FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Para estudiantes de Educación Básica y Media.
UNIVERSIDAD DE CHILE

Clase V-II: *Clustering* en *Machine Learning*

FM849 Proyecto de Ciencia de Datos: Inteligencia Artificial (IA) y sus aplicaciones

Máximo Flores Valenzuela (`mflores@dcc.uchile.cl`)

Universidad de Chile • 23 de agosto de 2025

Roadmap de la clase

1. Breve recuerdo de *clustering*.
2. Modelos de *clustering*: *K*-means, DBSCAN y *clustering* jerárquico.

Breve recuerdo de *clustering*

Clustering

El *clustering* o agrupamiento es una técnica de aprendizaje **no supervisado**. La diferencia con el aprendizaje supervisado es que ahora no necesitamos datos etiquetados con una clase.

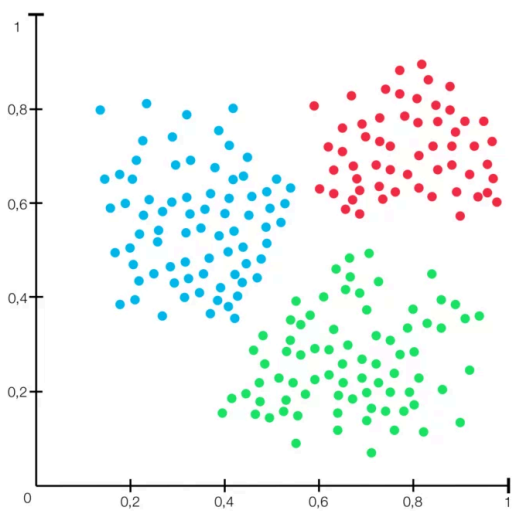

Figura 3: Ejemplo de *clustering* con 3 grupos.

Problema de ambigüedad

¿Cuál es la mejor manera de agrupar la información?

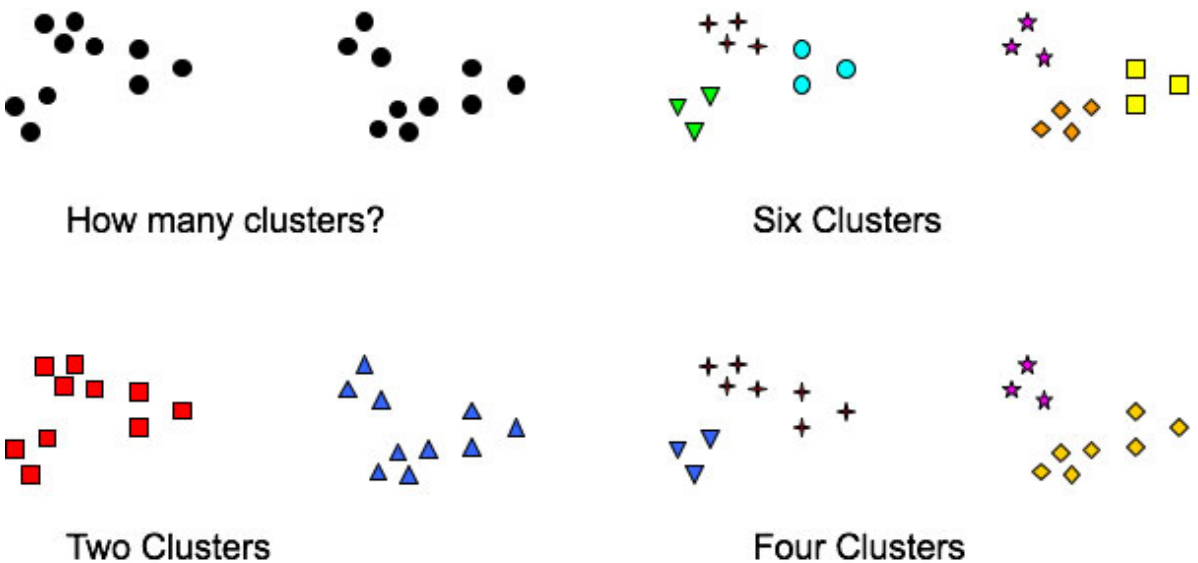

Figura 4: Ambigüedad en la generación de *clusters*.

Idea central

Al realizar agrupamiento o *clustering*, la idea es que la distancia:

- «intracluster», es decir, entre puntos del mismo grupo, sea mínima.
- «intercluster», es decir, entre puntos de grupos distintos, sea máxima.

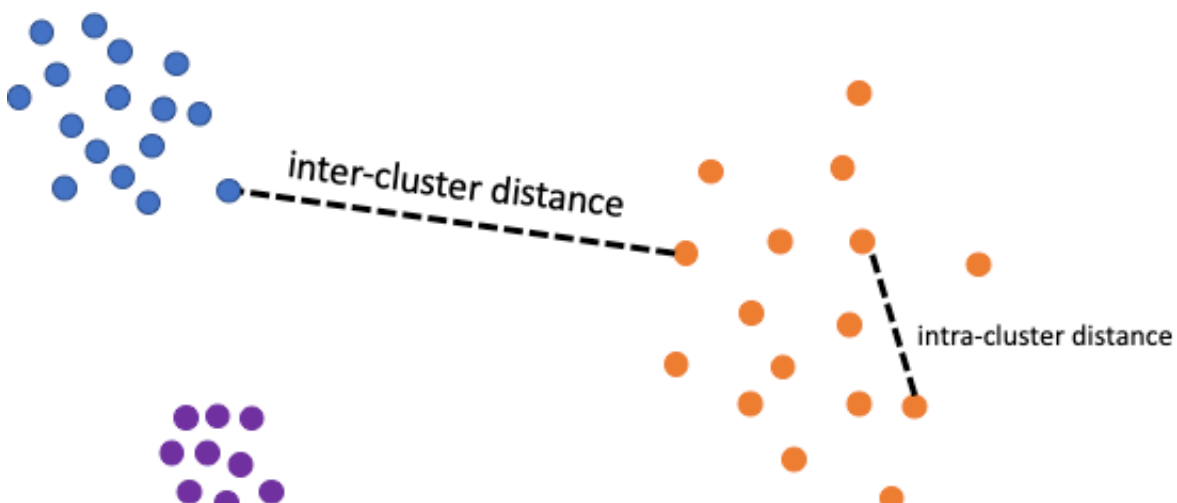

Figura 5: Distancias intracluster e intercluster.

Modelos de *clustering*

K-Means

La idea central es separar los datos en K grupos usando centroides (promedio de posiciones).

1. Se eligen K puntos aleatorios de los datos $X_{(1)}, ..., X_{(K)}$ como los centroides iniciales.
2. Cada punto se asigna al centroide más cercano usando una métrica de disimilitud (usualmente, distancia euclidiana).
3. Una vez se hayan asignado todos los puntos, se recalcula el centroide μ_k, $k \in [K]$ de cada *cluster* como el centro de masa del grupo.
4. Se repite desde el paso 2 hasta que la métrica SSE deje de disminuir de manera significativa:

$$\text{SSE}(\mathcal{C}_1, ..., \mathcal{C}_k) = \sum_{k=1}^{K} \left(\underbrace{\sum_{i \in \mathcal{C}_k} \underbrace{\|X_i - \mu_k\|^2}_{\text{distancia euclidiana}}}_{\text{distancia total intracluster}} \right)$$

Problemas de K-Means

El muestreo aleatorio $X_{(1)}, ..., X_{(K)}$ que se realiza al inicio depende del azar.

Esto genera que distintas ejecuciones del algoritmo entreguen distintos resultados, que en algunos casos, generan diferencias significativas.

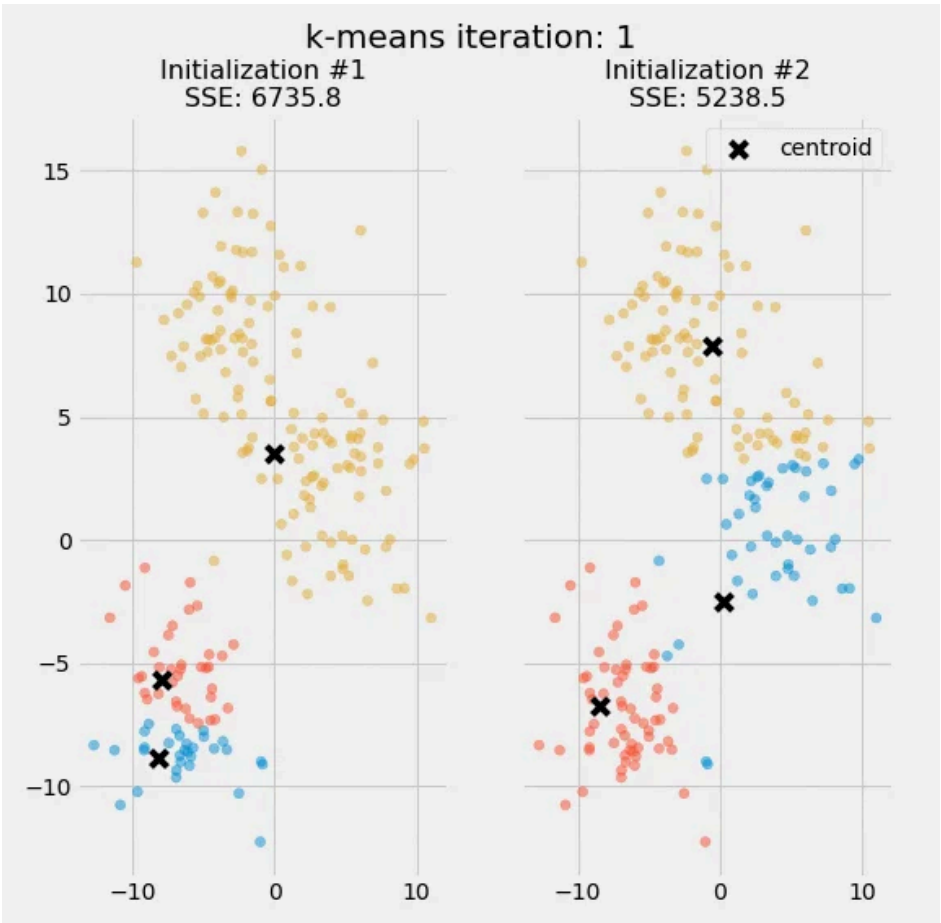

Figura 6: Diferentes inicializaciones de *K*-means.

Número de *clusters*

El número de *clusters* es un hiperparámetro del modelo *K*-means. Una técnica para encontrar el valor óptimo de *K* se llama «método del codo», y consiste en mirar el punto donde las diferencias en la métrica SSE ya no son significativas.

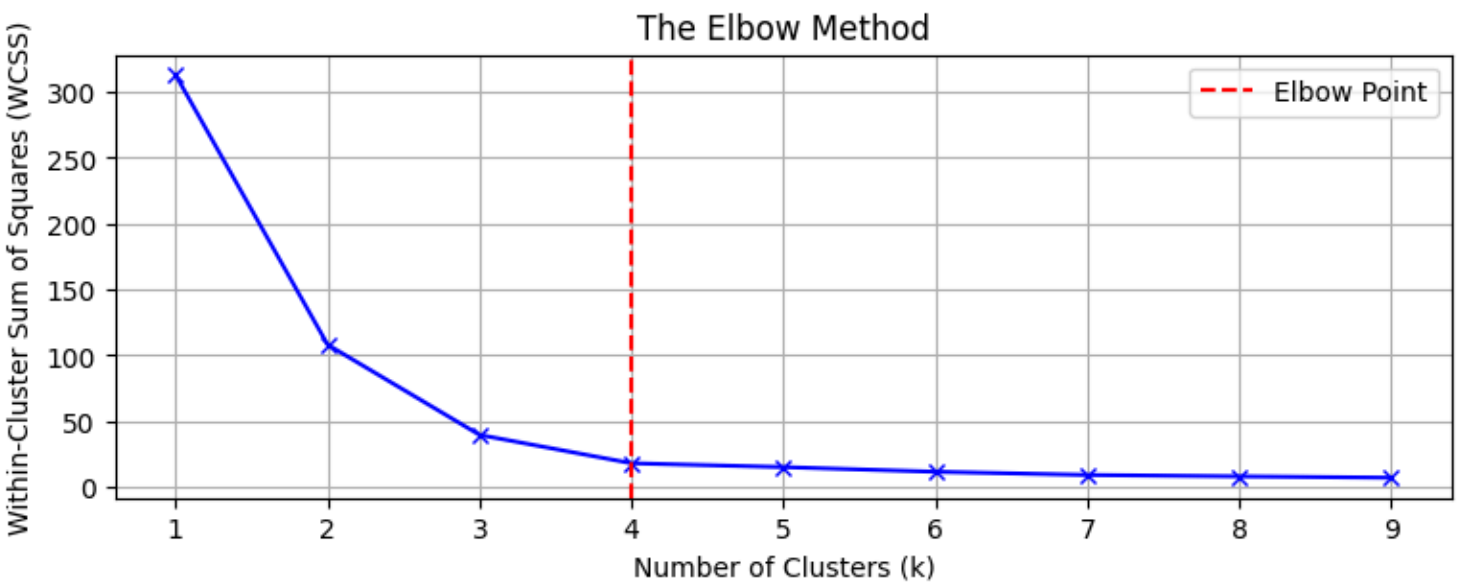

Figura 7: Método del codo para la elección del valor de *K*.

DBSCAN

Corresponde a un método para encontrar *clusters* buscando regiones con alta densidad de puntos, y marcando los puntos dispersos como «ruido».

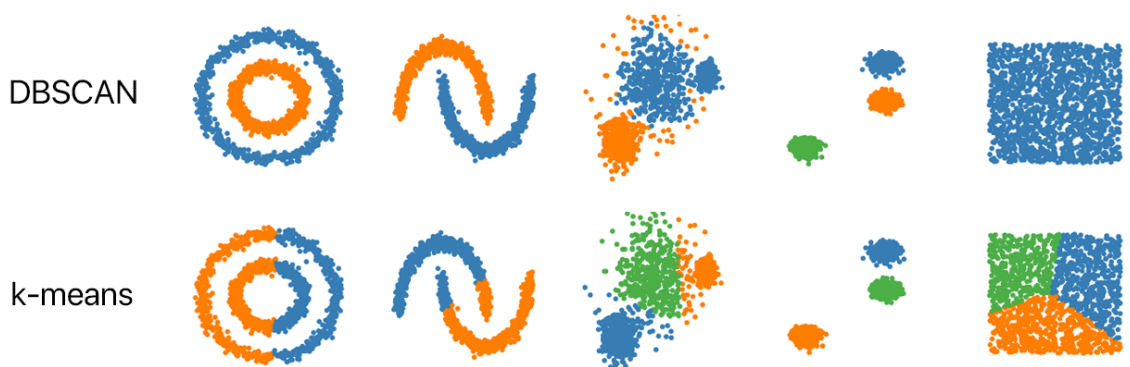

Figura 8: DBSCAN vs. *K*-means (visto anteriormente).

Ideas clave

Para este método, existen 3 conceptos clave:

- **Puntos fundamentales (núcleos)**: tienen al menos un número de puntos M a una distancia ε.
- **Puntos de frontera**: están a una distancia ε de un punto fundamental, pero no tienen al menos M vecinos a dicha distancia.
- **Puntos de ruido**: no cumplen la caracterización ni del primer punto ni del segundo.

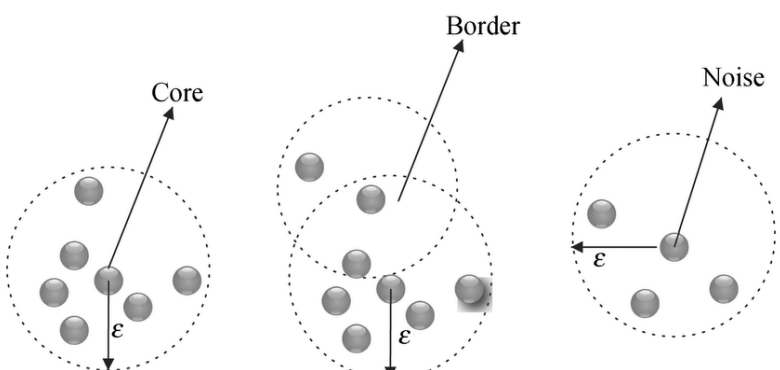

Figura 9: Separación de los 3 conceptos clave descritos arriba.

Configuración del modelo (`MinPts` y ε)

- Para obtener distintas configuraciones, se puede adaptar la distancia ε y la mínima cantidad de puntos `MinPts` (M).
- Para datos de d dimensiones, se recomienda que $\texttt{MinPts} \geq d + 1$. Esta elección depende de la densidad espacial de los datos: si hay mucho ruido o muchos datos, es mejor usar un valor de `MinPts` más alto. Si hay pocos datos, es mejor ajustarse a la cota.
- Para elegir ε, se puede realizar como sigue:
  1. Para cada punto, se calcula la distancia a su k-ésimo vecino más cercano ($k = \texttt{MinPts}$).
  2. Se ordenan las distancias obtenidas de menor a mayor.
  3. Se grafican las distancias, y se usa el método del codo visto anteriormente.

Elección de ε: gráfico *k-dist*

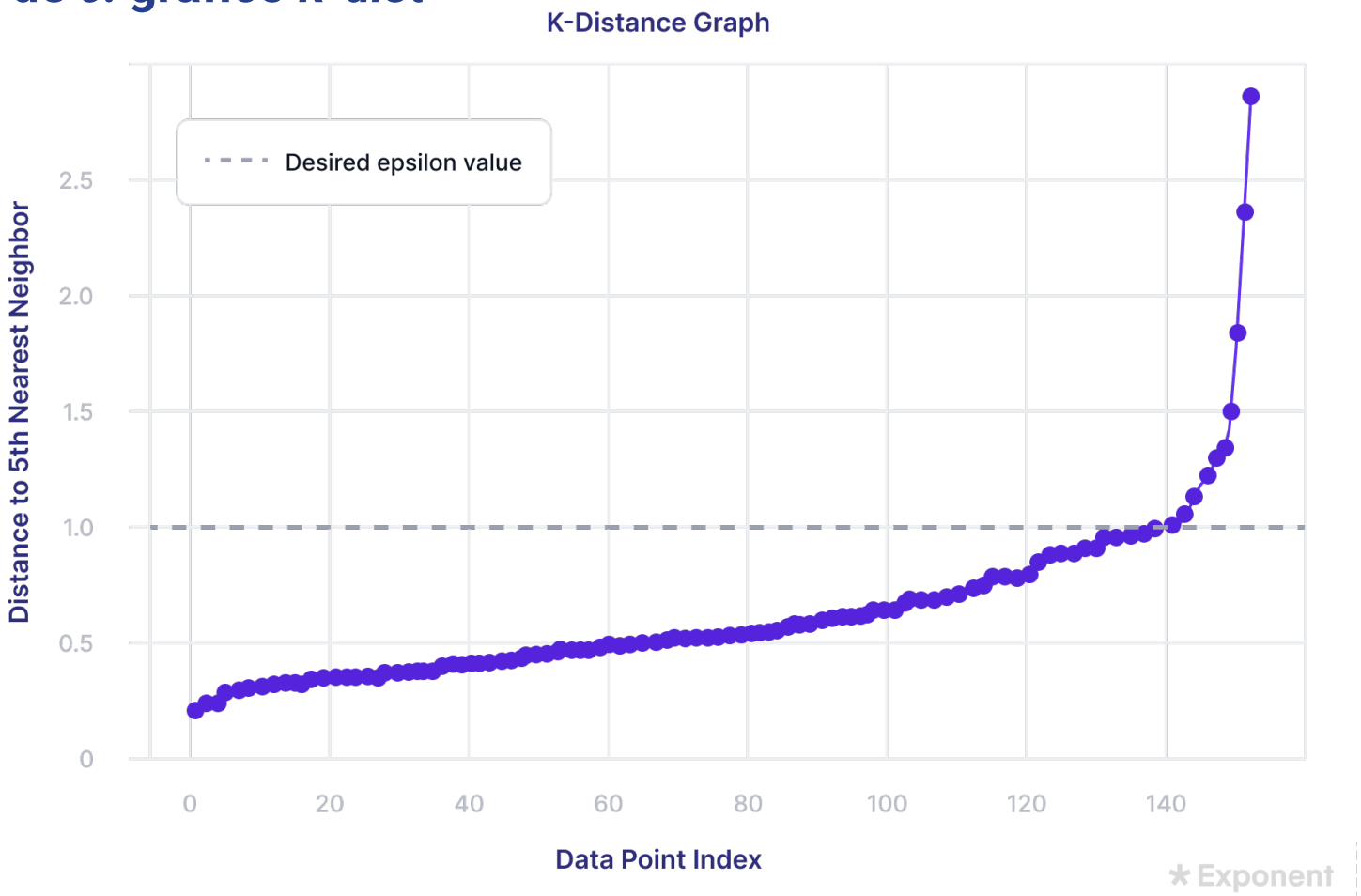

Figura 10: Método del codo para ver dónde el crecimiento empieza a ser notorio.

Clustering jerárquico

El *clustering* jerárquico consiste en una familia de métodos que construyen una estructura en forma de árbol (dendrograma).

Se divide en dos clases: jerárquico aglomerativo y jerárquico divisivo. También existen métodos híbridos.

En este curso, veremos los siguientes:

- Estrategia de *clustering* jerárquico aglomerativo (HAC).
- *Bisecting K-Means* (*clustering* jerárquico divisivo).
- HDBSCAN (híbrido).

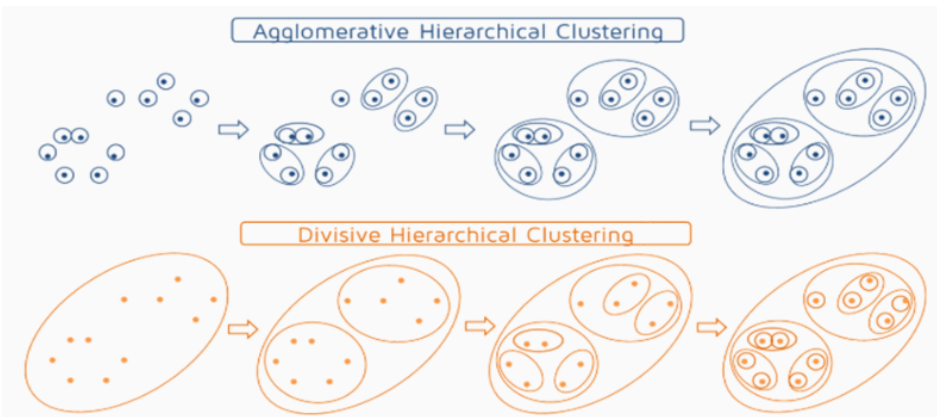

Figura 11: Los dos tipos de *clustering* jerárquico.

Clustering jerárquico aglomerativo

La idea se basa en el siguiente algoritmo:

1. Cada punto empieza como un propio *cluster* (hojas del árbol).
2. Se unen iterativamente los *clusters* más parecidos según un criterio hasta formar uno sólo (formando nodos internos, hasta la raíz del árbol).

Para el punto 2, existen varias variantes que miden la similitud de distintas formas.

Cuantificación de la disimilitud de *clusters*

Una pregunta natural es: ¿cómo cuantificamos la distancia entre dos *clusters* distintos? Para esto, se definen los siguientes enfoques:

- *Simple linkage*: menor distancia entre pares de puntos.
- *Complete linkage*: mayor distancia entre pares de puntos.
- *Average linkage*: promedio de distancias entre pares de puntos.
- *Centroid linkage*: distancia entre los centroides.
- Método de Ward: une los *clusters* que minimizan el aumento de SSE.

Cada uno de estos incide en el paso 2 descrito anteriormente de una forma distinta.

¿Cuál es la gracia entonces?

La raíz tiene el *cluster* completo. Si elegimos este *cluster*, entonces no habremos separado nada. La idea es realizar un corte en el árbol para obtener un punto de la división donde habían *clusters* descriptivos.

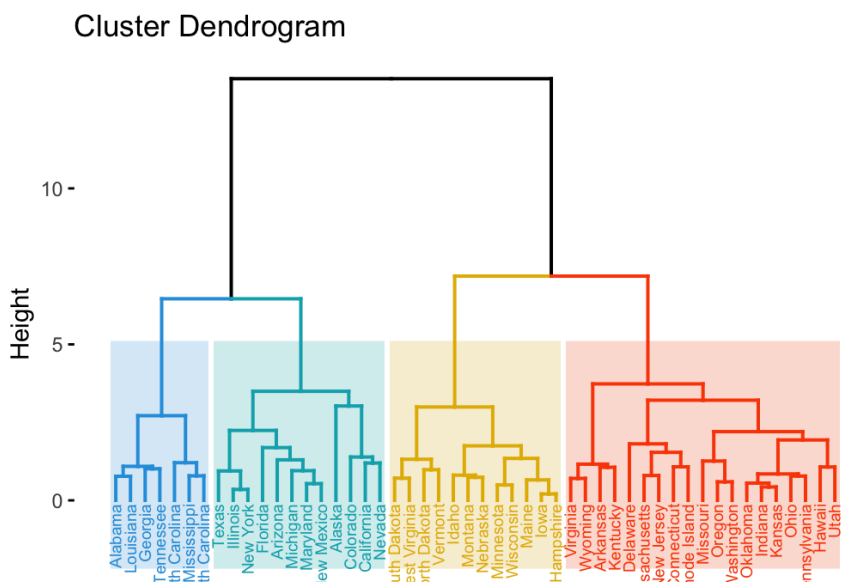

Figura 12: Dendrograma con estados de EE. UU.

Elección del corte

En el gráfico anterior, la altura indicada en el eje Y corresponde a la medida de disimilitud elegida para comparar dos *clusters*. Si dos *clusters* se fusionan a una altura h, entonces la disimilitud entre ellos era h en ese paso.

- La elección del corte depende de la disimilitud máxima que queramos admitir en la última fusión.
- Si se conoce de antemano la cantidad de *clusters* K que se quieren formar, se puede buscar alguna altura h tal que en la última fusión se hayan formado exactamente K *clusters* (o algo cercano a K).
- Se puede usar el método del codo también: buscamos saltos grandes en altura y cortamos antes de que se fusionen.

Bisecting K-Means

Como vimos anteriormente, es un tipo de *clustering* jerárquico divisivo.

1. Se parte con todos los puntos dentro del mismo *cluster*.

Hasta que no se tiene la cantidad deseada de *clusters* K:

2. Se realiza una bisección, es decir, se aplica K-Means con $K = 2$ al *cluster* de mayor SSE.
    - Por defecto, al inicio se realiza la bisección sobre todo el *cluster*.

Noten que la cantidad de *clusters* K es alcanzable para cualquier $K \in \mathbb{N}$, $K \geq 1$, dado que en cada paso se obtiene sólo 1 nuevo *cluster*.

HDBSCAN

A pesar de que DBSCAN tiene ventajas, como detectar formas arbitrarias y no sólo esferas, o identificar *outliers*, es muy sensible a los parámetros ε y `MinPts`.

Más aún, si la densidad de los datos no es homogénea, un sólo valor de ε no es apropiado.

Una extensión jerárquica híbrida de DBSCAN que explora esto es HDBSCAN.

Si les interesa experimentar, pueden echarle un vistazo a la documentación.

¿Preguntas?