Clasificación

Machine Learning e Imágenes en Python

Clasificación No supervisada: Clustering

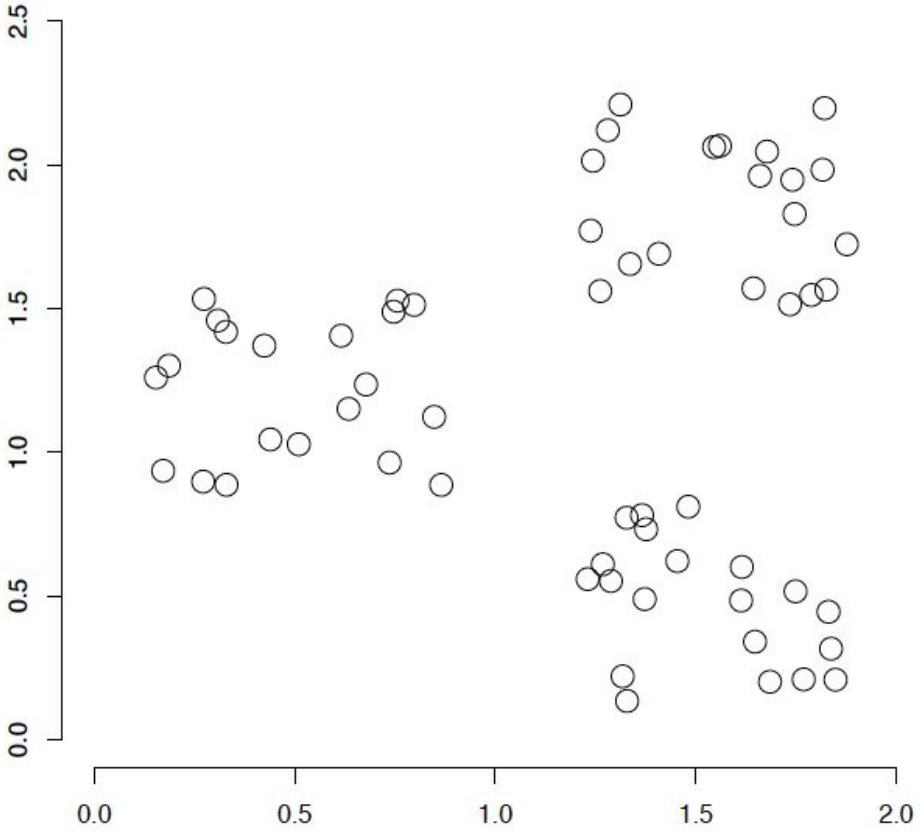
Notas basadas en notas de Laura Alonso Alemany y A. Georgina Flesia

Cómo funciona clustering

Agrupar objetos semejantes

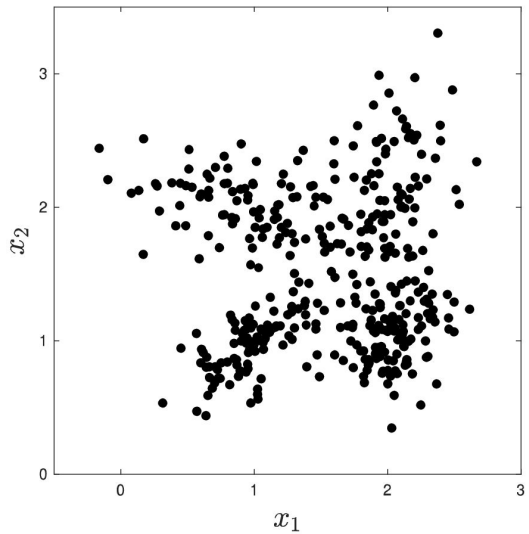
- Entrada: objetos en un espacio n-dimensional
- Salida: una **solución** con grupos (**clusters**) de objetos semejantes ➜ cercanos en el espacio
  - Se minimiza la distancia entre los objetos de un mismo grupo
  - Se maximiza la distancia entre los objetos de distintos clusters
- Los centros de cada cluster son los **centroides**

Dataset con clara estructura de clusters



¿Cómo sería un algoritmo para encontrar clusters en este espacio?

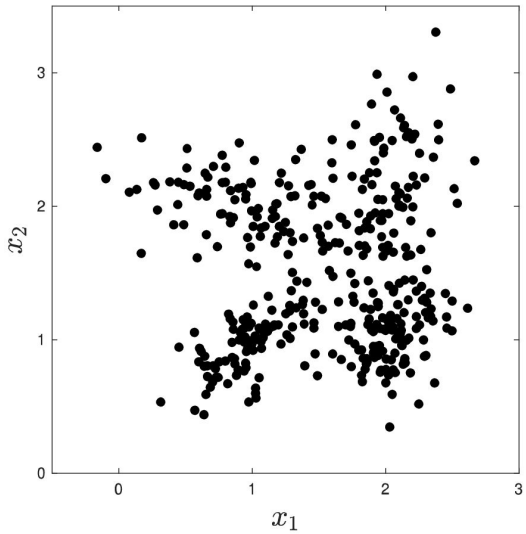
Dataset con no tan clara estructura de clusters



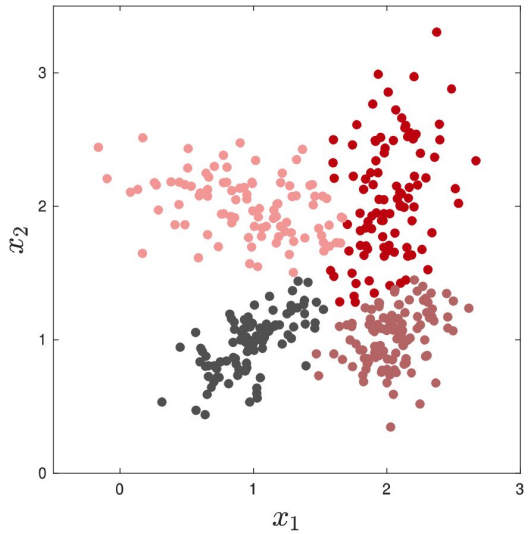
(a)

¿Cómo sería un algoritmo para encontrar clusters en este espacio?

Dataset con no tan clara estructura de clusters



(a)

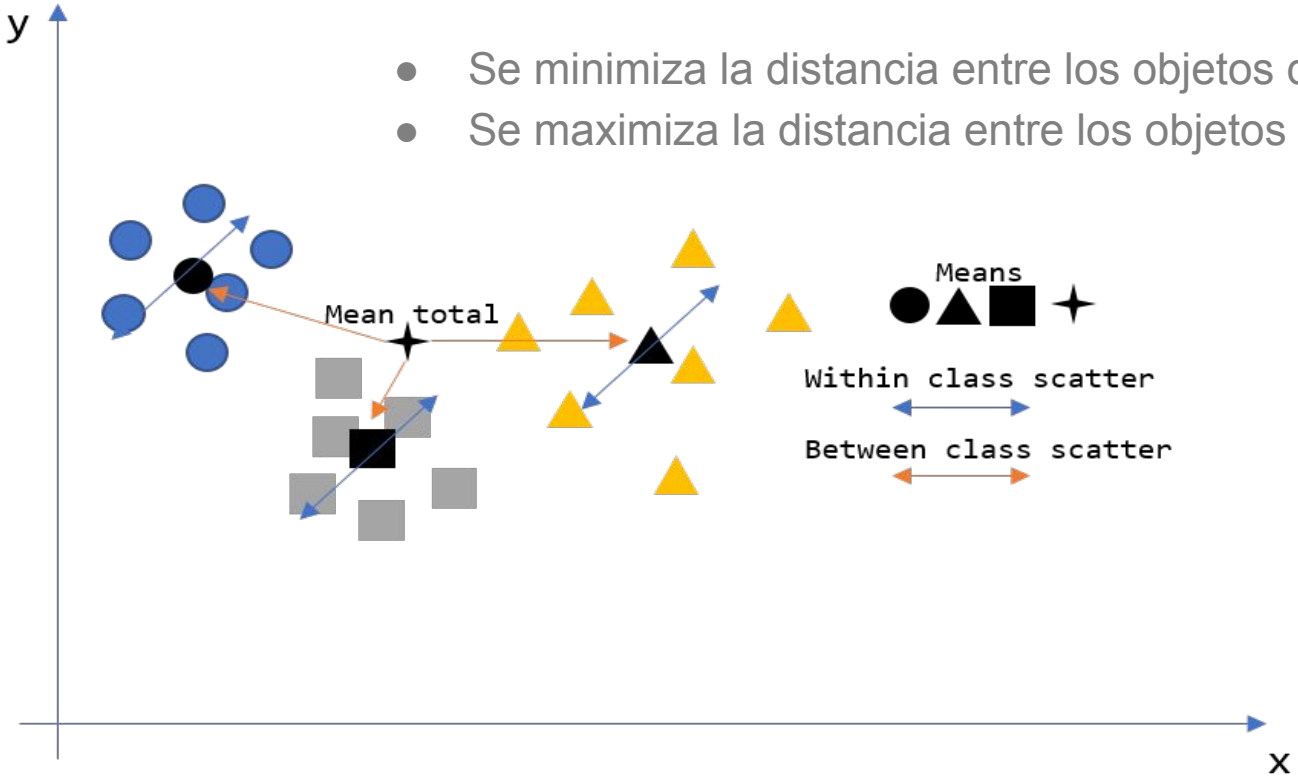


(b)

¿Cómo sería un algoritmo para encontrar clusters en este espacio?

Cómo funciona clustering

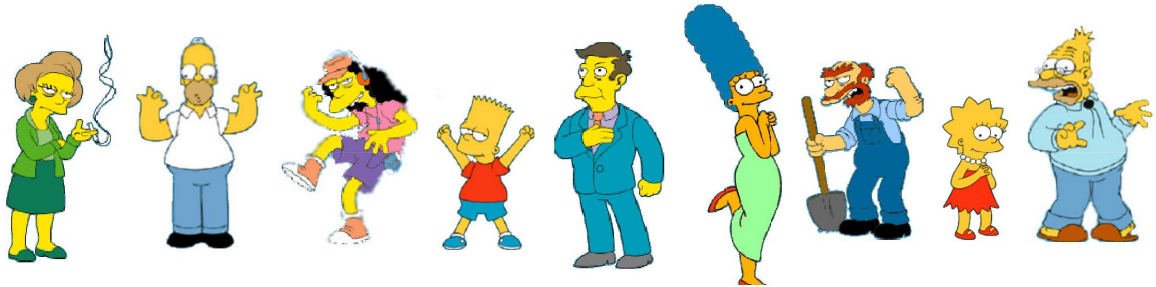
- Se minimiza la distancia entre los objetos de un mismo grupo
- Se maximiza la distancia entre los objetos de distintos clusters¶



Cuestiones cruciales

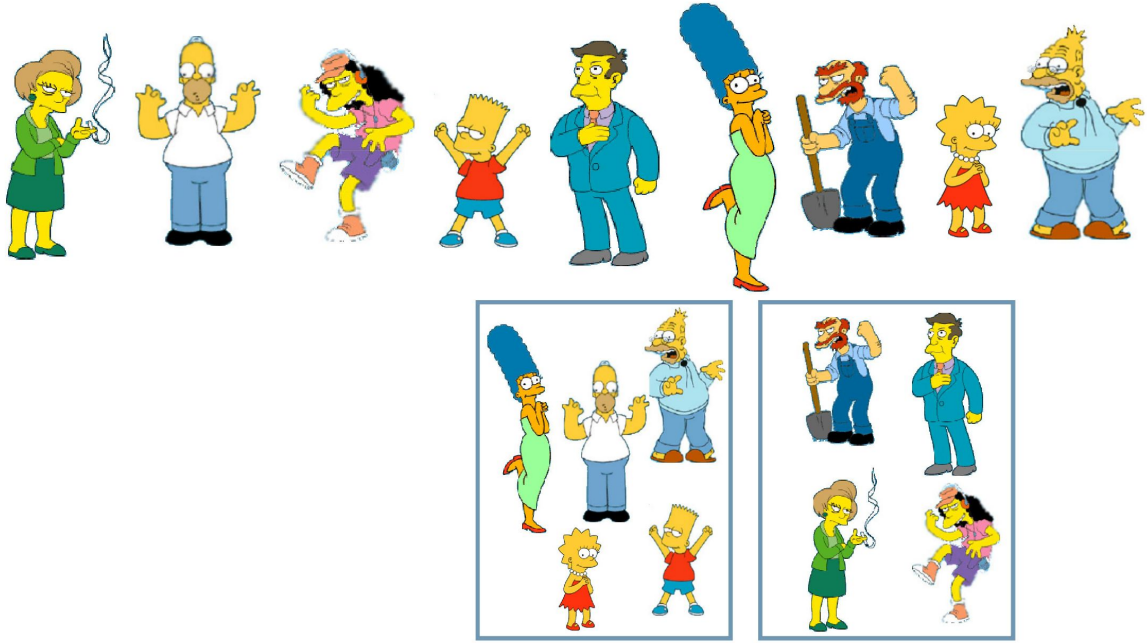
- ¿Cómo es el espacio? ¿Cómo represento mis problemas?
- ¿Cómo se calcula la distancia (semejanza) en este espacio?
- ¿Cuántos clusters quiero distinguir?
- ¿Qué distribución tienen estos clusters?
- ¿Cómo veo qué hay en cada cluster?
- ¿Cómo evalúo la bondad de cada solución?

Datos



¿Cómo los agrupo?

Datos agrupados según algún criterio



Escalado, Estandarización o Normalización

- Atributos continuos

  - Para evitar que unas variables dominen sobre otras los valores de los atributos se escalan, estandarizan o normalizan a priori
  - `sklearn.preprocessing`: Preprocessing and Normalization
  -
    - StandarScaler (z-score), cada columna de media 0 y varianza 1
    - MinMaxScaler
    - normalize (por defecto por fila, c/vector de norma 1, unitario)

Distancias: datos continuos

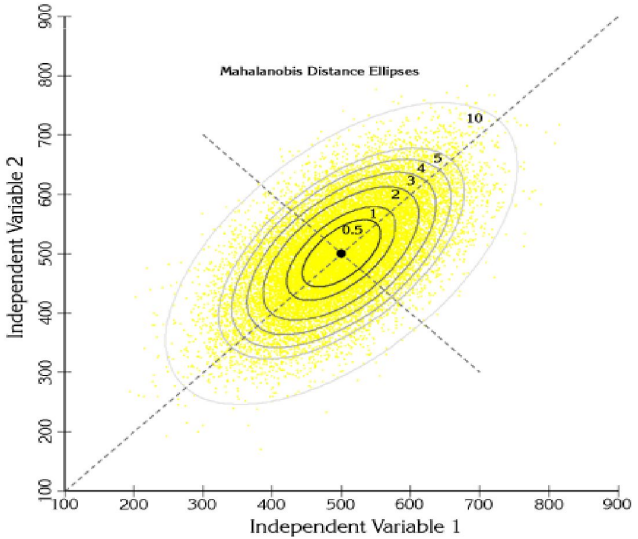
- Euclídea
- Distancia de Manhattan
- Distancia de mahalanobis
- “distancia” del Coseno ➔ primero normalizado por longitud de cada vector/fila,
- Similitud del coseno: considera el producto punto entre vectores, es alta cuando están alineados

Distancias: datos continuos

Distancia de Mahalanobis

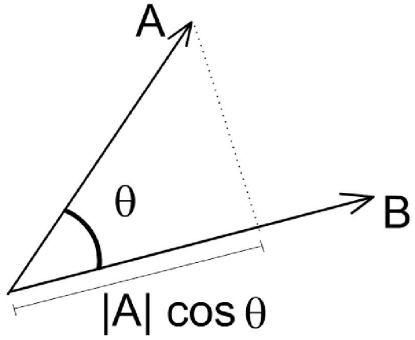
$$d(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T \Sigma^{-1} (\vec{x}-\vec{y})}.$$

- Considera las correlaciones entre variables.
- No depende de la escala de medida.



Similaridades

Medidas de correlación



- Producto escalar

$$S.(x, y) = x \cdot y = \sum_{j=1}^{J} x_j y_j$$

- "Cosine similarity"

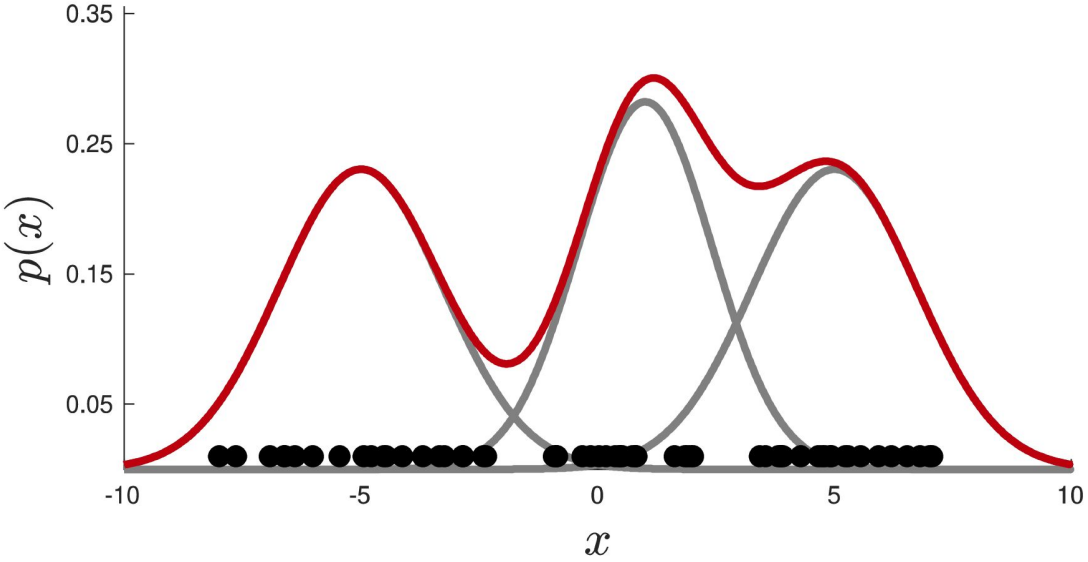
$$\cos(\vec{x}, \vec{y}) = \sum_i \frac{x_i \cdot y_i}{\sqrt{\sum_i x_i^2} \cdot \sqrt{\sum_i y_i^2}}$$

- Coeficiente de Tanimoto

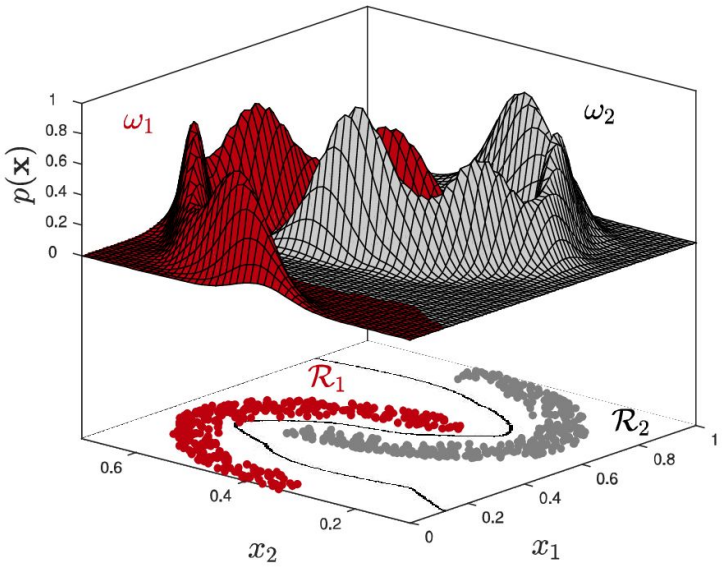
$$s(\vec{X}, \vec{Y}) = \frac{\vec{X}^t \cdot \vec{Y}}{\vec{X}^t \cdot \vec{X} + \vec{Y}^t \cdot \vec{Y} - \vec{X}^t \cdot \vec{Y}},$$

Mezcla de Gaussianas

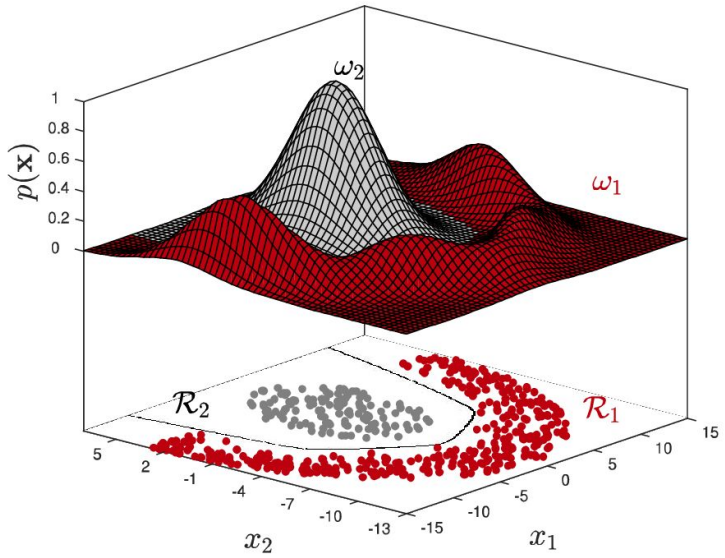
- Supongamos tener alguna información
  - Consideremos que estos datos son reales,
  - puedo trabajar con la distancia Euclídea.
  - datos producidos por una densidad mezcla de Gaussianas,



Mezcla de Gaussianas



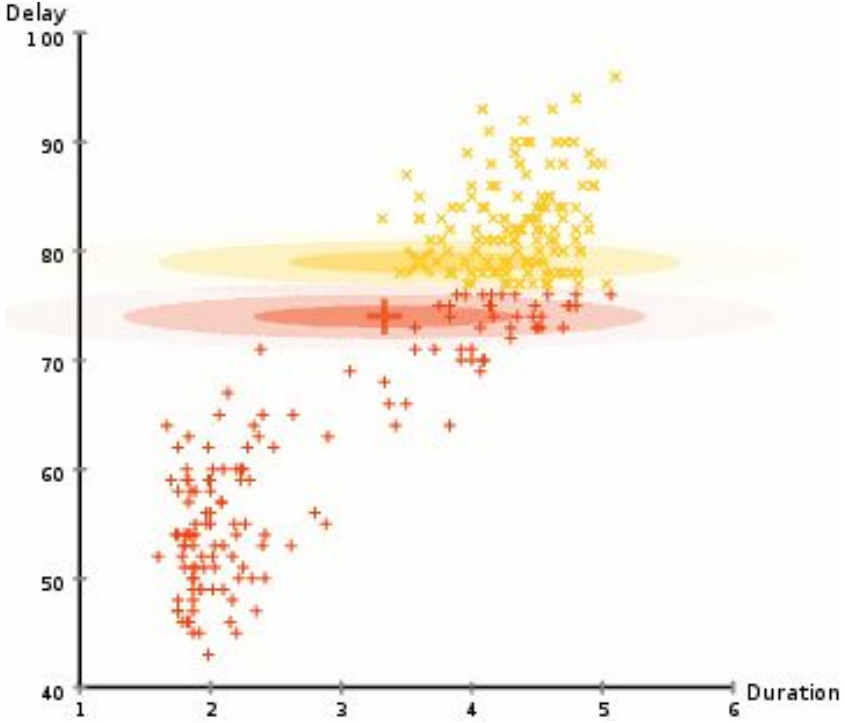
(a)



(b)

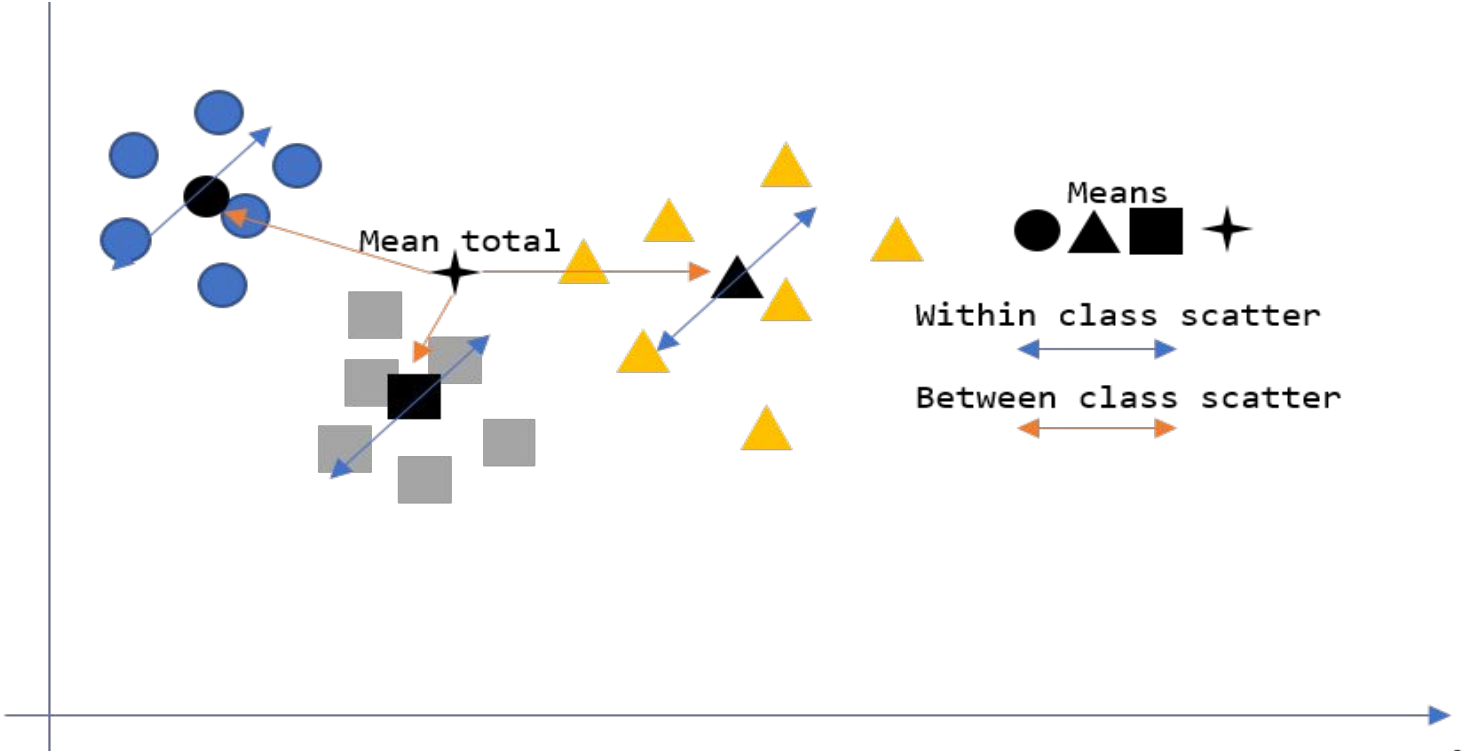
Como funciona el GMM?

- Comenzamos con una partición aleatoria de la cual se sacan los parámetros de inicio y desde allí se itera

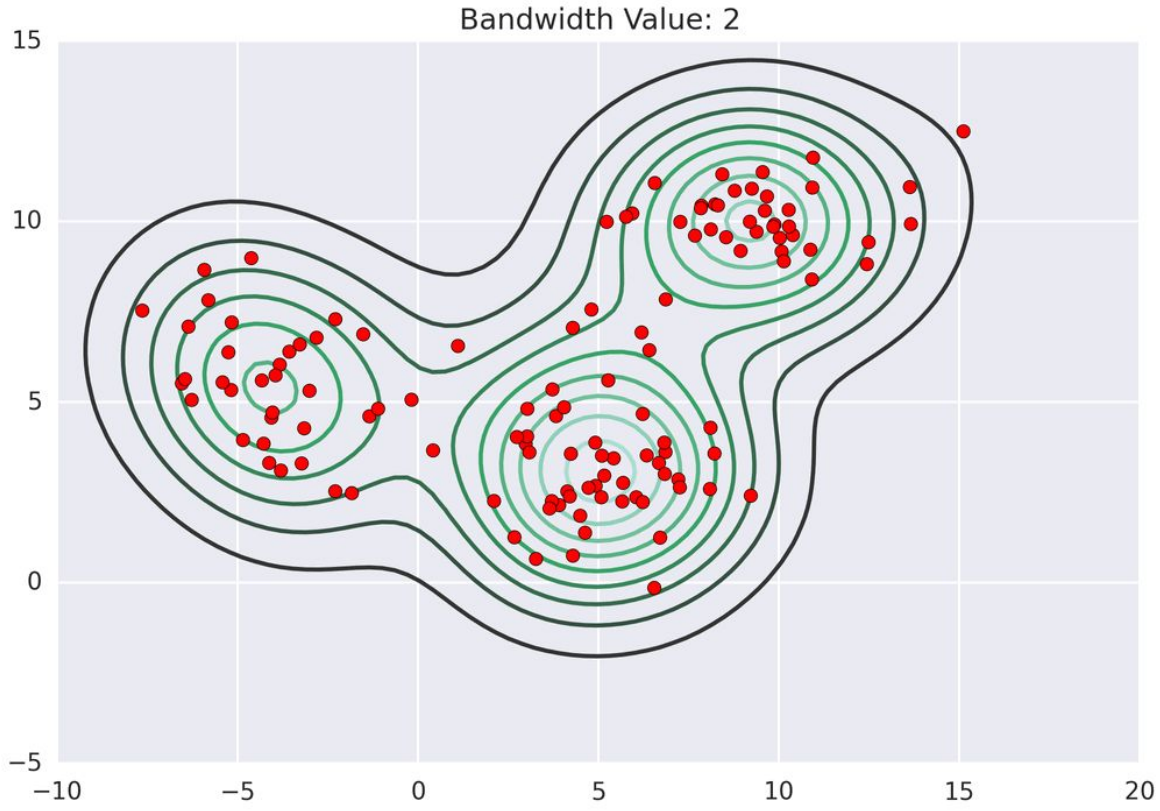


K-means

- Particiona usando distancia, sin pensar en densidades ni distribuciones de probabilidad.



Mean Shift Algorithm



Mean Shift Algorithm

Mean Shift Algorithm

- Parámetro bandwidth puede ser fijado a priori.
  - ➢
- No tiene sentido usar BIC o AIC pues no se está fijando el modelo paramétrico
  - ➢
- Puede ser estimado utilizando la teoría no paramétrica, dependiendo de que kernel se use.
  - ➢
- Note_fig4.ipynb tiene un ejemplo de Mean Shift automático y con k fijo.

Dbscan

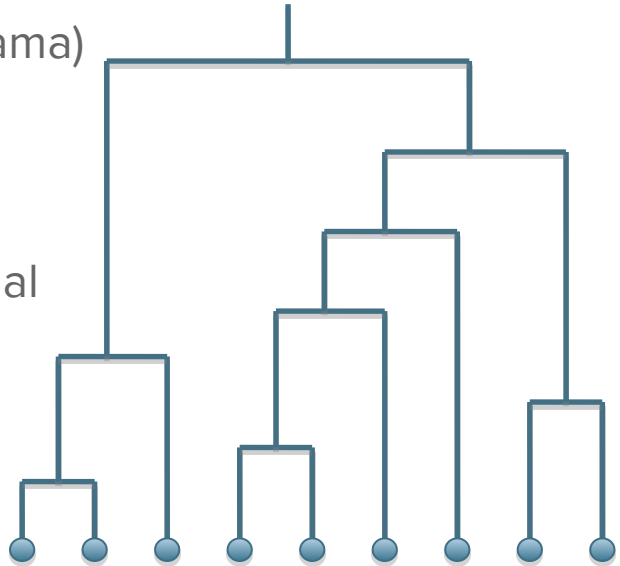
Clustering jerárquico

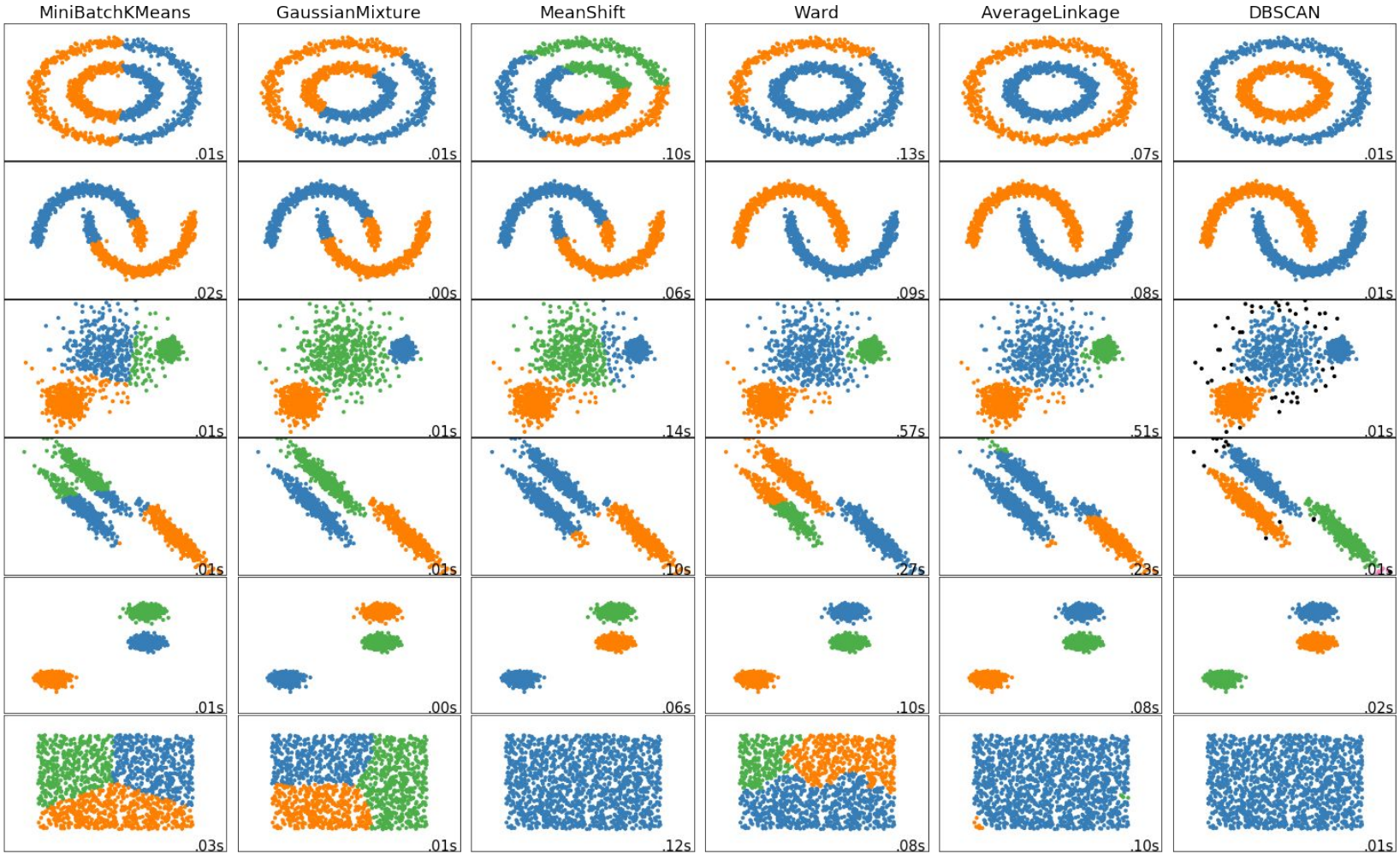
Si no queremos especificar k...

Algoritmos jerárquicos que generan una

taxonomía jerárquica de clusters (dendrograma)

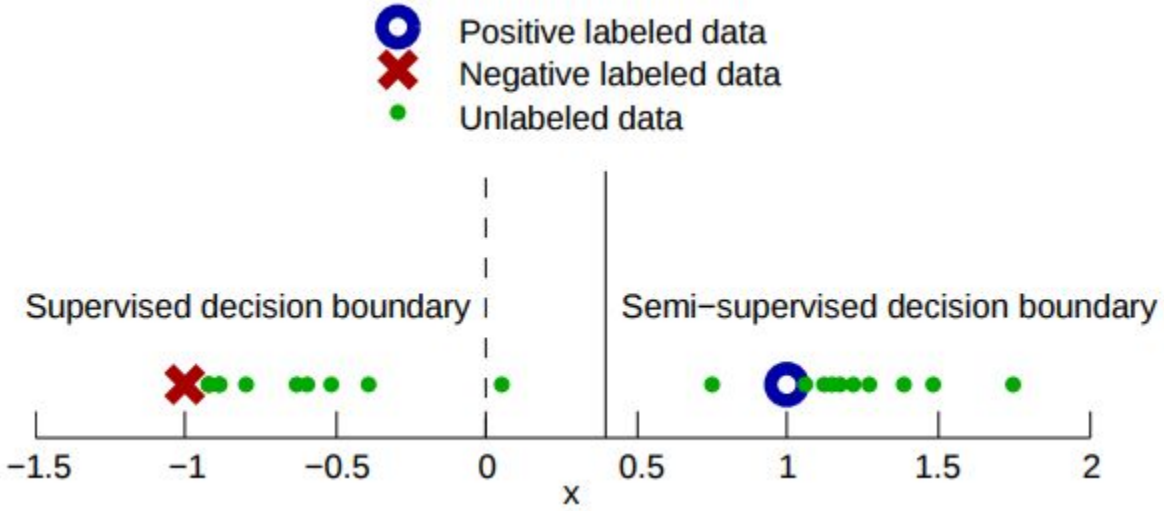
- Interpretación más rica
- Más difícil de interpretar
- El corte del árbol tiene que ser ortogonal




MiniBatchKMeans
GaussianMixture
MeanShift
Ward
AverageLinkage
DBSCAN
.01s
.01s
.10s
.13s
.07s
.01s
.02s
.00s
.06s
.09s
.08s
.01s
.01s
.01s
.14s
.57s
.51s
.01s
.01s
.01s
.10s
.27s
.23s
.01s
.01s
.00s
.06s
.10s
.08s
.02s
.03s
.01s
.12s
.08s
.10s
.01s

Semi-supervisado

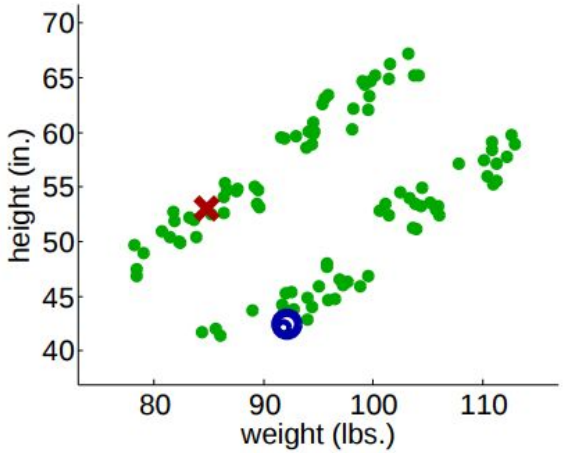
Semi Supervisado



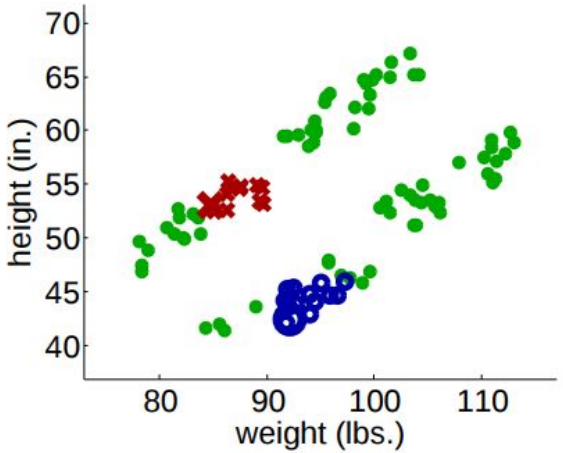
Algoritmo de autoaprendizaje

1. Obtener un conjunto pequeño de datos etiquetados
2. Aprender un clasificador de los datos etiquetados
3. Aplicar el clasificador sobre datos no etiquetados
4. Incorporar datos etiquetados automáticamente al conjunto de entrenamiento
5. Volver a 2.

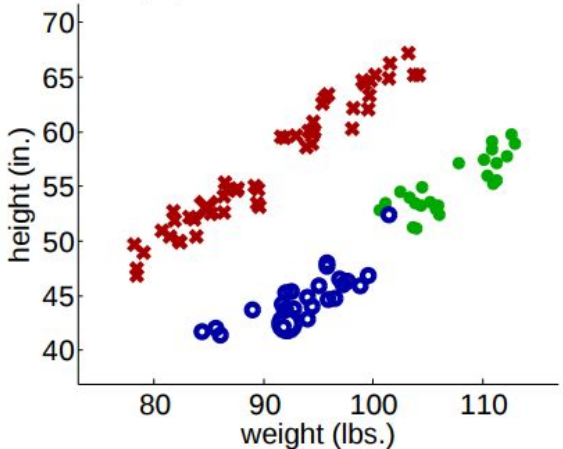
- ¿Qué ejemplos etiquetados automáticamente incorporamos?
  - Mayor confianza
  - Los n mejores
  - Todos

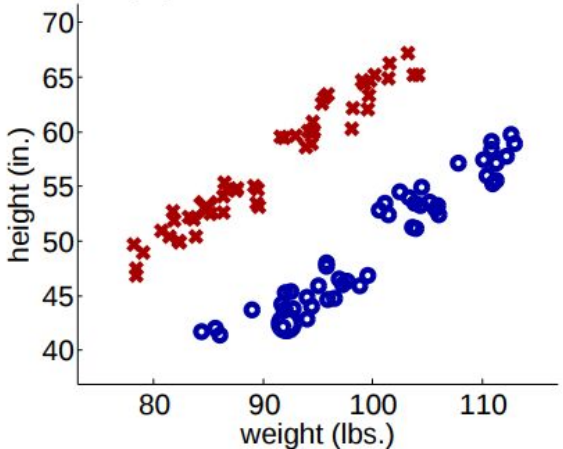


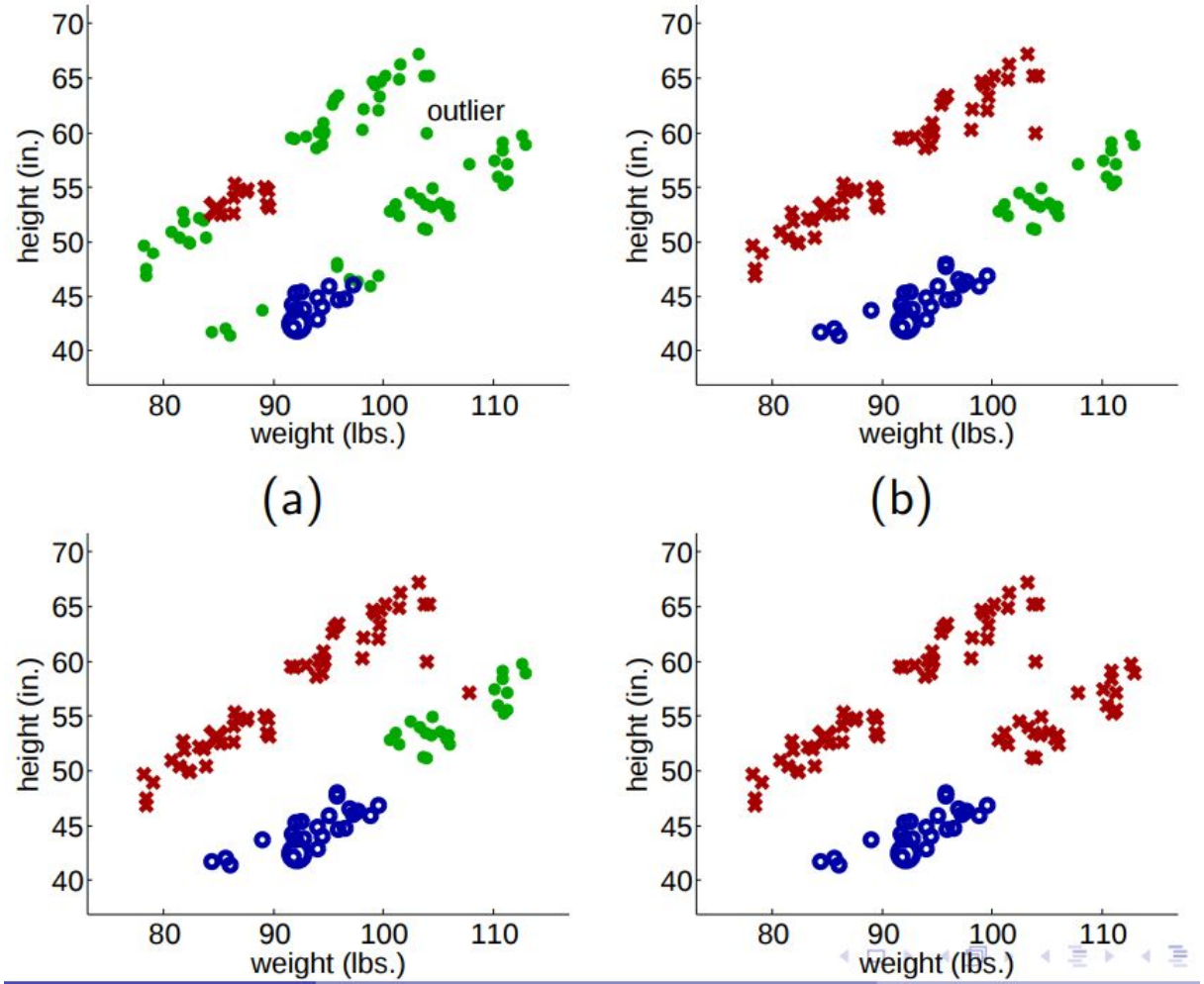
(a) Iteration 1



(b) Iteration 25






outlier
height (in.)
weight (lbs.)
70
65
60
55
50
45
40
80
90
100
110
(a)
(b)