Minería de datos: aprendizaje no supervisado y detección de anomalías

Clustering

00. Presentación

Isaac Triguero Velázquez

Presentación

Isaac Triguero Velázquez

Investigador Senior Distinguido
Departamento de Ciencias de la Computación e Inteligencia Artificial
E-mail: triguero@decsai.ugr.es

Guía de la asignatura

https://masteres.ugr.es/datcom/docencia/plan-estudios/guia-docente/M51/56/3/7

***Módulo** : Clustering*

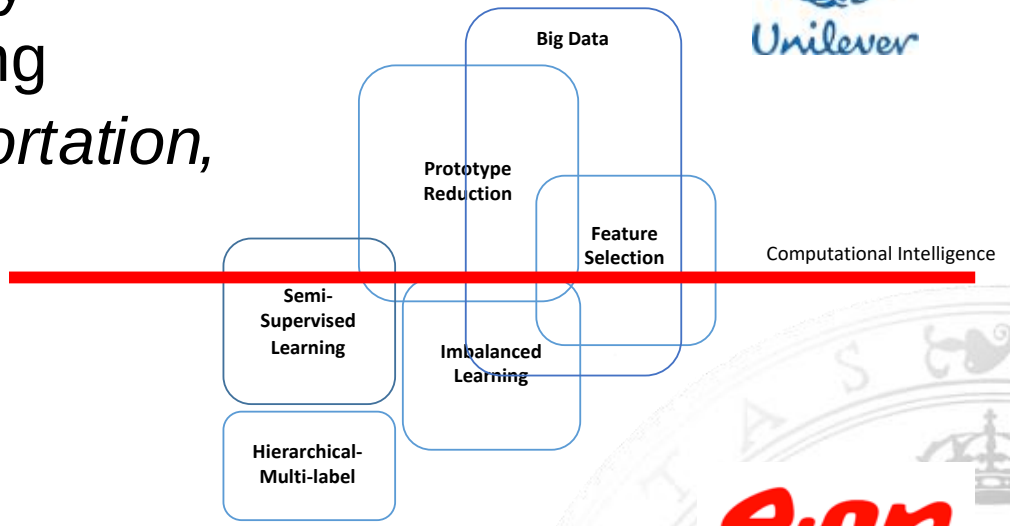
***Contenidos:** Introducción a los métodos de agrupación dentro del paradigma del aprendizaje no supervisado*

Mi investigación

- *Big data research*
  - Data Pre-processing
- *Computational intelligence techniques*
  - Fuzzy logic, Evolutionary algorithms, Deep learning
- *Healthcare, Energy, Transportation, Hospitality…*
- *Where I am heading:*
  - Sustainability
  - Green AI
  - General-Purpose AI

Fullstep
INSPIRE PROGRESS

Unilever



e·on

ArcelorMittal

Presentación

Horarios de clase

3 sesiones

Sesión 1:	*lunes 4/11*	*de 18:00 a 20:30*
Sesión 2:	*martes 5/11*	*de 15:30 a 18:00*
Sesión 3:	*miércoles 6/11*	*de 18:00 a 20:30*

Tutorías

https://decsai.ugr.es/informacion/directorio-personal/isaac-triguero-velazquez

Lunes de 11-12, Jueves 11-12, Edf Mecenas, Facultad de ciencias.

Google Meet: ***concertar cita previa por e-mail***

Metodología docente

¿Qué vamos a hacer?

Aprender el concepto, las técnicas y las aplicaciones del clustering

Aplicar los conocimientos teóricos a problemas prácticos

No vamos a estudiar reducción de dimensionalidad

¿Qué herramientas vamos a utilizar?

Ordenador (personal)

Python + Jupyter Notebooks/Lab (se recomienda usar anaconda)

Numpy, Scikit-learn library

¿Qué se va a evaluar?

Trabajo de análisis de datos con un dataset diferente a los usados en clase (como un Jupyter notebook)

Temario

Sesión 1

¿Qué es clustering?

Métodos basados en centroides: k-means

Implementación básica con Python

Sesión 2

Elementos básicos de un algoritmo de clustering

Implementación con Python y scikit-learn

Parámetros y medidas de calidad del clustering

Otros algoritmos basados en centroides

Sesión 3

Clustering basado en densidad

Clustering jerárquico

Implementación con scikit-learn – caso de estudio

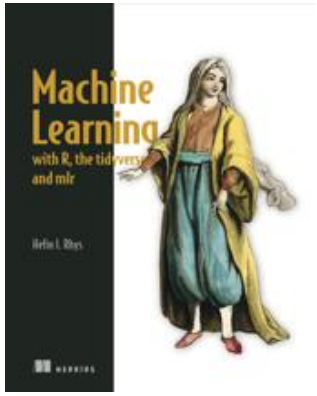
Bibliografía

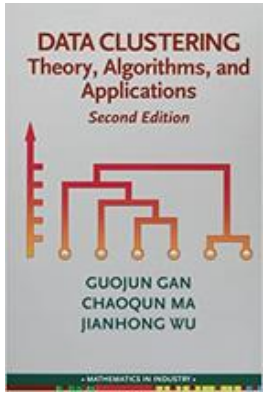
H.I. Rhys. Machine Learning with R, the tidyverse, and mlr. Manning, 2020.

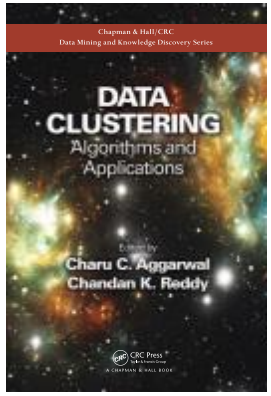
G. Gan, C. Ma, J. Wu. Data Clustering: Theory, Algorithms and Applications. SIAM, 2007.

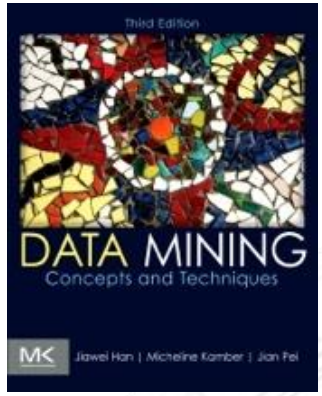
C.C. Aggarwal, C.K. Reddy. Data Clustering: Algorithms and applications. CRC Press, 2014.

J. Han, M. Kamber, J. Pei. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2011.









DECSAI

Minería de datos: aprendizaje no supervisado y detección de anomalías

Clustering

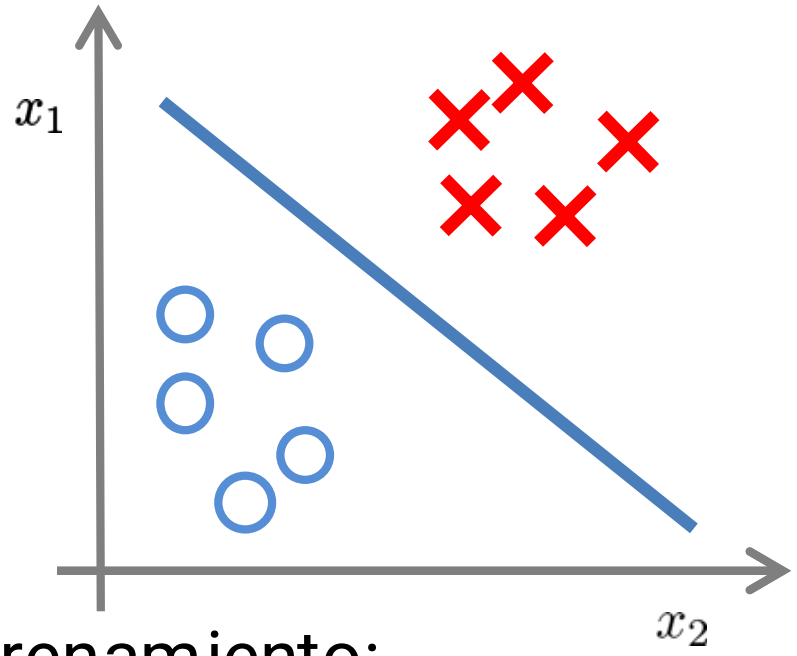
01. Introducción al clustering

Isaac Triguero Velázquez

Índice

Aprendizaje Supervisado vs. No supervisado

- *En **aprendizaje supervisado** buscamos la frontera de decisión que separe las dos clases*
- *Para cada ejemplo, tenemos su etiqueta de clase*



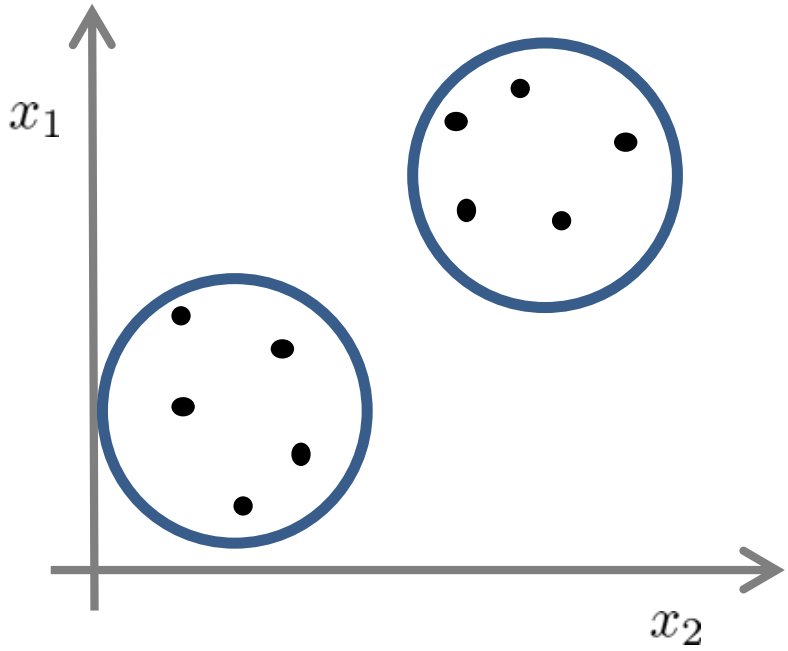
Conjunto de entrenamiento:

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$$

¿Qué es el *clustering*?

Definición

- *El algoritmo* ***busca una estructura en los datos***
- *Buscamos grupos de ejemplos similares (clusters)*
- ***No hay etiquetas para los ejemplos***



Conjunto de entrenamiento: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

¿Qué es el *clustering*?

Definición

Es una técnica de aprendizaje no supervisado

No se dispone de clases predefinidas ni de ejemplo etiquetados; es decir, no se conocen las agrupaciones para ningún subconjunto de individuos

Búsqueda de agrupaciones en datos

Proceso de agrupar un conjunto de objetos ***descritos*** *mediante propiedades en grupos (o clústeres), de forma que un clúster contiene objetos similares entre sí y diferentes a los de otros clústeres*

Suele realizarle en las primeras etapas del proceso de análisis de datos

Los clústeres sirven para resumir los datos, de forma que se pueden utilizar las agrupaciones como representación colectiva de los individuos

¿Qué es el *clustering*?

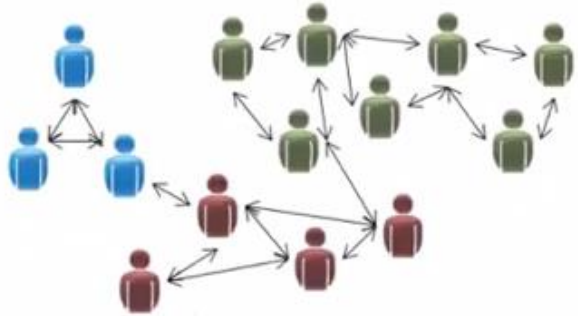
Aplicaciones

Agrupación de…

Segmentación de mercado

Grupos de clientes
Campañas de publicidad dirigida

Análisis de redes sociales



Grupos de amigos

Organización de clusteres

Unión de racks/nodos que trabajan conjuntamente

Análisis de imágenes astronómicas

Comprender cómo funciona la galaxia

¿Qué es el *clustering*?

Retos

- *Dada una serie de ejemplos*
  - Idea: Dividirlos en subconjuntos de ejemplos que son similares entre si

Retos:

¿Cómo medimos esa similitud?

¿Cómo evaluamos la calidad de los resultados?

Índice

K-means clustering

El algoritmo clásico

Aproximación básica *(algoritmo de Lloyd)*

Entrada: *k (número de clústeres), m objetos*

Procedimiento:

1. elegir aleatoriamente los centros de los clústeres

2. repetir mientras haya cambios:

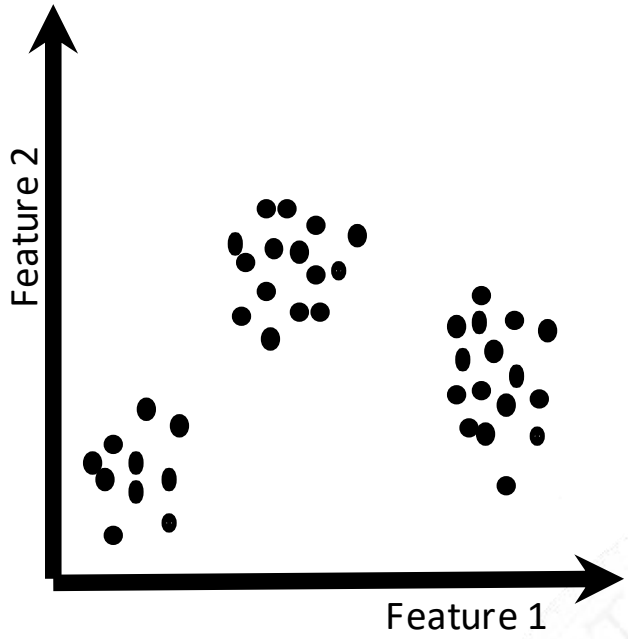
2.1 (re)asignar cada objeto al clúster con centro más cercano

2.2. recalcular los centros como el punto medio de cada clúster

- *Algoritmo basado en particiones*
- *Necesita indicar número de clusters a encontrar*
- *Cada cluster está representado por su centroide*

K-means
Ejemplo

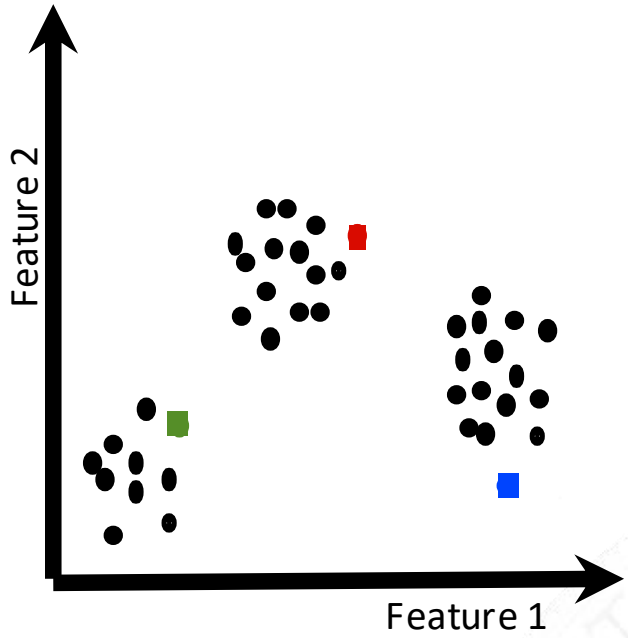
- Initialize K cluster centers
- Repeat until convergence:
  Assign each data point to the cluster with the closest center.
  Assign each cluster center to be the mean of its cluster's data points.



Animation from Databricks

K-means
Ejemplo

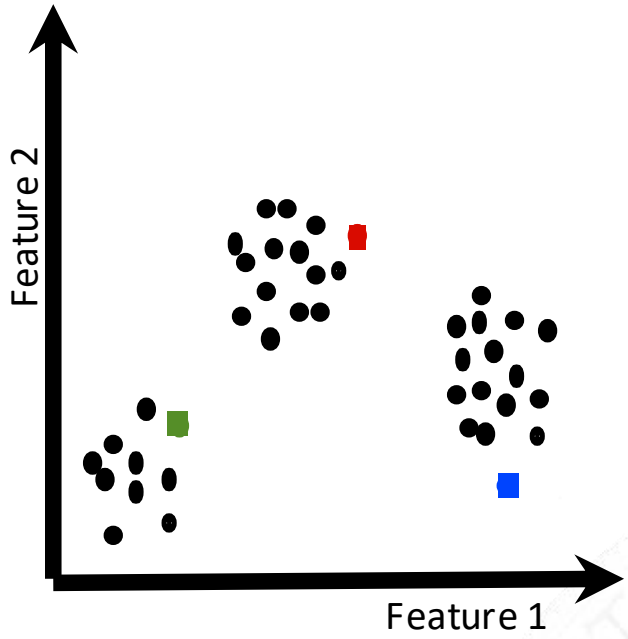
- Initialize K cluster centers
- Repeat until convergence:
  Assign each data point to the cluster with the closest center.
  Assign each cluster center to be the mean of its cluster's data points.



Animation from Databricks

K-means
Ejemplo

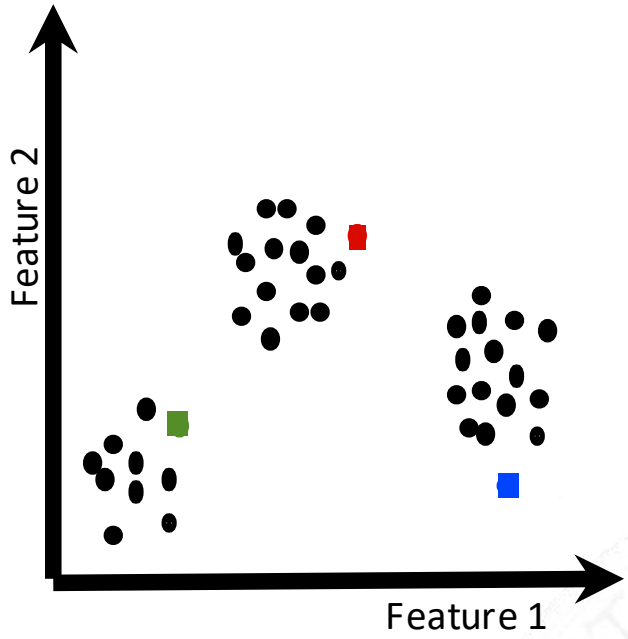
- Initialize K cluster centers
- Repeat until convergence:
  - Assign each data point to the cluster with the closest center.
  - Assign each cluster center to be the mean of its cluster's data points.



Animation from Databricks

K-means
Ejemplo

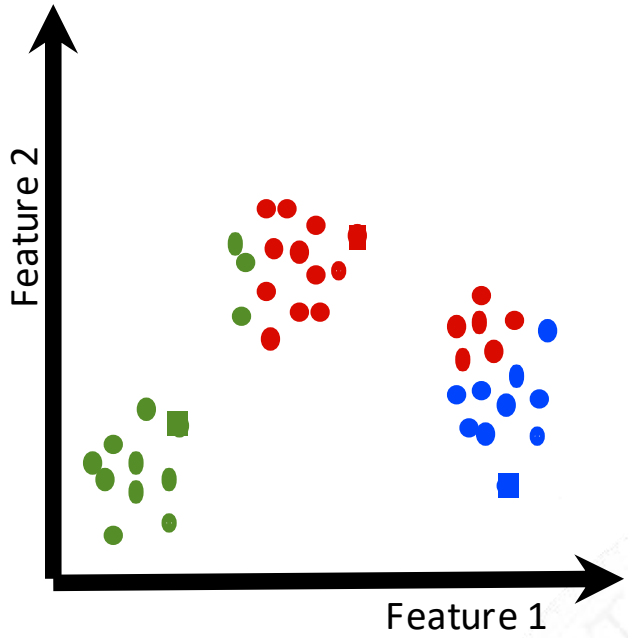
- Initialize K cluster centers
- Repeat until convergence:
  - Assign each data point to the cluster with the closest center.
  - Assign each cluster center to be the mean of its cluster's data points.



Animation from Databricks

K-means
Ejemplo

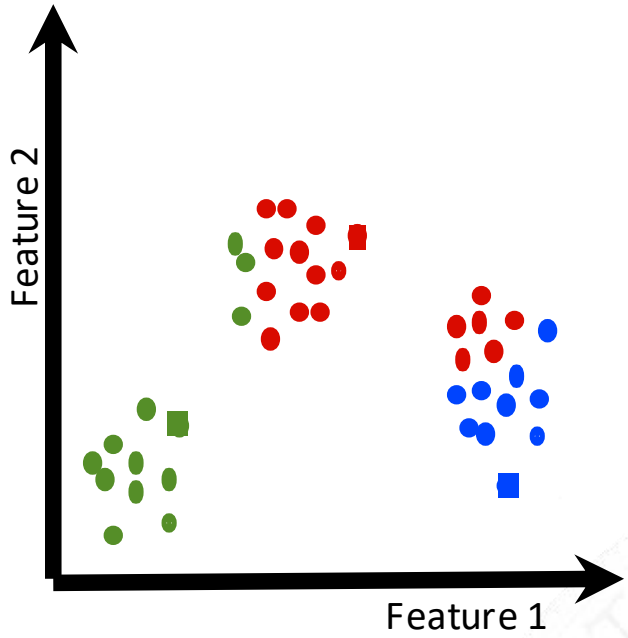
- Initialize K cluster centers
- Repeat until convergence:
  - Assign each data point to the cluster with the closest center.
  - Assign each cluster center to be the mean of its cluster's data points.



Animation from Databricks

K-means
Ejemplo

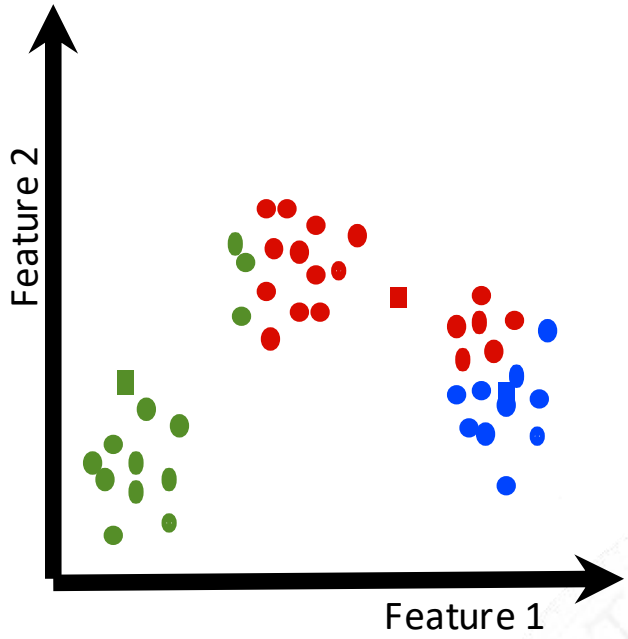
- Initialize K cluster centers
- Repeat until convergence:
  - Assign each data point to the cluster with the closest center.
  - Assign each cluster center to be the mean of its cluster's data points.



Animation from Databricks

K-means
Ejemplo

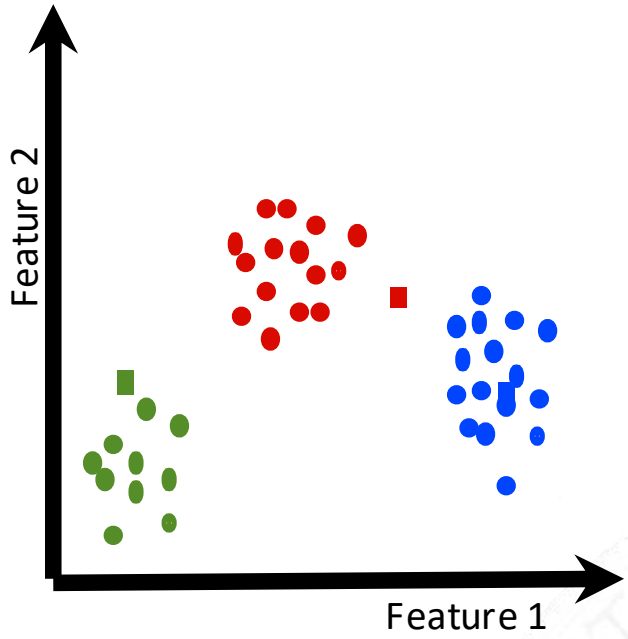
- Initialize K cluster centers
- Repeat until convergence:
  - Assign each data point to the cluster with the closest center.
  - Assign each cluster center to be the mean of its cluster's data points.



Animation from Databricks

K-means
Ejemplo

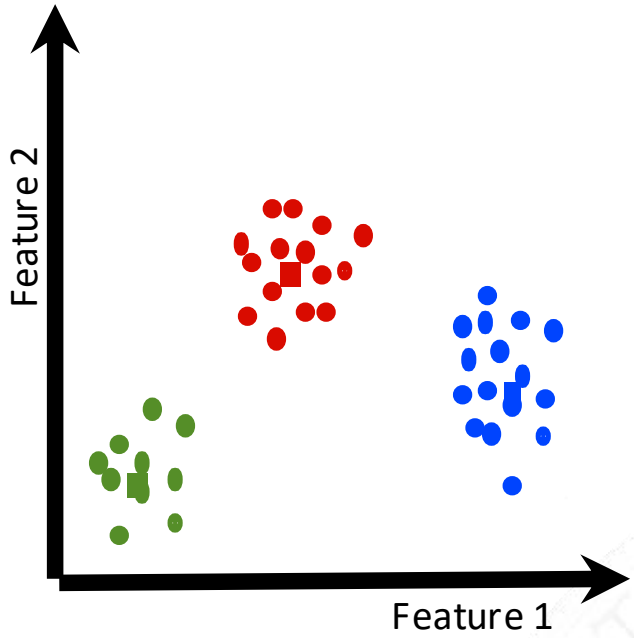
- Initialize K cluster centers
- **Repeat until convergence:**
  Assign each data point to the cluster with the closest center.
  Assign each cluster center to be the mean of its cluster's data points.



Animation from Databricks

K-means
Ejemplo

- Initialize K cluster centers
- Repeat until convergence:
  Assign each data point to the cluster with the closest center.
  Assign each cluster center to be the mean of its cluster's data points.

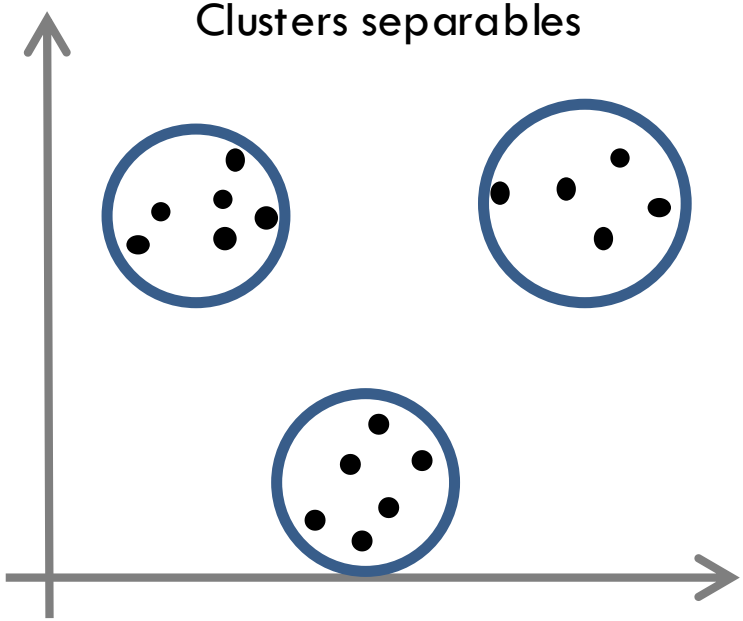


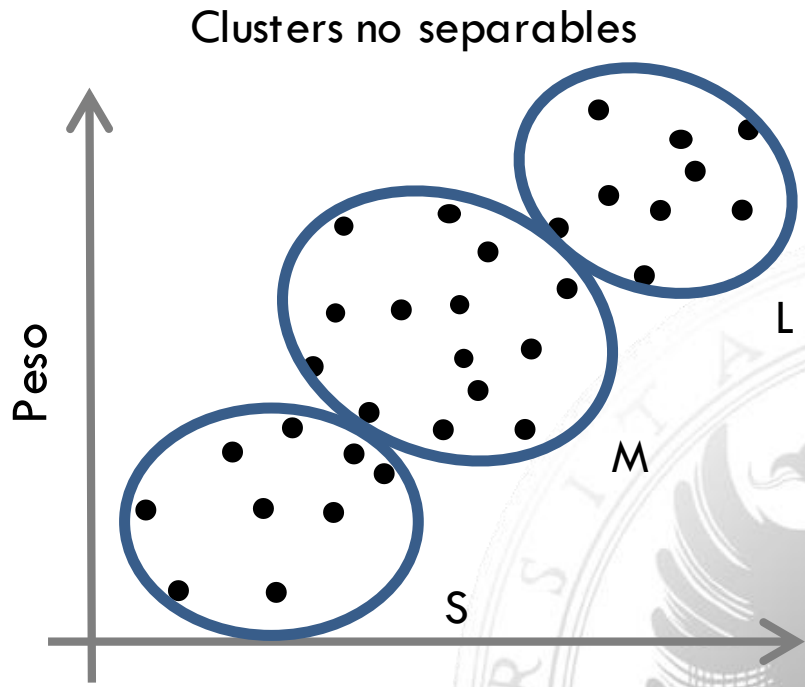
Animation from Databricks

Índice

K-means con clusters no separables

- *Aunque los clústeres no sean claramente separables*
  - K-means creará los grupos de ejemplos similares
- ***Ejemplo***
  - Crear grupos de **tallas para camisetas**





Función objetivo (de coste)

$c^{(i)}$ = índice del cluster (1,2,..., K) al que pertenece el ejemplo $x^{(i)}$ actualmente

μ_k = centroide del cluster k ($\mu_k \in \mathbb{R}^n$)

$\mu_{c^{(i)}}$ = centroide del cluster al que ha sido asignado el ejemplo $x^{(i)}$

Objetivo de la optimización:

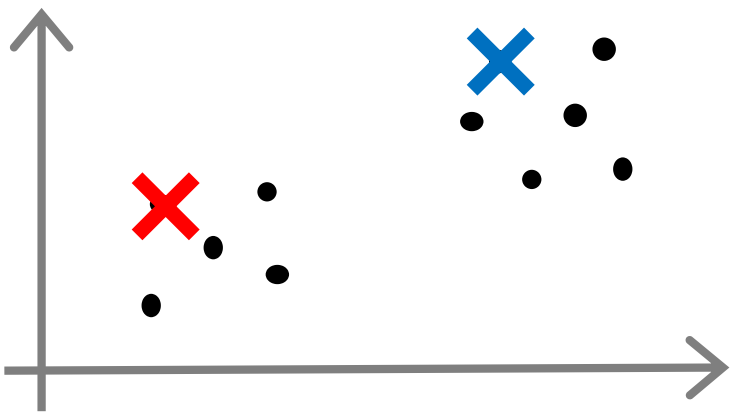
$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

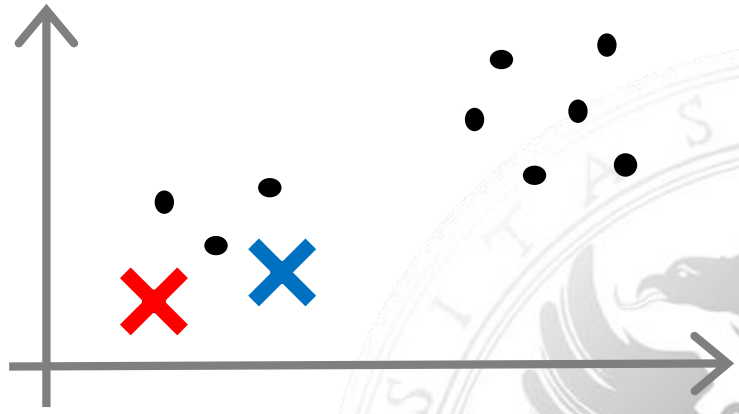
$$\min_{\substack{c^{(1)},\ldots,c^{(m)},\\ \mu_1,\ldots,\mu_K}} J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$

Minimizar la suma de las distancias entre cada ejemplo y el clúster al que pertenece

Inicialización aleatoria

- *Los centros de k-means se inicializan de manera aleatoria*
  - Lo más conveniente es **coger K ejemplos aleatorios como centroides** $(K < m)$

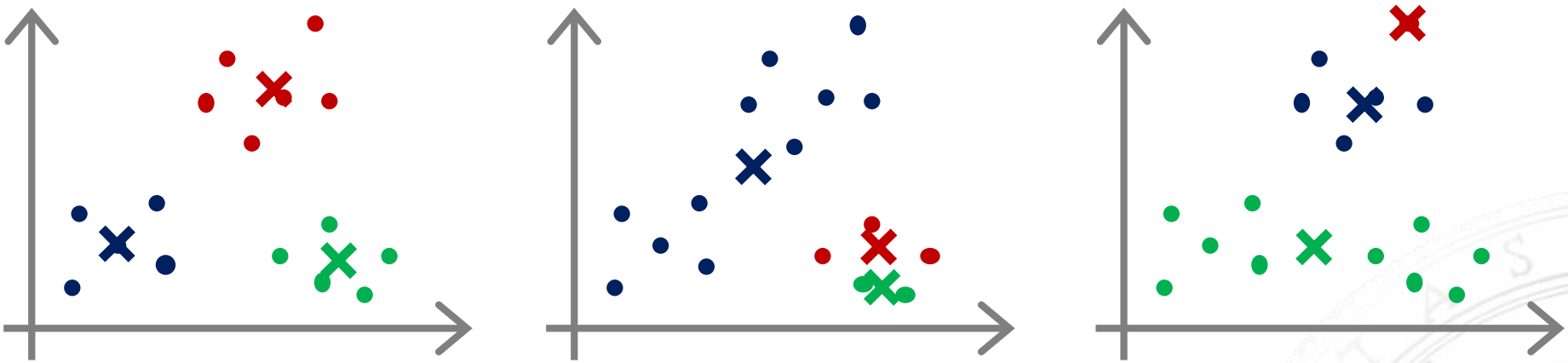




¡OJO! Podemos caer en mínimos locales

Mínimos locales

Según la inicialización podemos encontrar una u otra solución ***(mínimos locales)***



Posible solución: Repetimos k-means varias veces con diferente inicialización y nos quedamos con aquella que alcanza el menor coste

Inicialización aleatoria

For i = 1:100{

Inicializar aleatoriamente K-means

Ejecutar K-means y obtener

$c^{(1)}, \ldots, c^{(m)}, \mu_1, \mu_2, .., \mu_k$

Calcular $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \mu_2, .., \mu_k)$

}

Seleccionar el Clustering de menor coste

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \mu_2, .., \mu_k)$$

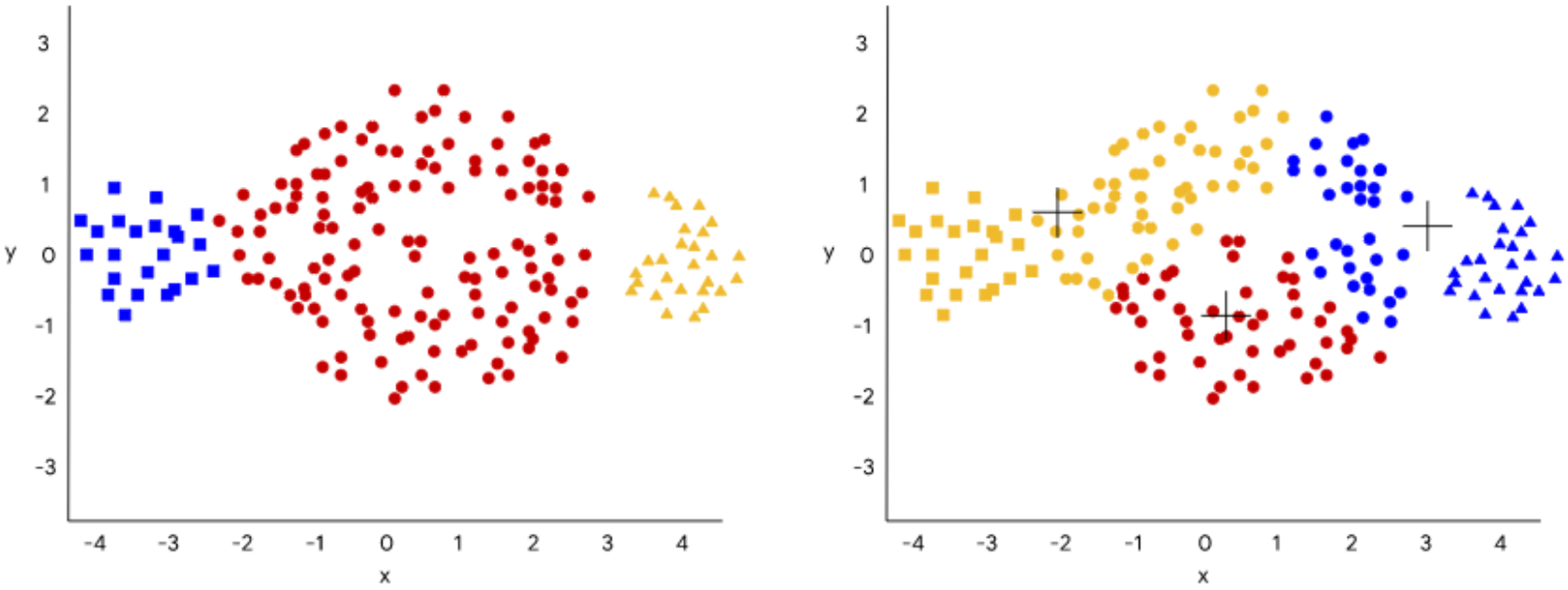
K-means: fortalezas y debilidades

- ***Fortalezas***:
  - Simple y detecta clústeres esféricos
  - Es un algoritmo relativamente eficiente O(k*m*iteraciones)

- ***Debilidades:***
  - Mínimos locales
  - ¿Cómo manejamos datos nominales?
  - ¿Determinar el valor de k?
  - Sensibilidad al ruido
  - Hay tipos de clusters que no podrán ser encontrados (p.ej non-convex)

K-means - densidades y tamaños

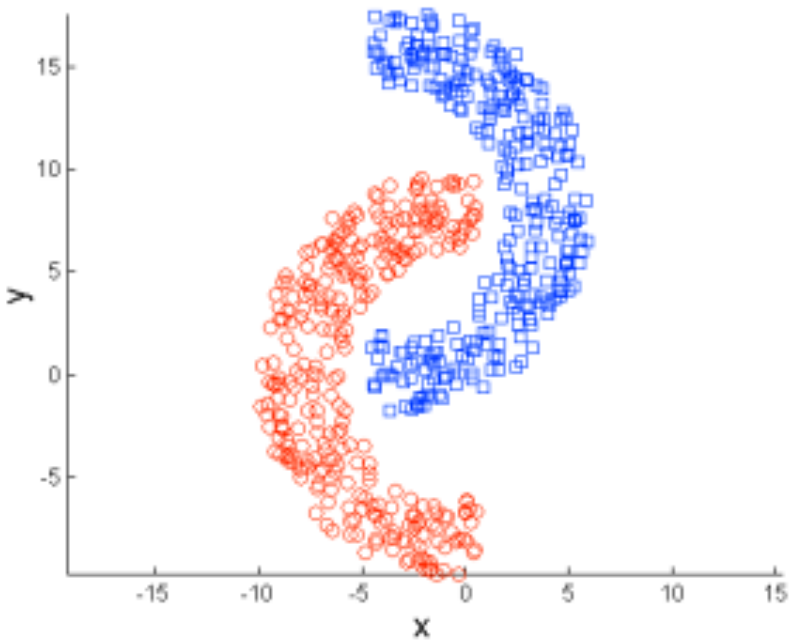
- El k−means puede encontrar problemas con diferentes densidades y tamaños

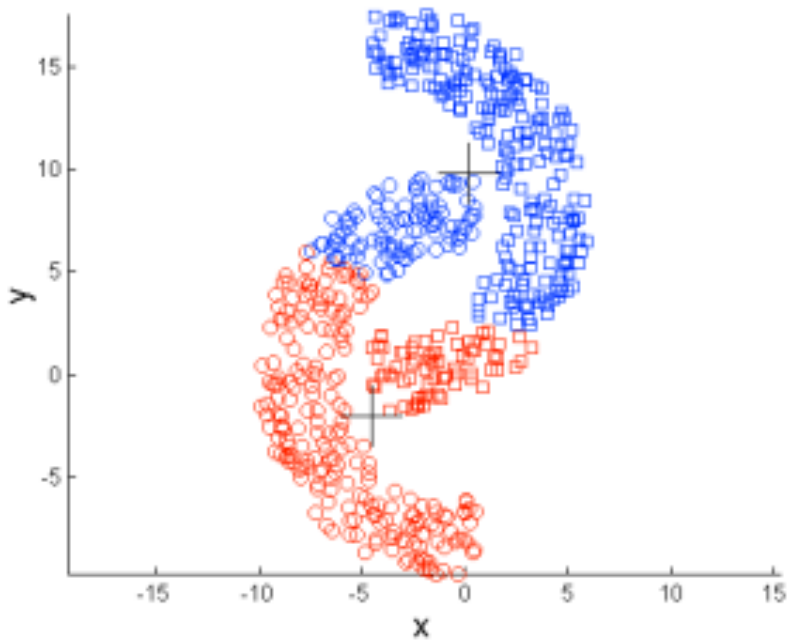


https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages

K-means - clusters no esféricos

- El k−means puede encontrar problemas con clusters no esféricos





Take-home message

- Clustering como técnica no supervisada para describir datos.
- El algoritmo k-means y cómo implementarlo
- Las debilidades del algoritmo del k-means