6.1 Introducción

Los datos (conjunto de entrenamiento) al final son los que nos aportan información para poder construir una solución al problema. La diferencia es que en este caso los datos no están etiquetados, por lo tanto, no tenemos a priori identificado como debería responder el sistema, sino que al final todo se basa en buscar proximidad, en buscar una forma de medir o cuantificar la similitud entre los datos, para poder agrupar aquellos que se parezcan entre sí de acuerdo con ese criterio. El criterio empleado debe tener sentido dado el problema a abordar.

Es el diseñador el que le atribuye significado y valor al resultado. La máquina aprende a agrupar los datos, quien les asocia un significado es el diseñador. En el aprendizaje supervisado ya había claramente una proyección sobre el conjunto de lo que era el significado de estos dentro del problema a resolver. Si partimos de un conjunto de datos y un conjunto de resultados, que deberían estar asociados a ellos, estos reultados ya son el significado de estos datos. Por lo tanto, en estos conjuntos de entrenamiento ya va a una buena parte de información significativa desde el punto de vista de la salida de vuelta por un sistema de aprendizaje supervidado.

En el aprendizaje no supervidado, no existe esta información. El sistema únicamente creará los agrupamientos y tendrá que ser el diseñador el que les dé significado.

Después del supervisado es el tipo de aprendizaje más utilizado.

2. Método K-medias

Es el método de aprendizaje no supervisado más intruitivo y simple. Tiene multitivos de variantes, pero vamos a ver el más elemental.

Suponemos que tenemos un conjunto de datos como el de la foto caracterizados a partir de dos variables, como la altura y el peso de un conjunto de personas.

A ojo podemos disintiguir agruapaciones en la imagen. Es un criterio empírico, no sabemos cómo realizamos estas agrupaciones en nuestra cabeza. ¿Cómo lo haría el método k-medias?

Suponemos que queremos identificar 2 grupos de datos. Partimos de los llamados centroides de cada uno de los grupos y los situamos aleatoriamente.

A partir de estos puntos de referencia, vemos que puntos están más cerca de uno y cuales más cerca del otro, identificando cada punto con un centroide. Depende del tipo de distancai que usemos y esta es una cuestión muy relevante que dependerá, en la mayoría de los casos, del problema que queremos resolver.

La iamgen anterior sería una primera división del conjunto de putnos de partida en los dos grupos que tratamos de identificar utilzando distnacia euclídea. EL poceso lógicamente no termina aquí, va iterando: - Ahora que ya tenemos disitnguidos dos conjutnos, vamos a calcular sus nuevos centroides: buscamos el punto medio de cada conjunto. - AL calcularlo, se van moviendo los centroides y puede ser que algún dato cacmbie de conjunto al estar más cerca del nuevo centroide del otro grtupo - Se repite el proceso hasta convergencia

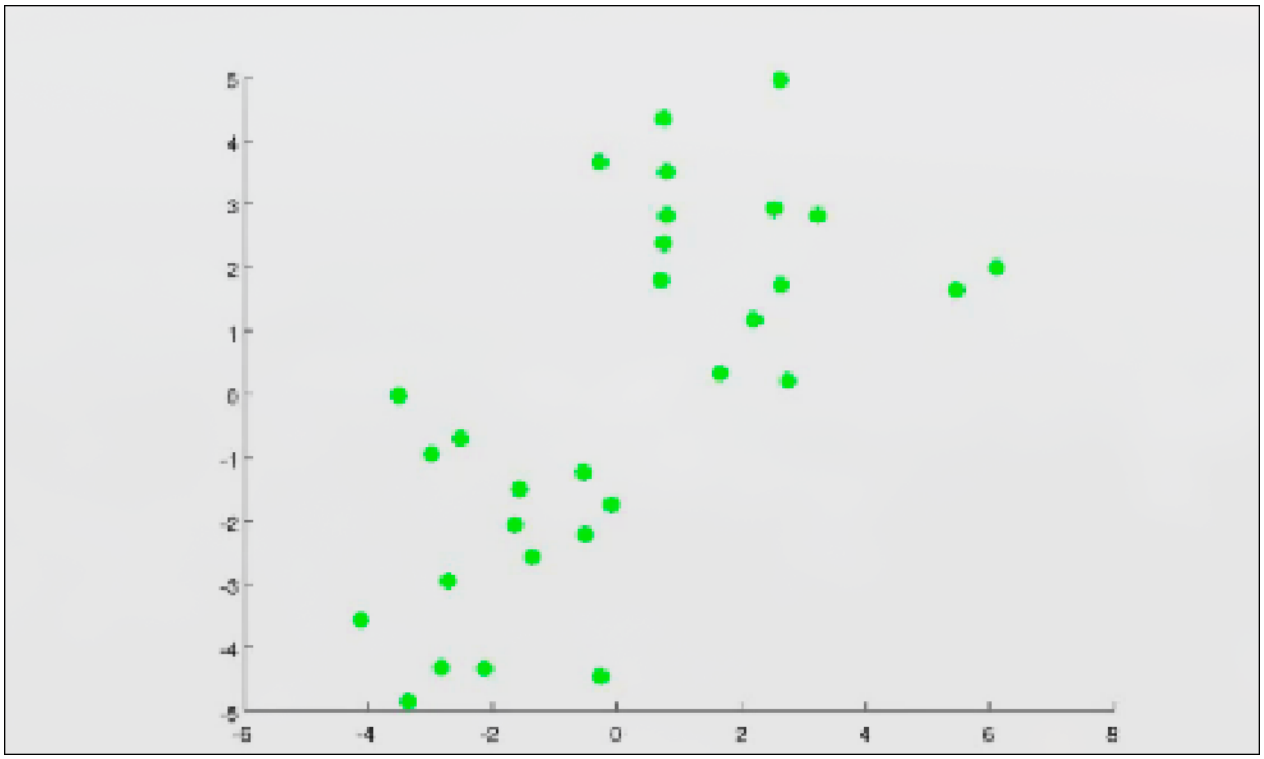

Figure 1: Pasted image 20251109161223.png

La base de operaciones de este algoritmo es muy simple, si se analiza en detall y se utilizan otro ditpo de distancias, este puede llegar a ser más complejo. Algunas cuestiones: - Numero de agrupamientos en los que dividimos, nos lo pueden dar predefinido o no. Si no nos lo dan, veremos un criterio para elegirlo. - Donde inicializamos los centroides, ya que podrían variar el resultado. Se suele hacer aleatoriamente e influirá en la velocidad del algoritmo (número de iteraciones necesarias para converger). Por ejemplo, en la imagen - 1) A ojo, podemos distinguir 3 agrupamientos - 2) Si tenemos la suerte de inicializar bien los centroides desde un principio, el proceso será óptimo y rápido. - 3) Si inicializamos así los centroides, el rojo y el verde se repartirán esos 5 puntos y el resto se quedarán siempre en el azul. Mala distribución. - 4) Si partimos de esos centroides, no sería correcta la agrupación porque el cluster rojo quedaría sin puntos asociados.

Para evitar problemas respecto a la inicialización de los clusters, podemos repetir el proceso muchas (cientos o miles, considerando el coste computacional que esto puede suponer) veces partiendo de distintos centroides y quedarnos con la mejor solución.

¿Cuál es esta mejor solución? A ojo, podemos ver en la imagen anterior cual es la mejor solución, la 2. Pero esto no nos vale computacionalmente. Deberemos tener un criterio de coste o rendimiento. Podemos suponer que este criterio normalmente es la solución que mejora una función de coste: por ejemplo, podemos decir que el coste o error será mayor cuanto más separados estén los puntos de un agrupamiento respecto a su centroide. Sumando las distancias euclídeas, en este caso, de los puntos a su centroide, podemos tener una medida del coste de la solución. Así también podríamos ver computacionalmente que la mejor solución es la 2, ya que es la que minimiza la

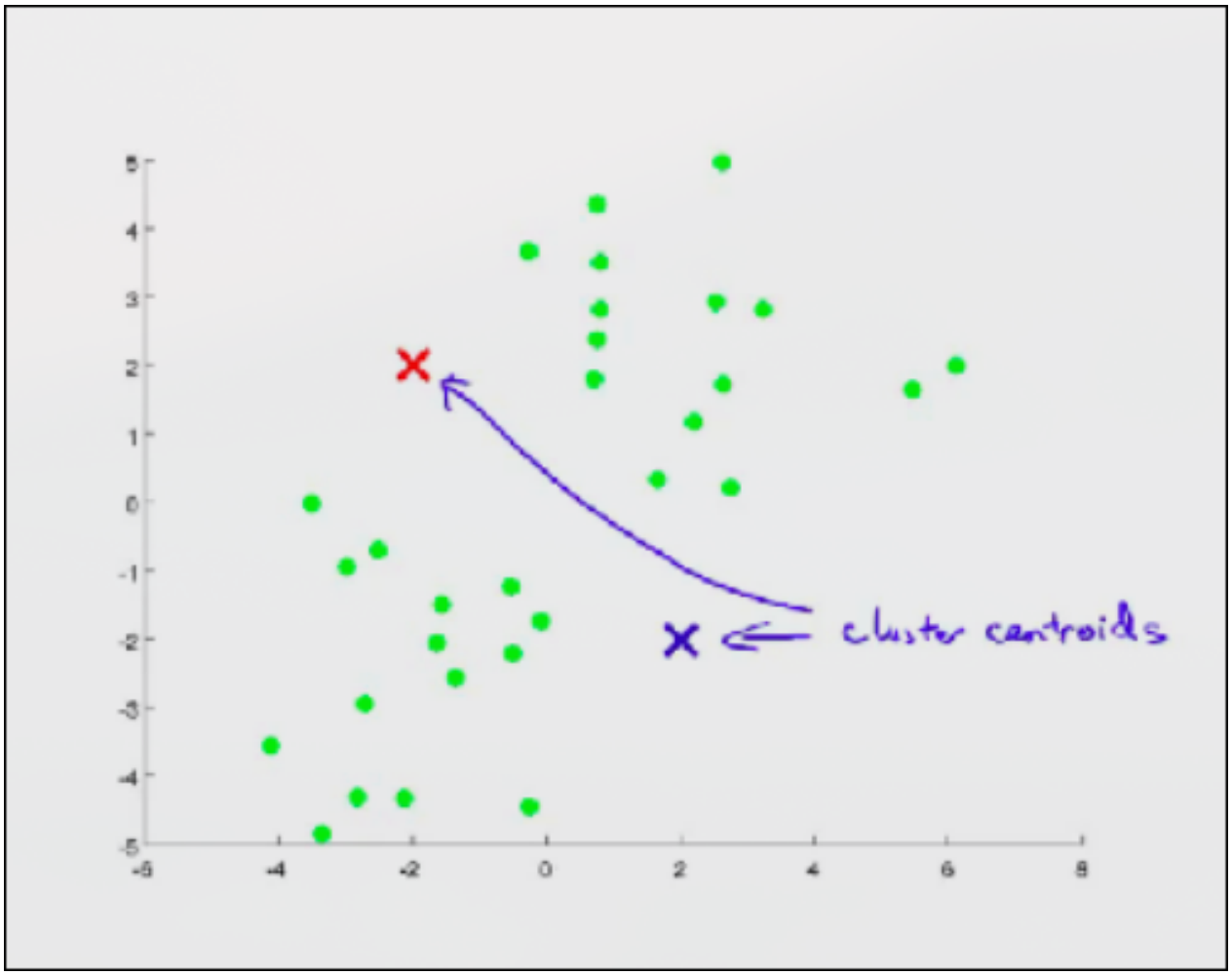

Figure 2: Pasted image 20251109161341.png

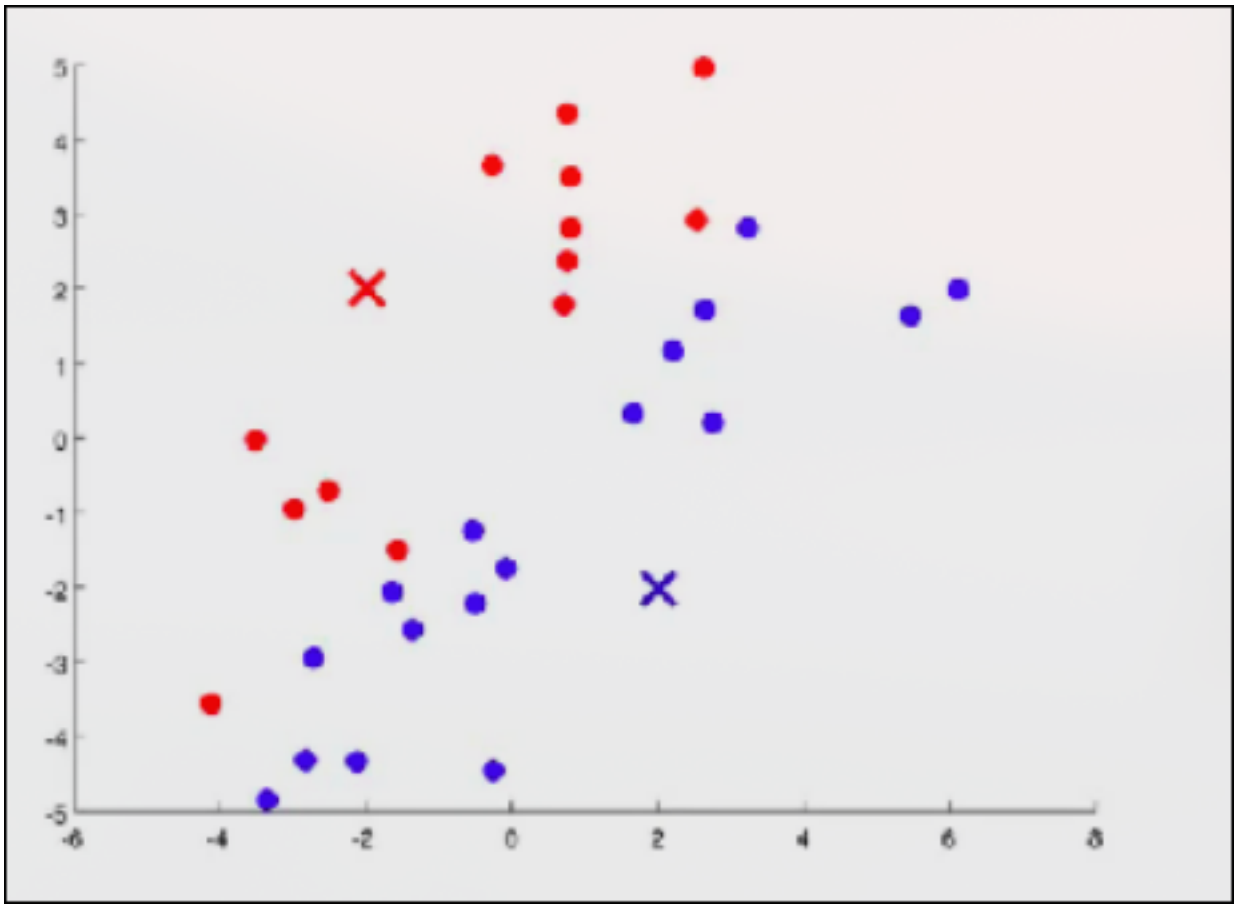

Figure 3: Pasted image 20251109161444.png

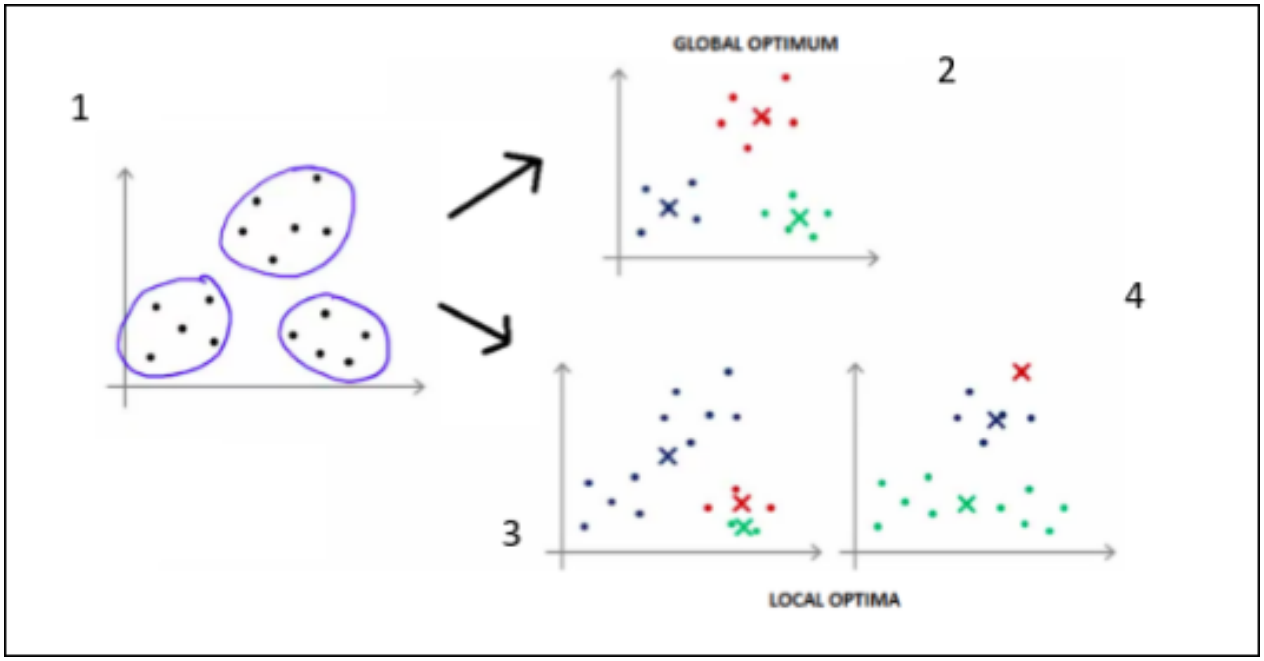

Figure 4: Pasted image 20251109161815.png

función de error en este caso.

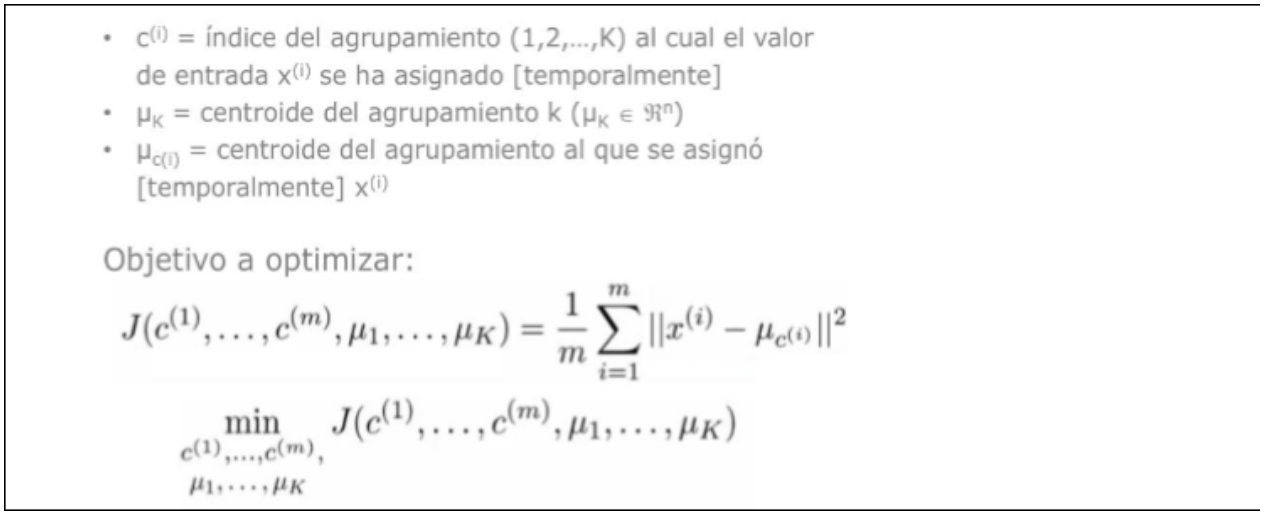

Figure 5: Pasted image 20251109161932.png

Tenemos un proceso de agrupamiento cuyo objetivo es maximizar la semejanza entre sí de los puntos pertenecientes a un agrupamiento, es decir, minimizar la distancia entre los puntos asociados en un agrupamiento y maximizarla con los de otros agrupamientos.

Pensando así podríamos decir que cada punto sea un agrupamiento distinto y por lo tanto sería optimo, ya que no habría distancia entre los puntos del agrupamiento, al ser solo un punto. Evidentemente, no diseñamos un sistema para que al final nos diga que tenemos tantos agrupamientos como puntos del conjunto de datos. Habrá que llegar a una solución de compromiso. ¿Cuántos clusters vamos a diferenciar? Podemos tener un número predefinido o obtener dicho número utilizando un criterio, que minimice o maximice, dependiendo de cada caso, el criterio establecido.