4 Implementation - Gaussian Mixtures

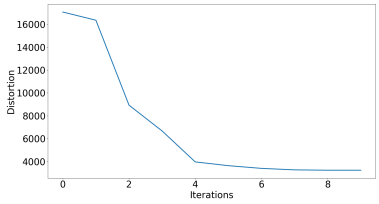
(a) Implement the K-means algorithm. Represent graphically the training data, the cluster centers, as well as the different clusters. Try several random initializations and compare results (centers and distortion measures).

Figure 1 shows the distortion measure $J(\mu, z) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_i^k ||x_i - \mu_k||^2$ against the number of iterations from three initialization strategies:

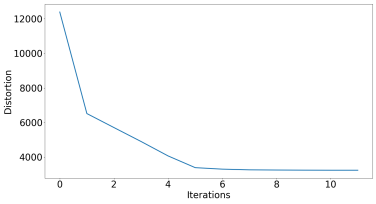
(i) Zero initialization: $\forall k,\ \mu_k = 0$
(ii) Random initialization: $\forall k,\ \mu_k \propto \mathcal{U}([x_{min}, x_{max}]) \times \mathcal{U}([y_{min}, y_{max}])$
(iii) Random sampling from the data points

In each case, we do 10 random restarts and keep the partition which minimizes the distortion. Table 1 shows the final distortion and the centers of the clusters for the different initialization strategies.

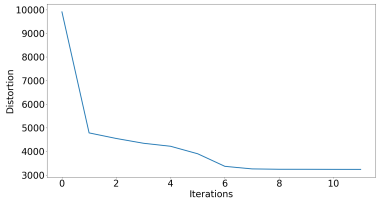
One can see that the random initialization strategy is the one minimizing the distortion in the training data. Nevertheless, all strategies lead to similar results, in terms of centroids and distortion, in the same amount of iterations (around 10 iterations). As a conclusion, we use the random strategy for further work. Figure 2 shows the training data, cluster centers and the different clusters.



(a) Zero initialization



(b) Random initialization



(c) Random from data initialization

Figure 1: Distortions against the number of iterations with different initialization strategies

Initializations	Zero	Random	Random from data
Distortion	3241.22	**3237.67**	3240.59
μ_1	$[3.48, -2.82]$	$[3.37, -2.67]$	$[3.61, -2.89]$
μ_2	$[-2.14, 3.97]$	$[-2.24, 4.16]$	$[-2.16, 4.11]$
μ_3	$[-3.80, -4.25]$	$[-3.78, -4.22]$	$[-3.64, -4.05]$
μ_4	$[3.80, 5.03]$	$[3.80, 5.10]$	$[3.79, 5.00]$

Table 1: Final distortion and centers of the clusters with different initialization strategies

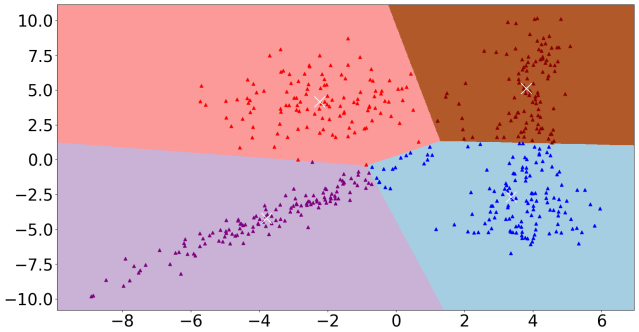


Figure 2: Training data, cluster centers and the different clusters.

(b) Consider a Gaussian mixture model in which the covariance matrices are proportional to the identity. Derive the form of the M-step updates for this model and implement the corresponding EM algorithm. Represent graphically the training data, the centers, as well as the covariance matrices

Consider the spherical Gaussian Mixture Model where we have n i.i.d data points $(x_i)_{i=1\dots n}$ and we want to estimate the latent features $(z_i)_{i=1\dots n}$ under the following assumptions:

$$\begin{aligned} X|Z=k &\propto \mathcal{N}(\mu_k, \sigma_k^2 \mathrm{Id}) \\ Z &\propto \mathcal{M}(1, \pi_1, \dots, \pi_K) \\ \theta &= \{\pi_k, \mu_k, \sigma_k,\ k = 1 \dots K\} \end{aligned}$$

The likelihood of this model is:

$$\begin{aligned} p_\theta(x) &= \sum_{k=1}^{K} p_\theta(x|z=k) p_\theta(z=k) \\ &= \sum_{k=1}^{K} \pi_k \times \mathcal{N}(x, \mu_k, \sigma_k^2 \mathrm{Id}) \end{aligned}$$

First, we can note:

$$\begin{aligned} p_\theta(z_i = k|x_i) &= \frac{p_\theta(x|z=k)p_\theta(z=k)}{\sum_{k'=1}^K p_\theta(x|z=k')p_\theta(z=k')} \\ &= \frac{\pi_k \mathcal{N}(x_i, \mu_k, \sigma_k^2 \mathrm{Id})}{\sum_{k'=1}^K \pi_{k'} \mathcal{N}(x_i, \mu_{k'}, \sigma_{k'}^2 \mathrm{Id})} \\ &= \tau_k^i(\theta) \end{aligned}$$

Now let's have a look at the Expectation-Maximization algorithm. Given θ_t at step t, we want to find θ_{t+1} such that:

$$\begin{aligned} \theta_{t+1} &= \arg\max_\theta \mathcal{L}(p_{\theta_t}(z|x), \theta) \\ &\text{with } \mathcal{L}(q,\theta) = \sum_z q(z) \log p_\theta(x,z) - \sum_z q(z) \log q(z) \\ \theta_{t+1} &= \arg\max_\theta \sum_z p_{\theta_t}(z|x) \log p_\theta(x,z) \\ &= \arg\max_\theta \mathbb{E}_{Z|X}[l_c(\theta)] \ \text{(E step)} \\ &\text{with } l_c(\theta) = \log p_\theta(x,z) = \sum_{i=1}^n \log p_\theta(x_i, z_i) = \sum_{i=1}^n \sum_{k=1}^K \left[z_k^i \log \pi_k + z_k^i \log \mathcal{N}(x_i, \mu_k, \sigma_k^2 \mathrm{Id}) \right] \\ \theta_{t+1} &= \arg\max_\theta \sum_{i=1}^n \sum_{k=1}^K \left[\tau_k^i(\theta_t) \log \pi_k + \tau_k^i(\theta_t) \log \mathcal{N}(x_i, \mu_k, \sigma_k^2 \mathrm{Id}) \right] \\ &= \arg\max_\theta J(\theta) \ \text{(M step)} \end{aligned}$$

The maximization of the π_k subject to $\sum_k \pi_k = 1$ gives us:

$$\pi_k^{(t+1)} = \frac{1}{n} \sum_{i=1}^n \tau_k^i(\theta_t)$$

which is the soft mean of the number of data points assigned to the cluster k.

The maximization of the μ_k gives us:

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n \tau_k^i(\theta_t) x_i}{\sum_{i=1}^n \tau_k^i(\theta_t)}$$

which is the mean of the data points assigned to the cluster k weighted by $\tau_k^i(\theta_t)$.

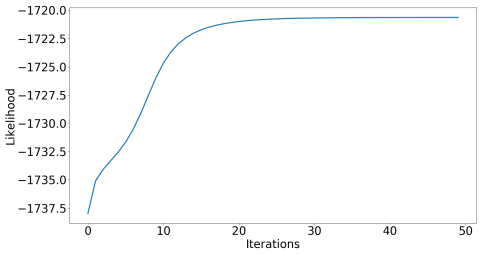
Finally, let's compute the gradient to obtain the expression for $\sigma_k^{(t+1)}$:

$$
\begin{aligned}
J &= \text{cte} + \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k^i(\theta_t) \left[-\log\det(\sigma_k^2 \mathrm{Id})^{\frac{1}{2}} - \frac{1}{2\sigma_k^2}(x_i - \mu_k)^\intercal (x_i - \mu_k) \right] \\
\nabla_{\sigma_k} J &= \sum_{i=1}^{n} \tau_k^i(\theta_t) \left[-\frac{d}{\sigma_k} + \frac{1}{\sigma_k^3}(x_i - \mu_k)^\intercal (x_i - \mu_k) \right] \\
&= -\frac{1}{\sigma_k} \sum_{i=1}^{n} d \times \tau_k^i(\theta_t) + \frac{1}{\sigma_k^3} \sum_{i=1}^{n} \tau_k^i(\theta_t)(x_i - \mu_k)^\intercal (x_i - \mu_k) \\
&= 0 \\
\implies \sigma_k^{2\,(t+1)} &= \frac{1}{d} \frac{\sum_{i=1}^{n} \tau_k^i(\theta_t)(x_i - \mu_k)^\intercal (x_i - \mu_k)}{\sum_{i=1}^{n} \tau_k^i(\theta_t)}
\end{aligned}
$$

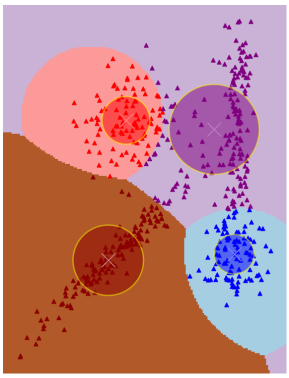
where d is the dimension of the data points (2 in our case). Hence, the estimate is the mean of the variance of each data point weighted by $\tau_k^i(\theta_t)$.

Figure 3 shows the log-likelihood against the number of iterations and represents the training data, cluster centers, covariance matrices and clusters. One can see that the clusters don't seem to fairly represent the real distribution of the data. Our spherical assumption doesn't seem appropriate, especially for the brown and purple clusters.

One should note that to represent the clusters, we assign to a point in the plane the cluster k where $k = \arg\max_{k'} \tau_{k'}$. The difference with k-means is that τ is not a hard but a soft-assignment. Hence, this cluster assignment method is just a convenient way to represent the latent parameters but one should look in details to τ to understand the effect of each gaussian on a particular data point.



(a) Log-likelihood against the number of iterations



(b) Training data, cluster centers, covariances and the different clusters.

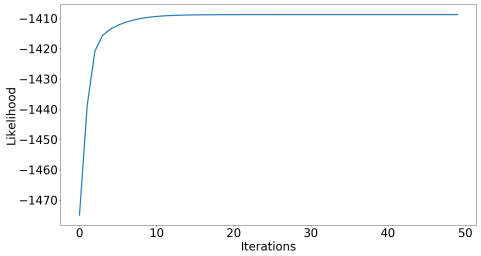
Figure 3: Spherical Gaussian Mixture Model

(c) Implement the EM algorithm for a Gaussian mixture with general covariance matrices. Represent graphically the training data, the centers, as well as the covariance matrices.

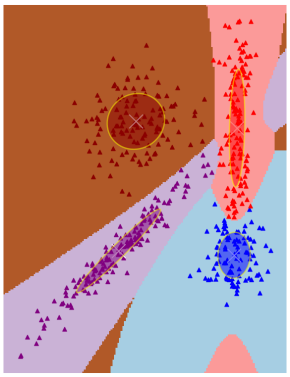
In this case, the computations are similar to the spherical gaussian mixture except that we do not make any assumptions on the form of the covariance matrices. Hence, its estimates becomes as follows:

$$\Sigma_k = \frac{\sum_{i=1}^{n} \tau_k^i (x_i - \mu_k)(x_i - \mu_k)^\intercal}{\sum_{i=1}^{n} \tau_k^i}$$

Figure 4 shows the log-likelihood against the number of iterations and represents the training data, cluster centers, covariance matrices and clusters. One can see that the learned decision boundaries are much more complex and seem to better encode the distribution of the data.



(a) Log-likelihood against the number of iterations



(b) Training data, cluster centers, covariances and the different clusters.

Figure 4: General Gaussian Mixture Model

(d) Comment the different results obtained in earlier questions. In particular, compare the log-likelihoods of the two mixture models on the training data, as well as on test data.

Table 2 shows the log-likelihood of the two mixture models on the training data, as well as on test data. One can directly see that the general model performs much better than the spherical one for both training and test data. This is coherent with our observations on Fig. 3b and Fig. 4b.

Model	Training set	Test set
Spherical	-1720.63	-1695.66
General	**-1408.78**	**-1490.04**

Table 2: Log-likelihood of the spherical and the general mixture models on the training and the testing data.