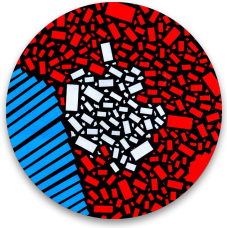


Fundamentos de Aprendizaje Automático y Reconocimiento de Patrones

Actividades en clase: Práctico 7

Graciana Castro, Martín Schmidt, Federico Lecumberry, Guillermo Carbajal

Instituto de Ingeniería Eléctrica

2024

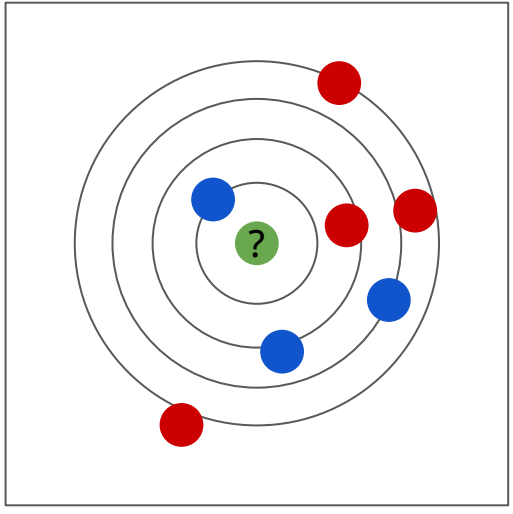
Objetivos

1. Aplicar *k-vecinos más cercanos* para clasificar entre las 10 categorías de dígitos. Optimizar el parámetro *k* utilizando la biblioteca *scikit-learn*.
2. Estimar densidades de probabilidad mediante el método de *Ventanas de Parzen*.
3. Implementar el algoritmo de clustering *k-means* y aplicarlo en datos sintéticos. Analizar su funcionamiento.
4. Realizar *agrupamiento de datos* mediante *Mezcla de Gaussianas*. Implementarlo utilizando el esquema *Expectation-Maximization* para encontrar los parámetros. Comparar este agrupamiento con el de *k-means*.

Tabla de contenido

Ejercicio 1: k-NN

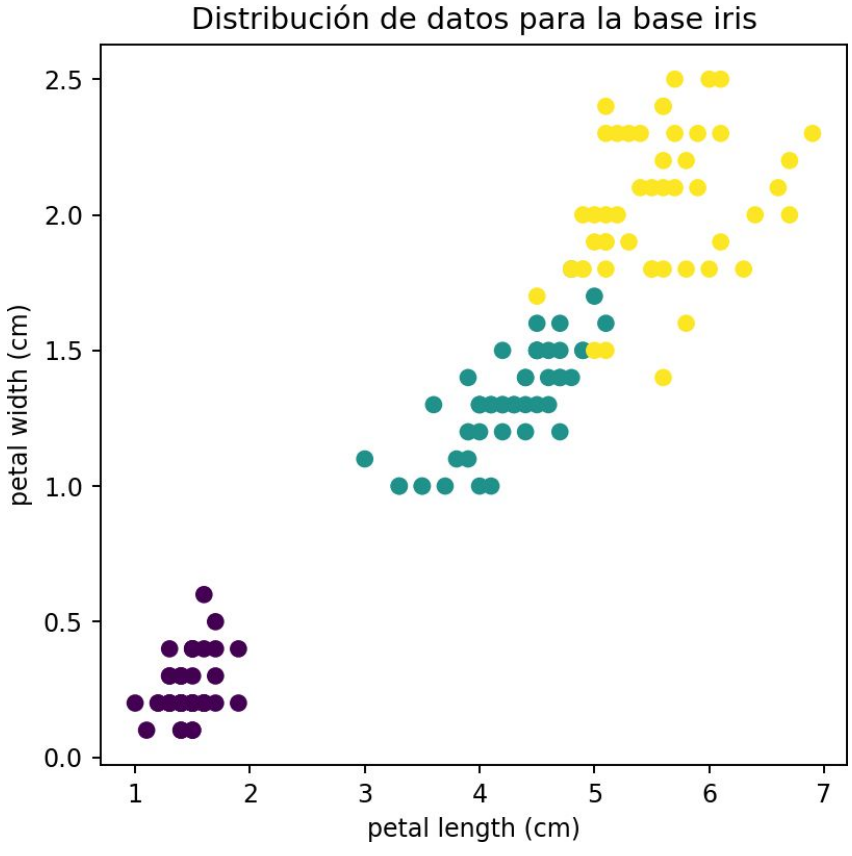
- Clasificación de dígitos utilizando *k-NN*
- Optimización del hyperparámetro *k*
  - mediante conjunto de validación
  - mediante validación cruzada



$k = 1, 3, 5, 7$

Ejercicio 2: ventanas de Parzen

- Datos: Flores de la base Iris



Ejercicio 2: ventanas de Parzen

- Datos: Flores de la base Iris
- Estimación de densidades

$$P(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} K(\mathbf{x}, \mathbf{z_i})$$

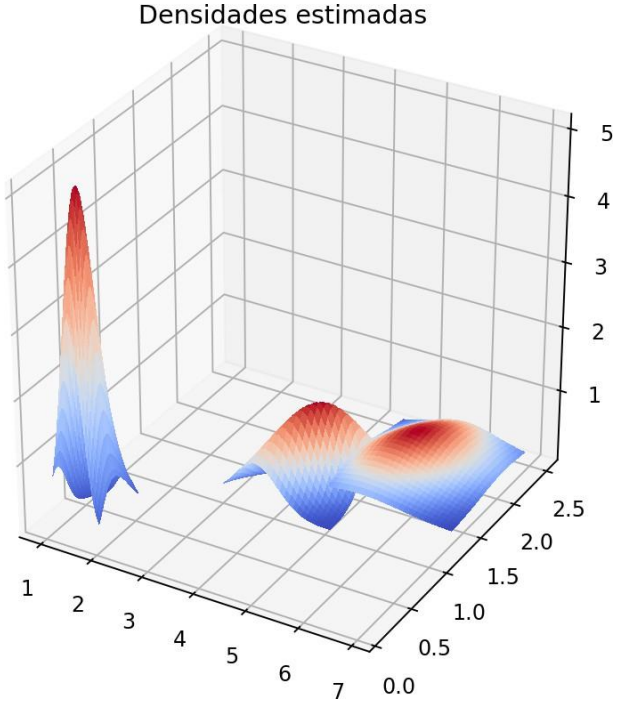
- Se utilizará un kernel gaussiano

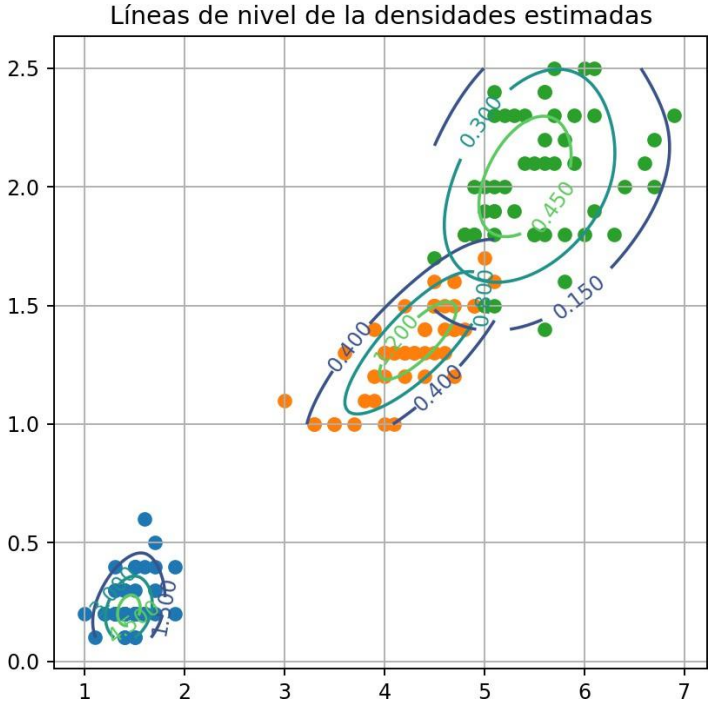
$$K(\mathbf{x}, \mathbf{z_i}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \mathbf{z_i})^T \Sigma^{-1} (\mathbf{x} - \mathbf{z_i}) \right)$$

con Σ de la forma $\Sigma = r^2 \Sigma_c$ siendo Σ_c la matriz de covarianza de los datos.

Ejercicio 2: ventanas de Parzen

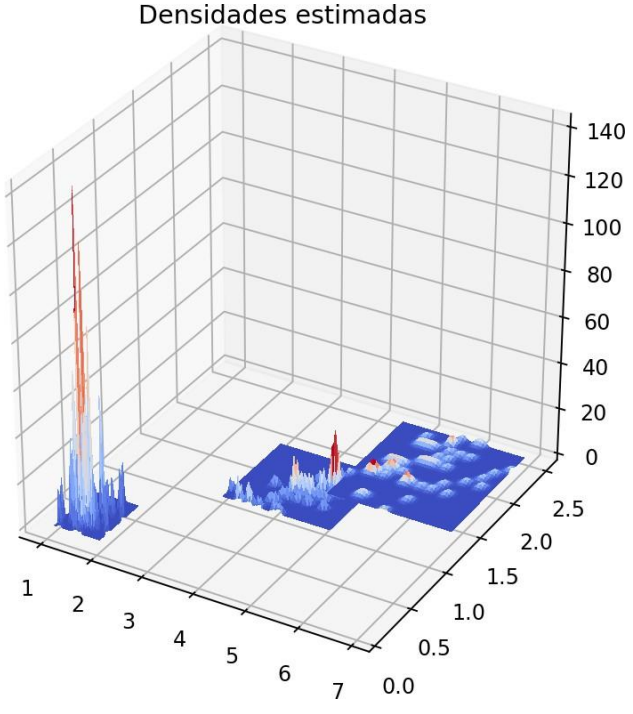
Efecto del ancho del kernel $r = 1$

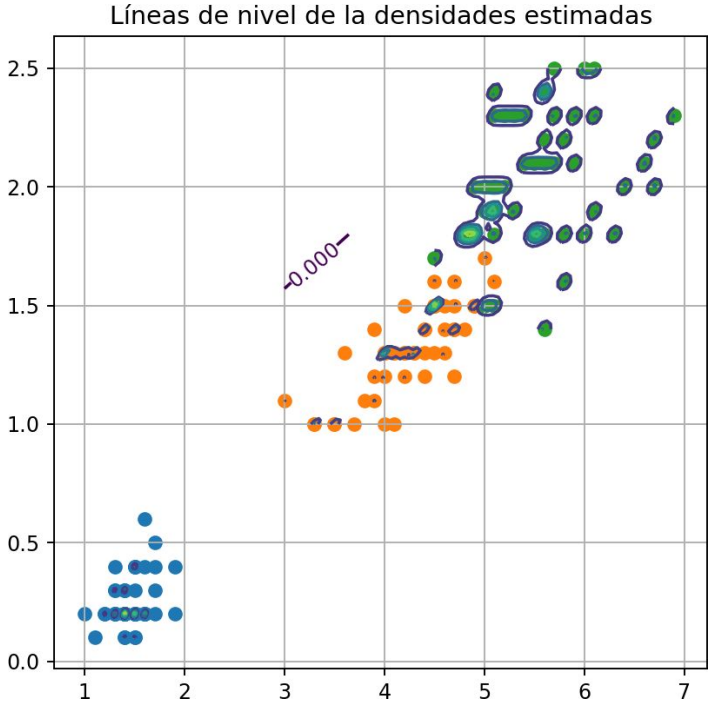




Ejercicio 2: ventanas de Parzen

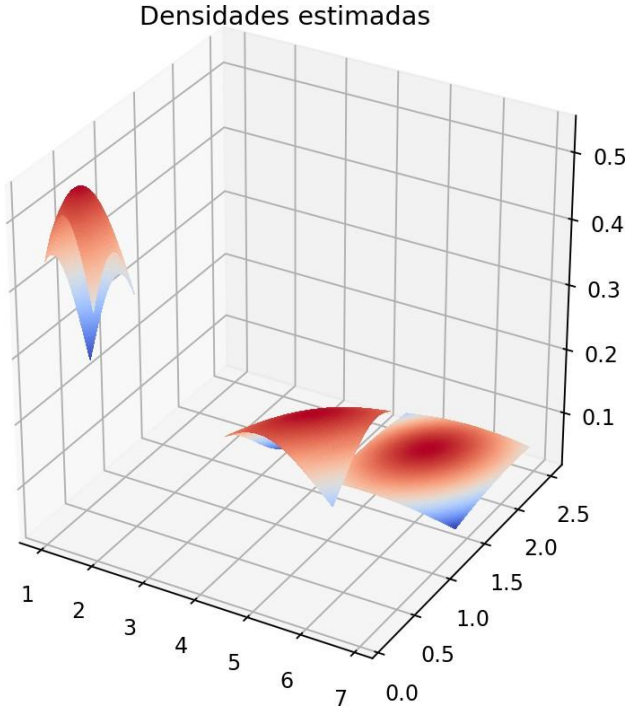
Efecto del ancho del kernel $r = 0.1$

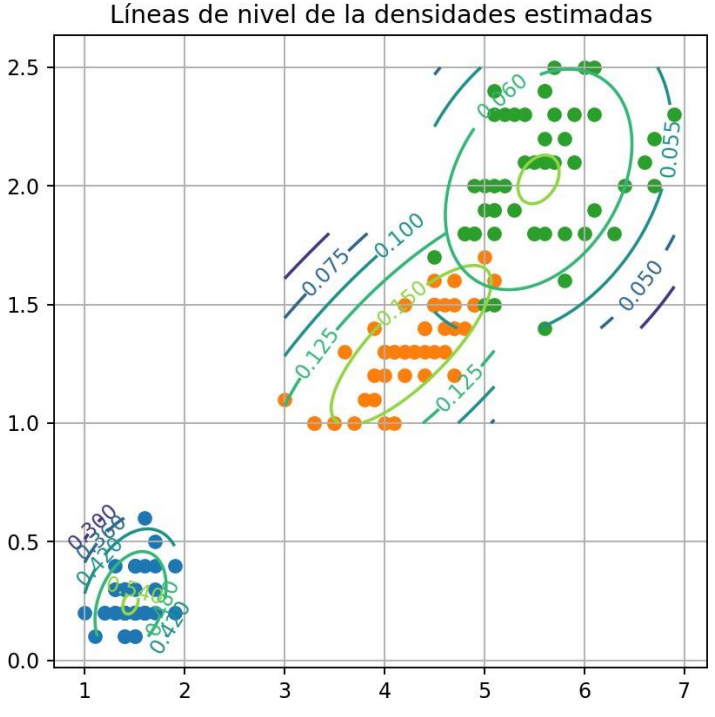




Ejercicio 2: ventanas de Parzen

Efecto del ancho del kernel $r = 4$





Ejercicio 3: K-means

- Técnica de agrupamiento de datos.
- Objetivo: Dividir los datos de entrada $x_1,...,x_N$ en K subconjuntos $S_1,...,S_K$ y elegir centros $\mu_1,...,\mu_K$ para cada cluster.
- Se busca minimizar

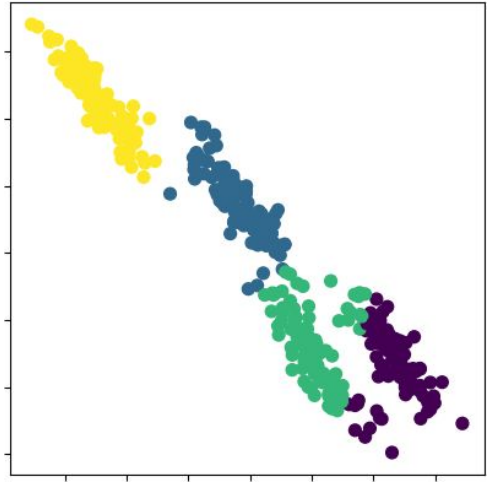
$$E_{\text{in}(S_1,...,S_K,\mu_1,...,\mu_K)} = \sum_{k=1}^{K} \sum_{\mathrm{x}_n \in S_k} \|\mathrm{x}_n - \mu_k\|^2$$

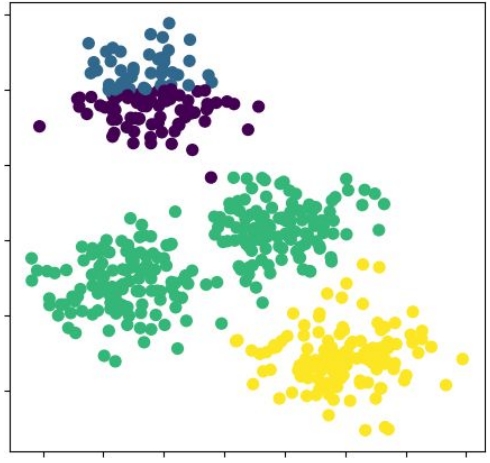
El algoritmo de Lloyd consta de los siguientes pasos:

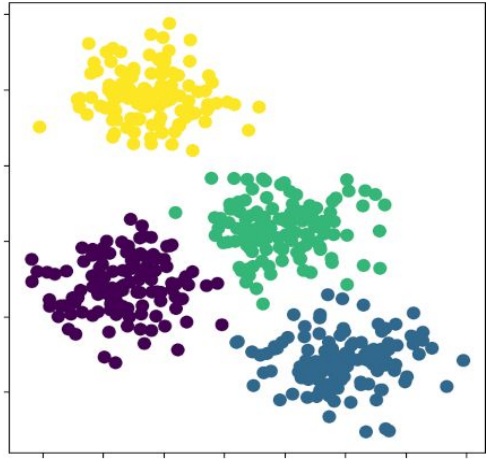
1. Se inicializan los centros μ_j de los clusters
2. Se construye el cluster S_j con todos los puntos cuyo centro m´as cercano es μ_j
3. Se actualizan los centros μ_j al centroide de los puntos pertenecientes a S_j
4. Se repiten los pasos anteriores hasta que el algoritmo converge

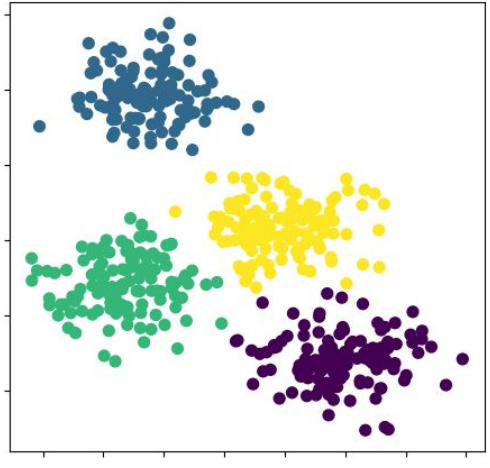
Ejercicio 3: K-means

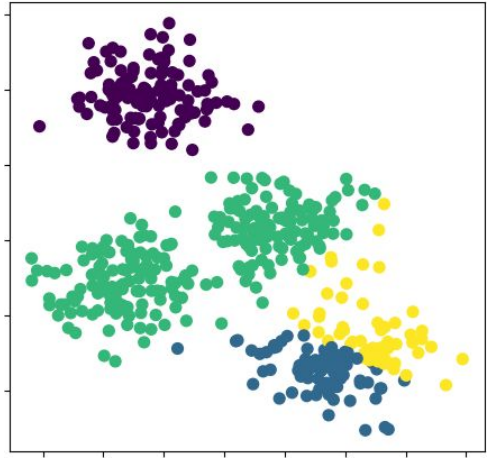
Resultados variando la inicialización.



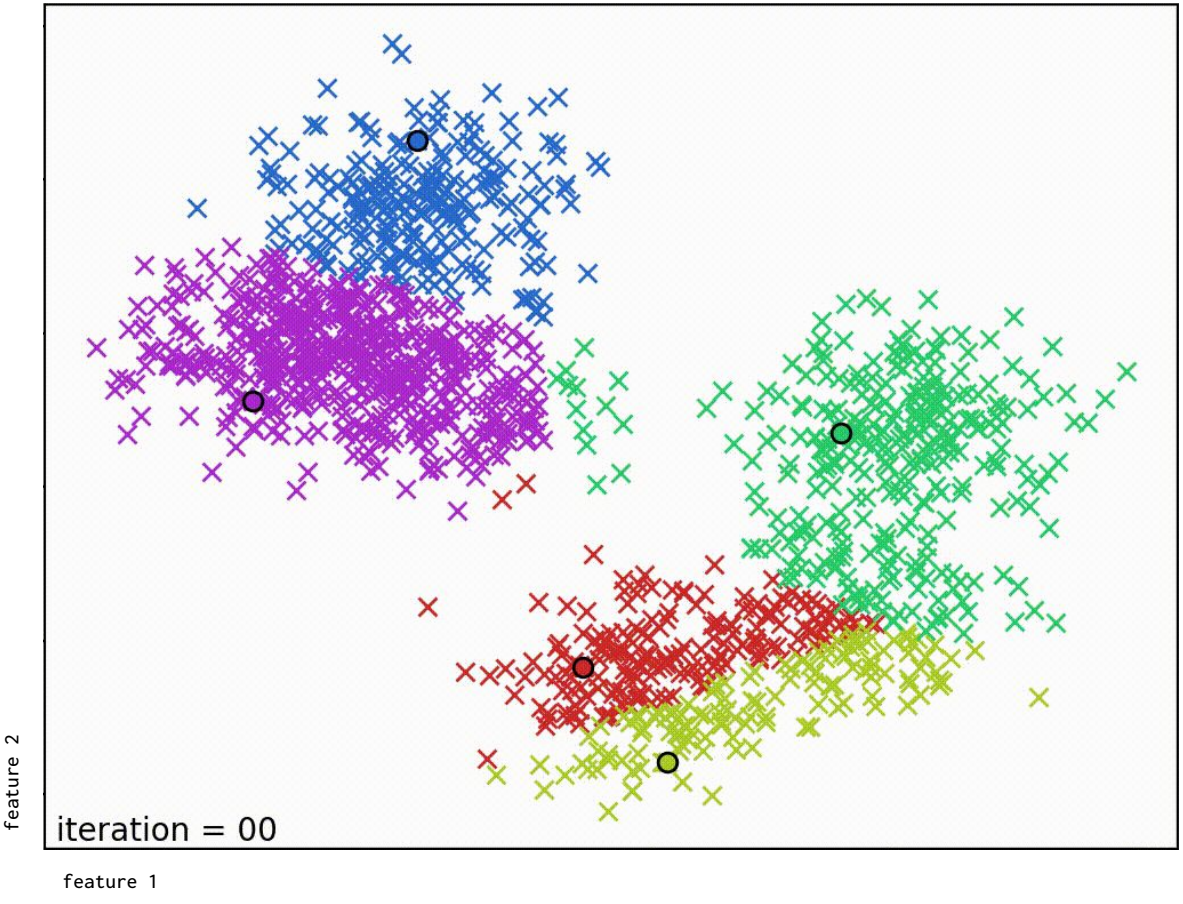




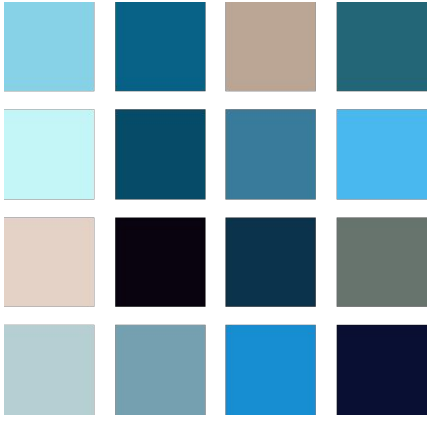




Ejercicio 3: K-means



↓ 16 color centroids



Color quantization (~K-means)

EM para mezcla de gaussianas

El algoritmo EM para encontrar los parámetros del modelo GMM consta de los siguientes pasos:

1. Se inicializan $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$ y $\boldsymbol{w}$.
2. Para cada muestra $\mathbf{x}_n$ se calcula la probabilidad γ_{nj} de pertenencia a cada *cluster j*.

$$\gamma_{nj} = \frac{w_j \mathcal{N}\left(\mathbf{x}_n | \mu_j, \boldsymbol{\Sigma}_j\right)}{\sum_{l=1}^{L} w_l \mathcal{N}\left(\mathbf{x}_n | \mu_l, \boldsymbol{\Sigma}_l\right)}$$

3. Se actualizan los centros $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$ y $\boldsymbol{w}$:

$$N_j = \sum_{n=1}^{N} \gamma_{nj} \qquad \mu_j^{new} = \frac{1}{N_j} \sum_{n=1}^{N} \gamma_{nj} \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_j^{new} = \frac{1}{N_j} \sum_{n=1}^{N} \gamma_{nj} \left(\mathbf{x}_n - \mu_j\right) \left(\mathbf{x}_n - \mu_j\right)^T$$

$$w_j^{new} = \frac{N_j}{N}$$

4. Se repiten los pasos anteriores hasta que el algoritmo converge.

Ejercicio 4: mezcla de Gaussianas

Resultados variando la inicialización.