# Clustering

## Aprendizaje no-supervisado

Hugo Franco

Universidad Central - Ingeniería de Sistemas

9 de noviembre de 2022

UNIVERSIDAD
CENTRAL

# Repaso

# Definición de aprendizaje automático (*machine learning*)

## Samuel, A.L. (1959), primeras exploraciones

«El aprendizaje automático es el campo de estudio que le da al computador la capacidad de aprender **sin ser programado explícitamente**.»

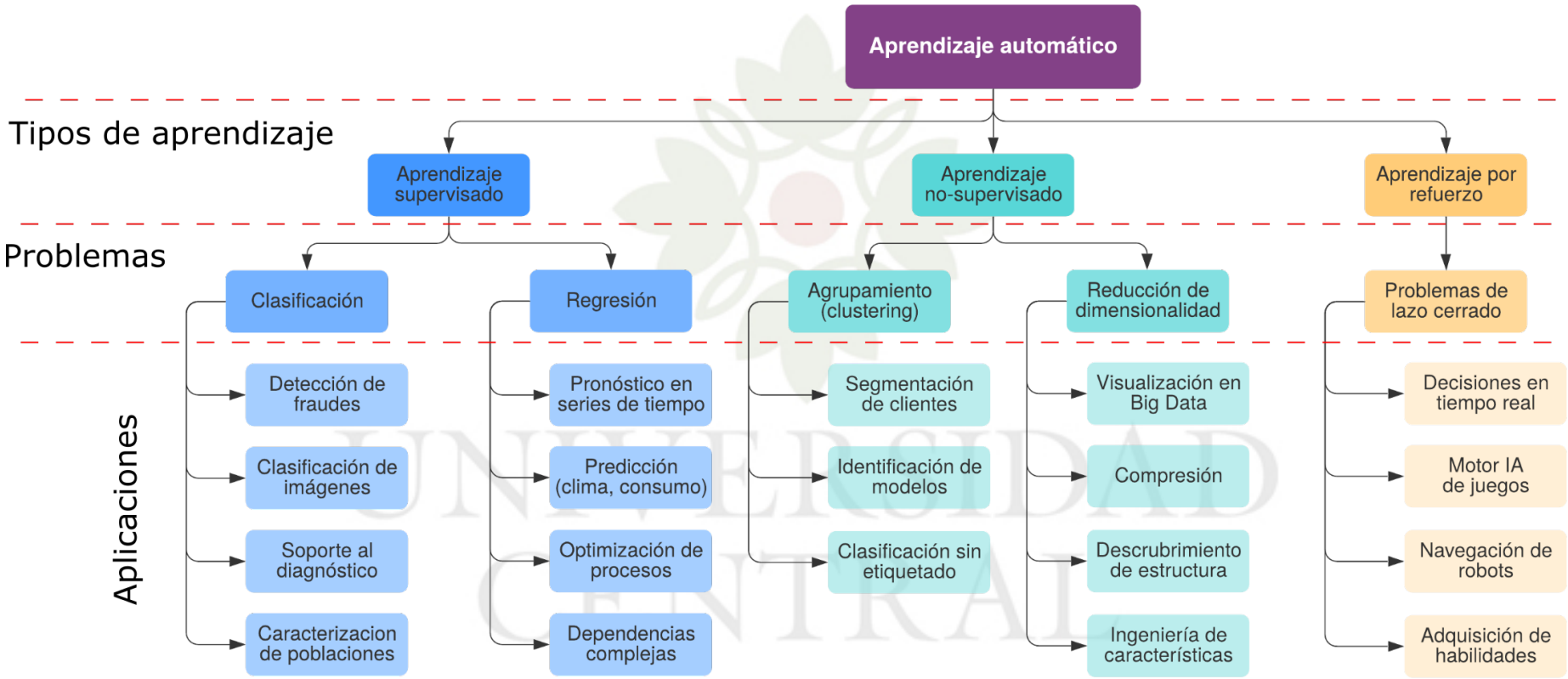
## Mitchell, T.M. y otros (1998), aproximación desde IA

«Se dice que un programa de computador aprende de la experiencia $E$ con respecto a alguna clase de tareas $T$ y una medida de rendimiento $P$ si su desempeño en las tareas $T$, medido por $P$, **mejora con la experiencia** $E$.»

## Murphy (2012), aproximación probabilística

«[...] un conjunto de métodos que pueden **detectar patrones en los datos automáticamente**, y luego usar los patrones descubiertos para predecir datos futuros, o para llevar a cabo otro tipo de **toma de decisiones** bajo incertidumbre [...]»

# Tipos de Aprendizaje Automático

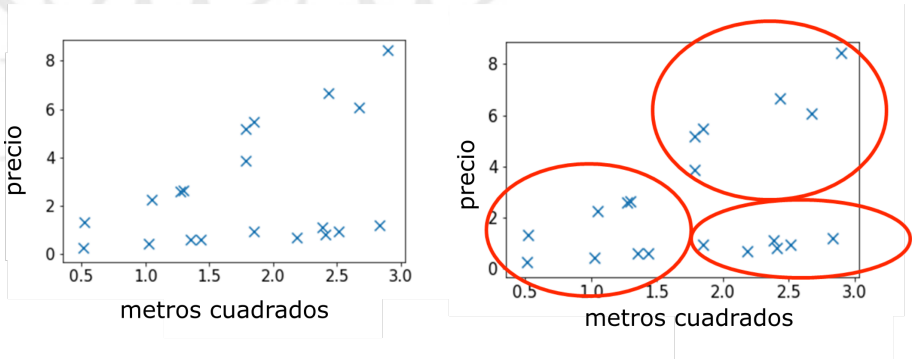


# Aprendizaje no-supervisado

- En aprendizaje no supervisado, el dataset consiste en un conjunto $\{\mathbf{x}_1 \ldots, \mathbf{x}_n\}$ con $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)} ..., x_i^{(m)})$, dadas $n$ muestras y $m$ características.
- Objetivo: encontrar estructuras relevantes en los datos partiendo exclusivamente de sus valores
- Se pueden encontrar patrones en los datos usando estrategias como la minimización distancia entre elementos de cada grupo (cluster) o la distancia de cada punto a un conjunto finito de $k$ elementos representativos

- Los modelos de distancia o relación entre elementos pueden variar (euclidea, L1 o «Manhattan», modelos basados en diferencias de gaussianas)

Ej. El dataset no contiene etiquetas. Encontrar estructuras (relaciones internas) en los datos es un problema de aprendizaje no-supervisado



# *Aprendizaje no-supervisado: Clustering* (agrupamiento)

# Métodos y técnicas de clustering

Aunque existe una gran variedad de métodos de agrupamiento (incluso algunas técnicas más recientes basadas en *kernels* o teoría de grafos), algunos de uso más frecuente son los siguientes:

- **Jerárquico**
  - Top-Down (divisivo)
  - Bottom-Up (aglomerativo)
- **Basado en Centroides**
  - **K-means**
  - K-medians
  - K-medoids
- Basado en densidades
  - Mean shift
  - **DBSCAN**
  - OPTICS
- Basado en modelos
  - **Gaussian Mixture models (probabilístico)**
  - COWEB (basado en heurísticas para crear árboles de categorización)

# Métodos jerárquicos

# Métodos jerárquicos

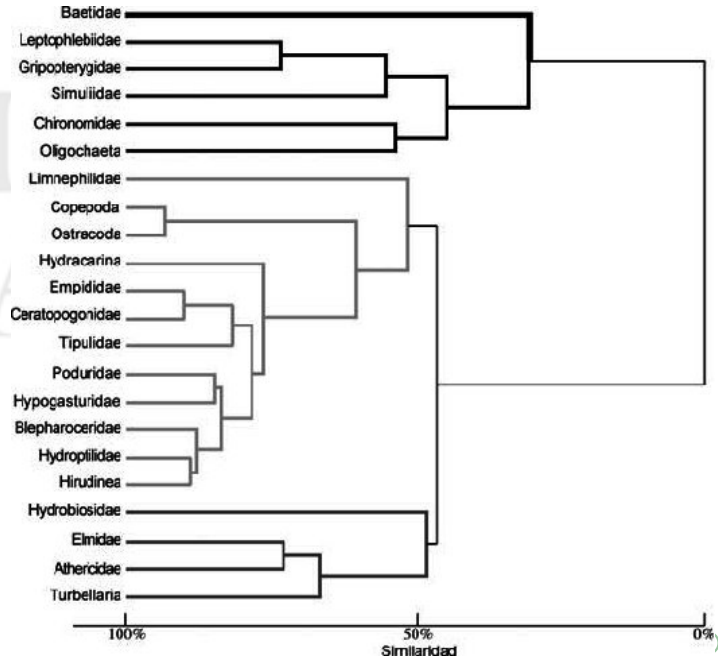
## Top-Down (divisivo):

Comienza por regiones (clusters) más gruesas y luego va afinando la partición (aumentando el número de clusters), subdividiendo cada cluster grueso en otros más finos allí donde se encuentra una distancia suficientemente grande entre grupos

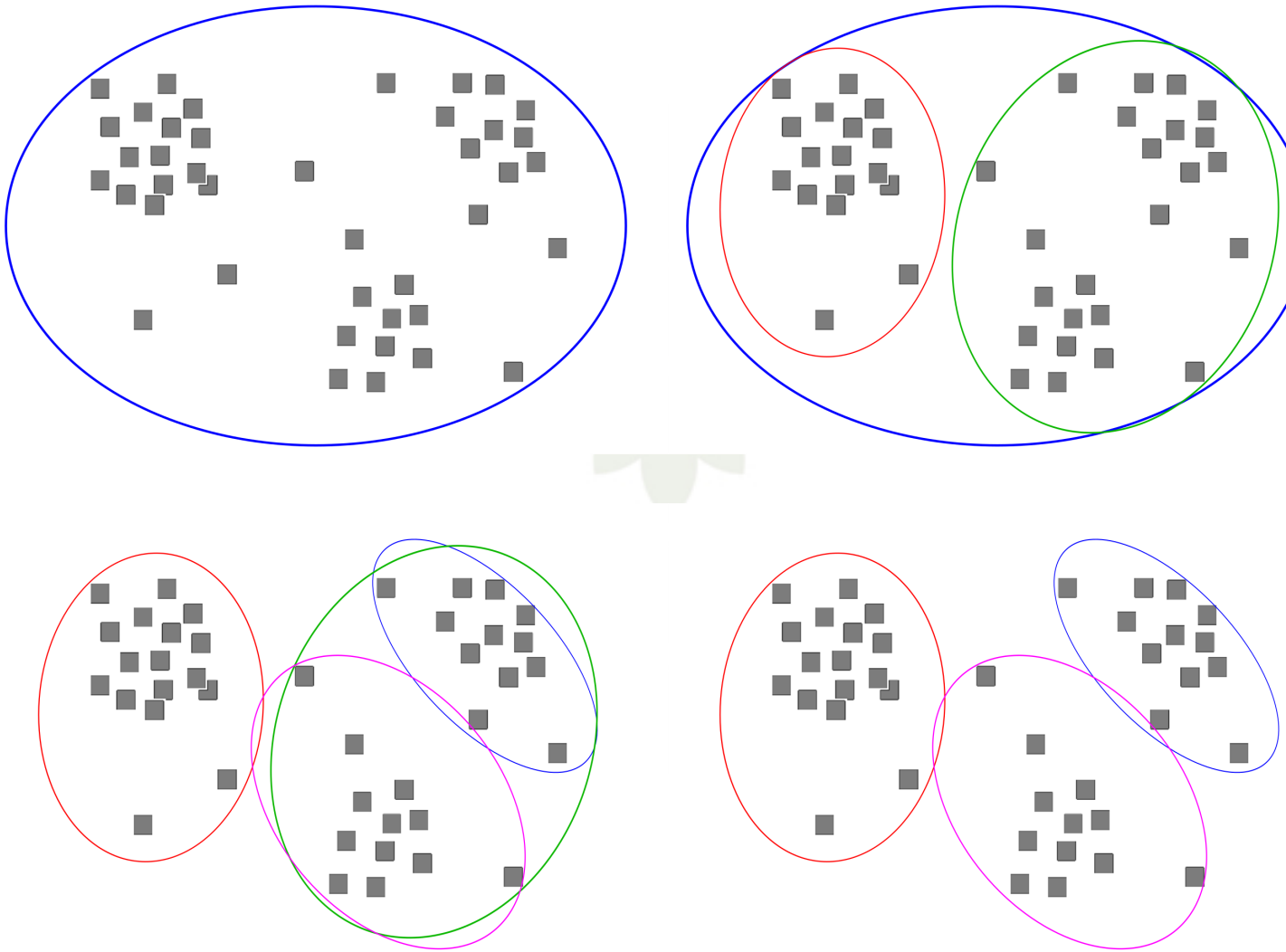
## Bottom-Up (aglomerativo):

Parte de todos los elementos del dataset como grupos independientes y luego va reduciendo el número de grupos por asociaciones entre elementos y grupos. Permite controlar el número de clusters estableciendo un punto de corte en el dendrogramaBottom-Up (aglomerativo):

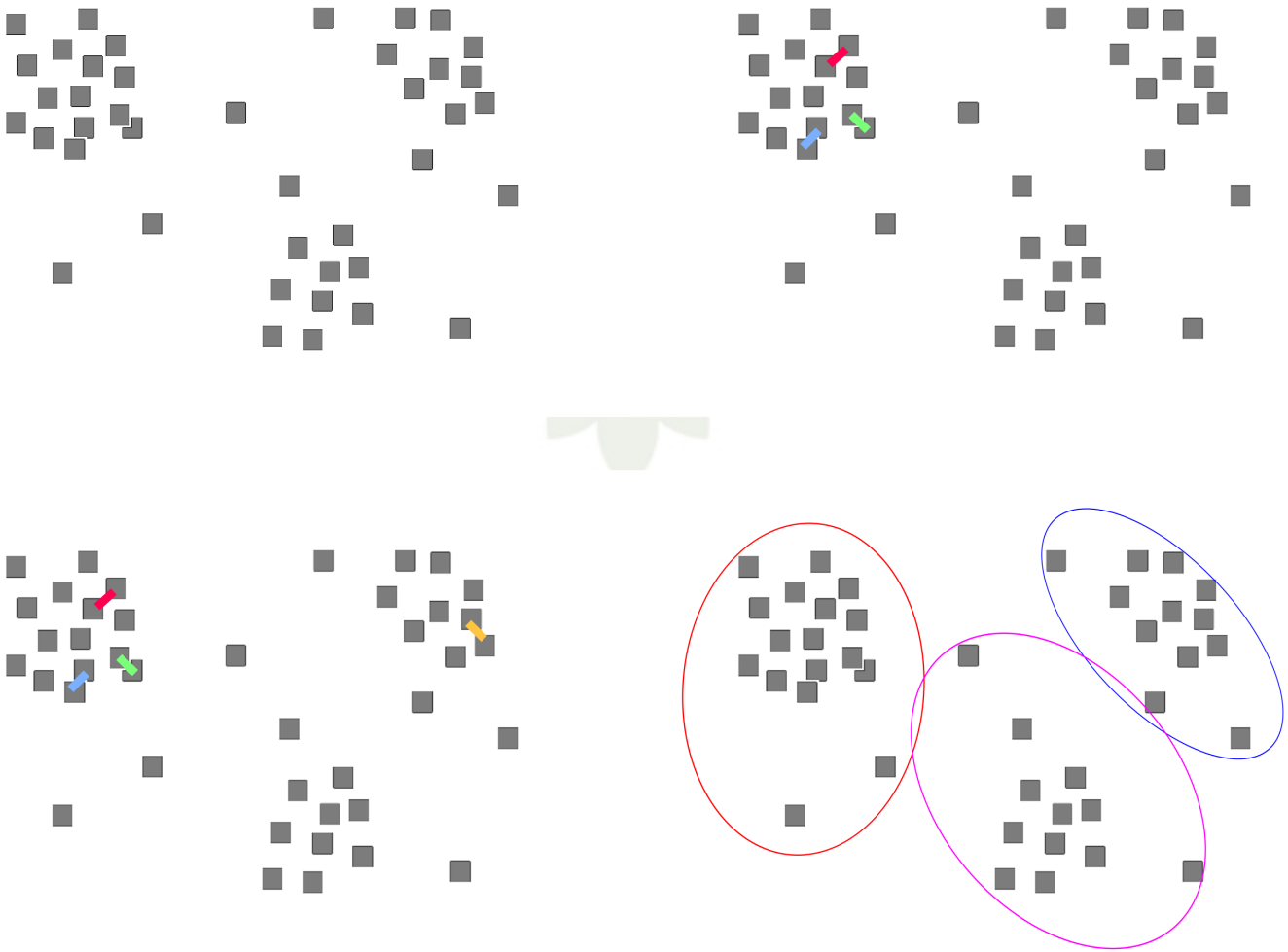
**Dendrograma:** el punto de corte establece el número de clusters y determina los elementos de cada cluster según las asociaciones determinadas. *Ejemplo: distribución geográfica de invertebrados en Bolivia (Molina et al., 2008)*



# Ejemplo: Top-Down (divisivo)



# Ejemplo: Bottom-Up (aglomerativo)



# Métodos basados en centroides

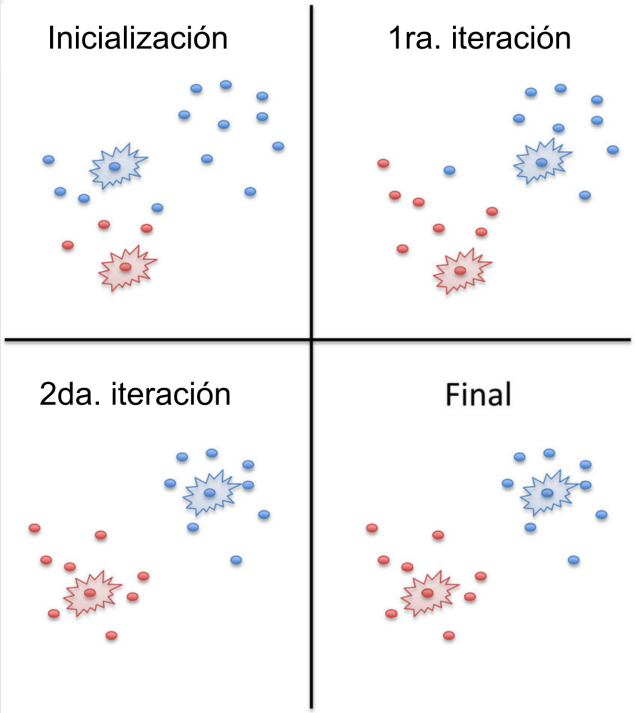
# Agrupamiento basado en «centroides»

- ***K-means***: cada elemento del dataset se agrupa en el clúster cuyo *centroide* se encuentra más cerca al elemento en el espacio de características. El parámetro K indica el número de clústers (y por lo tanto, el número de centroides) se emplean en el agrupamiento
  - *Centroide*: posición en el espacio de características en el cual cada una de sus componentes corresponde al valor medio de dicha componente calculado sobre los elementos asignados al clúster correspondiente (luego no es parte del dataset)
- ***K-medians***: variación de K-means en la que el elemento representativo se determina asignando a cada una de sus componentes el valor de la mediana de dicha componente sobre los elementos asignados al clúster (tampoco pertenece al dataset)
- ***K-medoids***: en vez de escogerse un valor arbitrario para las componentes del elemento representativo de cada *clúster* , se escoge el elemento dentro del dataset más cercano a todos los elementos de su clúster
- En datasets que incluyen características tanto continuas como categóricas, se pueden usar algoritmos como *centroides difusos, K-modes* o *ROCK (Robust Clustering Algorithm for Categorical attributes)*

# Algoritmo K-medoids

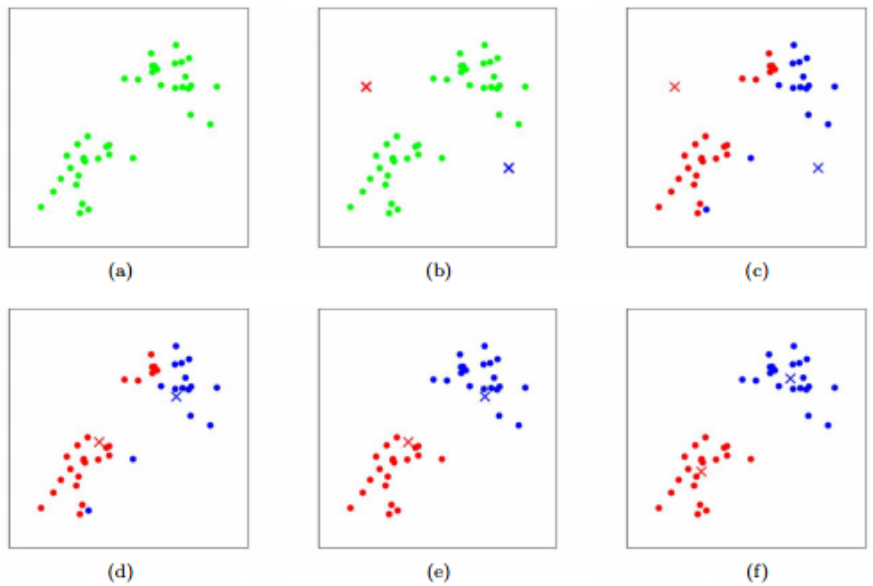
Dada una función de distancia definida sobre el espacio de características y una función de coste (p.ej. suma de distancias cuadráticas de cada elemento a su *medoide* correspodiente):

- Seleccionar los K medoides iniciales al azar (dentro del dataset)
- **Mientras** la función de costo esté por encima de un valor de tolerancia:
  - Para cada *cluster*, evaluar la función de coste de cada cluster si se intercambia cada elemento del cluster con el «medoide» actual
    - En caso de que el intercambio reduzca el valor de la función de costo para todos los elementos asignados al *medoide*, actualizar el *medoide* al elemento evaluado
  - Actualizar la etiqueta de cada elemento del dataset al *cluster* cuyo medoide se encuentre más cerca al elemento



# Método $K$-means

- $K$-means es un algoritmo de clasificación no supervisada (agrupamiento o *clustering* ) que agrupa objetos en K *clusters* (clases) basándose en sus características.
- El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster.
- Se suele usar la distancia cuadrática, pero esto puede variar según la naturaleza del problema.



# Algoritmo de K-means

**función** $K$-means: retorna una lista de elementos clasificados y una función de costo $J$

$K$: número de centroides; $d$: función de distancia; $X = \{\mathbf{x}_j\}\, j = 1 \ldots n$ : dataset

Definir aleatoriamente los valores para los centroides $\mathbf{c}_1, \mathbf{c}_2 \ldots, \mathbf{c}_K$

**mientras** que haya cambios en los centroides $\mathbf{c}_i$

Recorrer los elementos de $X$, etiquetando $\mathbf{x}_j^{(i)}$(i.e.$\mathbf{x}_j$ pertenece al cluster $i$), tal que se minimiza la distancia $d(\mathbf{c}_i, \mathbf{x}_j)$

**para** $i$ de 1 a $K$

Reemplace $\mathbf{c}_i$ con la media de todas las muestras para el cluster $i$-ésimo, $\bar{\mathbf{x}}_j^{(i)}$

**fin_para**

**fin_hasta**

calcular función de costo $\left(J\left(\mathbf{c}_1, \mathbf{c}_2 \ldots, \mathbf{c}_K\right) = \frac{1}{n} \sum_{j=1}^{n} \left\| \mathbf{x}_j^{(i)} - \mathbf{c}_i \right\|^2 \right.$

**retornar** lista de elementos etiquetados $X' = \{\mathbf{x}_j^{(i)}\}$ y el valor $J$ de la función de costo

# Identificación del mejor modelo en K-means

# Técnica del «codo»

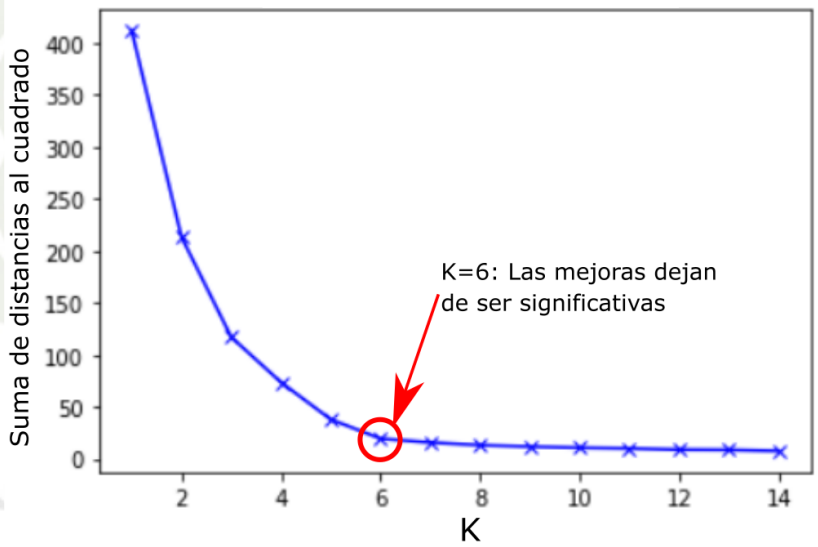
Para hacer la selección del mejor modelo de $K$-Means, dado que se trata de aprendizaje no supervisado, es necesario inferir el número de grupos (clusters) $K$ sin conocimiento previo del mismo

- Se puede usar una exploración sencilla del parámetro $K$, usando para evaluar visualmente el desempeño una medida de dispersión de los clusters
  - P. ej. la suma de distancias al cuadrado - SDC: $\sum_{i=1}^{n} \left\| \mathbf{x}_i^j - \mathbf{k}_j \right\|^2$, de acuerdo con el número de valores medios ($K$) empleados, siendo $\mathbf{k}_j$ el centroide del cluster $j$-ésimo, en el que quedó clasificado elemento $\mathbf{x}_i^j$ del dataset.

# Técnica del «codo» - implementación

**procedimiento** codo
    dataset: lista

    valores_SDC = lista()
    **Para** K=1 **hasta** max_clusters **hacer**
        iniciar(clasificador_km, $K$)
        entrenar(clasificador_km, dataset, sdc)
        valores_SDC.agregar($K$, sdc)
    **fin_para**
    graficar(valoresSDC)




En `sklearn`, el atributo `inertia_` del clasificador `KMeans` contiene la SDC

# Métodos basados en densidades

# Problemas de los métodos basados en centroides

- Cuando los grupos tienen medias muy cercanas, aunque la distribución de los elementos sea geométricamente diferente, K-means no logra separar correctamente las clases
- Cuando la superposición de los elementos en determinadas dimensiones (componentes) de los elementos dificulta la separación, la linealidad de la función «distancia» impide una separación detallada, aunque los clusters estén bien diferenciados

# Agrupamiento por desplazamiento de medias *(Mean-shift clustering)*

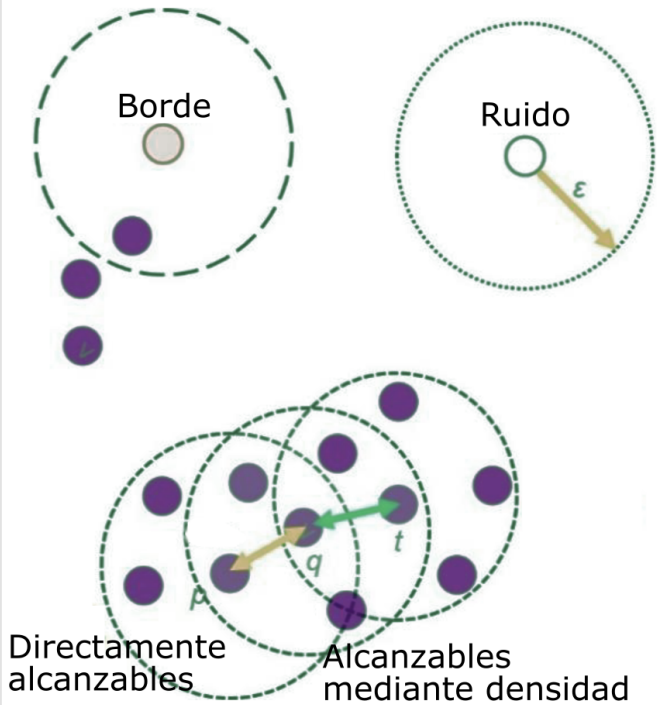
Ventanas deslizantes permiten buscar áreas de mayor *densidad* ($\#\ ^{\text{elementos}}/_{\text{u. área}}$). Las ventanas candidatas se filtran para eliminar «duplicadas» (cercanas). Desventaja: su complejidad computacional es alta

1. Se inicializa una ventana deslizante (circular) centrada en un punto no agrupado $C$ (definido al azar) con radio fijo $r$.
2. La ventana cambia su $C$ hacia algún vecino donde la ventana obtenga mayor densidad. Cada nuevo vecino visitado se etiqueta como parte del *cluster actual.*
3. El desplazamiento continúa hasta que no haya vecinos que permitan mejorar el valor de la densidad (volver a 2).
   - Cuando varias ventanas se superponen, se conserva la de mayor densidad (unir ventanas).
4. Al lograr una densidad máxima, se inicializa una nueva ventana para otro cluster (volver a 1)

# Agrupamiento espacial basado en distancias para aplicaciones con ruido - DBSCAN

Basado en la densidad de ventanas (vecindades). No requiere un número fijo de clusters al incio, identifica los valores atípicos como ruido y puede generar clusters de tamaño y forma arbitrarios. Sin embargo, tien bajo desempeño cuando los clústeres son de densidad variable o con datos de alta dimensionalidad.

1. Se inicia con un elemento arbitrario $p$ no visitado. La vecindad $b$ de $p$ se determina usando una distancia $\varepsilon$ (los vecinos de $p$ dentro de una esfera de radio $\varepsilon$).
2. Si hay una cantidad *minPts* de elementos en $b$, comienza el agrupamiento y $p$ es el primer punto del nuevo cluster; *de lo contrario*, $p$ se etiqueta como ruido ($p$ puede re-etiquetarse después); $p$ se etiqueta como "visitado".
3. Para todos los elementos en $b$ repite el mismo procedimiento, hasta que todos los $p'$ dentro de $b$ se han visitado y etiquetado.
4. Al terminar la visita a todos los elementos en $b$, se toma un nuevo elemento no visitado del dataset (regresar a 1), hasta que todos los elementos del *dataset* sean visitados.



# Métodos basados en modelos: Mezcla de Gaussianas

# Agrupamiento por mezcla de gaussianas

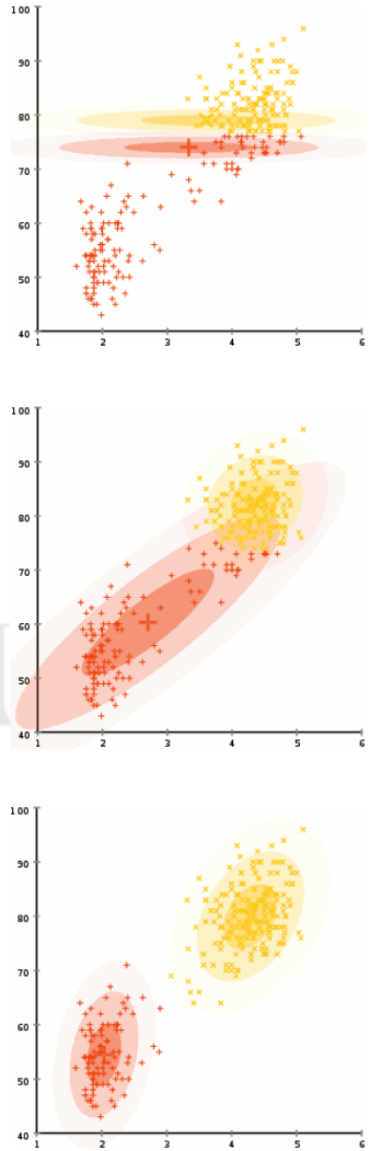
- La mezcla de modelos gaussianos (GMMs) ofrece más flexibilidad que K-means.
  - Se asume una distribución gaussiana de los elementos (datos) alrededor de cada «centroide», añadiendo otro parámetro al método, la dispersión (desviación estándar) además de la media (para cada dimensión).

$$\phi(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \; x \in \mathbb{R}.$$

$$\phi(\mathrm{x},\mu,\Sigma): \; \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\top}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

  donde $\Sigma$ es la matriz de covarianza diagonal (sus eigenvalores contienen las medidas de dispersión para cada componente)

- Así, se asigna una distribución gaussiana («normal») diferente para cada cluster. Para encontrar los parámetros $(\mu_i, \sigma_i)$ de cada cluster se usa un algoritmo denominado *Expectation–Maximization*



# *Expectation Maximization* para mezcla de gaussianas en agrupamiento

1. Seleccionar el número de clusters (como K-Means) e inicializar aleatoriamiente los parámetros de la distribución de cada cluster.
2. Dadas tales distribuciones, calcular la probabilidad de que cada elemento del dataset pertenezca a un cluster determinado.
   - Entre más cercano sea un punto al centro de una gaussiana, con mayor probabilidad pertenecerá al cluster.
3. Con base en las probabilidades estimadas, se calcula un nuevo conjunto de parámetros para las gaussianas de modo que se maximicen las probabilidades de los elementos del dataset dentro de cada cluster.
   - Los nuevos parámetros se calculan a partir de la suma ponderada de la posición de los elementos. Los pesos son las probabilidades de un elemento de pertenecer a un cluster particular.
   - De lo anterior, la desviación estándar cambia para crear un elipsoide más ajustado a los elementos más cercanos el centroide de cada cluster.
4. Los pasos 2 y 3 se repiten hasta que se cumple un criterio de convergencia (p.ej. una tasa de cambio definida sobre el centroide y la desviación estándar)

Dado que la probabilidad de pertenencia a cada cluster está definida para todos los elementos del dataset, cada elemento se etiqueta al cluster que maximiza su probabilidad de pertenencia, o se reporta la probabilidad de pertenencia a cada cluster (membresía mezclada).