Agrupamiento

Aprendizaje Automático

Juan David Martínez
jdmartinev@eafit.edu.co

Agenda

- Agrupamiento
- Métodos basados en distancias
- Métodos basados en densidades

Agrupamiento

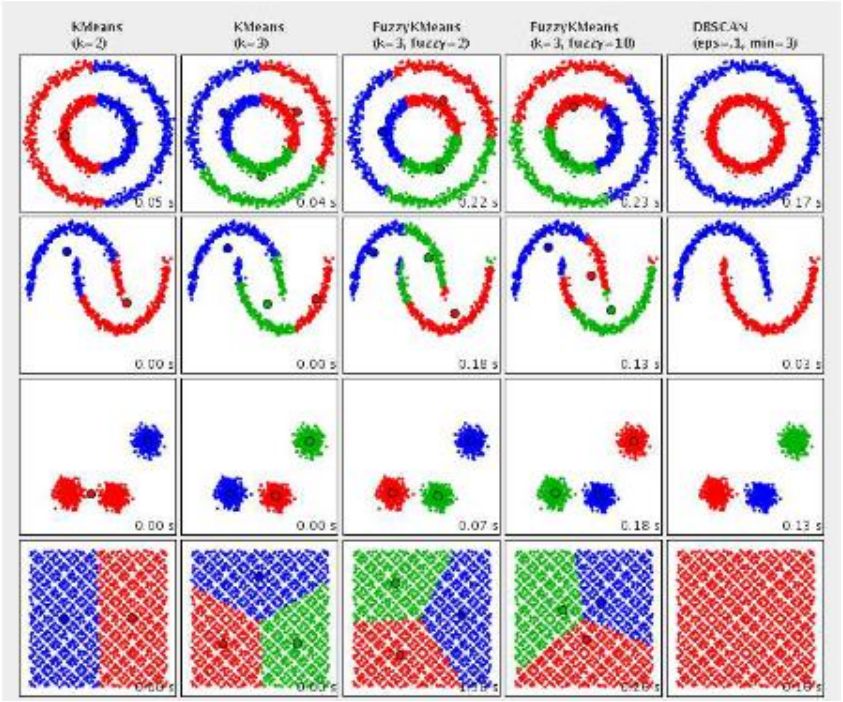
La mayoría de problemas de **Machine Learning** se basan en aprendizaje supervisado.

Problema: La gran mayoría de datos disponibles no están etiquetados

Tenemos las características $\boldsymbol{X}$ pero no tenemos las etiquetas $\boldsymbol{y}$

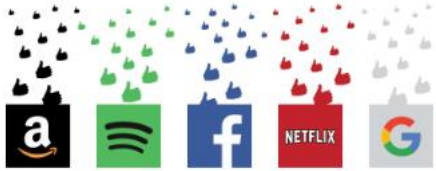
Agrupamiento

Clustering es la tarea de identificar muestras similares y asignarlas a agrupaciones, es decir, generar grupos de muestras similares.

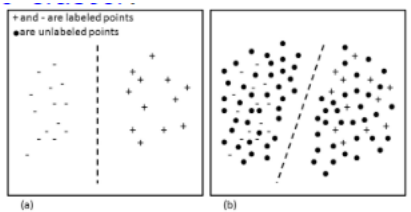


Aplicaciones

Segmentación de clientes

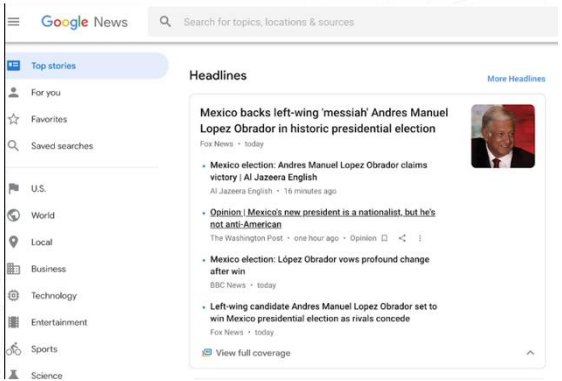


Aprendizaje semi-supervisado



Segmentación de imagen

Segmentación de contenido



K-Means: Definición

Escoger de forma aleatoria los puntos representativos de cada cluster $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(K)}$

Iterar:

a. Dado $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(K)}$, asignar $\mathbf{x}^{(i)}$ al cluster más cercano

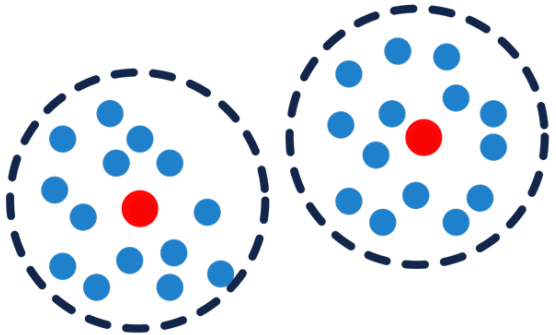
b. Dado $C_1, \ldots, C_k$,hallar los mejores puntos representativos de cada cluster

c. Calcular el costo para la nueva partición

d. El algoritmo para cuando:

   i. Los puntos representativos no cambien

   ii. El costo no cambie



K-Means: Función de costo

$$cost(C_1, \ldots, C_k) = \sum_{j=1}^{K} cost(C_j)$$

Homogeneidad de cada cluster

- Diámetro: Distancia del centro al punto más lejano
- Distancia promedio
- **Distancia promedio de un punto representativo a todos los elementos del cluster**

$$cost(C_j, \mathbf{z}^{(j)}) = \sum_{i \in C_j} dist(\mathbf{x}^{(i)}, \mathbf{z}^{(j)})$$

K-Means: Función de costo

Similitud de coseno

$$\cos(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \frac{\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}}{||\mathbf{x}^{(i)}|| ||\mathbf{x}^{(j)}||}$$

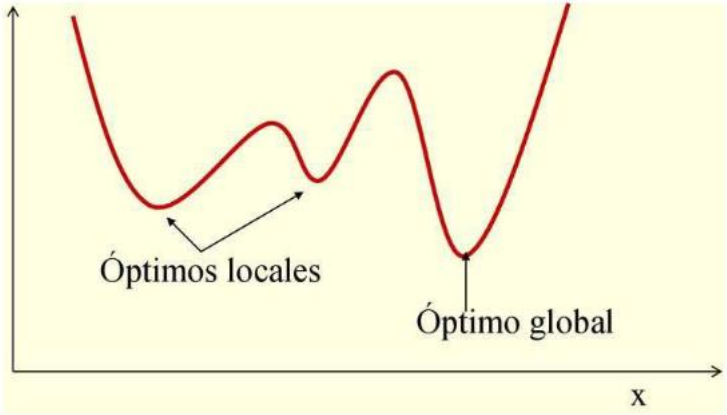
Distancia euclidea

$$dist(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = ||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||^2$$

$$cost(C_1, \ldots, C_K, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(K)}) = \sum_{j=1}^{K} \sum_{i \in C_j} ||\mathbf{x}^{(i)} - \mathbf{z}^{(j)}||^2$$

K-Means: Variabilidad

Aunque se garantiza que el algoritmo converge (encuentra centroides que no cambian), puede no converger a la solución correcta. Esto depende de la inicialización del centroide



K-Means ++

En lugar de inicializar los centroides al azar, es preferible inicializarlos usando un algoritmo llamado K-Means ++

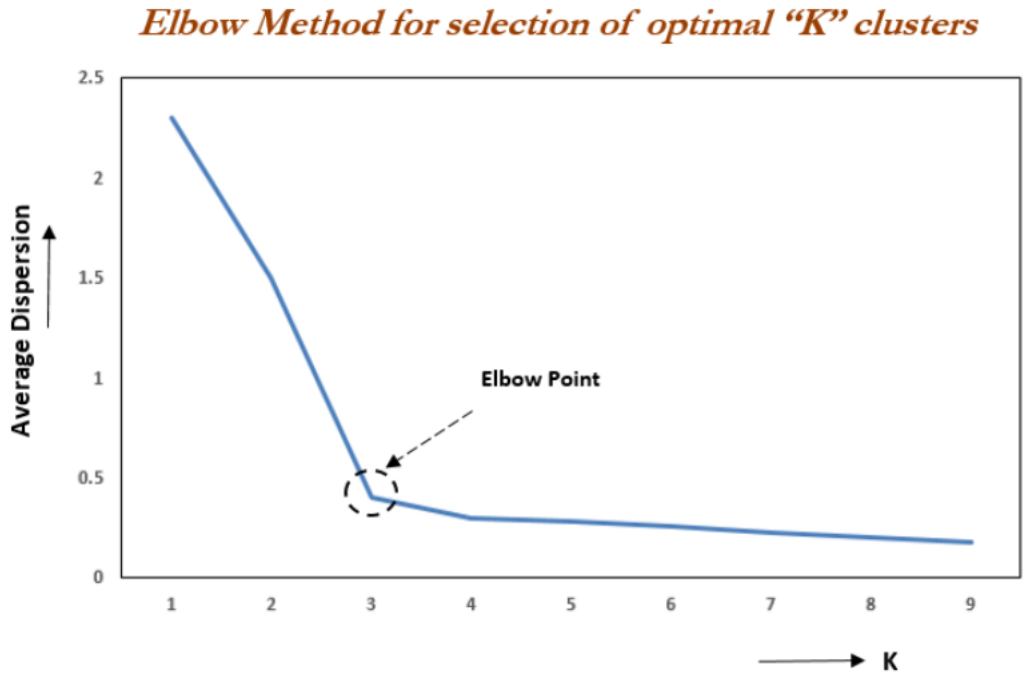
- Tome un centroide $\boldsymbol{z_1}$ elegido uniformemente al azar del conjunto de datos
- Tome un nuevo centroide $\boldsymbol{z_i}$ eligiendo una instancia $\mathbf{x_i}$ con probabilidad:

$$\frac{D(\boldsymbol{x}_i)^2}{\sum_{j=1}^{m} D(\boldsymbol{x}_j)^2}$$

- $D(\boldsymbol{x}_i)^2$ es la distancia entre la instancia $\boldsymbol{x}_i$ y el centroide más cercano que ya se eligió. Esto asegura que las instancias que están más lejos de los centroides ya elegidos sean mucho más probables como centroides
- Repita el paso anterior hasta que se hayan elegido todos los centroides k

Número óptimo de clusters

Se utiliza el criterio del codo o L-curve



Silohouette score

- a: distancia media dentro del grupo
- b: distancia media a las instancias del grupo más cercano

Este score puede variar entre -1 y 1:

- ≈ 1: Instancia está bien en su grupo
- ≈ 0: Instancia cerca de un límite
- ≈ -1: Instancia posiblemente mal asignada

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

Limitaciones

A pesar de que el algoritmo es rápido y escalable, no es perfecto:

- Es necesario ejecutar varias veces el algoritmo para evitar soluciones sub-óptimas
- Se necesita especificar el número de clusters
- No se comporta bien cuando los grupos tienen diferentes tamaños, diferentes densidades o formas no esféricas

Dependiendo de los datos, puede haber diferentes algoritmos que funcionen mejor.

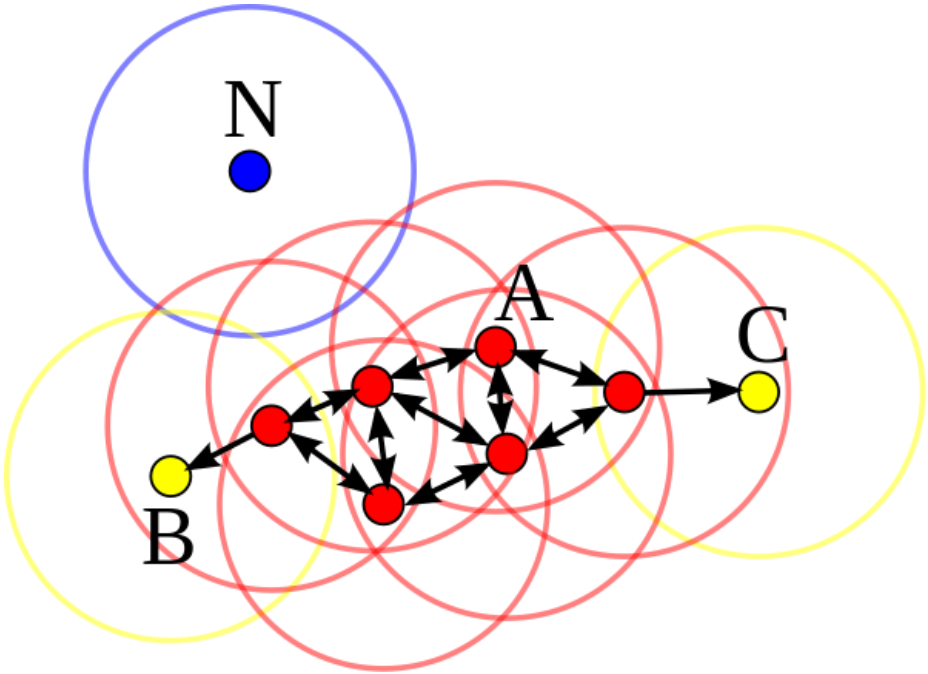
DBSCAN

Este algoritmo define grupos como regiones continuas de alta densidad:

1. Para cada instancia, el algoritmo cuenta cuántas instancias se encuentran dentro de una pequeña distancia ϵ (ϵ-neighborhood)
2. Si una instancia tiene al menos min_samples en su vecindario ϵ (incluida ella misma), entonces se considera una **instancia central**
3. **Todas las instancias cercanas a una instancia central pertenecen al mismo cluster.** Esto puede incluir otras instancias centrales
4. Cualquier instancia que no sea central y que no tenga una en su vecindario se considera una **anomalía**

Este algoritmo funciona bien si todos los grupos son lo suficientemente densos y están bien separados por regiones de baja densidad

DBSCAN



Algoritmo simple pero capaz de identificar cualquier número de clusters, de cualquier forma, es robusto para atípicos, tiene dos hiperparámetros

Si la densidad varía significativamente entre los grupos, puede ser imposible capturar los grupos correctamente

DBSCAN