TP2: Computational Statistics Course

Yassine ZAOUI
yassine.zaoui@ensta-paris.fr

Exercise 1: Discrete Distribution

Question 1: How to generate a random variable X?

To generate a random variable X having the discrete distribution:

$$P(X = x_i) = p_i \quad \text{for } i \in \{1, 2, \ldots, n\},$$

we use the **inversion method**. The procedure is as follows:

- Compute the cumulative distribution function (CDF):

$$F(k) = \sum_{j=1}^{k} p_j, \quad \text{for } k = 1, 2, \ldots, n.$$

- Sample a uniform random variable $U \sim \text{Unif}(0, 1)$.
- Find the smallest k such that:

$$F(k) \geq U,$$

  which corresponds to:

$$F^{-1}(U) = \min \left\{ k \in \{1, 2, \ldots, n\} \mid \sum_{j=1}^{k} p_j \geq U \right\}.$$

- Return the value k_{min}.

This ensures that X is generated according to the desired distribution.

Question 2: Corresponding Python Algorithm

You can check the source code for implementation.

Question 3: Empirical vs. Theoretical Distribution

For N large, we generate a sequence of N i.i.d. random variables and compare the empirical distribution with the theoretical distribution.

When N is sufficiently large (e.g., $N = 10,000$), the histogram of the empirical distribution should closely match the theoretical distribution $P(X = x_i) = p_i$. The result is given by the following plot where the red bars represent the theoretical probabilities, and the blue bars represent the frequencies observed in the generated samples.

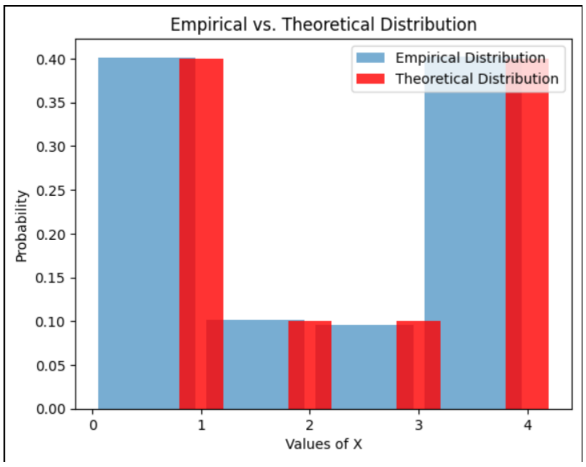

Figure 1: Empirical vs theoretical distribution

Indeed, as expected, the empirical distribution is roughly identical with the theoretical one, proving the success of our method to generate the discrete random variable X.

Exercise 2: Gaussian Mixture Model and the EM Algorithm

Question 1

Given n observations $\{x_1, \dots, x_n\}$, the likelihood of the parameters $\theta = (\alpha_1, .., \alpha_m, \mu_1, .., \mu_m, \Sigma_1, ..., \Sigma_m)$ is:

$$L(\theta; x) = \prod_{i=1}^{n} f_\theta(x_i),$$

where the marginal density $f_\theta(x_i)$ is:

$$f_\theta(x_i) = \sum_{j=1}^{m} \alpha_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j).$$

The log-likelihood is:

$$\ell(\theta; x) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{m} \alpha_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j) \right).$$

Question 2 & 3

To maximize the log-likelihood, we use the EM algorithm.

E-Step

Introduce latent variables $z_i \in \{1, \dots, m\}$ indicating the cluster of x_i. Define the posterior responsibility:

$$\gamma_{ij} = \Pr(z_i = j \mid x_i, \theta) = \frac{\alpha_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}{\sum_{k=1}^{m} \alpha_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}.$$

M-Step

Update the parameters θ to maximize the expected complete log-likelihood:

$$Q(\theta \mid \theta^{(t)}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_{ij} \log \left(\alpha_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j) \right).$$

This gives the following updates:

- Mixing weights:
$$\alpha_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} \gamma_{ij}.$$
- Means:
$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij} x_i}{\sum_{i=1}^{n} \gamma_{ij}}.$$
- Covariance matrices:
$$\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ij}(x_i - \mu_j^{(t+1)})(x_i - \mu_j^{(t+1)})^\top}{\sum_{i=1}^{n} \gamma_{ij}}.$$

See the implementation on the source code.

Question 4:

The estimated parameters are summarized in the table below, showing the differences between the true and estimated values:

Parameter	**Component 1 (True vs Est.)**	**Component 2 (True vs Est.)**	**Difference (L2 Norm)**
Weights	0.4000 vs 0.4011	0.6000 vs 0.5989	0.0011
Means	[0, 0] vs [-0.0123, -0.0075]	[3, 3] vs [3.0122, 3.0061]	0.0143, 0.0137
Covariances	See below	See below	0.0919, 0.0385

Table 1: Comparison of True and Estimated Parameters

Covariance Matrices:

- Component 1:
$$\text{True} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}, \quad \text{Estimated} = \begin{bmatrix} 1.0705 & 0.0388 \\ 0.0388 & 1.0216 \end{bmatrix}.$$
- Component 2:
$$\text{True} = \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{bmatrix}, \quad \text{Estimated} = \begin{bmatrix} 0.5079 & 0.0243 \\ 0.0243 & 0.5155 \end{bmatrix}.$$

The following figure plots the log-likelihood over EM iterations:

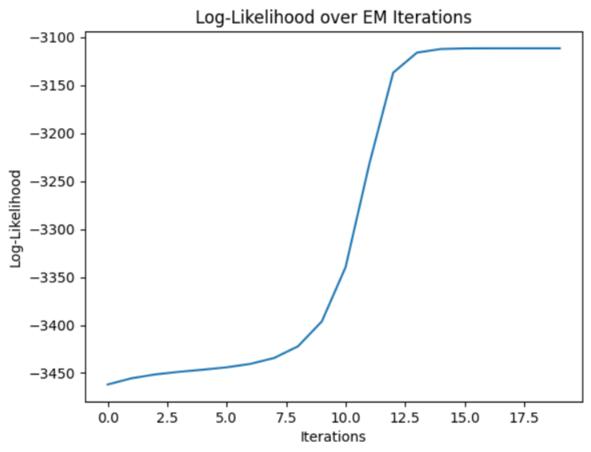

Figure 2: the log-likelihood over EM iterations

Conclusion: The estimated parameters are very close to the true ones, with minor differences in weights, means, and covariances. This indicates that the EM algorithm successfully estimated the parameters of the Gaussian mixture model.

Question 5: Application: Crude Birth/Death Rate

The data of Crude Birth Rate (CBR) and Crude Death Rate (CDR) for the year 2020 are analyzed using a Gaussian Mixture Model (GMM). The scatter plot below illustrates the relationship between CBR and CDR for the data.

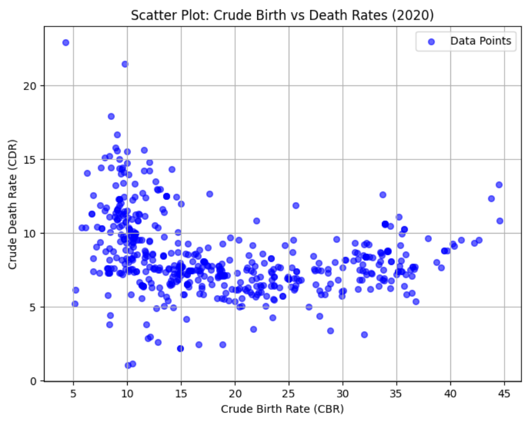

Figure 3: Scatter plot: Crude Birth Rate vs Death Rate for 2020

From the scatter plot, we observe that there might be multiple clusters in the data, which suggests that a Gaussian Mixture Model could be an appropriate model for clustering the data. The next step involves fitting the GMM to the data and evaluating the optimal number of clusters based on the Bayesian Information Criterion (BIC).

Question 6: Estimating Parameters and BIC Calculation

We estimate the parameters of the GMM for different values of m, the number of clusters, and compute the corresponding BIC for each model. The BIC scores are plotted below for different numbers of clusters, and the optimal number of clusters is selected based on the lowest BIC value.

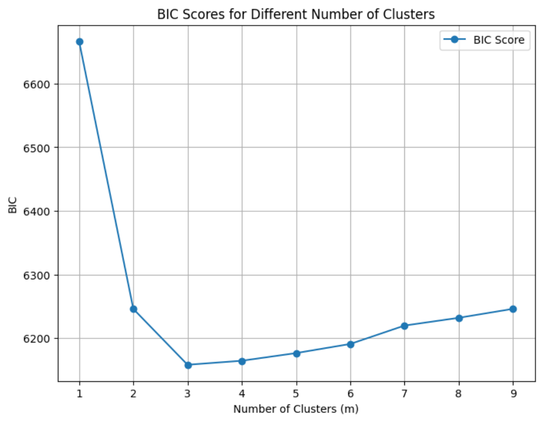

Figure 4: BIC Scores for Different Number of Clusters

As we can see from the BIC plot, the optimal number of clusters is 3, as it minimizes the BIC score.

The next plot shows the GMM clustering results for the optimal number of clusters $m = 3$, where the data points are colored based on their assigned cluster, and the cluster centroids are indicated by red crosses.

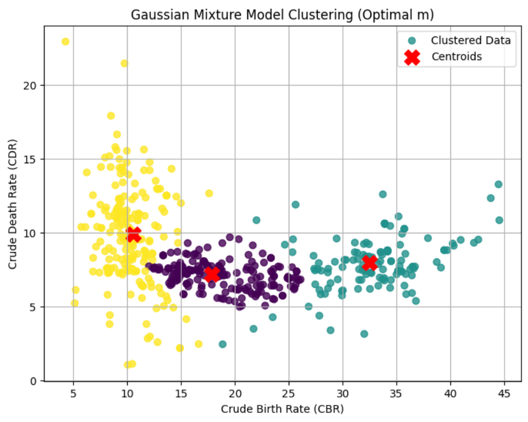

Figure 5: Gaussian Mixture Model Clustering with Optimal $m = 3$

Finally, the probability density function (PDF) of the GMM is plotted over the scatter plot using contour lines, which visually represents the density of the clusters.

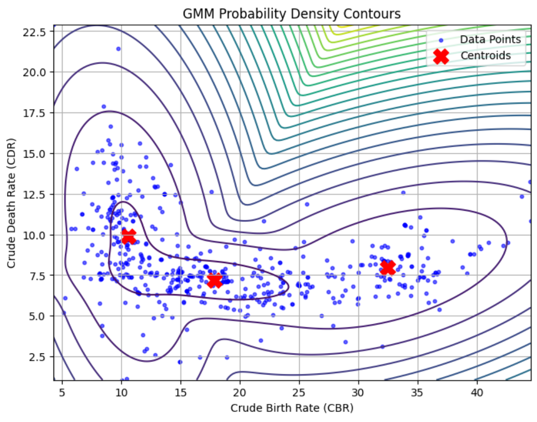

Figure 6: Gaussian Mixture Model Probability Density Contours

The contour plot shows the probability density function (PDF) of the GMM fitted to the data. The regions with higher density correspond to the clusters identified by the model. This confirms that the Gaussian Mixture Model is a suitable approach for clustering the Crude Birth Rate and Death Rate data.

Exercise 3: Importance Sampling

Exercise 3: Importance sampling

Question 1 & 2 &3

We calculate the expectation $\mathbb{E}_p[f(X)]$ where $f(x) = 2\sin\left(\pi\frac{x}{1.5}\right)\mathbb{1}_{\mathbb{R}^+}(x)$, using $p(x)$ and $q(x)$ defined as:

$$p(x) = x^{0.65}e^{-x^2/2} \quad \text{and} \quad q(x) = \frac{2}{\sqrt{2\pi \cdot 1.5}}e^{-\frac{(x-0.8)^2}{2\cdot 1.5}}.$$

The algorithm is implemented by generating samples from $q(x)$, discarding any negative samples since $p(x)$ is supported only on $\mathbb{R}^+$, and using importance sampling to estimate the expectation. Here, we just need to cautious as to normalize the weights $\frac{p(X_i}{q(X_i)}$ by their mean.
the code is given in the notebook.

Here we display the plots of the functions p and q for $\mu \in \{0.8, 6\}$:

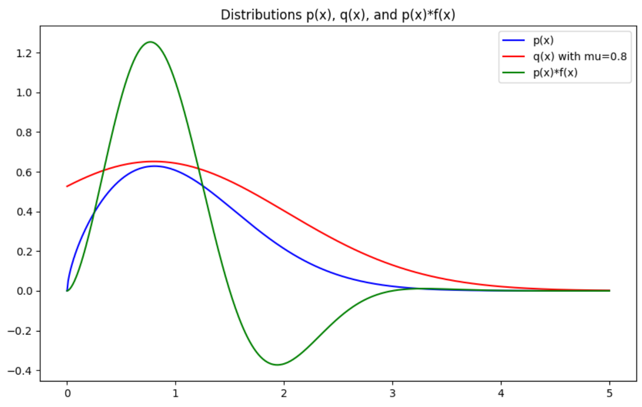

Figure 7: Ditributions p and q for $\mu = 0.8$

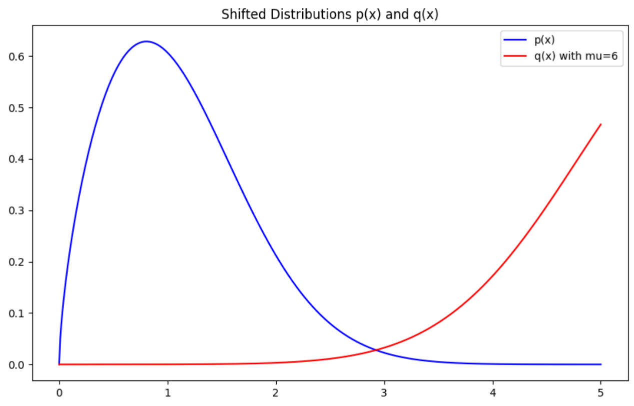

Figure 8: Ditributions p and q for $\mu = 6$

We have repeated the experiment with $\mu = 6$, shifting the proposal distribution $q(x)$ so that its center of mass is far from $p(x)$ as illustrated in the above plot. We here show the results in the following tables:

Table 2: Importance Sampling Results for Different Values of μ

Sample Size N	**Estimate** ($\mu = 0.8$)	**Variance** ($\mu = 0.8$)
10	-0.4219	0.2916
100	0.9077	0.0284
1,000	0.7773	0.0026
10,000	0.7851	0.0003

Sample Size N	**Estimate** ($\mu = 6$)	**Variance** ($\mu = 6$)
10	-1.0240	0.2916
100	0.9932	0.8495
1,000	-1.3792	1.3811
10,000	-0.0270	0.0841

We highlight that the value of the objective integral is ≈ 0.775. Now, according the tables, we see clearly that for $\mu = 0.8$, we have better estimation of the integral value and the variance of our estimator is consistently decreasing as N increases. Nevertheless, that's not the case for $\mu = 6$ where we have bad estimations that don't seem to converge at all, it's rather oscillating, which is the case for it's variance.

Question 4

The EM algorithm is employed in step (iii) of the Population Monte Carlo algorithm to maximize the empirical criterion:

$$\sum_{i=1}^{n} \tilde{\omega}_i \log \left(\sum_{j=1}^{M} \alpha_j \phi(X_i; \mu_j, \Sigma_j) \right),$$

where:

- $\phi(X_i; \mu_j, \Sigma_j)$ represents the density of a Gaussian distribution with mean μ_j and covariance Σ_j.
- $\tilde{\omega}_i$ are the normalized importance weights given by:

$$\tilde{\omega}_i = \frac{\omega_i}{\sum_{k=1}^{n} \omega_k}, \quad \text{with } \omega_i = \frac{p(X_i)}{q(X_i)}.$$

The EM algorithm alternates between the E-step and the M-step to maximize this criterion iteratively. Here, as seen in the class for the GMM model, to write our new parameters, we need to compute the γ_{ij}, which is the proportion of the contribution of the j-th Gaussian component to the i-th data point:

$$\gamma_{ij} = \frac{\alpha_j \phi(X_i; \mu_j, \Sigma_j)}{\sum_{k=1}^{M} \alpha_k \phi(X_i; \mu_k, \Sigma_k)}.$$

Now, using γ_{ij}, we update the parameters α_j, μ_j, and Σ_j to maximize the criterion. The updates are derived as follows:

Mixing Coefficients:

$$\alpha_j^{(t+1)} = \frac{\sum_{i=1}^{n} \tilde{\omega}_i \gamma_{ij}}{\sum_{i=1}^{n} \tilde{\omega}_i}.$$

Means:

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{n} \tilde{\omega}_i \gamma_{ij} X_i}{\sum_{i=1}^{n} \tilde{\omega}_i \gamma_{ij}}.$$

Covariance Matrices:

$$\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^{n} \tilde{\omega}_i \gamma_{ij} (X_i - \mu_j^{(t+1)})(X_i - \mu_j^{(t+1)})^\top}{\sum_{i=1}^{n} \tilde{\omega}_i \gamma_{ij}}.$$

3.C – Application to a Banana-shaped Density

5. Adaptive Sampling for Banana Density

The adaptive importance sampling algorithm was implemented for the banana-shaped density defined by:

$$\nu(x) \propto \Phi(x_1, x_2 + b(x_1^2 - 1), x_3, \ldots, x_d).$$

The following figure illustrates the resulting samples in the first two dimensions:

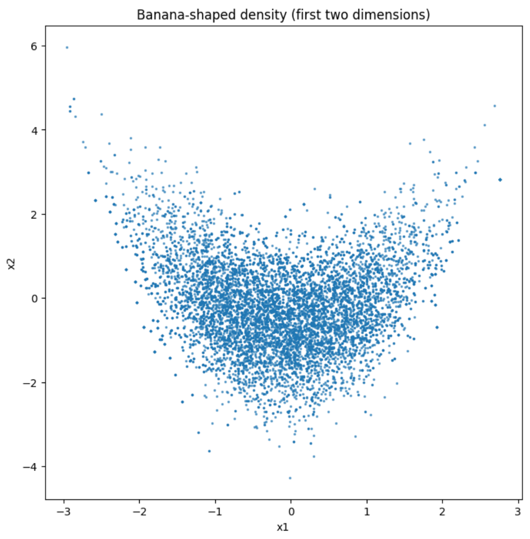

Figure 9: Resulting samples in the first two dimensions