UA

cursos d'estiu | cursos de verano
rafael altamira

Introducción al *Machine Learning*

Aprendizaje no supervisado

Profesor: Antonio Javier Gallego

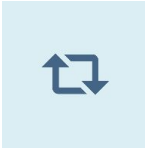
Contenidos

- Introducción al aprendizaje no supervisado
- Agrupamiento
- Análisis de componentes
- Aprendizaje de representaciones

Tipos de aprendizaje

- Aprendizaje supervisado

- Aprendizaje no supervisado



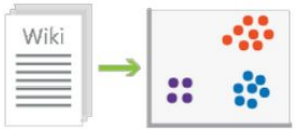
- Aprendizaje por refuerzo

Tipos de aprendizaje

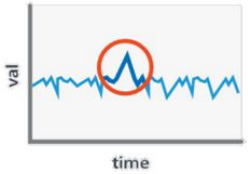
- **Aprendizaje no supervisado:**
  - El modelo se entrena con datos **no etiquetados**, lo cual significa que debe encontrar patrones y relaciones en los datos por sí mismo.
  - Ejemplos:
    - Agrupar documentos similares en función del texto que contienen.
    - Detectar detectar segmentos concretos de usuarios que acceden a una web en función de sus comportamientos e interacciones.

Tareas principales

- Agrupación (*clustering*).
- Detección de novedades o anomalías.
- Análisis de componentes.
- Aprendizaje de representaciones.



Clustering
(unsupervised – descriptive)

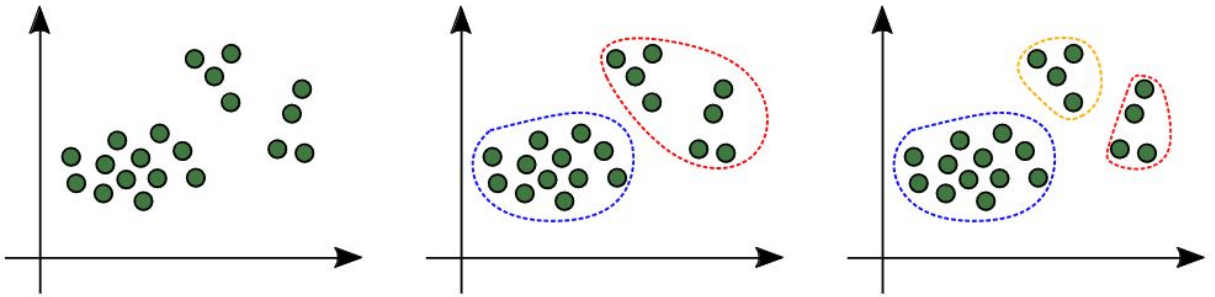


Anomaly Detection
(unsupervised – descriptive)

Agrupación

Agrupación

Tarea de agrupar las muestras o sus características en función de su similitud.



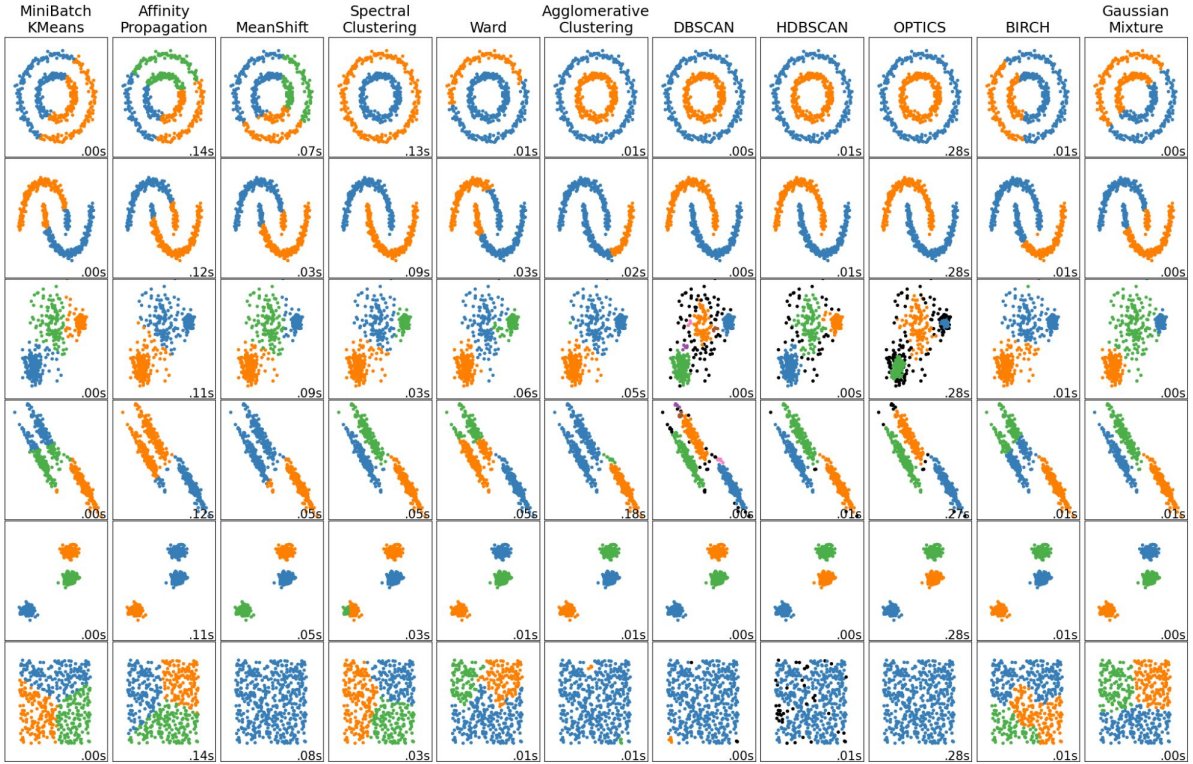
Algoritmos de *clustering*

- Encontramos numerosas propuestas para realizar esta tarea:
  - Basadas en conectividad o jerarquía.
  - Basadas en centroide.
    - *k-means*
  - Basadas en la distribución de los datos.
  - Basadas en la densidad.

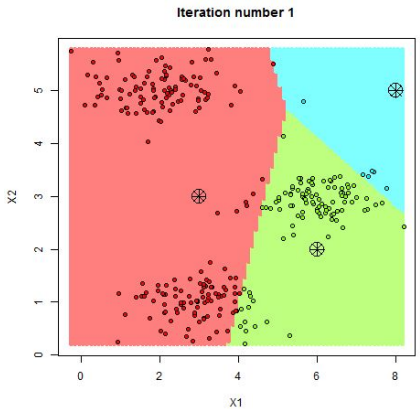
- Librería Sklearn:

  https://scikit-learn.org/stable/modules/clustering.html

Algoritmos de *clustering*

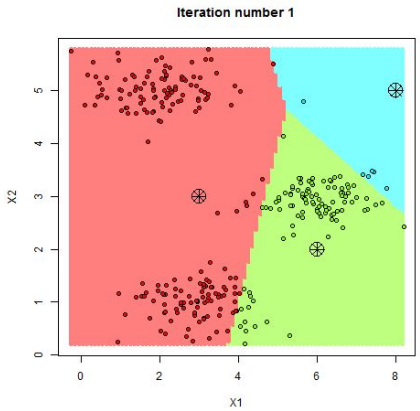


k-means



- Es uno de los algoritmos de *clustering* más conocidos.
- Se basa en la definición de **centroides** y en la agrupación según una **función de distancia**.
- El usuario tiene que establecer el número ***k*** de *clusters*.

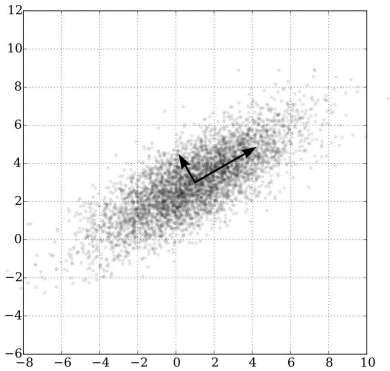
Pasos del algoritmo *k-means*



1. **Inicialización:** Se seleccionan aleatoriamente los *k* centroides iniciales.
2. **Asignación**: Cada muestra se asigna al centroide más cercano según la función de distancia.
3. **Actualización**: Se calculan nuevos centroides para cada *cluster* usando el promedio de las muestras.
4. **Reasignación**: Se repite el paso de asignación utilizando los nuevos centroides.
5. **Actualización y reasignación iterativa**: Los pasos 3 y 4 se repiten hasta que los centroides dejen de cambiar o se alcance el número máximo de iteraciones.

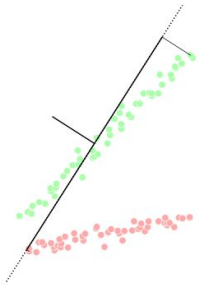
Análisis de componentes

Análisis de componentes



- Técnica no supervisada para reducir la dimensionalidad de un conjunto de datos.
- Uno de los algoritmos más utilizados es ***Principal Component Analysis*** **(PCA)**.
- PCA busca las direcciones principales a lo largo de las cuales los datos varían más.
- Convierte un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de variables sin correlación lineal llamadas **componentes principales**.

Pasos de algoritmo PCA



1. **Estandarización:** Si los datos no están en la misma escala, se estandarizan para que tengan media cero y varianza unitaria.
    - Esto asegura que todas las variables tengan la misma importancia en el análisis.
2. **Matriz de covarianza:** Se calcula esta matriz a partir de los datos estandarizados.
    - La covarianza mide cómo varían dos variables juntas.
3. **Cálculo de autovectores y autovalores:** Se calculan para la matriz de covarianza.
    - Los **autovectores** representan las direcciones principales de variación y los **autovalores** indican la cantidad de varianza explicada por cada autovector.
4. **Selección de componentes principales:** Se ordenan los autovectores según sus autovalores, de mayor a menor. Luego, se seleccionan los primeros *K* autovectores (donde K es el número de componentes principales deseadas) para formar una matriz de transformación.
    - La primera componente principal captura la mayor cantidad de variación en los datos
5. **Transformación de los datos:** Los datos originales se transforman utilizando la matriz de transformación formada por los autovectores seleccionados.
    - Esto proyecta los datos en el nuevo espacio definido por las componentes principales.

Ejemplos

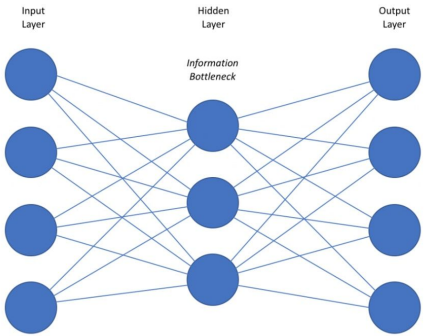
- La reducción de dimensionalidad lograda por PCA es útil en varias aplicaciones:
  - Visualización y análisis de datos en espacios de menor dimensión, lo que puede ayudar a identificar patrones y estructuras en los datos.
  - Eliminación de la correlación entre variables.
  - Compresión de datos.
  - Eliminación de ruido.

- Ejemplos:
  - Una análisis de 11 indicadores socioeconómicos de 96 países reveló que los resultados podrían explicarse a partir de solo dos componentes principales: el PIB y el índice de ruralidad.
  - https://scikit-learn.org/stable/auto_examples/decomposition/plot_faces_decomposition.html#sphx-glr-auto-examples-decomposition-plot-faces-decomposition-py

- Sklearn:
  https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

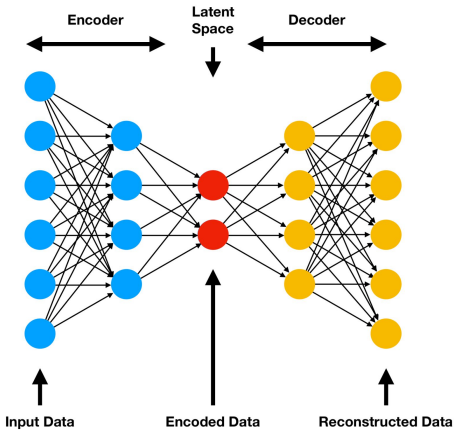
Aprendizaje de representaciones

Aprendizaje de representaciones



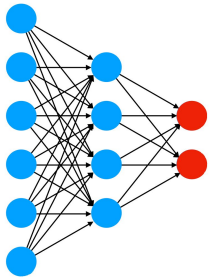
- Aproximación no supervisada para encontrar una representación adecuada, de tamaño reducido, para un conjunto de datos no etiquetado.
- Un ejemplo de este tipo de técnicas son los **Auto-Encoders**.

Auto-Encoder



- Arquitectura de red neuronal en la que se trata de reconstruir a la salida los datos recibidos como entrada.
- En esta arquitectura las capas intermedias tienen **un tamaño inferior** al de la entrada.
- Esta reducción fuerza a que la red tenga que extraer las **características más representativas** para posteriormente reconstruir la salida con el menor error posible.

Auto-Encoders



- Una vez entrenada, la parte del “decoder” se puede eliminar para así mapear directamente la entrada al nuevo espacio de representación.
- Si las activaciones son lineales se obtiene una reducción de dimensionalidad similar a PCA.
- Utilidades:
  - **(las mismas que PCA)**
  - Inicialización de métodos.
  - Aprendizaje auto-supervisado (coloreado, eliminar ruido, super-resolución…)

UA

cursos d'estiu | cursos de verano
rafael altamira

Introducción al *Machine Learning*

Aprendizaje no supervisado

Profesor: Antonio Javier Gallego