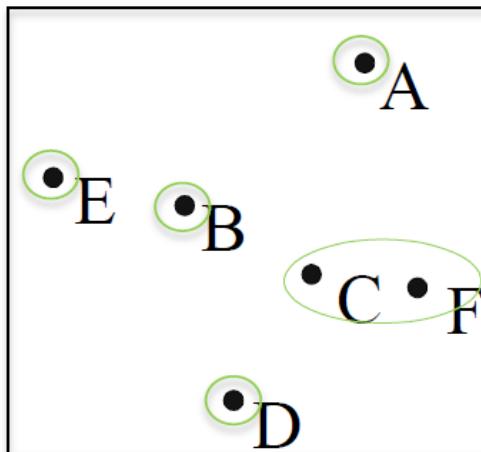
- Complejidad $O(2^n)$ para encontrar división óptima.
- Menos común en la práctica. Ejemplo: algoritmo DIANA.

Clustering aglomerativo

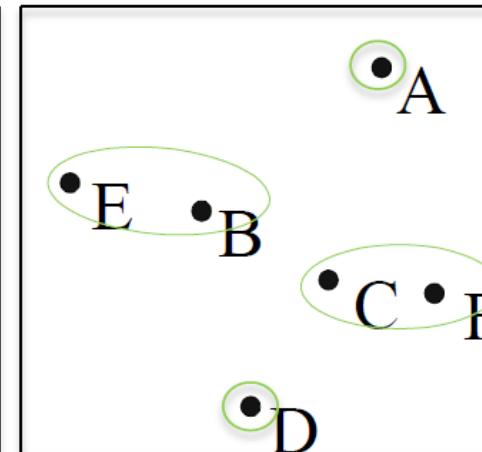
- Agrupa los elementos según niveles de proximidad /medidas de distancia
- Diferente medida de distancia = diferente agrupación



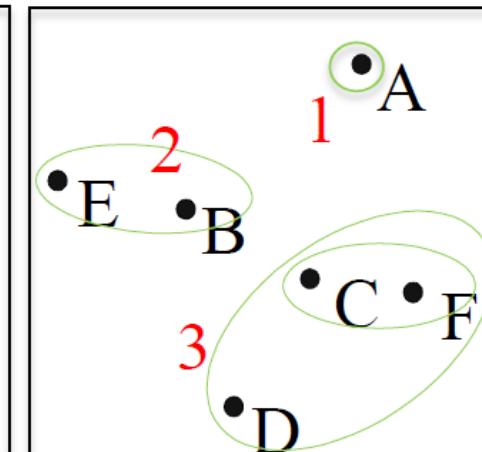
Stage 1



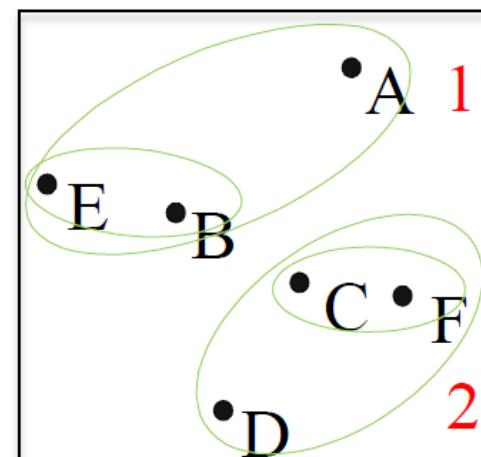
Stage 2



Stage 3



Stage 4
(three top-level clusters)



Stage 5 (two top-level clusters)

Algoritmo aglomerativo

Paso 1: Inicialización

Asignar cada observación a su propio cluster. Calcular matriz de distancias entre todos los pares.

Paso 2: Búsqueda

Identificar el par de clusters con menor distancia según el criterio de enlace seleccionado.

Paso 3: Fusión

Combinar los dos clusters más cercanos en un nuevo cluster. Actualizar la matriz de distancias.

Paso 4: Iteración

Repetir pasos 2 y 3 hasta que todos los puntos pertenezcan a un único cluster ($n-1$ fusiones).

Elemento clave: Criterio de enlace (Linkage)

Define cómo se calcula la distancia entre clusters. Diferentes criterios producen jerarquías diferentes. Los más comunes: Single, Complete, Average y Ward.



Criterios de enlace (*Linkage*)

Problema central

¿Cómo definir la distancia entre dos clusters cuando cada uno contiene múltiples puntos? El criterio de enlace determina esta definición.

Single Linkage

Distancia mínima entre cualquier par de puntos de ambos clusters.

Complete Linkage

Distancia máxima entre cualquier par de puntos de ambos clusters.

Average Linkage

Promedio de todas las distancias entre pares de puntos de ambos clusters (UPGMA).

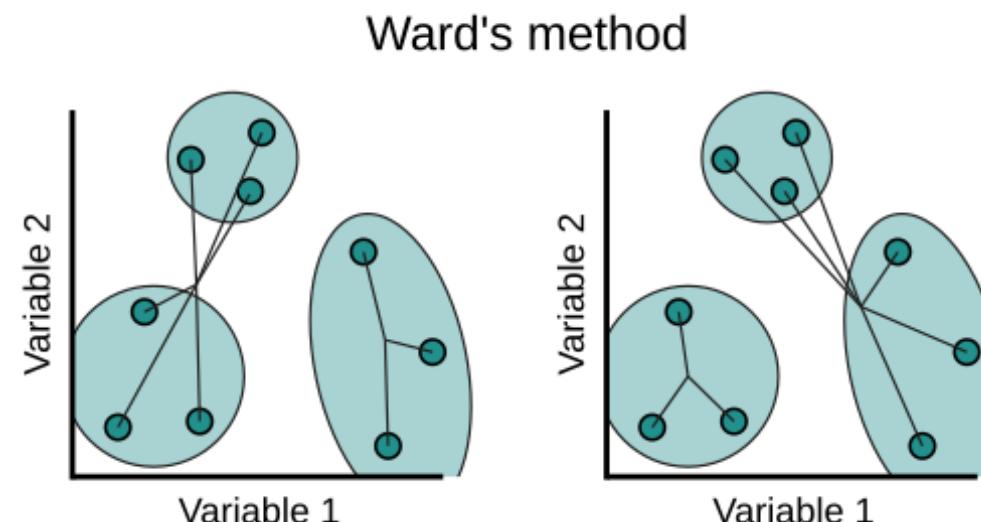
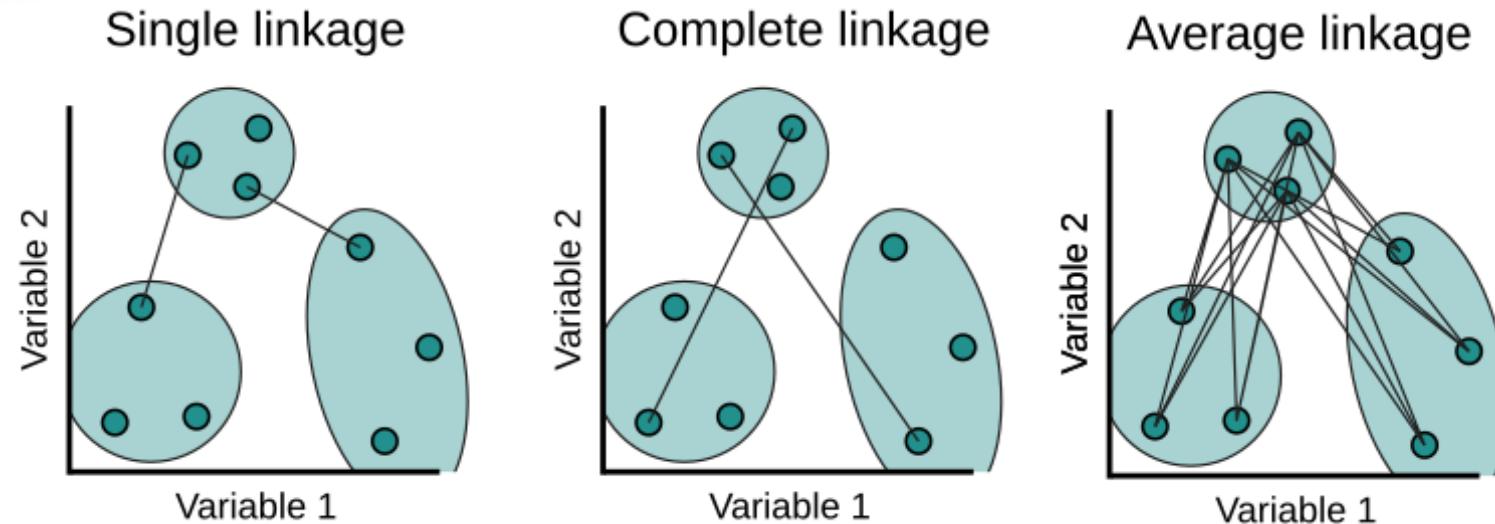
Ward's Method

Minimiza el incremento en la varianza total intra-cluster tras la fusión.

La elección del criterio de enlace tiene un impacto significativo en la forma y estructura de los clusters resultantes.



Criterios de enlace (*Linkage*)



Single Linkage (enlace simple)

Definición matemática

$$d(A,B) = \min\{d(a,b) : a \in A, b \in B\}$$

También conocido como: Nearest Neighbor

Comportamiento

Tiende a producir clusters elongados y cadenas. Puede detectar clusters de forma no convexa. Sensible al ruido y outliers (efecto cadena).

Ventajas

Detecta clusters no esféricos. Buen rendimiento con clusters bien separados. Complejidad reducida.

Limitaciones

Efecto cadena (chaining). Sensible a outliers. Clusters poco compactos.

Cuándo usar

Clusters con formas irregulares o alargadas. Datos sin ruido significativo. Cuando se busca conectividad entre puntos.

Produce dendrogramas con ramificaciones desiguales y extendidas.



Complete Linkage (enlace completo)

Definición matemática

$$d(A, B) = \max\{d(a, b) : a \in A, b \in B\}$$

También conocido como: Farthest Neighbor

Comportamiento

Tiende a producir clusters compactos y esféricos. Evita el efecto cadena del single linkage. Sensible a outliers de manera diferente.

Ventajas

Clusters compactos. Evita chaining. Buen control del diámetro del cluster.

Limitaciones

Sensible a outliers. Puede partir clusters grandes. Asume forma esférica.

Cuándo usar

Clusters aproximadamente esféricos. Se requieren clusters de diámetro controlado. Datos con pocos outliers extremos.

Produce dendrogramas más balanceados que single linkage.



Average Linkage (UPGMA)

Definición matemática

$$d(A,B) = (1/|A||B|) \sum d(a,b)$$

UPGMA: Unweighted Pair Group Method with Arithmetic Mean

Comportamiento

Compromiso entre single y complete linkage. Menos sensible a outliers que ambos extremos. Produce clusters de tamaño moderado.

Ventajas

Balance entre compacidad y conectividad. Robusto a outliers. Muy usado en biología.

Limitaciones

Mayor coste computacional. Puede no detectar clusters elongados.

Variantes

UPGMA: Promedio no ponderado (más común).

WPGMA: Promedio ponderado por tamaño de cluster.

Estándar en análisis filogenético y taxonomía.



Método de Ward

Criterio de fusión

$$\Delta E = E(A \cup B) - E(A) - E(B)$$

Minimiza el incremento en la suma de cuadrados intra-cluster (ESS)

Relación con K-Means

Misma función objetivo que K-Means (minimizar varianza intra-cluster).

Optimización jerárquica en lugar de iterativa. Produce clusters similares a K-Means.

Ventajas

Clusters compactos y esféricos.
Minimiza varianza total. Muy popular en práctica.

Limitaciones

Sensible a outliers. Tiende a clusters de igual tamaño. Solo distancia euclídea.

Cuándo usar

Cuando se esperan clusters compactos. Datos sin outliers extremos.
Como alternativa jerárquica a K-Means.

Generalmente considerado el método más efectivo para la mayoría de aplicaciones.

Comparación de criterios de enlace

Criterio	Forma clusters	Sensibilidad	Uso típico
Single	Elongados, cadenas	Muy alta a ruido	Formas arbitrarias
Complete	Compactos, esféricos	Moderada	Diámetro controlado
Average	Intermedios	Baja	Biología, taxonomía
Ward	Compactos, similares	Moderada a outliers	Uso general

Recomendación práctica

Comenzar con Ward para obtener una primera aproximación. Probar Average si se sospecha de formas irregulares. Usar Single solo para detectar estructuras de conectividad.

Dendrogramas

Definición

Diagrama de árbol que representa la secuencia de fusiones (o divisiones) realizadas durante el clustering jerárquico.

Elementos del dendrograma

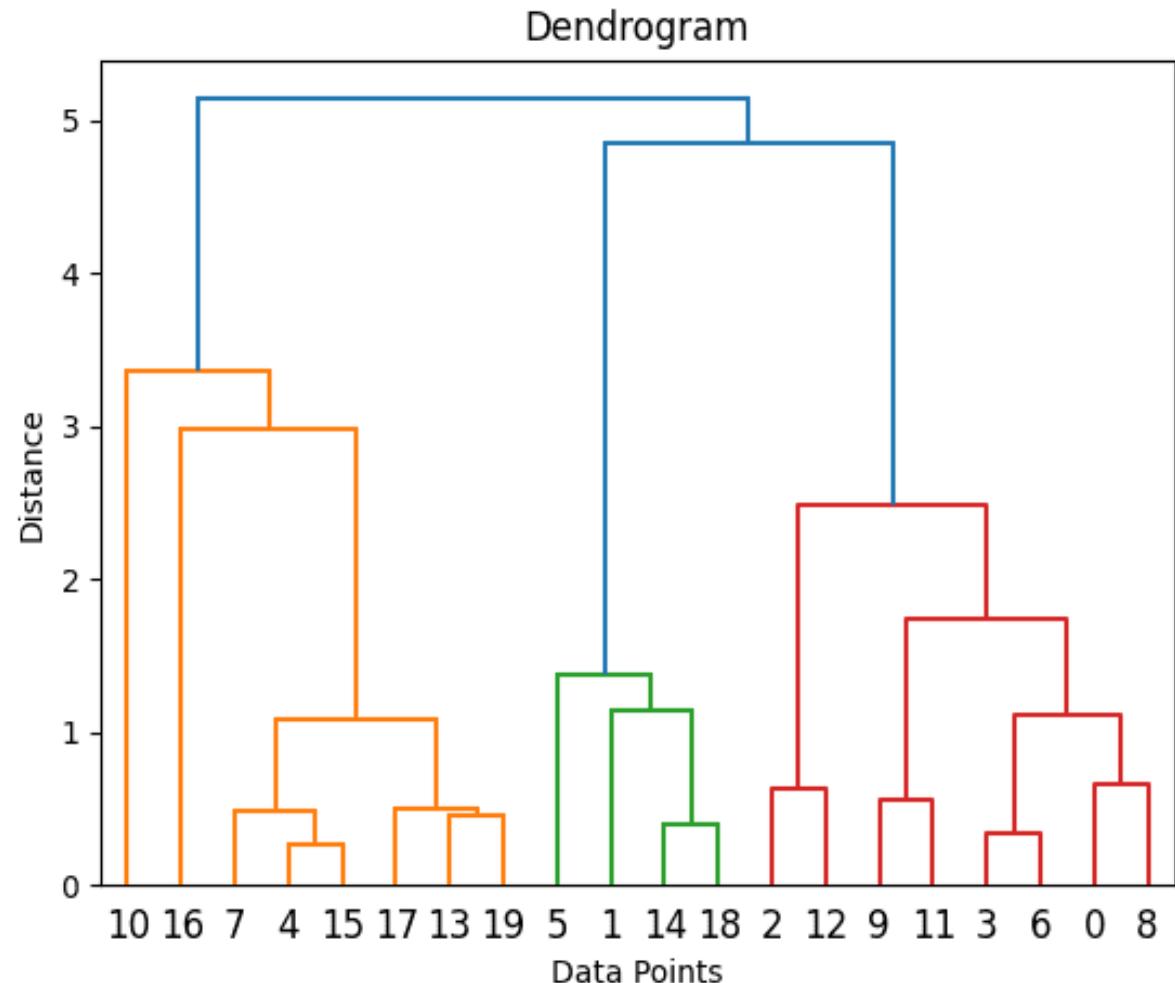
Hojas: Observaciones individuales (base del árbol).

Nodos: Puntos de fusión de clusters.

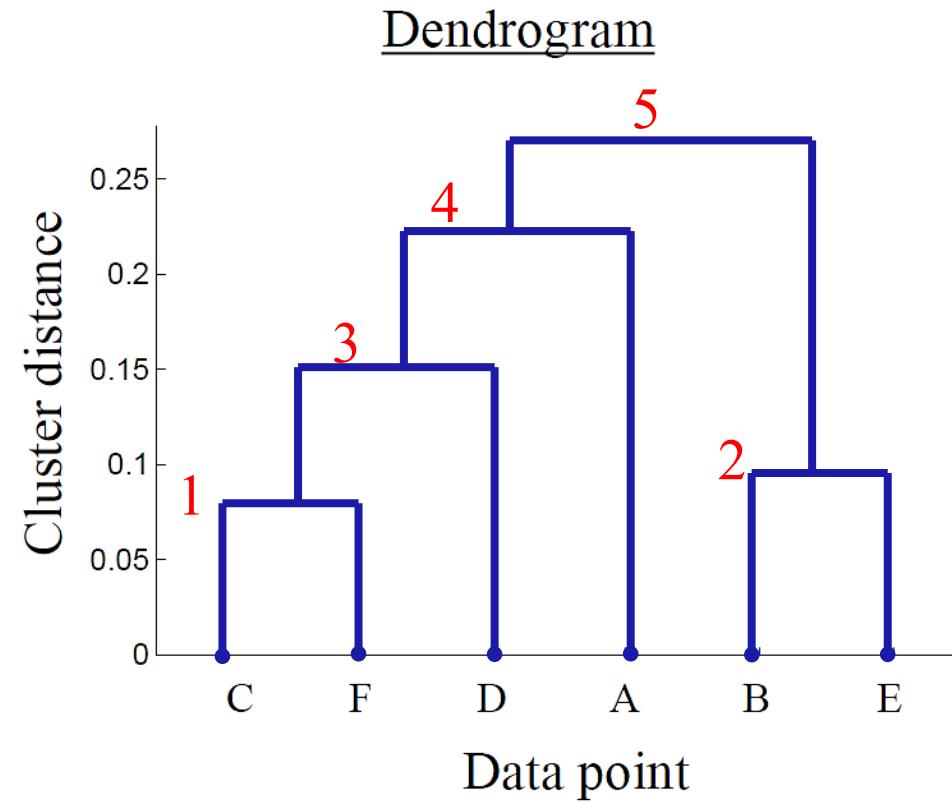
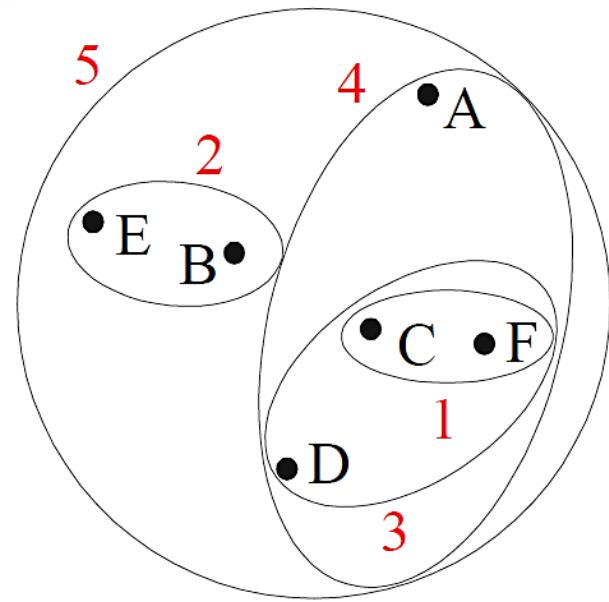
Altura: Distancia a la que ocurre cada fusión.

Información codificada

Orden de fusiones. Distancias entre clusters. Estructura jerárquica completa. Múltiples niveles de granularidad.



Dendrogramas: construcción



Proceso de construcción

1. Cada punto es una hoja en la base.
2. Conectar clusters fusionados con ramas.
3. Altura del nodo = distancia de fusión.
4. Continuar hasta la raíz (un cluster).

- A partir de alguna medida de distancia
- No requiere especificar número de clusters
- Presenta una “estructura” interna de relación de los elementos entre sí

Dendrogramas: interpretación

Lectura del eje vertical

La altura representa la distancia o disimilitud a la que se fusionan los clusters. Fusiones más altas indican clusters más diferentes.

Lectura del eje horizontal

Las observaciones se ordenan para minimizar cruces de ramas. La posición horizontal no tiene significado cuantitativo.

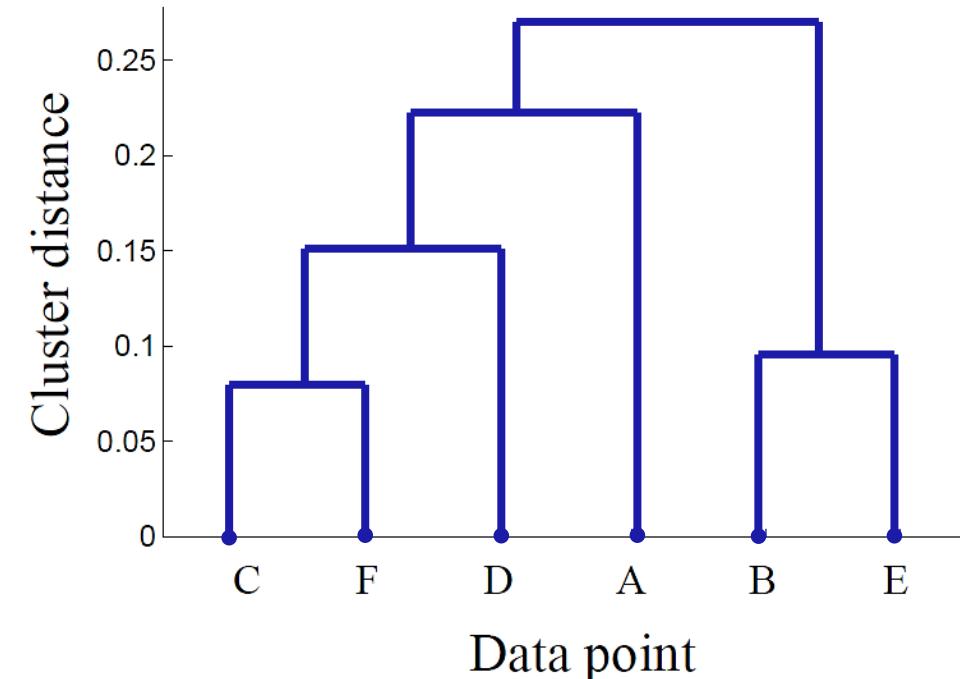
Identificación de clusters

Ramas verticales largas sugieren clusters bien separados. Ramas cortas indican observaciones similares.

Buenas prácticas

Comparar dendrogramas con diferentes criterios de enlace. Truncar para datasets grandes. Validar con métricas cuantitativas.

Dendrogram



Precauciones

La proximidad horizontal no implica similitud. Diferentes ordenaciones pueden dar el mismo dendrograma.

Corte del dendrograma

Concepto

Trazar una línea horizontal a cierta altura para obtener una partición específica. Los clusters resultantes son las ramas que cruza la línea.

Selección del número de clusters

Buscar el corte que maximice la distancia vertical antes de la siguiente fusión. Los saltos grandes en altura sugieren fronteras naturales entre clusters.

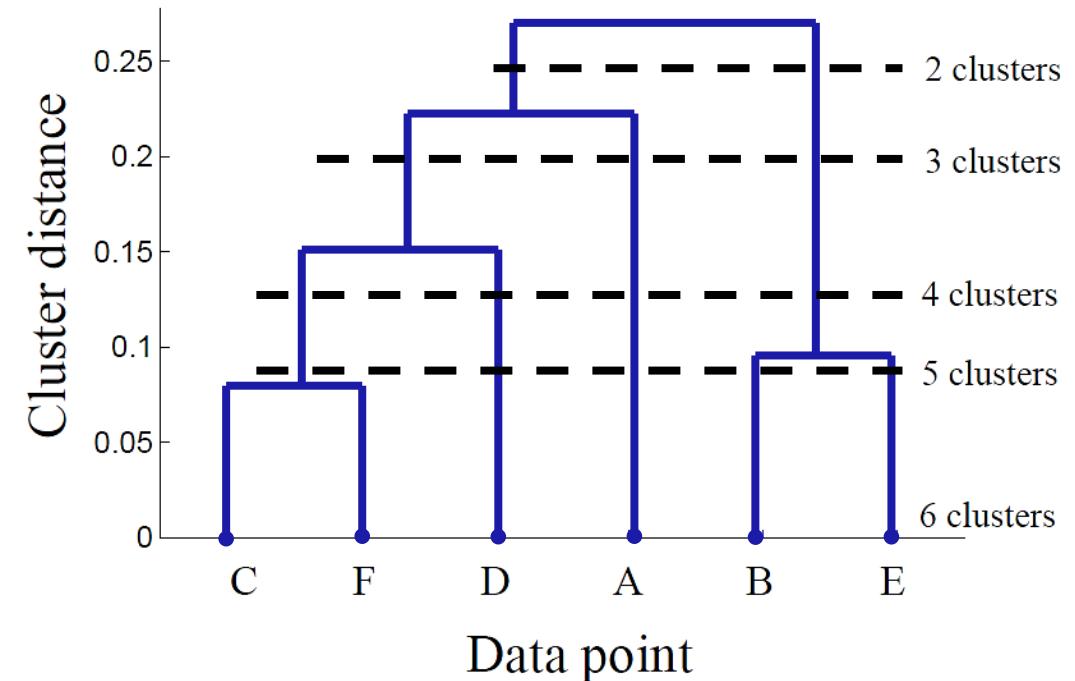
Métodos de selección de corte

Por altura: Especificar umbral de distancia máxima.

Por número: Especificar k clusters deseados.

Inconsistencia: Detectar saltos en distancias de fusión.

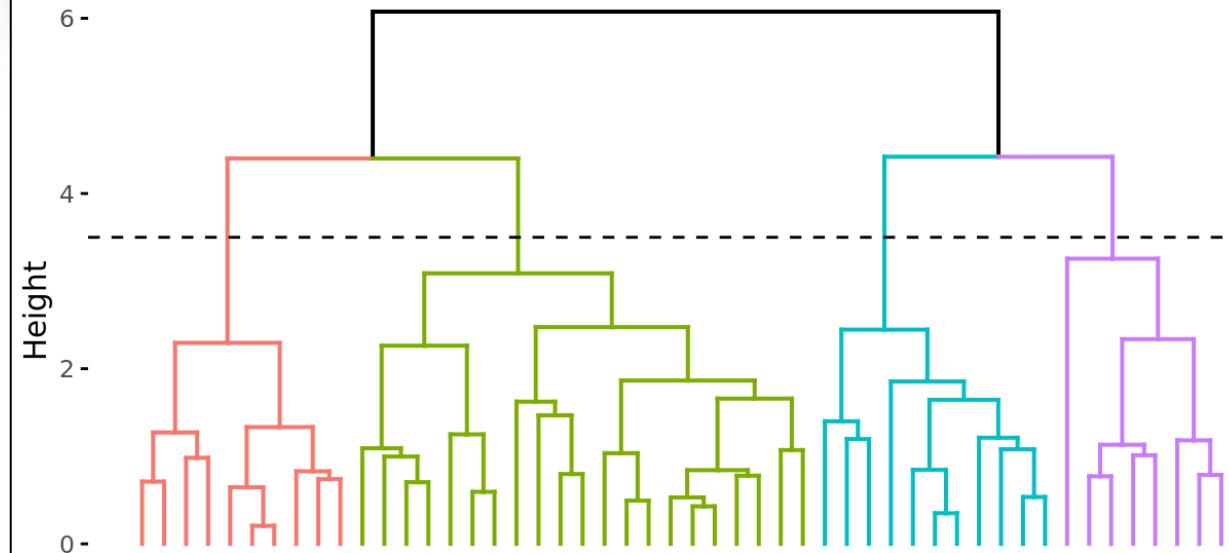
Dendrogram



Corte del dendrograma

Herarchical clustering

Distancia euclídea, Linkage complete, K=4



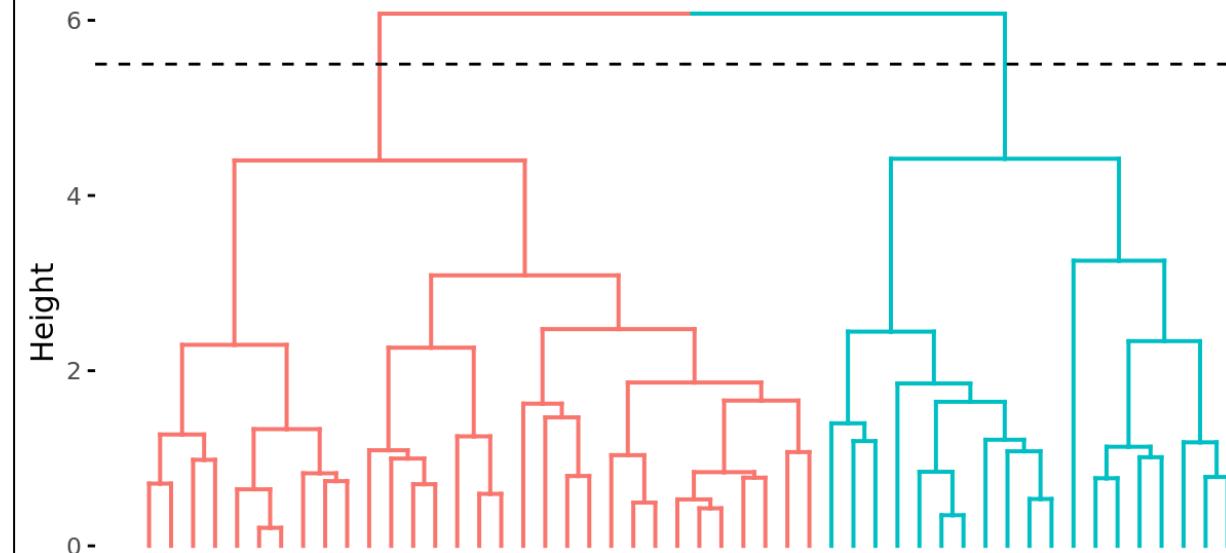
Fuente: <https://www.cienciadedatos.net/documentos/py20-clustering-con-python.html>

Criterio del salto máximo

Identificar la mayor diferencia entre alturas de fusión consecutivas. Cortar justo debajo del salto. Análogo al método del codo para K-Means.

Herarchical clustering

Distancia euclídea, Linkage complete, K=2



Coeficiente de inconsistencia

Compara la altura de cada nodo con la media de sus descendientes. Valores altos indican fusiones "inconsistentes" (buenos puntos de corte).

En la práctica

`scipy.cluster.hierarchy.fcluster` permite cortar por distancia (t) o número de clusters (k). Combinar con métricas de validación (silueta, etc.).

Complejidad computacional

Complejidad temporal

Naive: $O(n^3)$ - recalcular matriz completa en cada paso.

Optimizado: $O(n^2 \log n)$ - usando heap para búsqueda.

Single linkage: $O(n^2)$ - usando MST (Prim/Kruskal).

Complejidad espacial

$O(n^2)$ para almacenar la matriz de distancias. Puede ser prohibitivo para datasets grandes (n mayor que 10,000-50,000).

Limitaciones de escalabilidad

No escalable a big data sin modificaciones. La matriz de distancias crece cuadráticamente. Cada fusión requiere actualizaciones extensas.

Estrategias de escalabilidad

BIRCH: Pre-clustering para reducir n .

CURE: Representantes por cluster.

Muestreo: Clustering sobre subconjunto.

Para n mayor que 10,000, considerar K-Means o métodos aproximados.



MÓDULO 2.3

Clustering basado en densidad

Algoritmos de Particionamiento para Clustering

Motivación: clusters de forma arbitraria

Limitaciones de K-Means y Jerárquico

Asumen clusters de forma convexa (esférica). No manejan ruido ni outliers de forma natural. Requieren especificar k o interpretar dendrogramas. Sensibles a la inicialización o al criterio de enlace.

Ventajas del enfoque por densidad

Detecta clusters de forma arbitraria. Identifica automáticamente outliers como ruido. No requiere especificar k a priori. Robusto frente a puntos atípicos.

Idea central

Un cluster es una región de alta densidad de puntos, separada de otras regiones por zonas de baja densidad.

Aplicaciones típicas

Datos geoespaciales (GPS, mapas). Detección de anomalías. Segmentación de imágenes. Análisis de redes sociales. Bioinformática.

Los métodos basados en densidad son especialmente útiles cuando la estructura de los datos no se ajusta a formas geométricas predefinidas.



DBSCAN: Introducción

Density-Based Spatial Clustering of Applications with Noise

Propuesto por Ester et al. (1996). Algoritmo fundamental para clustering basado en densidad que agrupa puntos cercanos y marca como ruido los puntos aislados.

Parámetros principales

eps (ϵ): Radio de la vecindad para buscar puntos cercanos.

min_samples: Número mínimo de puntos para formar una región densa.

Intuición

Un punto pertenece a un cluster si tiene suficientes vecinos cercanos, o si está cerca de un punto que los tiene. Los puntos aislados son ruido.

Características

No requiere especificar k . Encuentra clusters de forma arbitraria. Identifica outliers automáticamente. Determinístico (excepto orden de procesamiento).

Complejidad

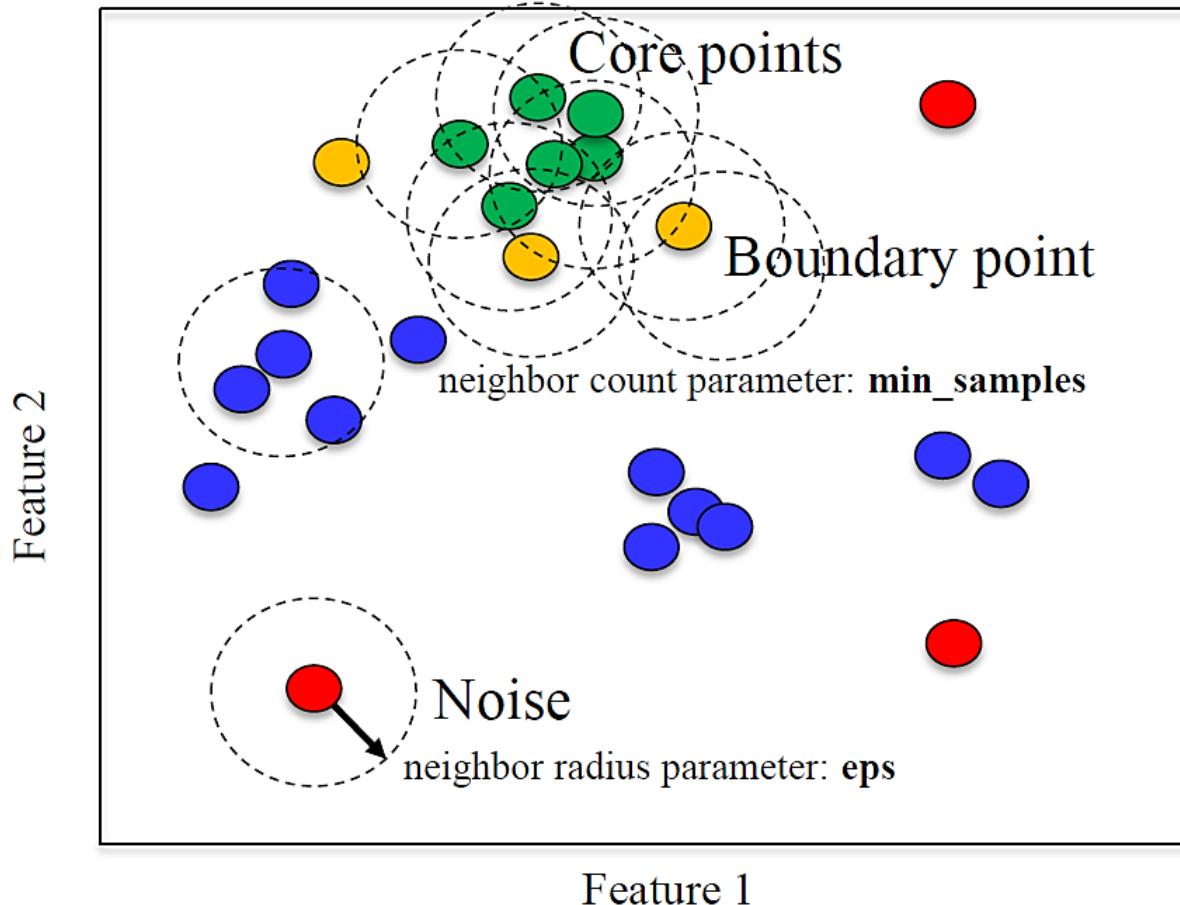
Tiempo: $O(n \log n)$ con índice espacial, $O(n^2)$ sin él.

Espacio: $O(n)$.

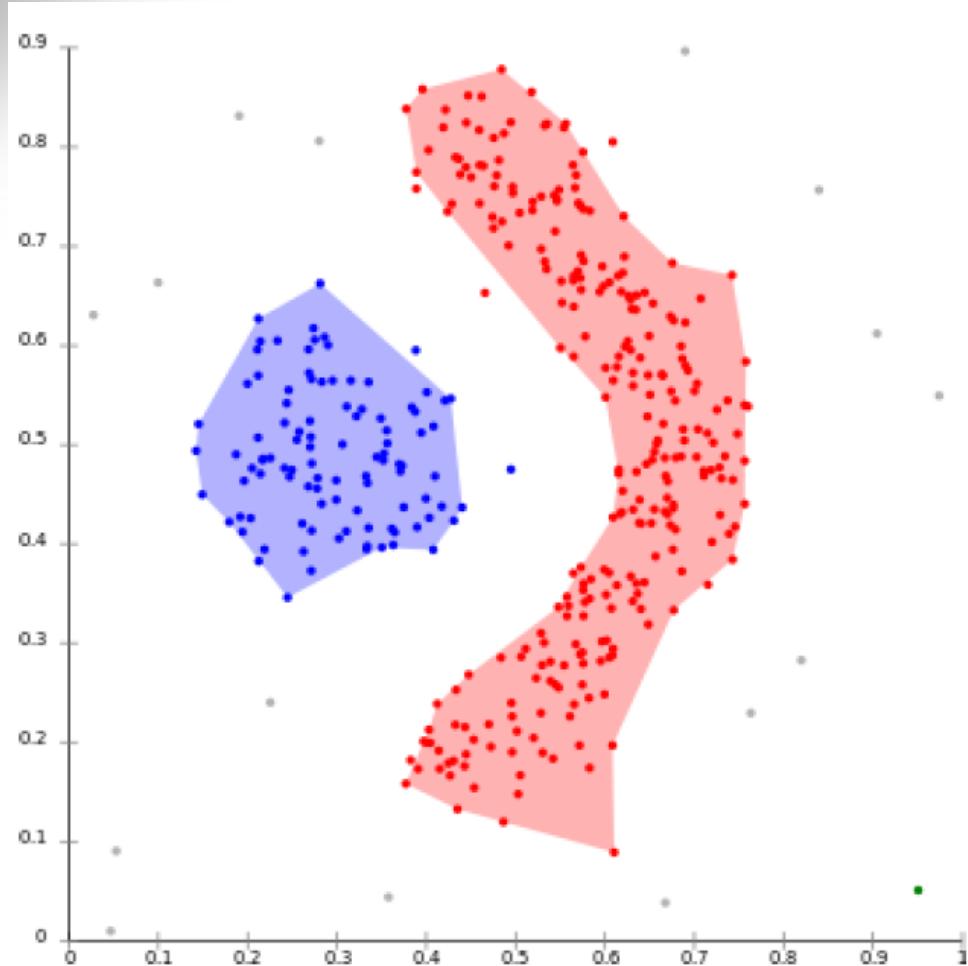


DBSCAN: Clustering

- No necesita especificar número de clústeres a buscar
- Eficiente: se puede emplear en datasets grandes
- Identifica **outliers/ruido**



DBSCAN: Clustering



- Encuentra clústeres no separables linealmente
- No depende de las condiciones iniciales
- Solo tiene dos parámetros que ajustar:
 - Umbral de distancia
 - Mínimo de vecinos
- Eficiente: se puede emplear en datasets grandes
- Identifica outliers

Concepto Epsilon-Vecindad (ε -neighborhood)

Definición formal

$$N_\varepsilon(p) = \{q \in D \mid \text{dist}(p, q) \leq \varepsilon\}$$

Conjunto de todos los puntos q cuya distancia a p es menor o igual a ε .

Interpretación geométrica

En 2D, la ε -vecindad es un círculo de radio ε centrado en el punto p.

En dimensiones mayores, es una hiperesfera.

Cardinalidad de la vecindad

El número de puntos en la ε -vecindad determina la densidad local:

$|N(p)|$ = número de puntos dentro del radio ε desde p (incluyendo p).

Rol del parámetro ε

ε pequeño: Vecindades pequeñas, muchos puntos aislados (ruido).

ε grande: Vecindades amplias, clusters pueden fusionarse.

La elección de ε es crítica para el resultado del algoritmo.

Tipos de puntos en DBSCAN

Punto Núcleo (Core Point)

$$|\text{N}_\epsilon(p)| \geq \text{min_samples}$$

Tiene al menos `min_samples` puntos en su ϵ -vecindad. Forma el núcleo de un cluster. Puede expandir el cluster a sus vecinos.

Punto Borde (Border Point)

$$|\text{N}_\epsilon(p)| < \text{min_samples}$$

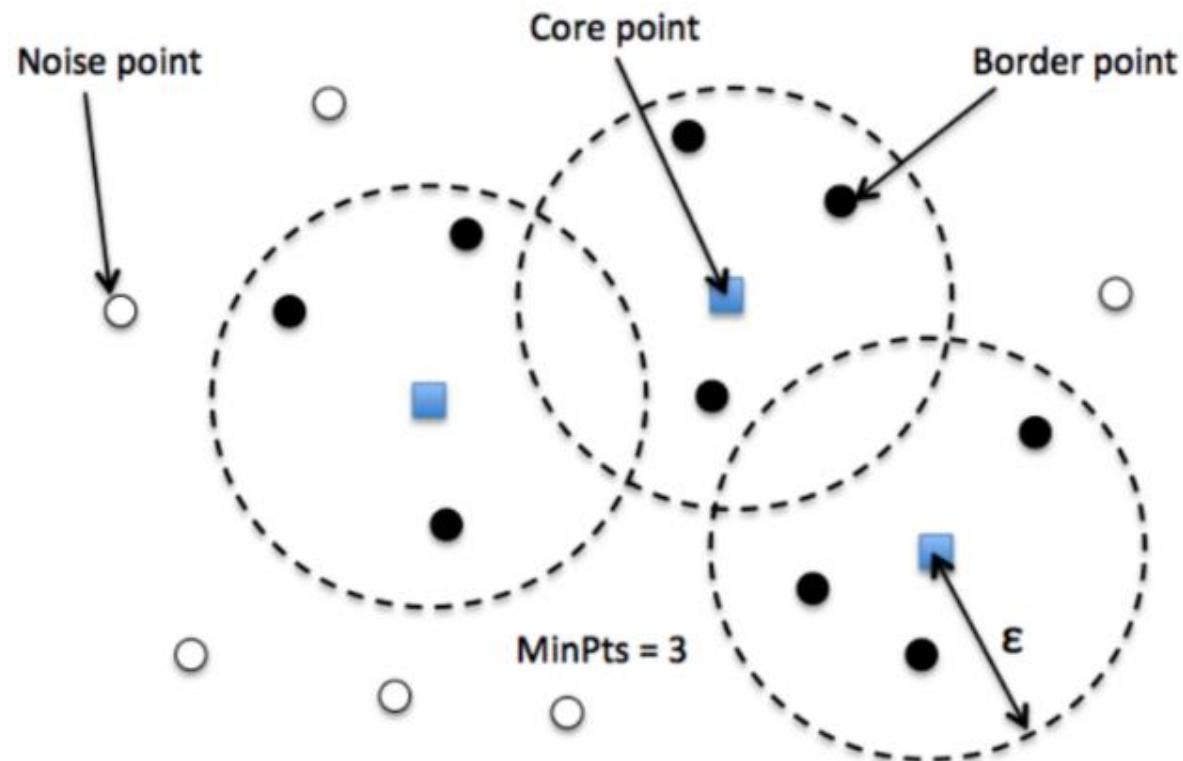
No es punto núcleo pero está en la ϵ -vecindad de uno. Pertenece al cluster pero no lo expande. Se encuentra en la periferia.

Punto Ruido (Noise Point)

No alcanzable

No es punto núcleo ni está en la vecindad de ninguno.

Considerado outlier o anomalía. Etiquetado como -1 en sklearn.



Algoritmo DBSCAN

Paso 1: Clasificación inicial

Para cada punto, calcular su ϵ -vecindad. Identificar puntos núcleo ($|N_\epsilon| \geq \text{min_samples}$).

Paso 2: Formación de clusters

Para cada punto núcleo no visitado, crear un nuevo cluster y expandirlo recursivamente.

Paso 3: Expansión del cluster

Añadir todos los puntos en la ϵ -vecindad al cluster. Si un vecino es punto núcleo, expandir recursivamente.

Paso 4: Etiquetado final

Puntos no asignados a ningún cluster se etiquetan como ruido (-1).

Pseudocódigo simplificado

```
Para cada punto p no visitado:  
    Marcar p como visitado  
    N = puntos en  $\epsilon$ -vecindad de p  
    Si  $|N| \geq \text{min\_samples}$ :  
        Crear nuevo cluster C  
        Expandir(p, N, C, eps, min_samples)  
    Sino: marcar p como ruido
```



Parámetros: *eps* y *min_samples*

eps (ϵ)

Radio máximo de la vecindad. Define la escala de densidad local.

ϵ muy pequeño: Pocos puntos núcleo, muchos clusters pequeños o ruido.

ϵ muy grande: Clusters se fusionan, se pierde estructura.

min_samples

Número mínimo de puntos para formar región densa. Controla la robustez al ruido.

min_samples pequeño: Más puntos núcleo, clusters menos compactos.

min_samples grande: Menos puntos núcleo, más ruido detectado.

Relación entre parámetros

eps y min_samples están interrelacionados. Un eps grande requiere min_samples mayor para mantener la misma definición de densidad.

Reglas heurísticas

min_samples $\geq d + 1$: donde d es la dimensionalidad.

min_samples = 2·d: recomendación común para datos ruidosos.

En scikit-learn: DBSCAN(eps=0.5, min_samples=5)

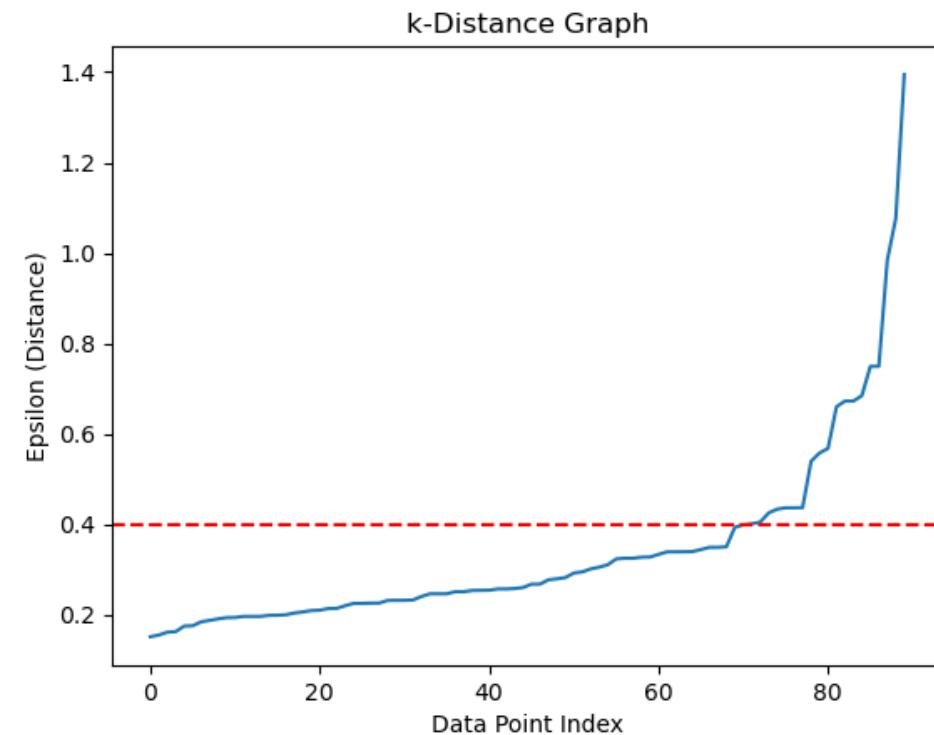
Selección de parámetros: k-Distance Graph

Método del k-distance

Para cada punto, calcular la distancia a su k-ésimo vecino más cercano ($k = \text{min_samples}$). Ordenar estas distancias de mayor a menor y graficarlas.

Procedimiento

1. Fijar $\text{min_samples} = k$ (ej: 4 o 5)
2. Calcular distancia al k-ésimo vecino para cada punto
3. Ordenar distancias de forma ascendente
4. Buscar el "codo" en la curva



El punto de inflexión (codo) indica el valor óptimo de eps .

Debajo del codo: Puntos en clusters (densidad alta).

Encima del codo: Puntos de ruido (densidad baja).

El método del k-distance es heurístico. En datasets complejos, puede ser necesario explorar múltiples valores de eps (grid search)

DBSCAN: Ventajas y limitaciones

Ventajas

Formas arbitrarias: Detecta clusters no convexos que K-Means no puede.

Detección de ruido: Identifica outliers automáticamente.

Sin k predefinido: El número de clusters emerge de los datos.

Eficiente: $O(n \log n)$ con índice espacial.

Determinístico: Resultado reproducible (excepto puntos borde).

Limitaciones

Densidad variable: Un solo eps no funciona si los clusters tienen densidades diferentes.

Alta dimensionalidad: La noción de densidad se degrada (maldición de la dimensionalidad).

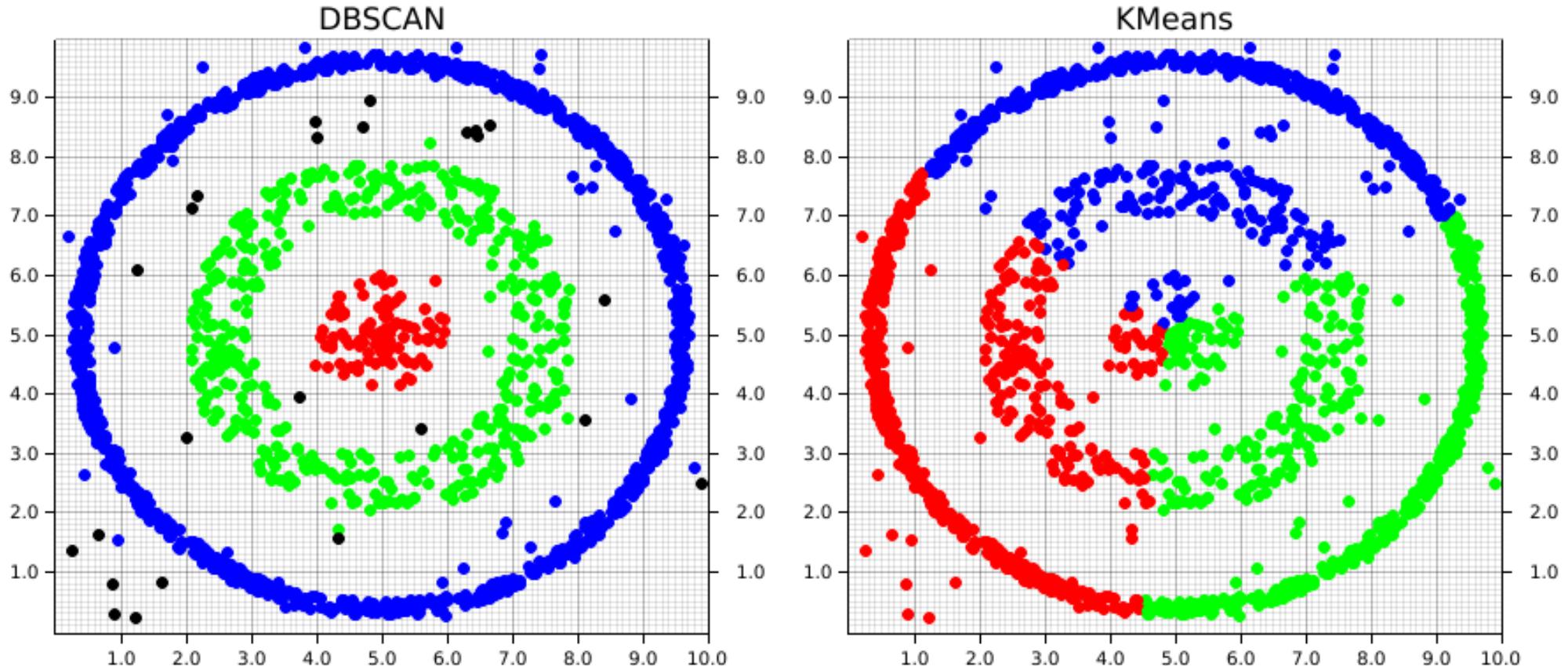
Sensibilidad a parámetros: eps y min_samples requieren ajuste cuidadoso.

Puntos borde: Su asignación puede variar según orden de procesamiento.

El problema de la densidad variable

Si un cluster tiene densidad alta y otro baja, un eps pequeño fragmentará el cluster denso, mientras que un eps grande fusionará el cluster disperso con el ruido. Este problema motiva extensiones como OPTICS y HDBSCAN.

DBSCAN: Ventajas y limitaciones



OPTICS: *Ordering Points To Identify Clustering Structure*

Motivación

DBSCAN requiere un único valor de eps , lo que es problemático cuando los clusters tienen densidades diferentes. OPTICS (Ankerst et al., 1999) resuelve esto creando un ordenamiento de los datos.

Conceptos clave

Core distance: Distancia mínima para que p sea punto núcleo.

Reachability distance: $\max(\text{core_dist}(o), \text{dist}(o,p))$ desde o hacia p .

Ordering: Secuencia de puntos procesados.

Idea principal

En lugar de asignar clusters directamente, OPTICS produce un ordenamiento lineal de los puntos que representa la estructura de densidad. Los clusters se extraen posteriormente.

Parámetros

min_samples: Igual que en DBSCAN.

max_eps: Límite superior (puede ser infinito).

OPTICS puede verse como una generalización de DBSCAN que funciona para un rango infinito de valores de eps simultáneamente.

OPTICS: Diagrama de alcanzabilidad (*reachability*)

Reachability Plot

Gráfico de barras donde el eje X es el orden de procesamiento y el eje Y es la reachability distance de cada punto.

Interpretación visual

Valles: Representan clusters (regiones de alta densidad).

Picos: Fronteras entre clusters o ruido.

Profundidad del valle: Indica densidad del cluster.

Extracción de clusters

Método ξ (xi): Detecta pendientes pronunciadas en el reachability plot.

Corte horizontal: Equivalente a ejecutar DBSCAN con ese eps.

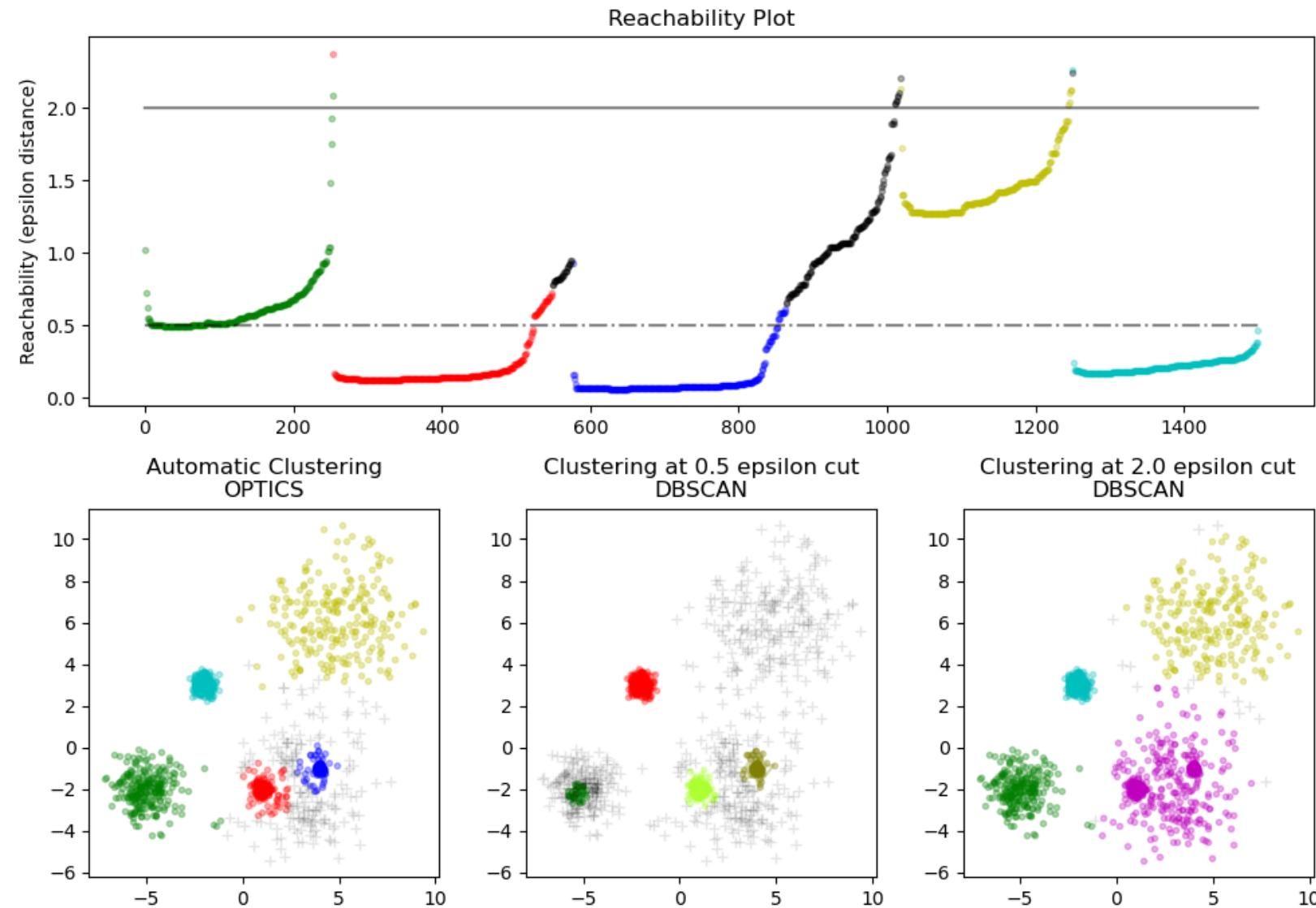
Implementación en sklearn

```
from sklearn.cluster import OPTICS
optics = OPTICS(min_samples=5, xi=0.05)
labels = optics.fit_predict(X)
reachability = optics.reachability_
```

Complejidad: $O(n^2)$ sin índice espacial, $O(n \log n)$ con índice.



OPTICS: Diagrama de alcanzabilidad (*reachability*)



HDBSCAN: Hierarchical DBSCAN

Definición

HDBSCAN (Campello et al., 2013) combina las ideas de DBSCAN con clustering jerárquico. Construye una jerarquía de clusters basada en densidad y extrae los clusters más estables.

Mutual Reachability Distance

$$\text{dmreach}(a,b) = \max(\text{core}(a), \text{core}(b), d(a,b))$$

Suaviza las distancias en regiones densas, manteniendo la estructura de densidad variable.

Proceso general

1. Transformar espacio usando mutual reachability distance.
2. Construir árbol de expansión mínima (MST).
3. Crear jerarquía de clusters conectados.
4. Condensar árbol y extraer clusters estables.

Parámetro principal

min_cluster_size: Tamaño mínimo de un cluster válido.

min_samples: Opcional, controla conservadurismo (por defecto = `min_cluster_size`).

HDBSCAN no requiere especificar eps, resolviendo el principal problema de DBSCAN.



HDBSCAN: Ventajas y características

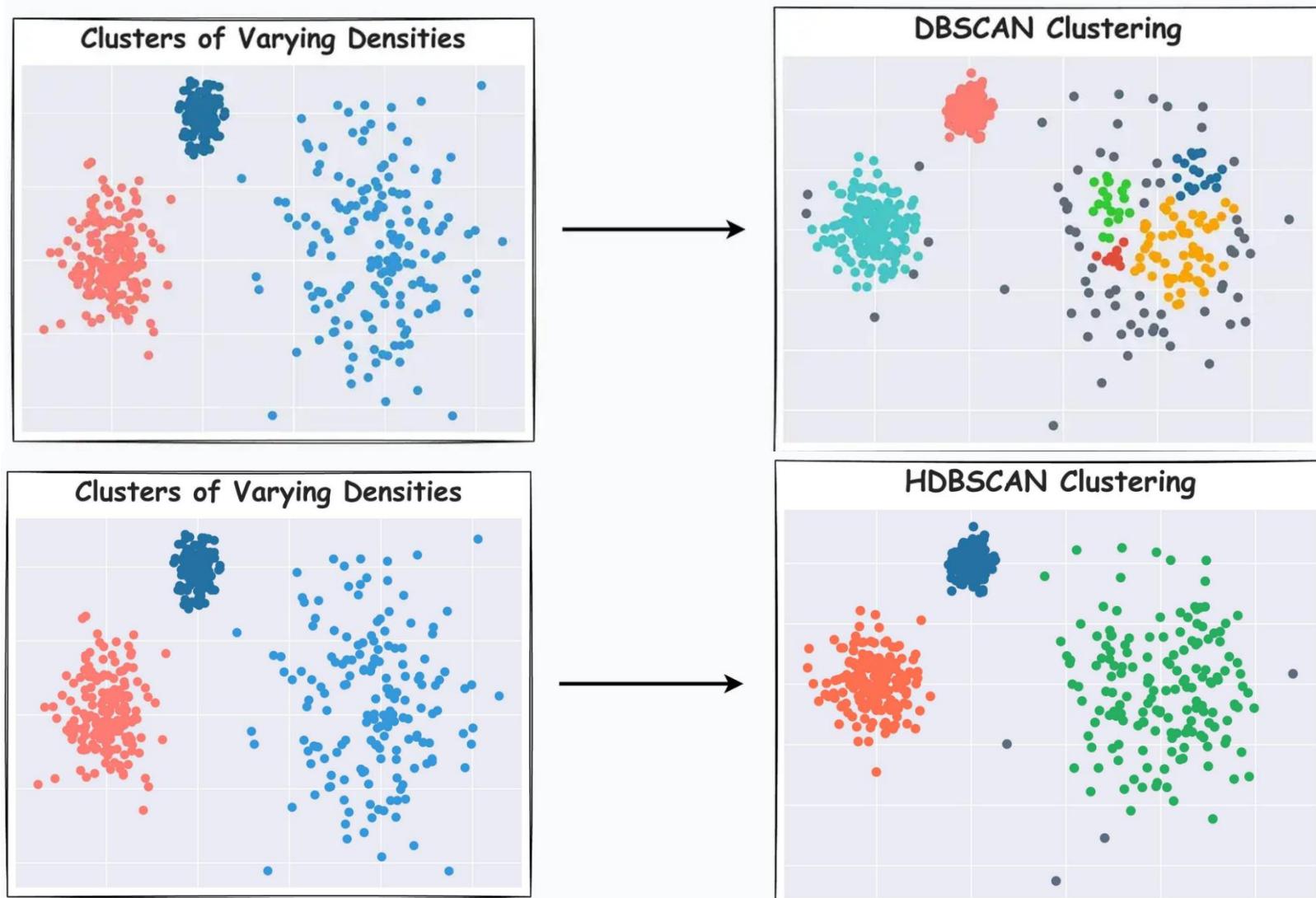
Limitaciones de DBSCAN

Densidad variable: Un solo eps no funciona si los clusters tienen densidades diferentes.

Alta dimensionalidad: La noción de densidad se degrada (maldición de la dimensionalidad).

Sensibilidad a parámetros: eps y min_samples requieren ajuste cuidadoso.

Puntos borde: Su asignación puede variar según orden de procesamiento.



HDBSCAN: Ventajas y características

Ventajas sobre DBSCAN

Sin eps: No requiere especificar radio de vecindad.

Densidad variable: Maneja clusters con diferentes densidades.

Selección automática: Determina número de clusters automáticamente.

Probabilidades: Proporciona probabilidad de pertenencia a cluster.

Soft clustering

HDBSCAN asigna probabilidades de pertenencia, permitiendo identificar puntos en fronteras de clusters.

Estabilidad de clusters

HDBSCAN selecciona clusters basándose en su estabilidad a través de diferentes niveles de la jerarquía. Clusters que persisten en muchos niveles son más confiables.

Implementación

```
import hdbscan
clusterer = hdbscan.HDBSCAN(
    min_cluster_size=15,
    min_samples=5
)
labels = clusterer.fit_predict(X)
probs = clusterer.probabilities_
```

Complejidad: $O(n \log n)$ tiempo, $O(n)$ espacio con optimizaciones.



Comparación de métodos basados en Densidad

Característica	DBSCAN	OPTICS	HDBSCAN
Parámetro principal	eps, min_samples	min_samples, xi	min_cluster_size
Densidad variable	No	Sí	Sí
Selección auto. de k	Sí	Sí	Sí
Probabilidades	No	No	Sí
Jerarquía de clusters	No	Sí (reachability plot)	Sí (condensed tree)
Complejidad tiempo	$O(n \log n)^*$	$O(n \log n)^*$	$O(n \log n)$

Recomendación: HDBSCAN para uso general. DBSCAN cuando la densidad es uniforme y se conoce un buen eps. OPTICS para análisis exploratorio de estructura jerárquica. (*con índice espacial)

MÓDULO 2.4

Modelos basados en modelos y otros algoritmos

Algoritmos de Particionamiento para Clustering

Gaussian Mixture Models (GMM)

Definición

Un GMM asume que los datos provienen de una mezcla de K distribuciones gaussianas, cada una con sus propios parámetros (media y covarianza). Es un modelo generativo probabilístico.

Parámetros del modelo

μ : Vector de medias del componente k

Σ : Matriz de covarianza del componente k

π : Probabilidad a priori del componente k

Modelo matemático

$$p(x) = \sum_{k=1}^K \pi_k \cdot N(x | \mu_k, \Sigma_k)$$

π_k : peso del componente k ($\sum \pi_k = 1$)

Ventajas sobre K-Means

Clusters elípticos (no solo esféricos). Soft clustering con probabilidades. Marco probabilístico formal. Permite selección de modelo con BIC/AIC.

K-Means es un caso especial de GMM con covarianzas esféricas iguales.



Algoritmo Espectation-Maximization (EM)

Paso E (Expectation)

Calcular las responsabilidades: probabilidad de que cada punto pertenezca a cada componente.

$$\gamma_{ik} = \pi_k N(x_i | \mu_k, \Sigma_k) / \sum_j \pi_j N(x_i | \mu_j, \Sigma_j)$$

Paso M (Maximization)

Actualizar los parámetros usando las responsabilidades como pesos:

$$N_k = \sum_i \gamma_{ik} \text{ (responsabilidad total)}$$

$$\mu_k = (1/N_k) \sum_i \gamma_{ik} x_i$$

$$\pi_k = N_k / N$$

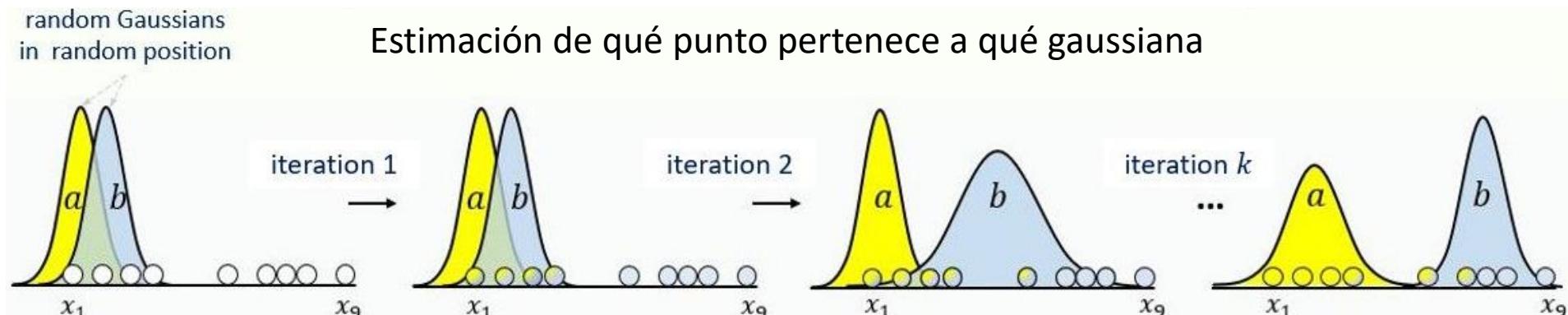
Proceso iterativo

1. Inicializar parámetros (μ, Σ, π)
2. Paso E: calcular responsabilidades
3. Paso M: actualizar parámetros
4. Repetir hasta convergencia

Convergencia

EM garantiza aumentar la log-verosimilitud en cada iteración. Converge a un máximo local (no necesariamente global).

Nota: Sensible a la inicialización. Usar múltiples inicializaciones (n_init en sklearn).



Tipos de Covarianza en GMM

Full (Completa)

Cada componente tiene su propia matriz de covarianza general.

Forma: Elipses orientadas arbitrariamente

Parámetros: $K \cdot d \cdot (d+1)/2$

Tied (Atada)

Todos los componentes comparten la misma covarianza.

Forma: Elipses con misma orientación y forma

Parámetros: $d \cdot (d+1)/2$

Diagonal

Matrices diagonales: variables independientes.

Forma: Elipses alineadas con ejes

Parámetros: $K \cdot d$

Spherical (Esférica)

$\text{Covarianza} = \sigma^2 I$ (varianza única por componente).

Forma: Círculos/esferas

Parámetros: K

Consideraciones para la selección

Más parámetros (full): Mayor flexibilidad pero riesgo de sobreajuste.
Requiere más datos.

Menos parámetros (spherical): Más robusto con pocos datos pero menos expresivo. Similar a K-Means.

En scikit-learn: `GaussianMixture(covariance_type='full'|'tied'|'diag'|'spherical')`

Selección del número de componentes

Como en otros algoritmos, utilizamos criterios de información para decidir el nº óptimo de componentes

BIC (Bayesian Information Criterion)

$$BIC = -2 \cdot \ln(L) + p \cdot \ln(n)$$

L: verosimilitud máxima. p: número de parámetros. n: número de observaciones. Penaliza más los modelos complejos.

Procedimiento

1. Ajustar GMM para $K = 1, 2, 3, \dots, K$
2. Calcular BIC (o AIC) para cada K
3. Seleccionar K que minimiza el criterio
4. Considerar el "codo" si no hay mínimo claro

AIC (Akaike Information Criterion)

$$AIC = -2 \cdot \ln(L) + 2 \cdot p$$

Penalización independiente del tamaño de muestra. Tiende a seleccionar modelos más complejos que BIC.

Recomendaciones

BIC: Preferido para selección de modelos. Más conservador.

AIC: Mejor para predicción. Puede sobreestimar K .

En sklearn: `gmm.bic(X)` y `gmm.aic(X)`

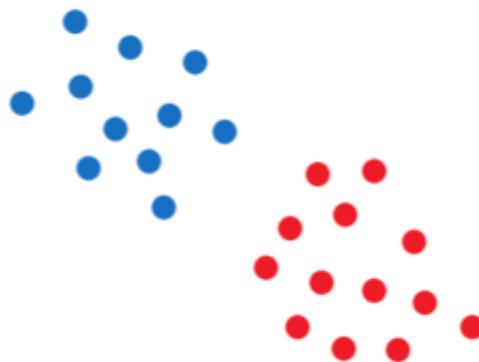
Hard Clustering VS Soft Clustering

Hard Clustering

Cada punto pertenece a exactamente un cluster. Asignación determinista.

Ejemplos: K-Means, DBSCAN, Jerárquico

Resultado: $y \in \{1, 2, \dots, K\}$



Limitaciones del Hard Clustering

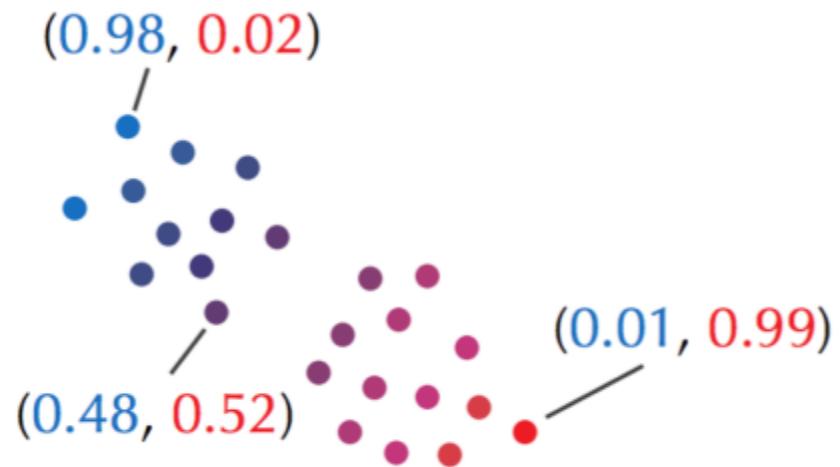
No cuantifica incertidumbre. Puntos en fronteras asignados arbitrariamente. Pérdida de información sobre la estructura.

Soft Clustering

Cada punto tiene probabilidad de pertenecer a cada cluster. Asignación probabilística.

Ejemplos: GMM, Fuzzy C-Means

Resultado: $P(y=k)$ para cada k



Ventajas del Soft Clustering

Cuantifica incertidumbre. Identifica puntos ambiguos. Más información para decisiones downstream.

Clustering Espectral

Idea central

En lugar de trabajar directamente en el espacio de características, transformamos los datos usando propiedades espectrales de un grafo de similitud, y luego aplicamos K-Means en el espacio transformado.

Proceso general

1. Construir grafo de similitud (matriz de afinidad)
2. Calcular matriz laplaciana del grafo
3. Extraer eigenvectores (representación espectral)
4. Aplicar K-Means en el nuevo espacio

¿Por qué funciona?

La transformación espectral "desenreda" estructuras no convexas. Puntos conectados en el grafo quedan cerca en el espacio espectral, incluso si estaban lejos en el espacio original.

Ventajas

Detecta clusters no convexos. No asume forma específica. Basado en teoría sólida de grafos y álgebra lineal.

Limitación: Complejidad $O(n^3)$ para el cálculo de eigenvectores. No escala bien a datasets grandes.



Matriz de afinidad y Laplaciano del grafo

Matriz de Afinidad (W)

Mide la similitud entre cada par de puntos. Valores altos = puntos similares.

RBF (Gaussiana):

$$W_{ij} = \exp(-||x_i - x_j||^2 / 2\sigma^2)$$

Matriz de Grado (D)

Matriz diagonal con la suma de afinidades de cada nodo.

$$D_{ii} = \sum_j W_{ij}$$

Captura cuánto está "conectado" cada nodo en el grafo. Puntos en zonas densas tendrán valores altos.

Laplaciano del Grafo (L)

No normalizado: $L = D - W$

Normalizado simétrico: $L = DLD^{-1}$

Normalizado RW: $L = DL^{-1}$

Propiedades del Laplaciano

Semidefinida positiva: eigenvalores ≥ 0

Número de eigenvalores = 0 indica componentes conexas

Los K eigenvectores correspondientes a los K menores eigenvalores codifican la estructura de clusters.

Algoritmo de Clustering Espectral

Paso 1: Matriz de Afinidad

Construir W usando kernel RBF o k-vecinos más cercanos. El parámetro γ (o σ) controla la escala de similitud.

Paso 2: Laplaciano

Calcular matriz de grado D y laplaciano L. Normalizar si es necesario (L_{sym} o L_{rw}).

Paso 3: Eigenvectores

Calcular los K eigenvectores correspondientes a los K menores eigenvalores de L. Formar matriz U $\in \mathbb{R}^{n \times K}$.

Paso 4: K-Means

Aplicar K-Means a las filas de U (cada fila = representación espectral de un punto). Asignar etiquetas.

Implementación en scikit-learn

```
from sklearn.cluster import SpectralClustering  
sc = SpectralClustering(n_clusters=3, affinity='rbf', gamma=1.0)  
labels = sc.fit_predict(X)
```

affinity='rbf': Kernel gaussiano, requiere ajustar gamma.

affinity='nearest_neighbors': Basado en k-NN, requiere ajustar n_neighbors.

Mean-shift

Concepto

Algoritmo basado en estimación de densidad. Cada punto se mueve iterativamente hacia la media de los puntos en su vecindad, convergiendo a los modos (máximos) de la densidad.

Algoritmo

1. Para cada punto, definir ventana de radio bandwidth
2. Calcular media ponderada de puntos en la ventana
3. Mover el centro de la ventana a la nueva media
4. Repetir hasta convergencia (modo local)

Parámetro: bandwidth

Radio de la ventana de búsqueda. Controla la escala de los clusters.
Puede estimarse automáticamente con `estimate_bandwidth()`

Ventajas

No requiere K. Encuentra número de clusters automáticamente.
Clusters de forma arbitraria.

Limitaciones

Costoso: $O(n^2)$. Sensible a bandwidth. No escala bien a alta dimensión.

`MeanShift(bandwidth=None, bin_seeding=True)` - `bin_seeding` acelera el proceso.

Affinity Propagation

Concepto

Algoritmo basado en paso de mensajes entre puntos. Los puntos "votan" para elegir ejemplares (representantes de clusters). No requiere especificar K.

Tipos de mensajes

Responsabilidad $r(i,k)$: Cuán adecuado es k como ejemplar de i, considerando otros candidatos.

Disponibilidad $a(i,k)$: Cuán apropiado es para i elegir a k como ejemplar, según otros puntos.

Parámetros

preference: Preferencia de cada punto a ser ejemplar. Valores más bajos = menos clusters.

damping: Factor de amortiguación [0.5, 1) para evitar oscilaciones.

Ventajas

No requiere K. Ejemplares son puntos reales. Basado en similitudes.

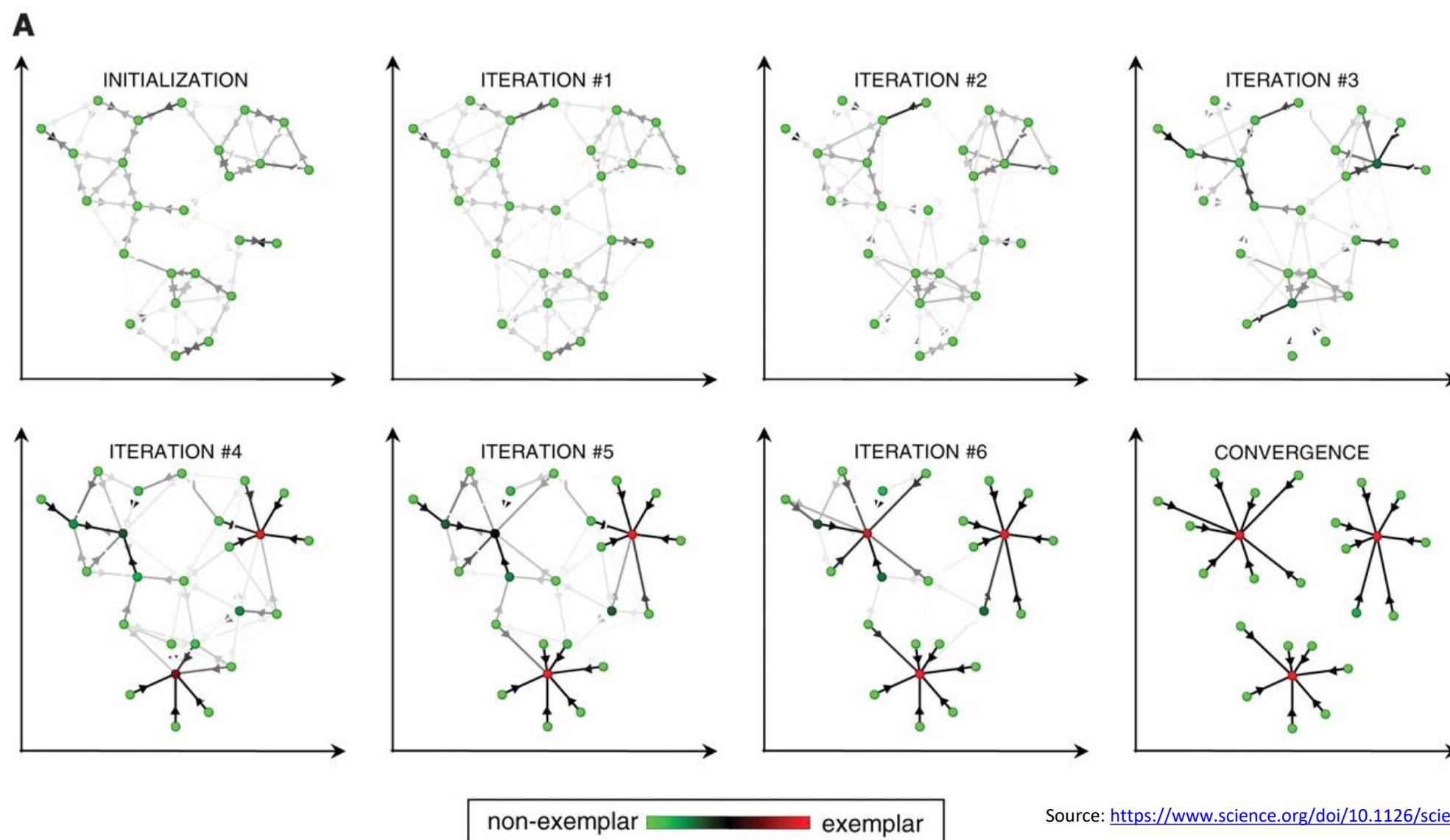
Limitaciones

$O(n^2)$ tiempo y espacio. Puede no converger. Sensible a preference.

AffinityPropagation(damping=0.9, preference=None)



Affinity Propagation



BIRCH

Balanced Iterative Reducing and Clustering using Hierarchies

Diseñado para Big Data

BIRCH está optimizado para datasets muy grandes que no caben en memoria. Construye un resumen compacto de los datos (CF Tree) con una sola pasada.

Clustering Feature (CF)

Resumen estadístico de un subcluster:

N: Número de puntos

LS: Suma lineal de puntos

SS: Suma de cuadrados

Parámetros principales

threshold: Radio máximo de un subcluster.

branching_factor: Máximo de hijos por nodo.

n_clusters: Número final de clusters (opcional).

Ventajas

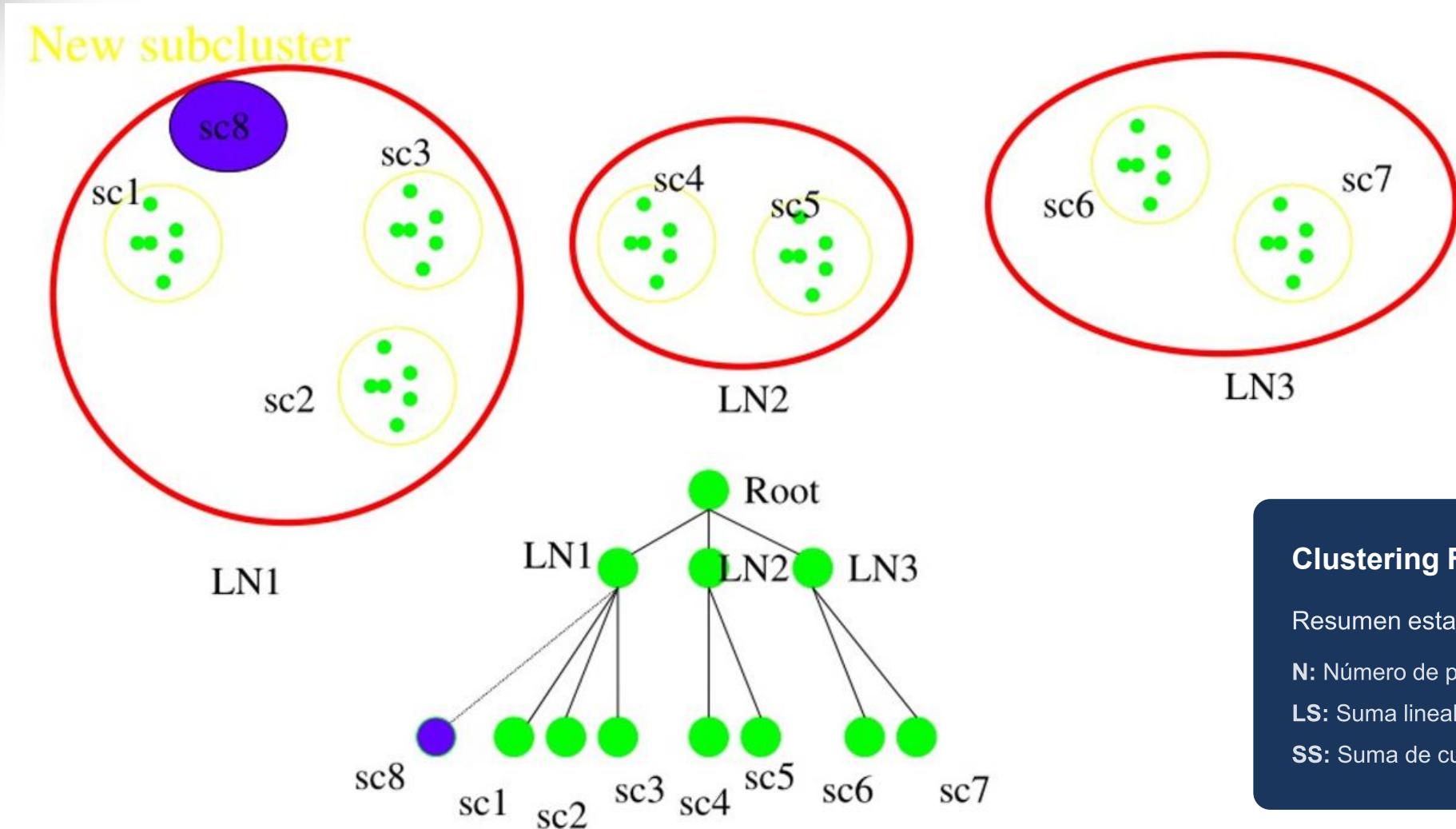
Muy escalable $O(n)$. Una pasada por datos. Bajo uso de memoria.

Limitaciones

Solo distancia euclíadiana. Sensible al orden de datos. Clusters esféricos.

`Birch(threshold=0.5, branching_factor=50, n_clusters=3)`

BIRCH



Clustering Feature (CF)

Resumen estadístico de un subcluster:

N: Número de puntos

LS: Suma lineal de puntos

SS: Suma de cuadrados

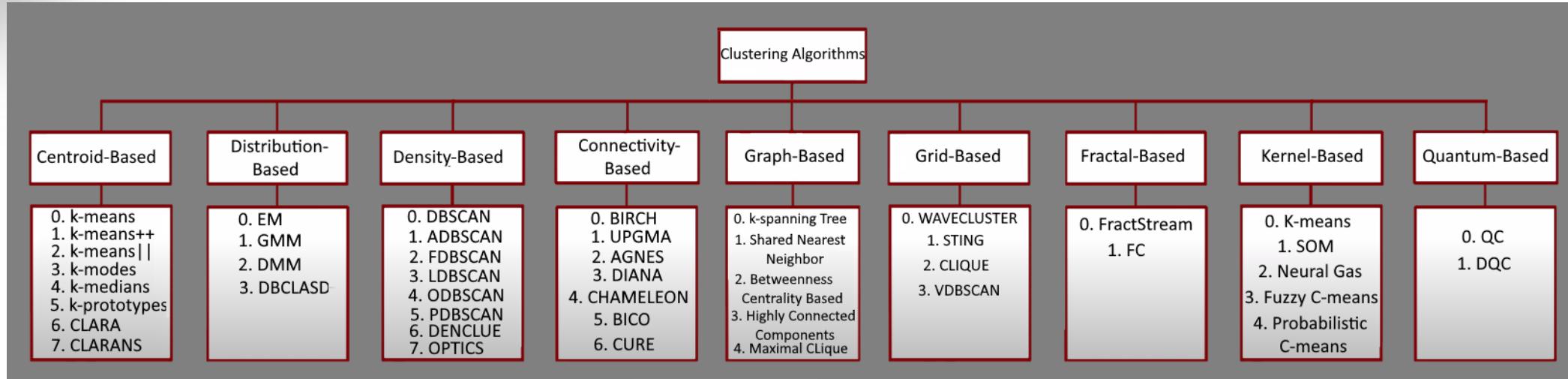
Comparación de algoritmos

Algoritmo	Requiere K	Forma clusters	Complejidad	Mejor uso
GMM	Sí (BIC/AIC)	Elíptica	$O(nK^2d)$	Soft clustering, datos gaussianos
Espectral	Sí	Arbitraria	$O(n^3)$	Clusters no convexos, n pequeño
Mean-Shift	No	Arbitraria	$O(n^2)$	K desconocido, n moderado
Affinity Prop.	No	Según similitud	$O(n^2)$	Ejemplares reales, n pequeño
BIRCH	Opcional	Esférica	$O(n)$	Datasets muy grandes

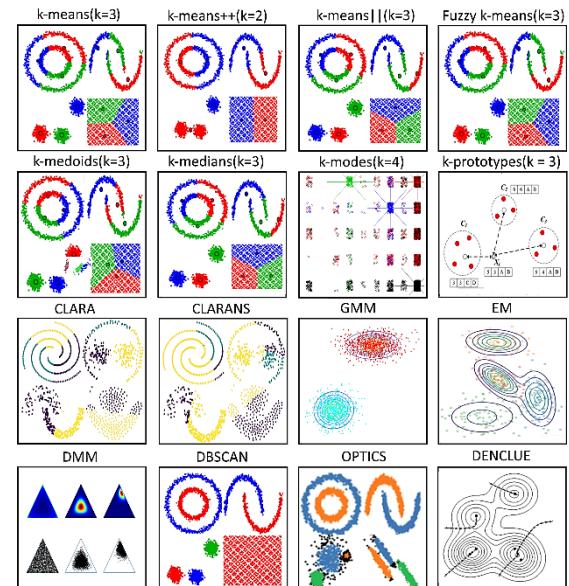
Recomendación general: GMM para clustering probabilístico con clusters elípticos. Espectral para formas complejas en datasets pequeños. BIRCH para grandes volúmenes de datos. Mean-Shift cuando K es desconocido.

Hay muchos tipos de clustering...

<https://towardsdatascience.com/17-clustering-algorithms-used-in-data-science-mining-49dbfa5bf69a>



Hay tantos tipos de clustering como medidas y técnicas para calcular distancias y asociaciones entre elementos y grupos

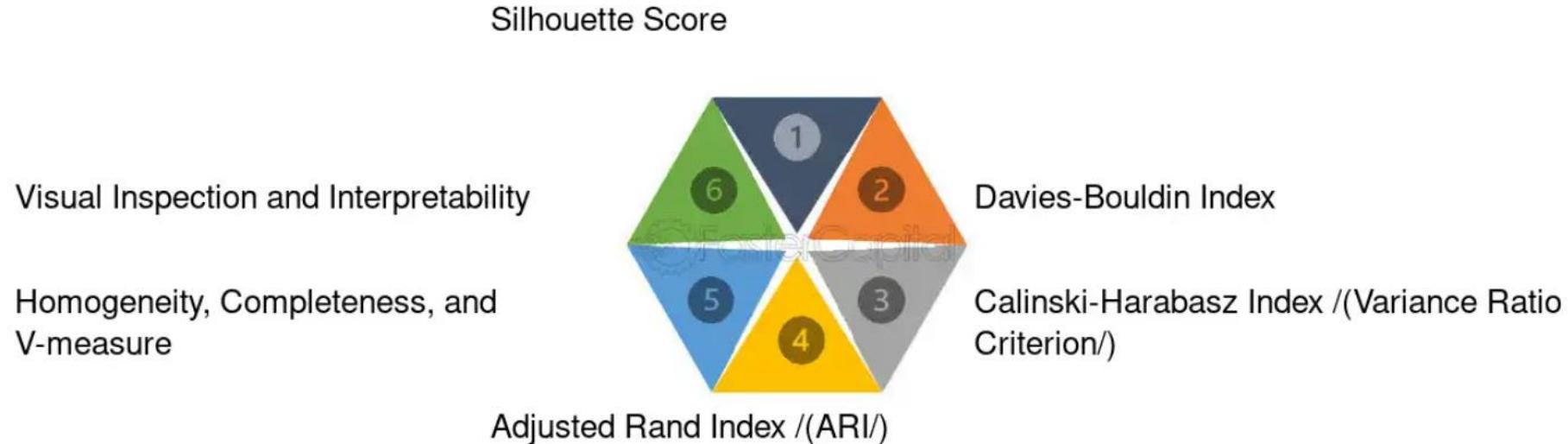


Hay muchos tipos de clustering...

<https://scikit-learn.org/stable/modules/clustering.html>

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large <code>n_samples</code> , medium <code>n_clusters</code> with <code>MiniBatch</code> code	General-purpose, even cluster size, flat geometry, not too many clusters, inductive	Distances between points
Affinity propagation	damping, sample preference	Not scalable with <code>n_samples</code>	Many clusters, uneven cluster size, non-flat geometry, inductive	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with <code>n_samples</code>	Many clusters, uneven cluster size, non-flat geometry, inductive	Distances between points
Spectral clustering	number of clusters	Medium <code>n_samples</code> , small <code>n_clusters</code>	Few clusters, even cluster size, non-flat geometry, transductive	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large <code>n_samples</code> and <code>n_clusters</code>	Many clusters, possibly connectivity constraints, transductive	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large <code>n_samples</code> and <code>n_clusters</code>	Many clusters, possibly connectivity constraints, non Euclidean distances, transductive	Any pairwise distance
DBSCAN	neighborhood size	Very large <code>n_samples</code> , medium <code>n_clusters</code>	Non-flat geometry, uneven cluster sizes, outlier removal, transductive	Distances between nearest points
OPTICS	minimum cluster membership	Very large <code>n_samples</code> , large <code>n_clusters</code>	Non-flat geometry, uneven cluster sizes, variable cluster density, outlier removal, transductive	Distances between points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation, inductive	Mahalanobis distances to centers
BIRCH	branching factor, threshold, optional global clusterer.	Large <code>n_clusters</code> and <code>n_samples</code>	Large dataset, outlier removal, data reduction, inductive	Euclidean distance between points

3. Métricas para clustering



¿Por qué evaluar clusters?

El problema fundamental

En clustering no supervisado, no existe una "respuesta correcta" predefinida.

Cualquier algoritmo producirá clusters, pero

¿son significativos?

¿Son útiles para el problema?



Preguntas clave

- ¿Los clusters están bien separados?
- ¿Son compactos internamente?
- ¿El número K es apropiado?
- ¿Los resultados son estables?

ATENCIÓN

Las métricas nos dan valores numéricos que permiten comparar y hacernos una idea de lo buenas que son las agrupaciones, pero **NO son una verdad absoluta**: La utilidad de los grupos y los modelos obtenidos es siempre **relativa**

Métricas para clustering

¿Como podemos evaluar el resultado de agrupamientos si no hay etiquetas para contrastar?

Métricas internas

Evalúan la calidad usando solo los datos y las asignaciones. No requieren etiquetas verdaderas. Útiles en escenarios reales:

- índice Davies-Bouldin
- índice de Dunn
- coeficiente de silueta
- etc.

Métricas externas

Comparan con etiquetas verdaderas (ground truth). Útiles para validación y benchmarking de algoritmos

- Adjusted Rank Index
- NMI
- Fowlkes-Mallows
- etc.



Inspección visual

imprescindible y muy informativa

- aspecto de los clústeres
- gráfico de silueta

Cohesión y separación

Cohesión (Compactness)

Mide cuán cercanos están los puntos dentro de un mismo cluster. Un cluster cohesivo tiene puntos muy similares entre sí.

$$\text{Cohesión} = \sum d(x, c_k)$$

Suma de distancias al centroide. Menor = mejor.

Separación (Separation)

Mide cuán distintos son los clusters entre sí. Clusters bien separados tienen centroides distantes.

$$\text{Separación} = \sum d(c_i, c_j)$$

Distancia entre centroides. Mayor = mejor.

Objetivo de la cohesión

Minimizar la variabilidad intra-cluster. Puntos del mismo cluster deben ser similares.

Objetivo de la separación

Maximizar la diferencia inter-cluster. Clusters diferentes deben ser distintos.

Principio fundamental

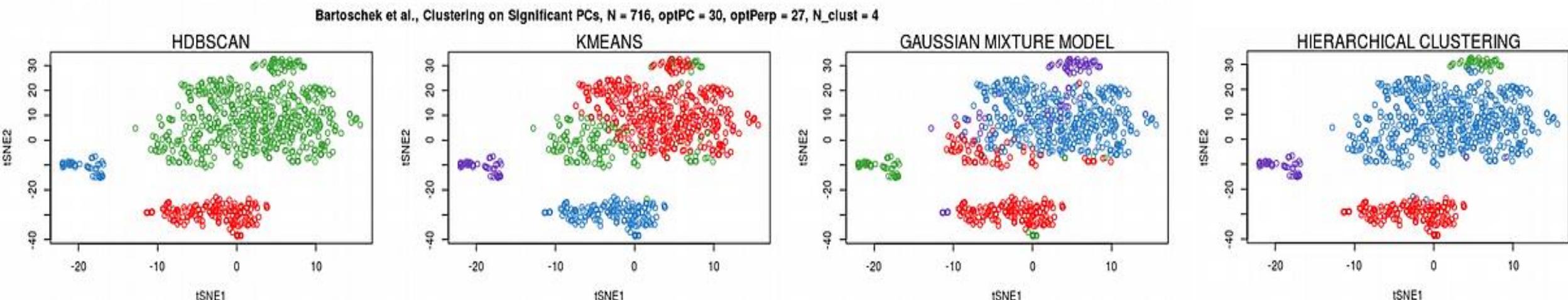
Un buen clustering maximiza la separación entre clusters y minimiza la dispersión dentro de cada cluster.

Métricas para clustering

Inspección visual: imprescindible y muy informativa

Aspecto de los clusters

- Se representan los grupos obtenidos sobre diferentes variables de los datos estudiados
- Se busca coherencia en las agrupaciones



Fuente: <https://towardsdatascience.com/>

¿Qué clustering es mejor? No está definido...

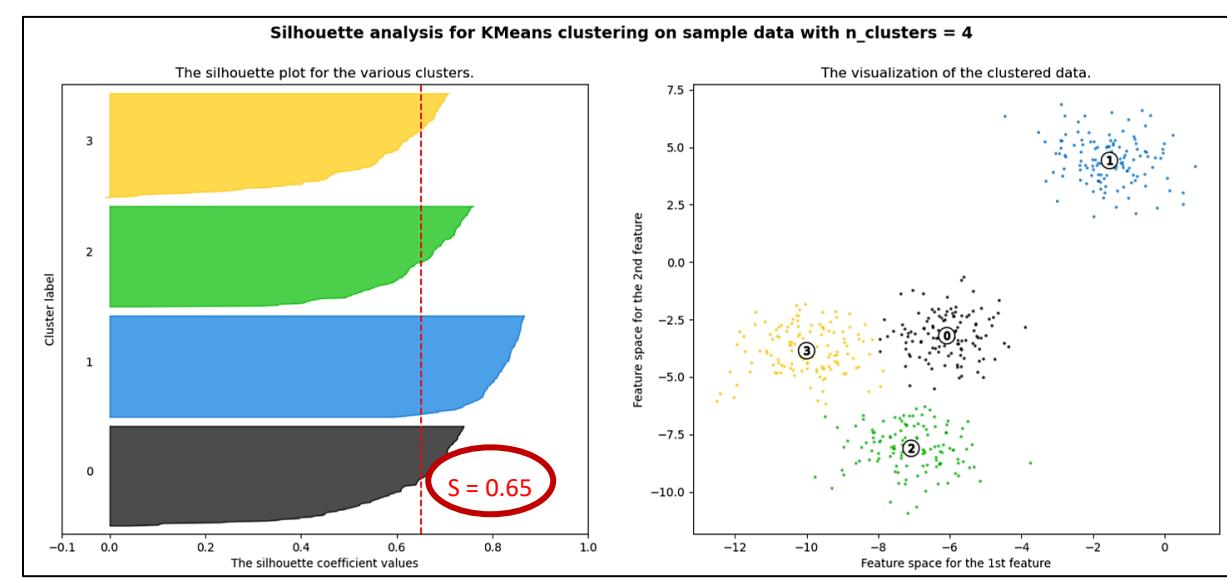
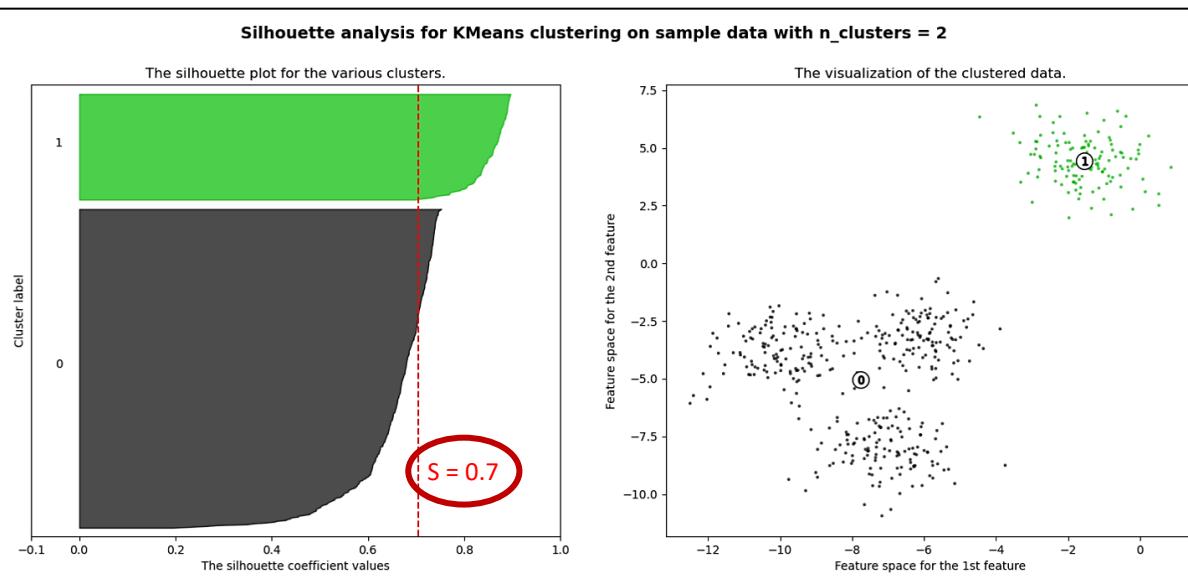
Métricas para clustering

Inspección visual: imprescindible y muy informativa

Análisis de silueta

- Se representa la distancia media entre un punto del cluster y todos los demás intracluster, dividida entre la distancia media entre ese punto y los puntos del cluster más cercano
- Puede ser entre 0 y 1. Cuanto mayor mejor separación entre clusters
- El **coeficiente de silueta (S)** es la media de los valores de todos los puntos analizados

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



Fuente: <https://towardsdatascience.com/>

Coeficiente de Silueta (Silhouette Score)

Definición para cada punto i

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

a(i): Distancia media a puntos del mismo cluster

b(i): Distancia media al cluster más cercano

Score global

Promedio de s(i) para todos los puntos. Rango: [-1, 1]. Valores más altos indican mejor clustering.

Guía de interpretación

0.71 - 1.00: Estructura fuerte

0.51 - 0.70: Estructura razonable

0.26 - 0.50: Estructura débil

< 0.25: Sin estructura clara

Interpretación del valor

s(i) ≈ 1: Punto bien asignado a su cluster

s(i) ≈ 0: Punto en frontera entre clusters

s(i) < 0: Posible asignación incorrecta

Ventaja: Permite visualizar silueta por cluster para identificar clusters problemáticos.

`silhouette_score(X, labels)`

Índice de Calinski-Harabasz (Variance Ratio)

Fórmula

$$CHI = \frac{\frac{B}{K-1}}{\frac{W}{n-k}}$$

B: Dispersión between-cluster (separación)

W: Dispersión within-cluster (cohesión)

K: Número de clusters, **n:** Número de puntos

Intuición

Ratio entre varianza inter-cluster y varianza intra-cluster, normalizado por grados de libertad. Similar a un estadístico F de ANOVA.

Interpretación

Rango: $[0, +\infty)$

Mayor valor = mejor clustering

Ventajas

Cálculo rápido $O(n)$. No tiene límite superior. Penaliza clusters poco densos.

Limitaciones

Asume clusters convexos.

Sensible a outliers. No comparable entre datasets.

`calinski_harabasz_score(X, labels)`

Índice de Davies-Bouldin

Fórmula

$$DB = \frac{1}{n} \sum_{i=1}^n \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

donde:

- n is the number of clusters, c_i is the **centroid** of cluster i
- σ_i is the average distance of all elements in cluster i to centroid c_i
- $d(c_i, c_j)$ is the distance between centroids c_i and c_j

Intuición

Promedio de la "similitud" máxima entre cada cluster y su vecino más parecido. Clusters compactos y separados tienen bajo DB.

Interpretación

Rango: $[0, +\infty)$

Menor valor = mejor clustering

(Opuesto a Silueta y Calinski-Harabasz)

Ventajas

Fácil de calcular. Interpretación intuitiva. Identifica clusters problemáticos.

Limitaciones

Usa solo centroides. Asume clusters convexos. Sensible a número de clusters.

`davies_bouldin_score(X, labels)`

Concepto similar: Índice de Dunn

- A partir de medidas intra-clusters y entre clusters
- Distancia entre clusters / distancia intra-cluster
- Cuanto más alto mejor

$$D = \frac{\min_{1 \leq i < j \leq n} d(i, j)}{\max_{1 \leq k \leq n} d'(k)}$$

$d(i, j)$ represents the distance between clusters i and j

$d'(k)$ measures the intra-cluster distance of cluster k .

Comparación de métricas internas

Métrica	Rango	Óptimo	Mejor para	Limitación principal
Silueta	$[-1, 1]$	Mayor	Diagnóstico por punto	Costoso para n grande
Calinski-Harabasz	$[0, +\infty)$	Mayor	Cálculo rápido	Asume convexidad
Davies-Bouldin	$[0, +\infty)$	Menor	Detectar clusters similares	Solo usa centroides

Uso para selección de K

Calcular métricas para $K=2,3,\dots,K_{max}$. Buscar el K que optimiza (máximo para Silueta y CH, mínimo para DB).

Recomendación práctica

Usar múltiples métricas. Si discrepan, priorizar la interpretabilidad del negocio y la visualización de los clusters.



Métricas externas: concepto

Definición

Métricas que comparan las etiquetas predichas con etiquetas verdaderas (ground truth). Miden qué tan bien el clustering recupera la estructura "real" de los datos.

Cuándo usarlas

Benchmarking de algoritmos en datasets conocidos
Validación contra clasificación experta
Investigación y comparación de métodos

Enfoque basado en pares

Evalúan si pares de puntos que deberían estar juntos/separados lo están en el clustering predicho.

Enfoque basado en información

Miden la información mutua entre las particiones. Cuánta información sobre las etiquetas reales proporciona el clustering.

Limitación: Requieren ground truth, que raramente está disponible en problemas reales de clustering.



Adjusted Rand Index (ARI)

Concepto

Mide la similitud entre dos particiones contando pares de puntos que están en el mismo/diferente cluster en ambas particiones.

Tabla de contingencia

- a:** Pares juntos en ambas particiones
- b:** Pares separados en ambas particiones
- c:** Juntos en real, separados en predicho
- d:** Separados en real, juntos en predicho

Rand Index original

$$RI(P, G) = \frac{a+d}{a+b+c+d}$$

Ajuste por azar

ARI corrige el RI para que asignaciones aleatorias tengan valor esperado 0, independientemente del número de clusters.

$$ARI = \frac{RI - E(RI)}{1 - E(RI)}$$

Interpretación

- ARI = 1:** Particiones idénticas
- ARI ≈ 0:** Asignación aleatoria
- ARI < 0:** Peor que aleatorio

```
adjusted_rand_score(labels_true, labels_pred)
```

Normalized Mutual Information (NMI)

Información Mutua (MI)

Mide cuánta información sobre una partición se obtiene al conocer la otra. Basada en teoría de la información.

$$MI(U,V) = \sum_{i,j} P(i,j) \log[P(i,j) / P(i)P(j)]$$

El NMI mide cuánta información compartida hay entre dos agrupaciones: la que produce tu algoritmo y la "verdadera" (ground truth).

Problema de MI

MI no está acotada y tiende a aumentar con el número de clusters, incluso para particiones aleatorias.

Solución : Normalized Mutual Information escala a 0-1

Normalización

$$NMI = MI(U,V) / \text{mean}(H(U), H(V))$$

$H(U)$, $H(V)$: Entropías de cada partición

Interpretación

Rango: [0, 1]

NMI = 1: Correspondencia perfecta

NMI = 0: Independencia total

Variante AMI: Adjusted Mutual Information corrige por azar.

```
normalized_mutual_info_score(labels_true, labels_pred)
```



Homogeneidad, Completitud y V-measure

Homogeneidad (h)

Un cluster es homogéneo si todos sus miembros pertenecen a la misma clase real.

$$h = 1 - H(C|K) / H(C)$$

$H(C|K)$: entropía de clases dado clusters

Completitud (c)

Un clustering es completo si todos los miembros de una clase están en el mismo cluster.

$$c = 1 - H(K|C) / H(K)$$

$H(K|C)$: entropía de clusters dado clases

V-measure

Media armónica de homogeneidad y completitud. Balanceo de ambos criterios.

$$V = 2 \cdot h \cdot c / (h + c)$$

Análogo a F1-score en clasificación

Rango e interpretación

Todas en $[0, 1]$. Valor 1 = perfecto. Permiten diagnosticar si el problema es de pureza (h baja) o fragmentación (c baja).

Fowlkes-Mallows Index (FMI)

Definición

Media geométrica de precisión y recall calculados sobre pares de puntos.

$$FMI = \sqrt{(PPV \times TPR)}$$

$$FM = \sqrt{\frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}}$$

Donde

PPV (Precision): $TP / (TP + FP)$

TPR (Recall): $TP / (TP + FN)$

TP: pares correctamente agrupados. FP: pares incorrectamente agrupados.
 FN: pares incorrectamente separados.

Interpretación

Rango: $[0, 1]$

FMI = 1: Clustering perfecto

FMI → 0: Clustering aleatorio

Ventajas

Interpretación clara. Balanceo de precision/recall. No sesgado por número de clusters.

Limitaciones

No ajustado por azar (aunque existe variante ajustada). Puede ser alto para clusterings triviales.

`fowlkes_mallows_score(labels_true, labels_pred)`

Estabilidad de clusters

Concepto

Un clustering estable produce resultados similares ante pequeñas perturbaciones en los datos. La inestabilidad sugiere que la estructura encontrada podría ser espuria.

Método Bootstrap

1. Generar B muestras bootstrap del dataset
2. Aplicar clustering a cada muestra
3. Comparar particiones resultantes (ARI)
4. Alta similitud = alta estabilidad

Subsampling

Alternativa al bootstrap: usar submuestras aleatorias (ej. 80% de los datos) y evaluar consistencia de asignaciones en la intersección.

Uso para selección de K

Calcular estabilidad para $K=2,3,\dots,K_{\max}$. El K óptimo suele tener máxima estabilidad (estructura robusta).

Ventaja: No requiere ground truth. Complementa métricas internas para validar que el clustering es robusto.



Comparación de métricas externas

Métrica	Rango	Ajustada	Enfoque	Uso recomendado
ARI	[-1, 1]	Sí	Pares de puntos	Benchmarking general
NMI	[0, 1]	No (AMI sí)	Información mutua	Comparar particiones
V-measure	[0, 1]	No	Entropía condicional	Diagnóstico h/c
FMI	[0, 1]	No	Precision/Recall	Balance precision/recall

Métricas ajustadas

ARI y AMI corrigen por : valor esperado 0 para asignaciones aleatorias.
 Preferibles para comparaciones justas.

Recomendación

Usar ARI como métrica principal. Complementar con V-measure para diagnóstico. NMI para comparar diferentes números de clusters.



Evaluación en clustering

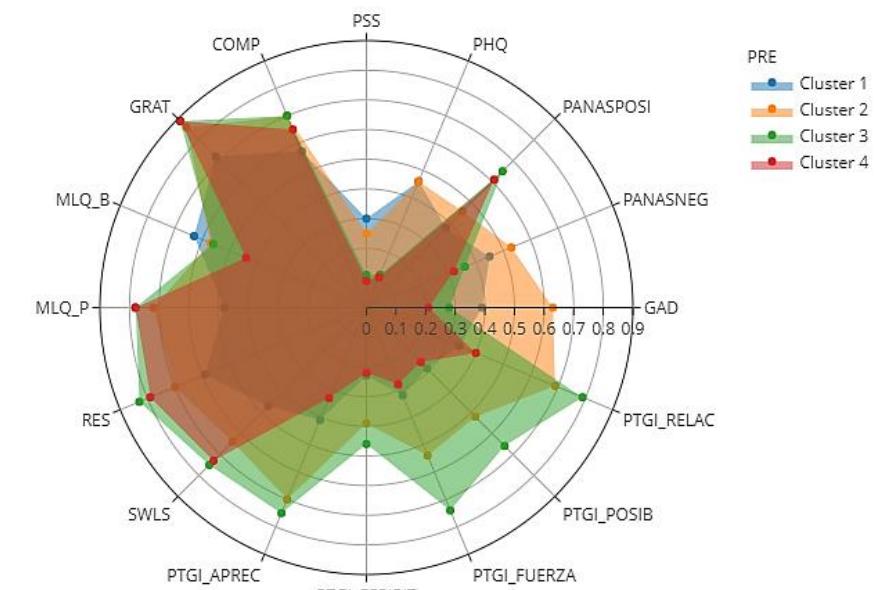
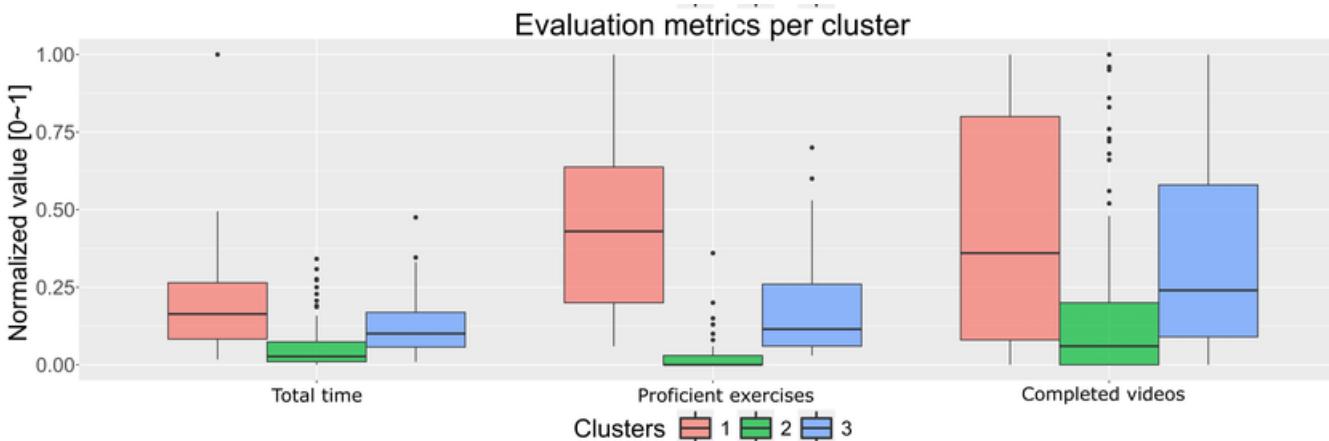
Tenemos medidas y gráficas de los clusters...

Elegimos el que creemos mejor...

¿Cómo sacamos valor a nuestro clustering?

Estadísticas y gráficos para cada cluster

- Medias, dispersión
- Distribuciones, boxplot
- Gráficos de radar multivariable



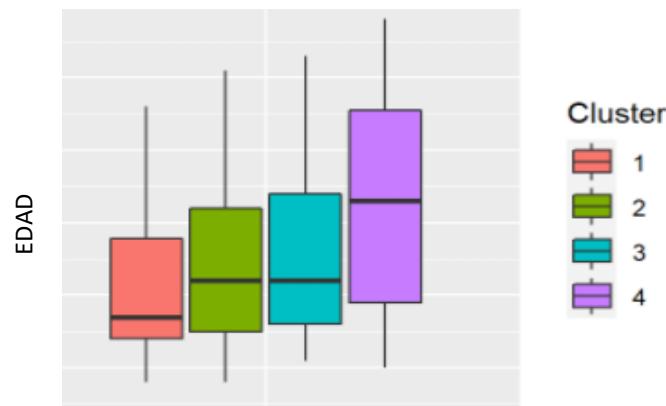
Evaluación en clustering

Tenemos medidas y gráficas de los clusters...

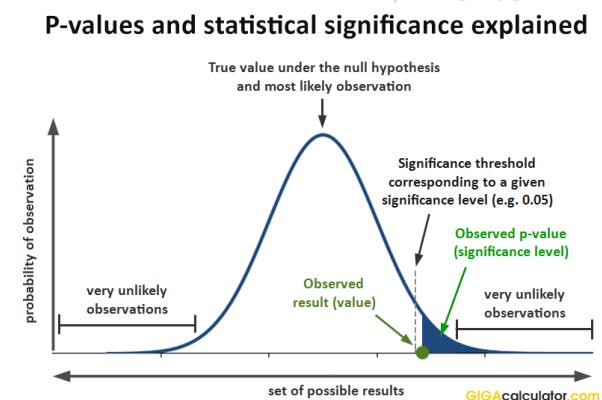
Elegimos el que creemos mejor...

¿Cómo sacamos valor a nuestro clustering?

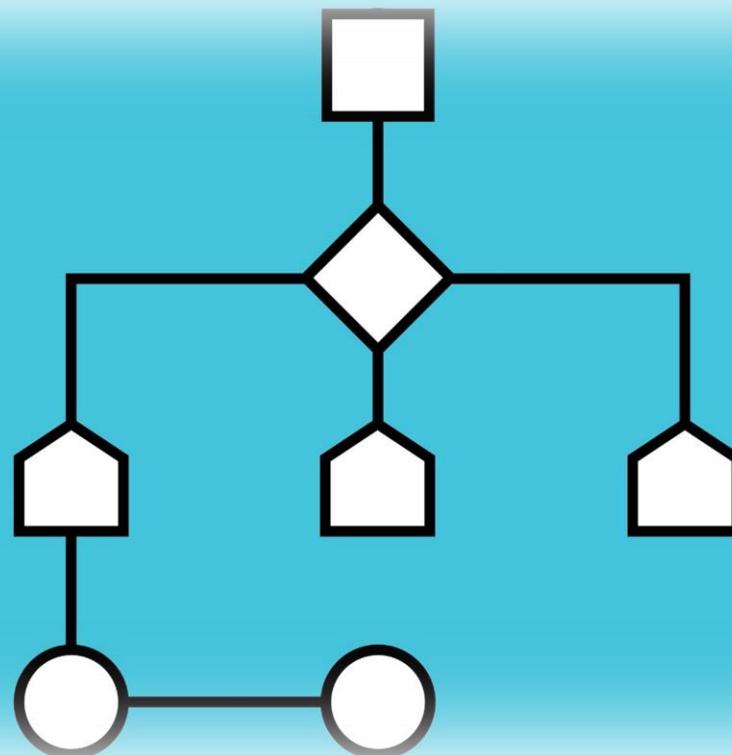
- Pruebas estadísticas de **independencia** entre clusters
- Confrontamos variables de diferentes clusters
- Contraste de hipótesis: ¿sus medias son iguales?
¿Misma varianza?
- ¿Podemos decir que son estadísticamente diferentes con un nivel de significación?



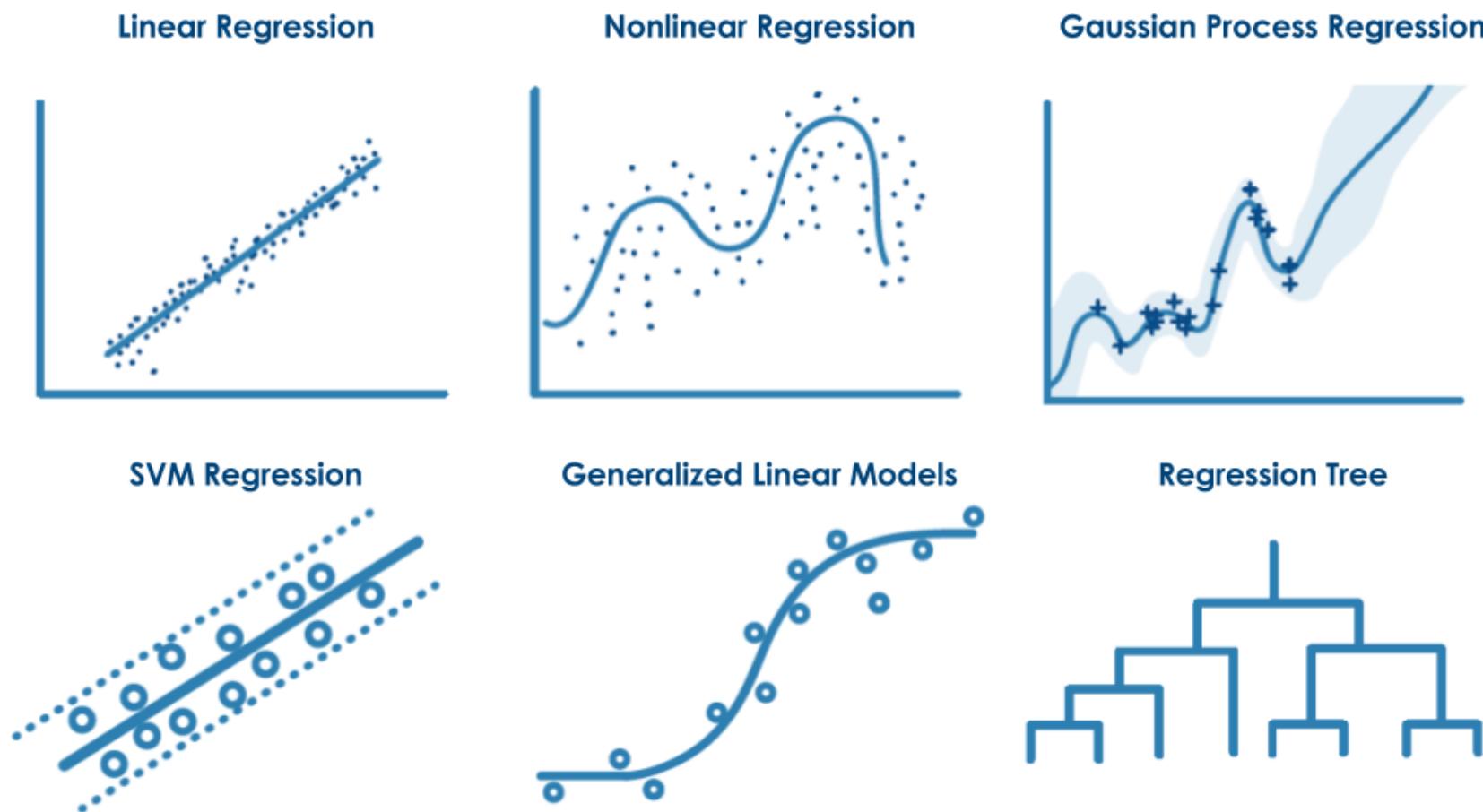
##	Variable	Mes	Chi2	P_valor
## [1,]	"PHQ"	"PRE"	"215.258927251455"	"2.12589678357261e-46"
## [2,]	"PHQ"	"SEM4"	"94.7701729924005"	"2.06874277592757e-20"
## [3,]	"PHQ"	"SEM8"	"89.8917530641123"	"2.31103742705943e-19"
## [4,]	"PHQ"	"POST"	"55.9661817692241"	"4.2713273472927e-12"
## [5,]	"GAD"	"PRE"	"152.830476433222"	"6.45866933385558e-33"
## [6,]	"GAD"	"SEM4"	"52.9552141104039"	"1.87411547332337e-11"
## [7,]	"GAD"	"SEM8"	"42.1578430226086"	"3.71430702074901e-09"
## [8,]	"GAD"	"POST"	"52.6702227404608"	"2.15551817999425e-11"
## [9,]	"PANASPOSI"	"PRE"	"215.754562824074"	"1.66115866134124e-46"
## [10,]	"PANASPOSI"	"SEM4"	"94.8257704920202"	"2.01260431208547e-20"
## [11,]	"PANASPOSI"	"SEM8"	"77.10963644102"	"1.27901628143038e-16"
## [12,]	"PANASPOSI"	"POST"	"75.5817669943294"	"2.71902070675866e-16"
## [13,]	"PANASNEG"	"PRE"	"178.89331727571"	"1.52916548516757e-38"
## [14,]	"PANASNEG"	"SEM4"	"65.9505267070525"	"3.14062301965757e-14"
## [15,]	"PANASNEG"	"SEM8"	"51.1566227756189"	"4.53027977591731e-11"
## [16,]	"PANASNEG"	"POST"	"51.6897907698203"	"3.48752776620395e-11"



Resumen de algoritmos

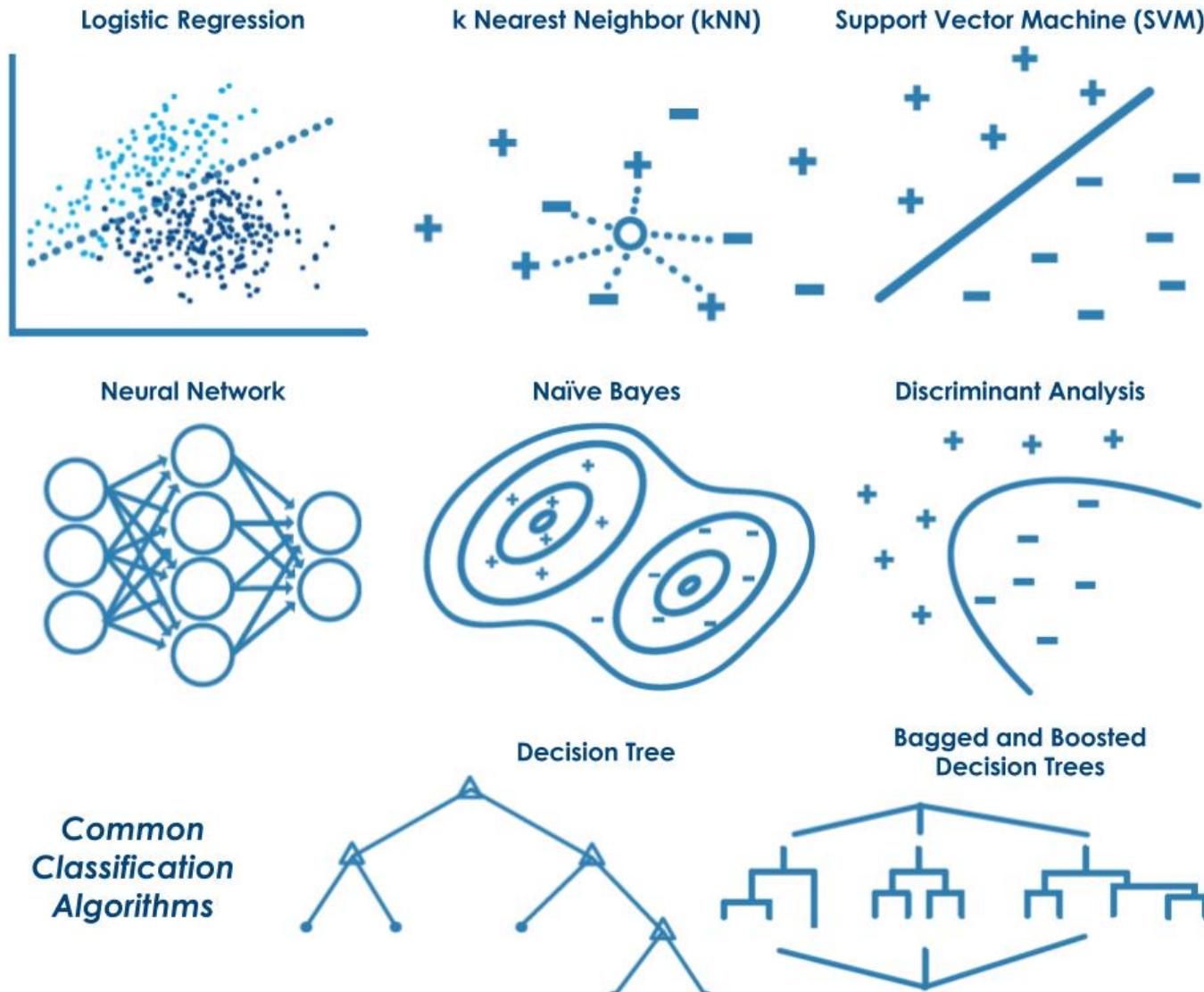


Técnicas para regresión: resumen

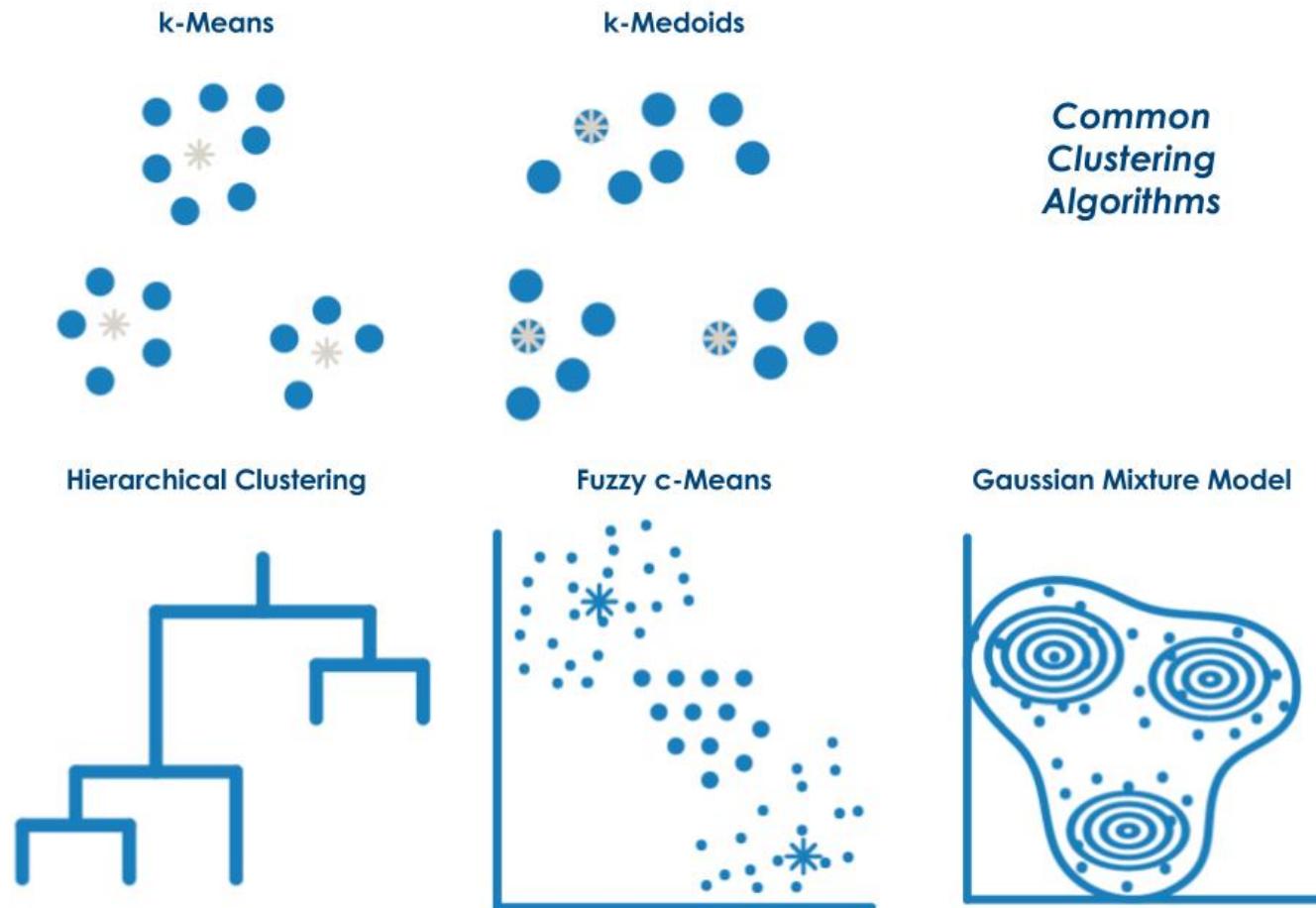


Common Regression Algorithms

Técnicas para clasificación: resumen



Técnicas para clustering: resumen



FIN

Clústering

9-10 enero, 2026

Juan José Garcés Iniesta
jjgarcesiniesta@gmail.com