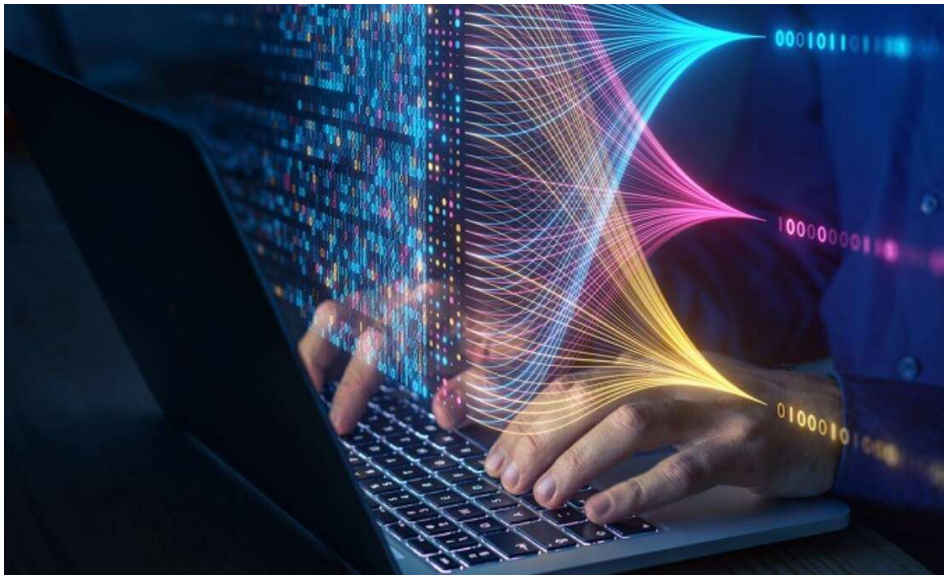
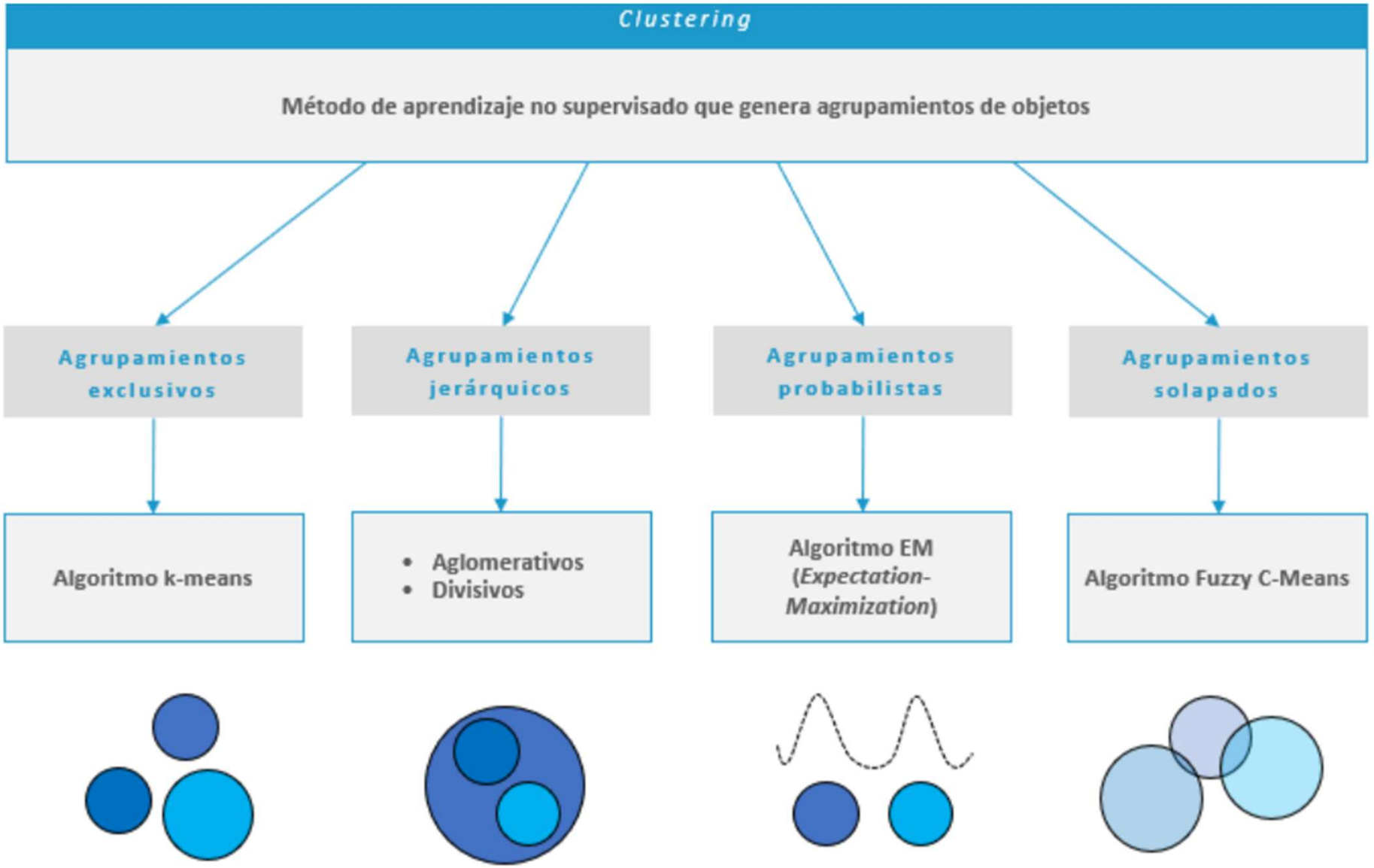
Técnicas de inteligencia artificial

Adriana Cervantes Castillo



SEMANA 10: CLUSTERING


Clustering
Método de aprendizaje no supervisado que genera agrupamientos de objetos
Agrupamientos exclusivos
Agrupamientos jerárquicos
Agrupamientos probabilistas
Agrupamientos solapados
Algoritmo k-means
Aglomerativos
Divisivos
Algoritmo EM (Expectation-Maximization)
Algoritmo Fuzzy C-Means

Clustering

Es un método de aprendizaje no supervisado que permite agrupar objetos en clústeres o agrupamientos, cuyos miembros son similares entre sí en cierto modo.

La clase es desconocida y es, el descubrimiento de esta clase el objetivo de este aprendizaje, a través de la agrupación de instancias en base a un esquema de similitud.

- El resultado de aplicar una técnica de *clustering* es una serie de **agrupamientos o clústeres**, los cuales son particiones de un conjunto de instancias

Clúster

Es una colección de objetos similares entre sí y diferentes a los objetos que pertenecen a otros clústeres.

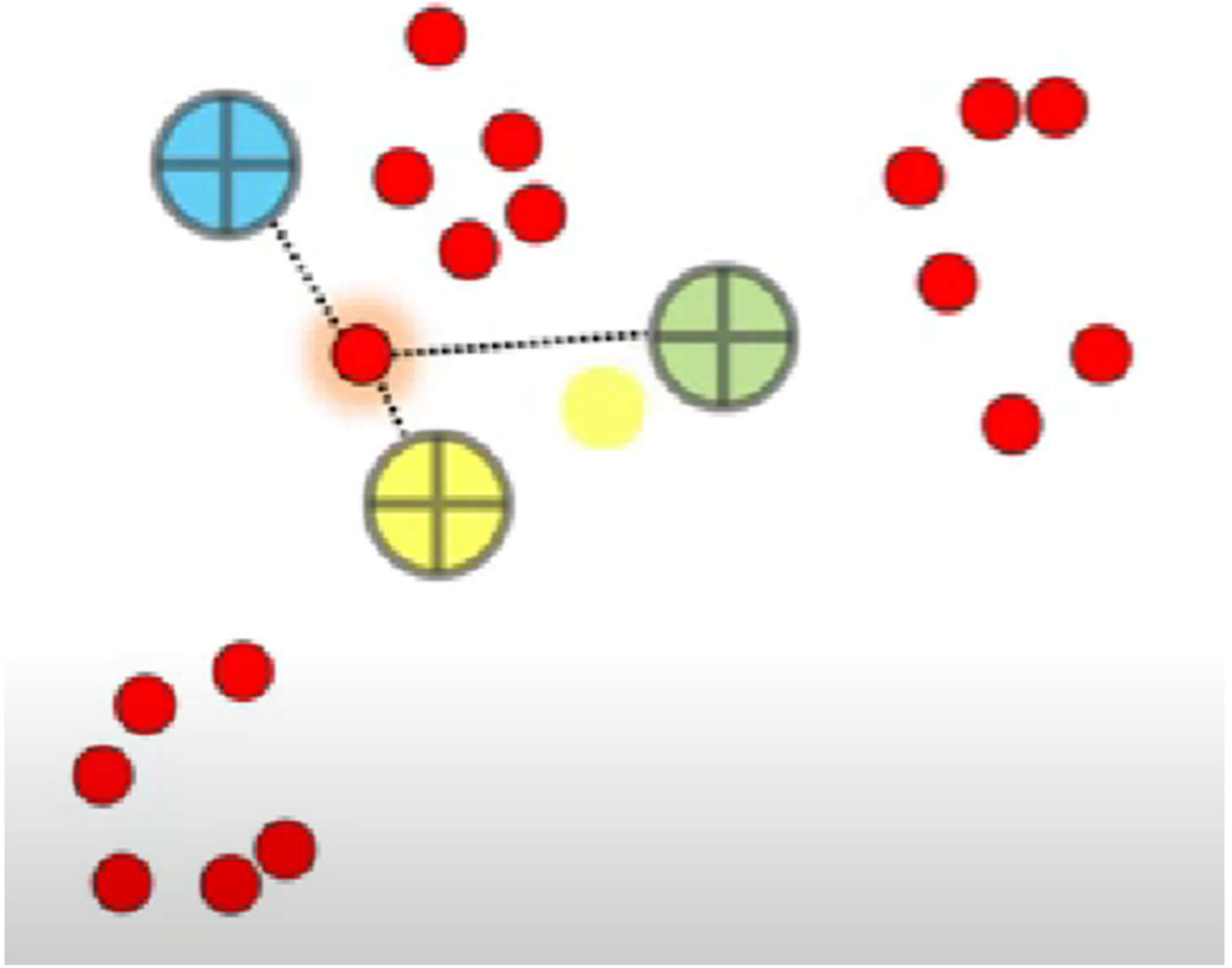
- La calidad de los agrupamientos creados dependerá de la aplicación final, por lo cual es el usuario el que determinará la calidad en base a si los agrupamientos creados son útiles y se ajustan a sus necesidades. Por lo tanto, según el objetivo del problema a tratar, habrá que seleccionar un algoritmo u otro.

Agrupamientos exclusivos

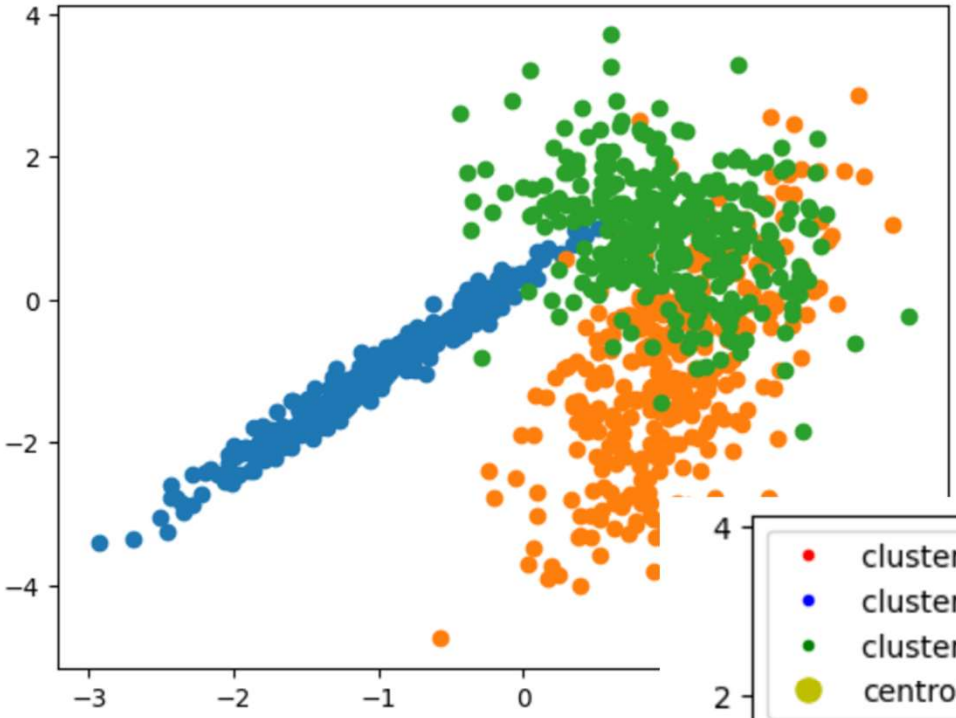
- Cada uno de los clústeres tiene al menos un objeto y los objetos se agrupan pudiendo pertenecer únicamente a un clúster.
- Tienen como punto de partida el **número de clústeres** que se desea generar.
- Generalmente utilizan una medida de distancia para generar los clústeres.

Algoritmo K-means

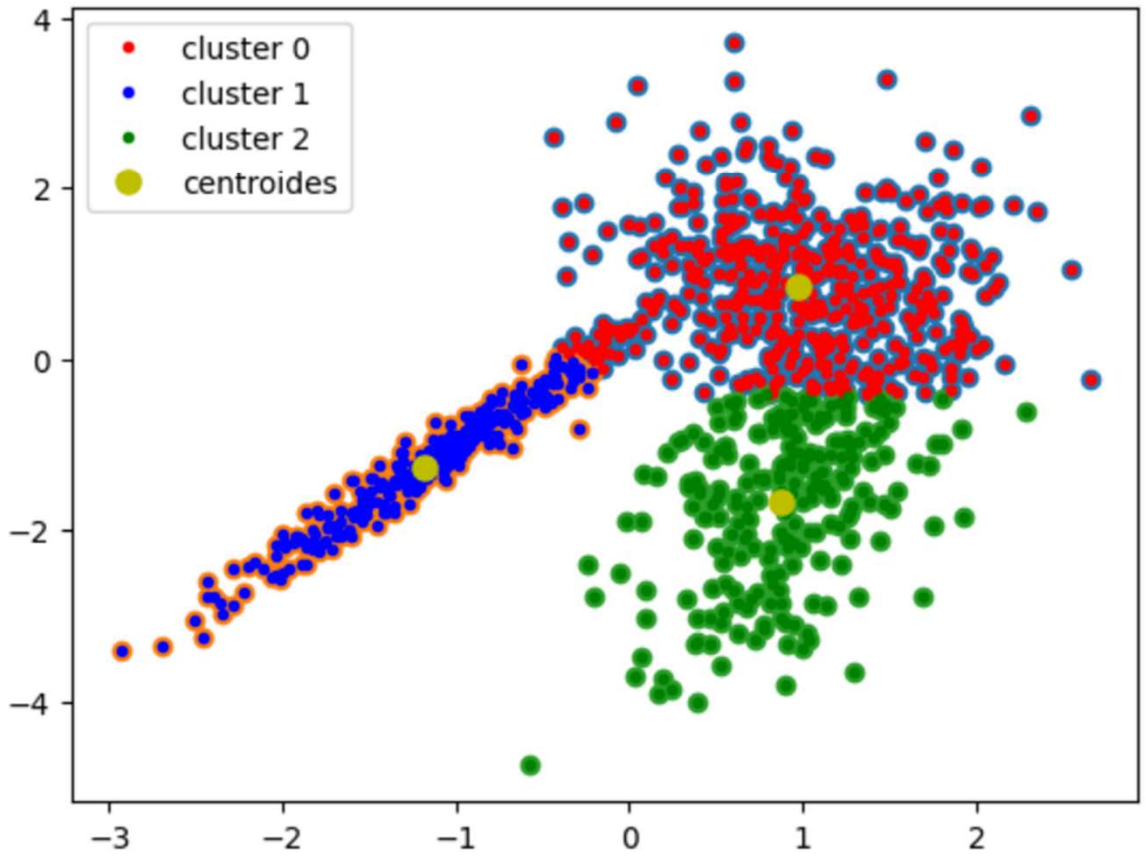
1. Selecciona *k* objetos del conjunto inicial de manera aleatoria. Cada uno de estos *k* objetos representan cada uno de los centroides de los *k* clústeres.
2. El resto de los objetos se asigna al clúster cuyo centroide es más similar. Esta similitud se calcula mediante una **medida de distancia** entre el objeto y el centroide. En este paso ya se obtienen *k* agrupamientos o clústeres iniciales.
3. Para cada clúster generado en el paso previo, se recalcula la posición del centroide o media de los puntos.
4. Los pasos 2-3 se repiten hasta que las posiciones de los centroides no varían.



Sin K-means



Con K-means



K-Means

- Mayormente aplicado a datos que tienen pocas dimensiones, son numéricos y se pueden dividir fácilmente.

- Ventajas
  - Sencillo, rápido y escalable
- Desventajas
  - Selección de k
  - Selección del centroide
  - Los valores atípicos disminuyen su rendimiento

```
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import KMeans
from matplotlib import pyplot

X, y = make_classification(n_samples = 1000, n_classes= 3,
                           n_features=2, n_informative=2,
                           n_redundant=0, n_clusters_per_class=1,
                           random_state=4)

model = KMeans(n_clusters = 3)
model.fit(X)

#model.labels_
predict = model.predict(X)
clusters = unique(predict)
colores = ["blue", "orange","green","red","purple","brown","pink","black" ]
for cluster in clusters:
    fila = where(predict == cluster)
    pyplot.scatter(X[fila,0], X[fila,1])

    pyplot.scatter(model.cluster_centers_[cluster][0],
                   model.cluster_centers_[cluster][1],
                   marker ="p", s= 280, color=colores[cluster])
```

Agrupamiento jerárquico.

- Algoritmos que generan jerarquías de clústeres.
- Estos algoritmos trabajan de manera incremental.

Existen dos categorías de este tipo de algoritmos

Algoritmos aglomerativos:

- Generan un clúster por cada objeto
- Iterativamente los clústeres se van agrupando entre si, generan clústeres de mayor tamaño
- Finaliza bajo una condición de paro o cuando todos los elementos se encuentran dentro de un solo clúster.

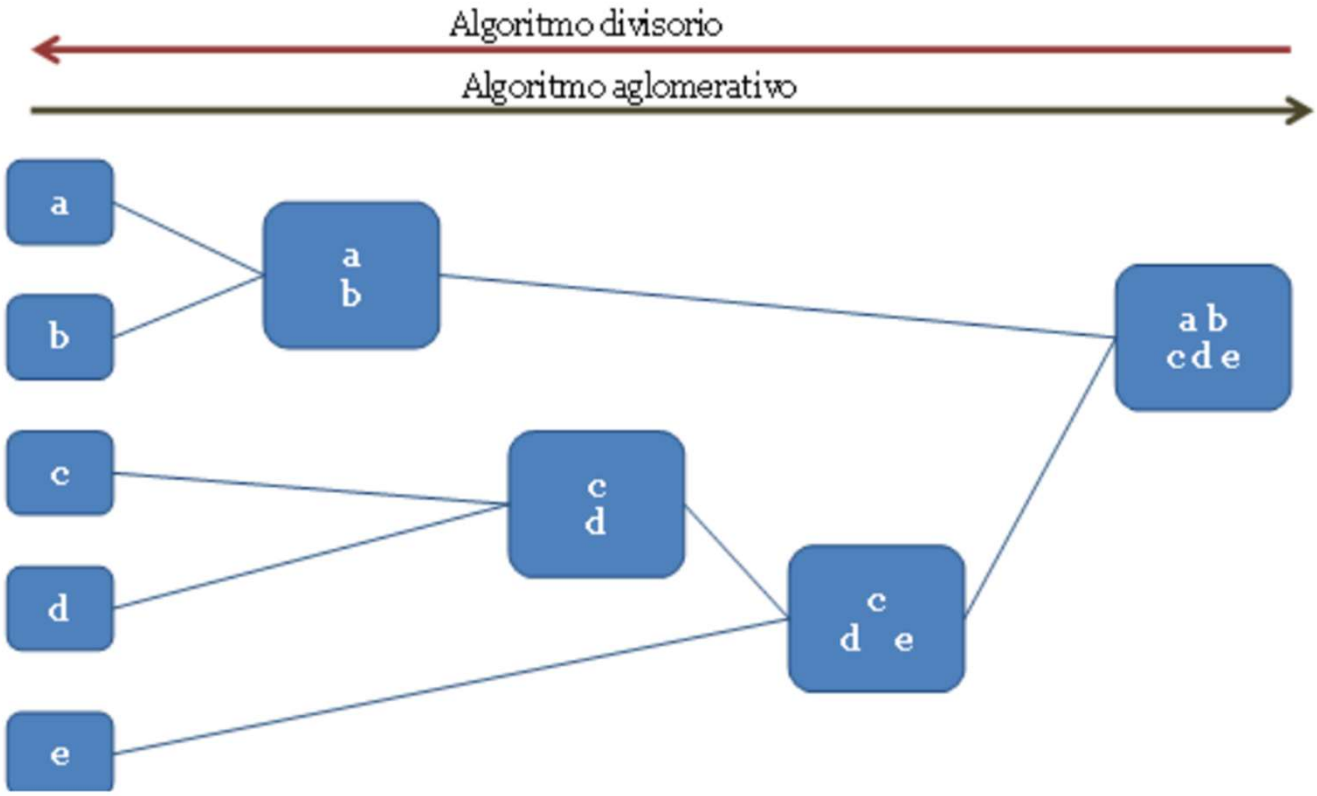
1. A cada punto se le asigna un clúster, de modo que inicialmente se tienen N clústeres de 1 elemento cada uno.

2. Se calcula la distancia entre los clústeres y, aquellos que estén más cercanos, se juntan en un único clúster.

3. Se calcula la distancia entre el nuevo clúster y el resto de clústeres.

4. Se repiten los pasos 2 y 3 hasta que todos los puntos estén agrupados en un único clúster.

- Algoritmos jerárquico divisorios
  - Parten de un clúster general que contiene todos los elementos
  - Posteriormente, se va particionando hasta tener clústeres más pequeños

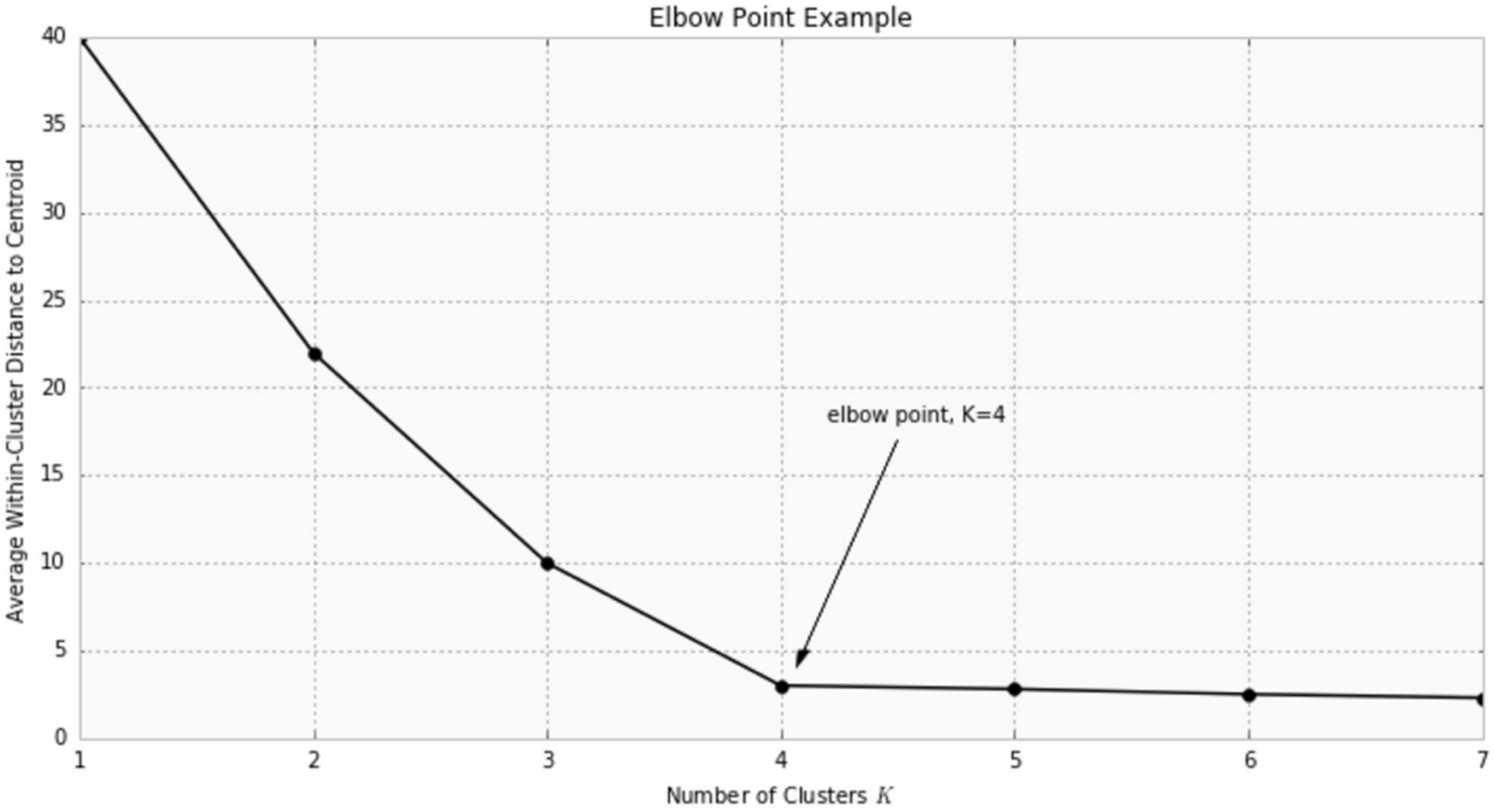


```
from sklearn.cluster import AgglomerativeClustering
```

```
model = AgglomerativeClustering(n_clusters=3)
```

```
model.fit_predict(X)
```

Método de Elbow



- Fuente: https://www.aprendemachinelearning.com/k-means-en-python-paso-a-paso/

WCSS

La **inercia** se define como la suma de las distancias cuadradas de los puntos a los centroides del clúster al que pertenecen:

$$\text{Inercia} = \sum_{k=1}^{K} \sum_{x \in C_k} \|x - \mu_k\|^2$$

Donde:

- K: Número de clústeres.
- C_k: Conjunto de puntos en el clúster k.
- μ_k: Centroide del clúster k.
- $\|x - \mu_k\|$: Distancia euclidiana entre el p[illegible]o x y el centroide μ_k.

- Que tan pegados (juntos) están los datos del centroide

Para que usar clustering

- Exploración de datos
  - Tendencias, patrones, valores atípicos
- Reducción de datos
  - Al realizar grupos, facilita su análisis
- Segmentación de clientes
  - Dirigir campañas a clientes específicos

Medidas de similitud entre clústeres

- La **Precision** cuenta los verdaderos positivos, cuántos ejemplos se clasifican correctamente dentro del mismo cluster:

$$Pr = \frac{T_P}{T_P + F_P}$$

- El **Recall** evalúa el porcentaje de elementos que se incluyen correctamente en el mismo clúster:

$$R = \frac{T_P}{T_P + F_N}$$

Gracias por su atención

www.unir.net