Master 2 MVA - Computational Statistics

TP2 : Expectation-Maximisation algorithm –
Importance sampling

Mathis LE BAIL

November 2023

Exercise 1 : Discrete distributions

Let $n \in \mathbb{N}^*$ and $X = \{x_1, \dots, x_n\}$ a set of n distinct real numbers. Let $(p_i)_{i \in [\![1,n]\!]}$ a sequence of real numbers such that :

$$\forall i \in [\![1,n]\!], \quad p_i > 0 \quad \text{and} \quad \sum_{i=1}^{n} p_i = 1$$

1. We want to generate a random variable X with the discrete distribution $(p_i)_{i \in \{1,\dots,n\}}$ on X :

$$\forall i \in \{1, ..., n\}, \qquad \mathbb{P}(X = x_i) = p_i$$

To do this, we use the Inverse transformation method. If we denote F_X the cumulative distribution function of X and $U \sim \mathcal{U}([0,1])$ then we have $X \sim F_X^{-1}(U)$. X is discrete so F_x is piecewise constant. For a realisation x of X and u of U, we have the formula :

$$x = \mathbb{1}_{[0,p_1]}(u)x_1 + \sum_{j=1}^{n-1} \mathbb{1}_{[\sum_{k=1}^{j} p_k, \sum_{k=1}^{j+1} p_k]}(u) \quad x_{j+1} \tag{1}$$

This formula allows for sampling according to the distribution of X defined by the $(pi)_i$ and $(x_i)_i$.

2. 3. One applies this formula in a custom Python algorithm. We use the latter to generate a sequence $(X_i)_{i \in \{1,\dots,N\}}$ of i.i.d random variables having the same distribution as X for N large and $(p_j)_j$ fixed. For some example distributions, we compare the empirically generated distribution with the theoretical one :

For a uniform distribution over a space $X = [\![0,9]\!]$, we have the following distributions :

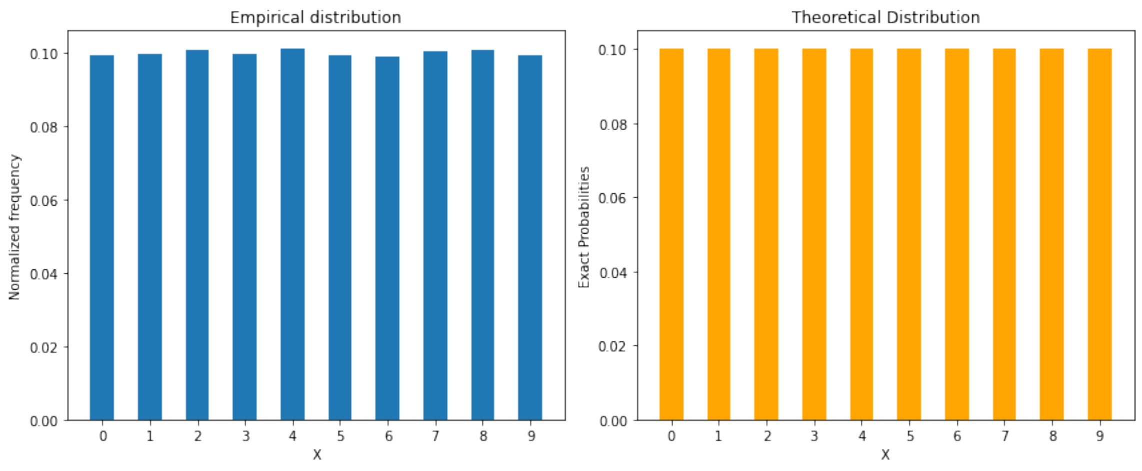


FIGURE 1 – Graphical comparison between the empirical uniform distribution generated by the sequence $(X_i)_{i \in [\![1,N]\!]}$ and the theoretical uniform distribution with $N = 100000$ and $n = 10$

We try the same with a more original distribution. We take $X = [\![1,8]\!]$ and $p = [0.1, 0.05, 0.3, 0.01, 0.15, 0.09, 0.25, 0.05]$, we obtain :

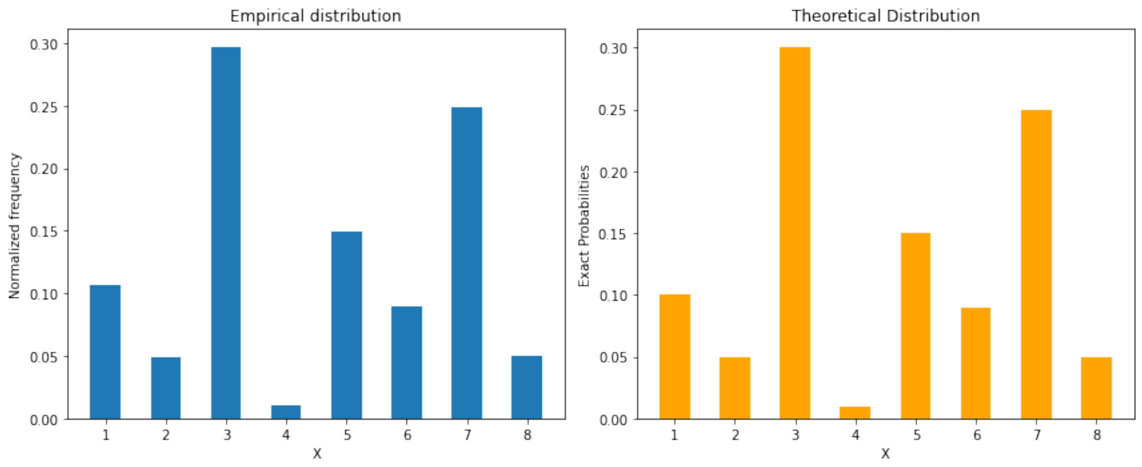


Figure 2 – Graphical comparison between the empirical distribution generated by the sequence $(X_i)_{i\in[\![1,N]\!]}$ and the theoretical distribution with $N = 10000$ and $n = 8$

In both cases, the empirical distribution obtained with the sequence $(X_i)_{i\in[\![1,N]\!]}$ generated by the algorithm appears to be quite close to the theoretical one for a sufficiently large N.

Exercise 2 : Gaussian mixture model and the EM algorithm

Let consider a n-sample. For each individual, we observe a random variable $X_i \in \mathbb{R}^d$ and we assume there exists a discrete unobserved variable Z_i which encodes the class of X_i. More formally, we consider a mixture of p Gaussians defined as follows. Let $(\alpha_1, \dots, \alpha_p) \in \mathbb{R}_+^p$ such that $\sum_{i=1}^p \alpha_i = 1$ and consider the following hierarchical model :

$$\forall i \in [\![1, n]\!], \quad \forall j \in [\![1, p]\!], \quad \mathbb{P}\left(Z_i = j\right) = \alpha_j$$

and

$$\forall i \in [\![1, n]\!], \quad \forall j \in [\![1, p]\!] \quad X_i \mid \theta, \{Z_i = j\} \sim \mathcal{N}\left(\mu_j, \Sigma_j\right).$$

Unless otherwise stated, we assume that p is fixed.

1. If we consider only the distribution of X then it depends of the parameters $\theta = \{(\alpha_j), (\mu_j), (\Sigma_j)\}_{j \in [\![1,p]\!]}$ and for the likelihood of θ given the outcomes of $(x_i)_{i \in [\![1,n]\!]}$ of the i.i.d n-sample $(X_i)_{i \in [\![1,n]\!]}$, it is defined by :

$$\begin{aligned}\mathcal{L}(x_1, ..., x_n; \theta) = p_\theta(x_1, ..., x_n) &= \prod_{i=1}^n p_\theta(x_i) \\ &= \prod_{i=1}^n \left(\sum_{j=1}^p p_\theta(x_i | Z_i = j) \mathbb{P}(Z_i = j) \right) \\ &= \prod_{i=1}^n \left(\sum_{j=1}^p \mathcal{N}(x_i; \mu_j, \Sigma_j) \alpha_j \right)\end{aligned} \tag{2}$$

where $\mathcal{N}(x_i; \mu_j, \Sigma_j)$ denotes the multivariate Gaussian density with parameters μ_j and Σ_j evaluated at $x_i \in \mathbb{R}^d$.

3. The EM algorithm decomposes into two main iterative steps :

- Expectation step (E step) : Compute $Q(\theta|\theta^t) = \mathbb{E}_{Z \sim p_{\theta_t}(.|X)}[\log(p_\theta(X, Z|\theta))]$ where $p_{\theta_t}(Z|X)$ is the current conditional distribution of Z given X and θ_t the current estimates of the parameters.
- Maximization step (M step) : Update $\theta_{t+1} = \arg\max\ Q(\theta|\theta^t)$

In our case where $d = 2$ and the number of clusters is denoted p, we compute Q as :

$$Q(\theta|\theta^t) = \sum_{i=1}^n \mathbb{E}_{Z_i \sim p_{\theta_t}(.|X_i = x_i)}[\log(p_\theta(x_i, Z_i))] \quad (X_i, Z_i) \text{ sequence of independent variables} \tag{3}$$

with

$$\begin{aligned}\forall j \in [\![1, p]\!], \quad & \log p_\theta(x_i, Z_i = j) \\ &= \log p_\theta(x_i | Z_i = j) + \log p_\theta(Z_i = j) \\ &= \log \mathcal{N}(x_i; \mu_j, \Sigma_j) + \log \alpha_j \qquad \text{as we are in a Gaussian mixture model}\end{aligned} \tag{4}$$

and

$$\begin{aligned}p_{\theta_t}(Z_i = j | X_i = x_i) &= \frac{p_{\theta_t}(x_i | Z_i = j) p(Z_i = j)}{p(x_i)} \\ &= \frac{\mathcal{N}(x_i; \mu_{jt}, \Sigma_{jt}) \alpha_{jt}}{\sum_{k=1}^p \mathcal{N}(x_i; \mu_{kt}, \Sigma_{kt}) \alpha_{kt}}\end{aligned} \tag{5}$$

We denote $\tau_{ij} = p_{\theta_t}(Z_i = j | X_i = x_i)$. This gives us :

$$Q(\theta|\theta^t) = \sum_{i=1}^n \sum_{j=1}^p (\log \mathcal{N}(x_i; \mu_j, \Sigma_j) + \log \alpha_j) \tau_{ij} \tag{6}$$

So one can see that the E step involves computing the (τ_{ij}) which completely determine Q with a fixed θ_t.

For the M step, finding $\theta_{t+1} = (\alpha_{t+1}, \mu_{t+1}, \Sigma_{t+1})$ amounts to solving the following problem :

$$\begin{aligned} \min_{(\alpha_j),(\Sigma_j),(\mu_j)} \quad & -\sum_{i=1}^{n}\sum_{j=1}^{p}(\log \mathcal{N}(x_i;\mu_j,\Sigma_j) + \log \alpha_j)\tau_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{p} \alpha_j = 1 \\ & -\alpha_j < 0 \\ & \Sigma_j \in S_d^+(\mathbb{R}) \end{aligned} \tag{7}$$

We take the Lagrangian

$$\mathcal{L}(\alpha,\theta,\lambda) = -\sum_{i=1}^{n}\sum_{j=1}^{p}(\log \mathcal{N}(x_i;\mu_j,\Sigma_j) + \log \alpha_j)\tau_{ij} + \lambda_1(\sum_{j=1}^{p}\alpha_i - 1) + \lambda_2^T\alpha + cst \tag{8}$$

$\lambda_2 = 0$ from KKT as the constraint $-\alpha_j < 0$ is never saturated.

For α^*, one then have

$$\begin{aligned} & \frac{\partial \mathcal{L}}{\partial \alpha_j} = 0 \\ \Leftrightarrow & -\sum_{i=1}^{n}\frac{\tau_{ij}}{\alpha_j} + \lambda_1 = 0 \\ \Leftrightarrow & \alpha_j\lambda_1 = \sum_{i=1}^{n}\tau_{ij} \end{aligned} \tag{9}$$

$\sum_{j=1}^{p}\alpha_j^* = 1$ gives $\lambda_1 = \displaystyle\sum_{j=1}^{p}\sum_{i=1}^{n}\tau_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{p}\tau_{ij} = \sum_{i=1}^{n} 1 = n$. Thus $\alpha_j^* = \frac{1}{n}\sum_{i=1}^{n}\tau_{ij}$

For μ^*,

$$\begin{aligned} & \frac{\partial \mathcal{L}}{\partial \mu_j} = 0 \\ \Leftrightarrow & \sum_{i=1}^{n}\frac{1}{2}\tau_{ij}(\Sigma_j^{-1}x_i - \Sigma_j^{-1}\mu_j) = 0 \\ \Leftrightarrow & \sum_{i=1}^{n}\tau_{ij}\Sigma_j^{-1}x_i = \sum_{i=1}^{n}\tau_{ij}\Sigma_j^{-1}\mu_j \\ \Leftrightarrow & \mu_j = \frac{\sum_{i=1}^{n}\tau_{ij}x_i}{\sum_{i=1}^{n}\tau_{ij}} \end{aligned} \tag{10}$$

For Σ^*,

$$\frac{\partial \mathcal{L}}{\partial \Sigma_j} = \frac{\partial}{\partial \Sigma_j}\left(-\frac{1}{2}\sum_{i=1}^{n}\tau_{ij}\log(\det(\Sigma_j)) - \frac{1}{2}\sum_{i=1}^{n}\tau_{ij}(x_i-\mu_j)^T\Sigma_j^{-1}(x_i-\mu_j)\right) \tag{11}$$

and

$$-\log(\det(\Sigma_j)) = \log(\det(\Sigma_j^{-1}))$$

and in fact we take the derivative with regard to Σ_j^{-1},

$$\begin{aligned}&\frac{\partial}{\partial \Sigma_j^{-1}}\left(\frac{1}{2}\sum_{i=1}^{n}\tau_{ij}\log(\det(\Sigma_j^{-1})) - \frac{1}{2}\sum_{i=1}^{n}\tau_{ij}(x_i-\mu_j)^T\Sigma_j^{-1}(x_i-\mu_j)\right)\\ =&\frac{1}{2}\sum_{i=1}^{n}\tau_{ij}\Sigma_j - \frac{1}{2}\sum_{i=1}^{n}\tau_{ij}(x_i-\mu_j)(x_i-\mu_j)^T \qquad \text{using that } \frac{\partial \log(\det(A^{-1})}{\partial A} = A^T\end{aligned} \tag{12}$$

Thus

$$\begin{aligned}&\frac{\partial \mathcal{L}}{\partial \Sigma_j} = 0\\ \Leftrightarrow &\frac{1}{2}\sum_{i=1}^{n}\tau_{ij}\Sigma_j - \frac{1}{2}\sum_{i=1}^{n}\tau_{ij}(x_i-\mu_j)(x_i-\mu_j)^T = 0\\ \Leftrightarrow &\Sigma_j = \frac{\sum_{i=1}^{n}\tau_{ij}(x_i-\mu_j)(x_i-\mu_j)^T}{\sum_{i=1}^{n}\tau_{ij}}\end{aligned} \tag{13}$$

To summarize, to update θ_t in the maximization step at a fixed (τ_{ij}), we choose :

$$\begin{aligned}\alpha_j^* &= \frac{1}{n}\sum_{i=1}^{n}\tau_{ij}\\ \mu_j^* &= \frac{\sum_{i=1}^{n}\tau_{ij}x_i}{\sum_{i=1}^{n}\tau_{ij}}\\ \Sigma_j^* &= \frac{\sum_{i=1}^{n}\tau_{ij}(x_i-\mu_j)(x_i-\mu_j)^T}{\sum_{i=1}^{n}\tau_{ij}}\end{aligned} \tag{14}$$

That is for the iteration steps used in our implementation of the algorithm. For the termination condition, one stops when the change in the data log-likelihood is "small enough" i.e. when $|l(x,\theta_{t+1}) - l(x,\theta_t)| \leq \epsilon$ where $\epsilon = 0.001$ for instance. For the initialization, we take the parameters θ_0 to some random values.

One makes an observation regarding the calculation of the (τ_{ij}). The formulation (5) can rise an underflow issue as both numerator and denominator can be quite low. To prevent this numerical issue in our implementation, we use the log-sum-exp trick. The detail of the calculation of the (τ_{ij}) is the following :

$$\log(\tau_{ij}) = \log(\mathcal{N}(x_i;\mu_{jt},\Sigma_{jt})) + \log(\alpha_{jt}) - \log(\sum_{k=1}^{p}\mathcal{N}(x_i;\mu_{kt},\Sigma_{kt})\alpha_{kt}) \tag{15}$$

$$\log(\sum_{k=1}^{p}\mathcal{N}(x_i;\mu_{kt},\Sigma_{kt})\alpha_{kt}) = \log(\sum_{k=1}^{p}\exp(\log(\mathcal{N}(x_i;\mu_{kt},\Sigma_{kt})\alpha_{kt}))) \tag{16}$$

and we denote $a_k = \log(\mathcal{N}(x_i;\mu_{kt},\Sigma_{kt})) + \log(\alpha_{kt})$ and $A = \max_{k\in\{1,\dots,p\}} a_k$. Thus, using the log-sum-exp trick :

$$\begin{aligned}\log(\sum_{k=1}^{p}\exp(\log(\mathcal{N}(x_i;\mu_{kt},\Sigma_{kt})\alpha_{kt}))) &= \log(\sum_{k=1}^{p}\exp(a_k))\\ &= \log(\sum_{k=1}^{p}\exp(a_k)\exp(A-A))\\ &= A + \log(\sum_{k=1}^{p}\exp(a_k - A))\end{aligned} \tag{17}$$

This way, we ensure that the largest positive exponential term is around 1.

Finally,

$$\tau_{ij} = \exp\left(\log(\mathcal{N}(x_i;\mu_{jt},\Sigma_{jt})) + \log(\alpha_{jt}) - A - \log(\sum_{k=1}^{p}\exp(a_k - A))\right) \tag{18}$$

This trick is also used to compute the data log-likelihood in the termination criteria.

To test our EM algorithm, one samples a set of observations according to a Gaussian mixture law when $d = 2$ with a number of clusters $p = 3$, a number of sampled points $n = 5000$, (μ_j, Σ_j) chosen randomly and $\alpha_j = \frac{1}{p}$ constant. Figure (3) presents the resulting distribution of the n sampled points.

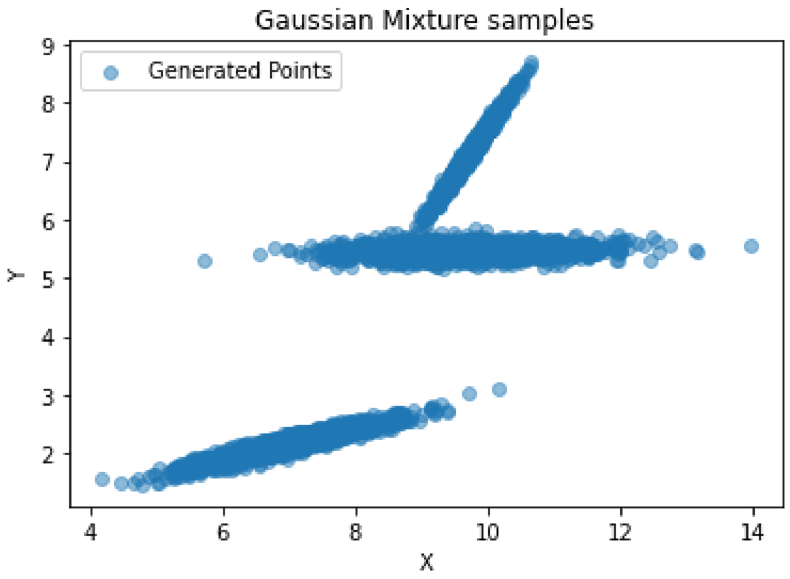


FIGURE 3 – Distribution of the points sampled according to our toy Gaussian mixture law

Figure (4) depicts the evolution of the log-likelihood over the number of iterations of our EM algorithm for the considered instance.

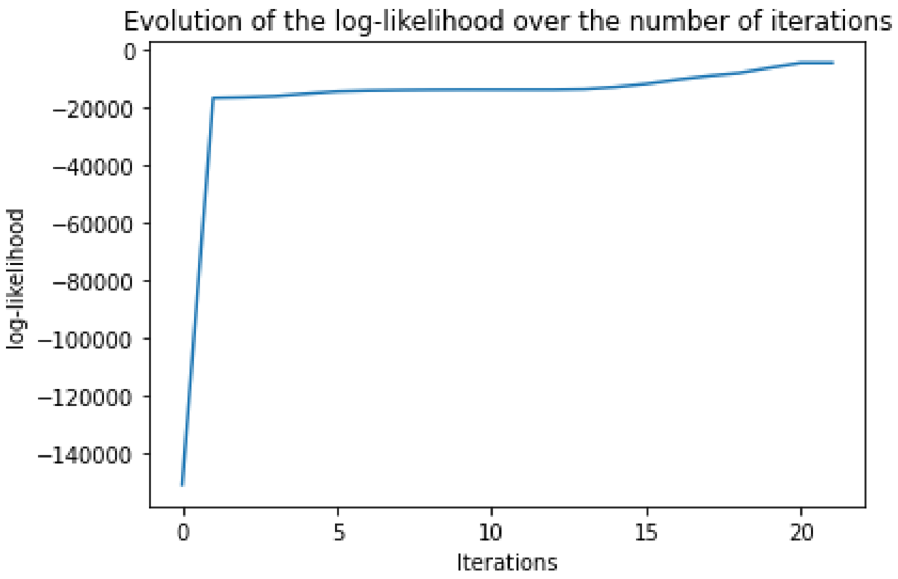


FIGURE 4 – Evolution of the log-likelihood over the number of iterations of the algorithm

One can see that the log-likelihood increases rapidly in a few iterations and then grows steadily until it converges after 22 iterations.

Figure (5) illustrates the prediction of the clusters. One can see graphically that the estimation seems close to the ground truth.

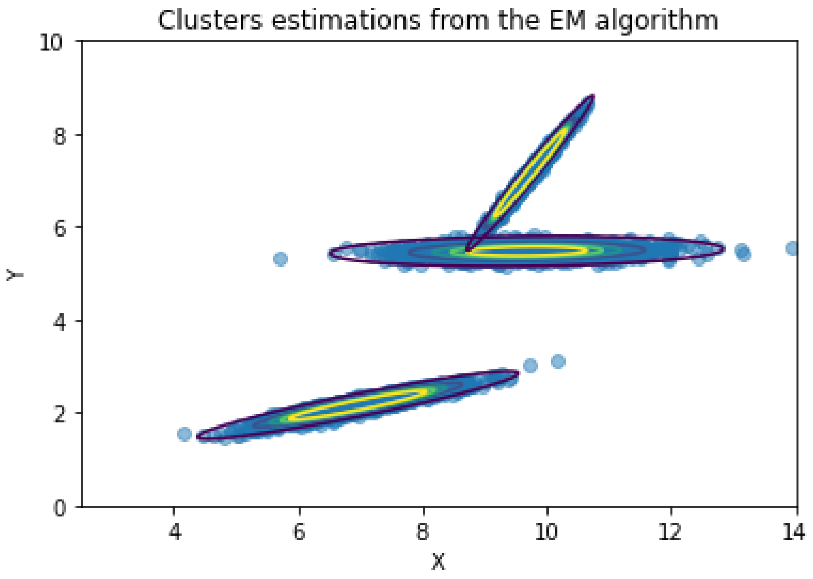


FIGURE 5 – Clusters estimations from the implemented EM algorithm for our toy Gaussian mixture distribution

A more detailed comparison between the original and estimated parameters yields :

```
   Original paramter alpha :
    [0.33333333 0.33333333 0.33333333]
  Estimated paramter alpha :
    [0.3368     0.33263935 0.33056065]
```

```
   Original paramter mu :
     [[9.67029839 5.47232249]
     [9.7268436  7.14815994]
     [6.97728825 2.16089496]]
  Estimated paramter mu :
     [[9.67819848 5.4709881 ]
     [9.73068448 7.15462061]
     [6.96039157 2.15799463]]
```

```
Original paramter Sigma :
 [[[0.96311181 0.01608244]
  [0.01608244 0.01003882]]

 [[0.07400008 0.11999459]
  [0.11999459 0.19904368]]

 [[0.61743774 0.16407237]
  [0.16407237 0.04907939]]]
Estimated paramter Sigma :
 [[[0.9845779  0.01481484]
  [0.01481484 0.01021869]]

 [[0.07773249 0.12579044]
  [0.12579044 0.20821874]]

 [[0.58948381 0.15680466]
  [0.15680466 0.04717793]]]
```

The estimated parameters seem to be really close to the original ones for each cluster. We can be confident that the algorithm has converged at least to a solution near the global minimum.

5.We will apply the implemented EM algorithm to the *Crude Birth/Death Rate* dataset to cluster the data. We are again in dimension $d = 2$, there are two variables 'CDR' and 'CBR'. Figure (6) gives the associated scatter graph.

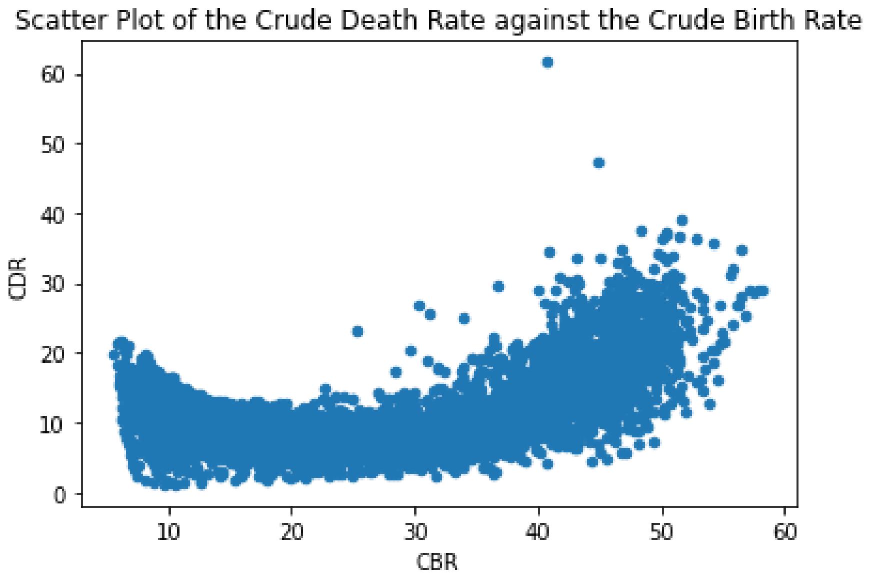


FIGURE 6 – Scatter Plot of the Crude Death Rate against the Crude Birth Rate

By visualizing the graph, we can indeed estimate that the distribution does not arise from a simple Gaussian law. A visual estimation might suggest that the distribution could be approximated by a Gaussian Mixture Model with 3 or 4 clusters.

6. This time we do not know beforehand the optimal number of clusters p. The BIC criterion is a useful tool to help with this decision. It is computed as follows :

$$\widehat{p} = \underset{p \geq 1}{\operatorname{argmin}} \left\{ -\log \mathcal{L}\left(x_1, \ldots, x_n; \theta\right) + \frac{\operatorname{df}(p)\log(n)}{2} \right\}$$

where df is the number of degrees of freedom of the mixture model with p clusters.

df is equivalent to the total number of free parameters in the model. We enumerate those for the parameters of the GMM :

- α has a total of $p - 1$ free parameters as we have the constraint $\sum_{j=1}^{p} \alpha_j = 1$
- $(\mu_j)_j$ has a total of pd free parameters
- $(\Sigma_j)_j$ is a sequence of p symmetric matrices of dimensions $d \times d$. So they each have $\frac{d(d+1)}{2}$ free parameters. So $(\Sigma_j)_j$ has a total of $p\frac{d(d+1)}{2}$ degrees of freedom.

We then conclude that $\operatorname{df}(p) = p - 1 + pd + p\frac{d(d+1)}{2}$. We run the implemented EM algorithm with $p \in \{2, 3, 5, 7, 9, 10\}$ and compute for each value of p the associated BIC on the instance.

Figure (7) shows the different values of the BIC criterion for the numbers of p tested on the instance. The BIC criterion keeps decreasing until we reach $p = 9$. Thus, 9 seems to be the value minimizing this metric. However, it should be noted that the BIC values become very close to the one obtained for $p = 9$ starting from $p = 5$ with a relatively small decrease beyond this threshold.

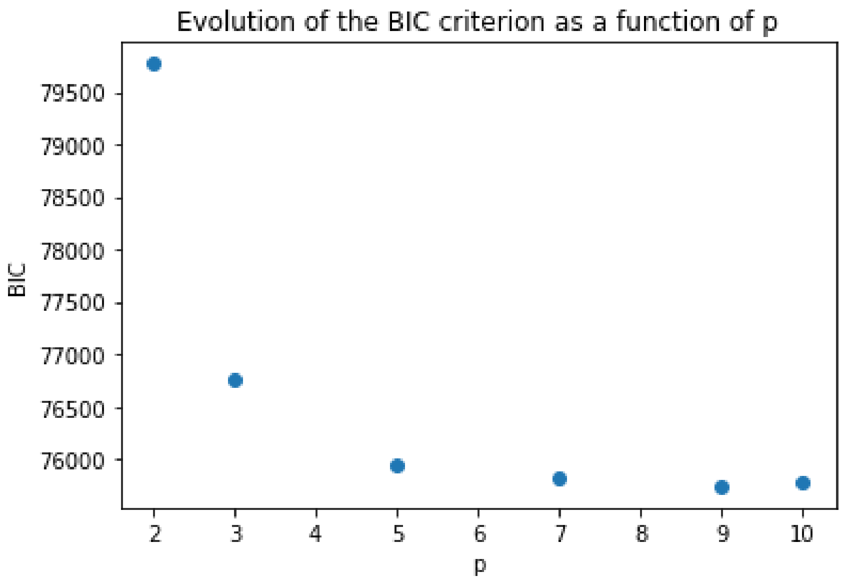
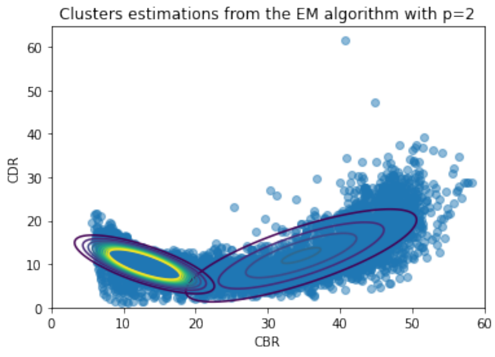
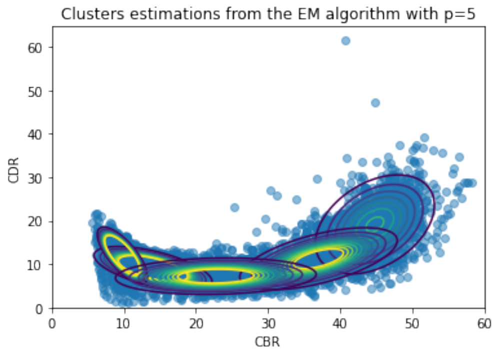


FIGURE 7 – Plot of the BIC criterion for different numbers of clusters

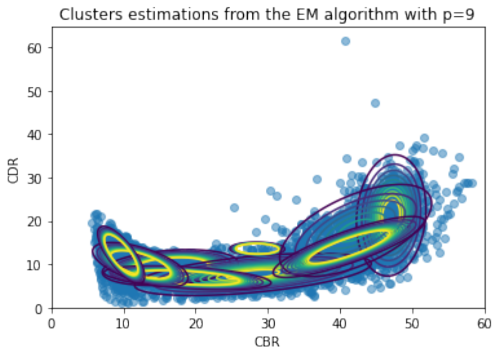
Here are a few representations of the estimated p.d.f. over the scatter plot for a few different p :



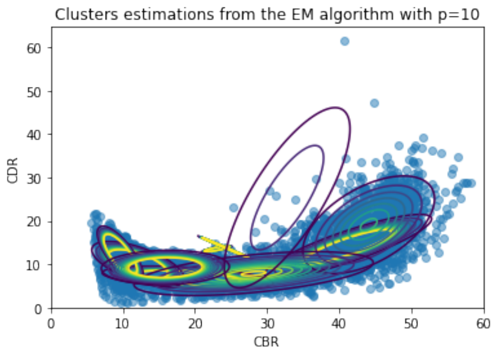
(a) Plot of the estimated p.d.f. over the scatter plot for $p = 2$



(b) Plot of the estimated p.d.f. over the scatter plot for $p = 5$



(c) Plot of the estimated p.d.f. over the scatter plot for $p = 9$



(d) Plot of the estimated p.d.f. over the scatter plot for $p = 10$

It appears that the clusters truly begin to conform to the shape of the plot only from $p = 5$. Based on the type of data, they cluster areas of the world at different period of time that exhibit 'CDR' and 'CBR' characteristics that are similar. (μ_j) are the centers of those clusters.

Exercise 3 : Importance sampling

3.A - Poor Importance Sampling

In this section, we will implement importance sampling in order to calculate the expectation of a function f defined by

$$f(x) = 2\sin\left(\frac{2\pi}{3}x\right)\mathbb{1}_{R^+}(x)$$

where x is distributed according to a unnormalized density p that is similar to a χ distribution. We will use a scaled normal distribution $\mathcal{N}(0.8, 1.5)$ as our sampling distribution where the parameters are chosen so that $p(x) < \gamma q(x)$ for all $x \in \mathbb{R}^+$where $\gamma \in \mathbb{R}^+$. Let consider

$$p(x) = x^{0.65}\mathrm{e}^{-\frac{x^2}{2}}\mathbb{1}_{\mathbb{R}+}(x) \quad \text{and} \quad q(x) = \frac{2}{\sqrt{2\pi(1.5)}}\mathrm{e}^{-\frac{((0.8)-x)^2}{2\times 1.5}}.$$

Note that neither p nor q are proper density functions here without normalization.

The importance weights are computed as follows :

$$\frac{p(x)}{q(x)} = \frac{\sqrt{3\pi}}{2}x^{0.65}\exp\left(-\frac{x^2+3.2x-1.28}{6}\right) \tag{19}$$

If $(X_1, \ldots, X_n)$ is a sample from q, as p is only known up to a normalizing constant, $\mathbb{E}_p[f(X)]$ can therefore be approximated by

$$\frac{1}{n}\sum_{i=1}^{n}\tilde{w}_i f(X_i) \quad \text{where} \quad \tilde{w}_i = \frac{w_i}{\frac{1}{n}\sum_{j=1}^{n} w_j} \quad \text{with} \quad \omega_i = \frac{p\,(X_i)}{q\,(X_i)} \tag{20}$$

$(\tilde{w}_i)_i$ are denoted the normalized importance weights. The simple importance sampling procedure is implemented for those functions. Figure (9) shows the evolution of the mean and the variance of the importance sampling estimator of $\mathbb{E}_p[f(X)]$ with a distribution q of mean $\mu = 0.8$. The variance of the estimator decreases as the sample size increases and we converge towards the value 0.6808 with a variance of 0.000233 with $N = 10^4$.

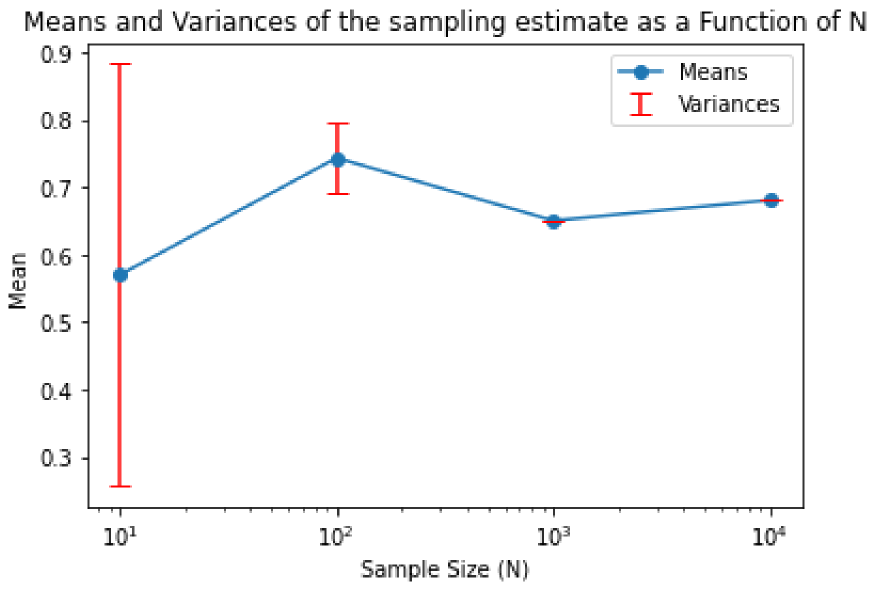


FIGURE 9 – The means and variances are computed by repeating the estimator calculation procedure 10 times for each sample size. The sample size axis is on a logarithmic scale

If we shift the mean q to $\mu = 6$ the centers of mass for each distribution become far apart compare to the previous case as shown in (Figure 10).

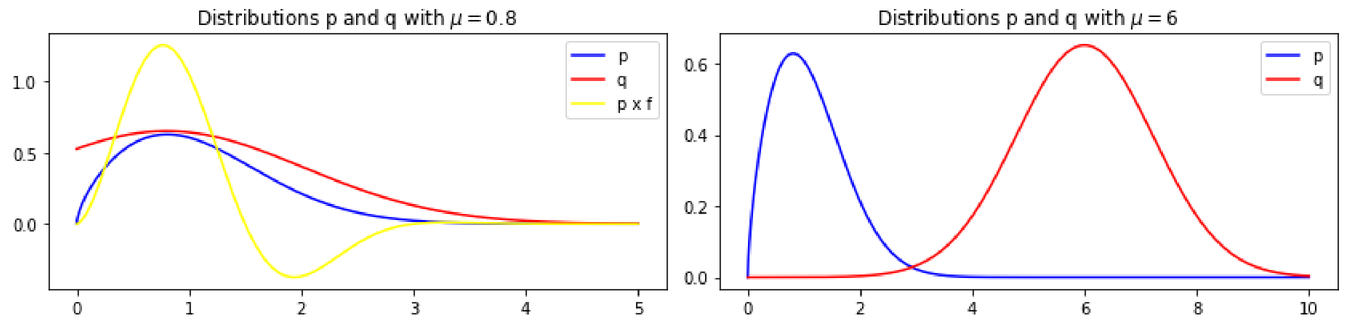


FIGURE 10 – Distributions p and q for two choices of mean μ.

In this configuration, the vast majority of points x sampled according to q will yield $p(x)f(x)$ and $p(x)$ values that are extremely low. We can predict that the normalized importance weights $(\tilde{w}_i)$ will then be quite low. We repeat the previous experiment, we obtain the results in Figure 11. This time, the method does not seem to work as there is convergence towards 0 with a decreasing variance. The main reason of this failure is that we do not have anymore $\text{Supp}(p \times f) \subset \text{Supp}(q)$. Thus, we are sampling far from crucial regions resulting in low $(\tilde{w}_i)$ values for a significant portion of samples (x_i).

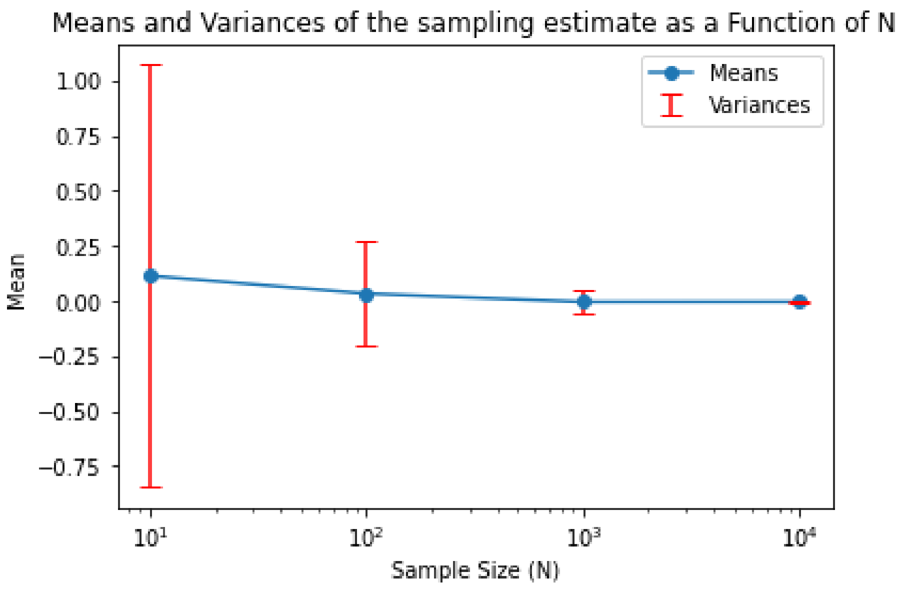


FIGURE 11 – The means and variances are computed by repeating the estimator calculation procedure 10 times for each sample size. The sample size axis is on a logarithmic scale

We compare the normalized importance weights for both values of μ :

```
    Normalized importance weights metrics for mu=0.8 and for N=10000

    Mean :  1.0000000000000002
    Variance :  0.7448347447448481
    1st Quantile :  0.0
    Median :  0.9874827254991875
    3rd Quantile :  1.8936084560613955


    Normalized importance weights metrics for mu=6 and for N=10000

    Mean :  1.0
    Variance :  14.724238098461251
    1st Quantile :  0.0025897915140211123
    Median :  0.034770452962260424
    3rd Quantile :  0.3355912150775743
```

We indeed observe that the distribution of normalized weights for $\mu = 6$ is predominantly centered near 0. It is mainly the outliers that have ventured into the region of interest that are contributing to the estimator. However, this contribution diminishes as N increases. It is also a good example that poor choice of the distribution q increases the variance of the weights.

3.B Adaptive Importance Sampling

4. The empirical criterion in step (iii) of the Population Monte Carlo Algorithm to maximize is similar to the data log-likelihood that we have to maximize in the EM but with weights $\tilde{w_i}^{(t)}$: $\sum_{i=1}^n \tilde{w_i}^{(t)} \log q_\theta(x_i^{(t)})$. Thus, Q would be rewritten like this :

$$Q(\theta|\theta^t) = \sum_{i=1}^n \mathbb{E}_{Z_i \sim p_{\theta_t}(.|X_i = x_i)}[\tilde{w_i} \log(p_\theta(x_i, Z_i))] \tag{21}$$

giving the new function to maximize in the M step :

$$Q(\theta|\theta^t) = \sum_{i=1}^n \sum_{j=1}^p (\log \mathcal{N}(x_i; \mu_j, \Sigma_j) + \log \alpha_j)\tilde{w_i}\tau_{ij} \tag{22}$$

We can easily derive the following parameter updates based on our previous results :

$$\begin{aligned} \hat{\alpha}_j &= \frac{1}{n}\sum_{i=1}^n \tilde{w_i}\tau_{ij} \\ \hat{\mu}_j &= \frac{\sum_{i=1}^n \tilde{w_i}\tau_{ij}x_i}{\sum_{i=1}^n \tilde{w_i}\tau_{ij}} \\ \hat{\Sigma}_j &= \frac{\sum_{i=1}^n \tilde{w_i}\tau_{ij}(x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^n \tilde{w_i}\tau_{ij}} \end{aligned} \tag{23}$$

as it is like we are dealing with a new $\tilde{\tau_{ij}} = \tilde{w_i}\tau_{ij}$. Just a remark for $\hat{\alpha}_j$, there is still a division by n since if we modify (9) with our new values, we have :

$$\lambda_1 = \sum_{j=1}^p \sum_{i=1}^n \tilde{w_i}\tau_{ij} = \sum_{i=1}^n \tilde{w_i} \sum_{j=1}^p \tau_{ij} = \sum_{i=1}^n \tilde{w_i} = n \tag{24}$$

because $\sum_{i=1}^n \tilde{w_i} = \frac{\displaystyle\sum_{i=1}^n w_i}{\frac{1}{n}\displaystyle\sum_{j=1}^n w_j} = n$

Hence, we can run the Population Monte Carlo Algorithm by solving a log-likelihood maximization at each step (iii) using the EM algorithm.

3.C Application to a "banana"-shaped density

5. We apply the Population Monte Carlo Algorithm to sample according to the density $\nu(x)$ which is known up to a constant. As explained in 4., we resolve each time the step (iii) with a modified EM algorithm. Step (ii) consists of computing the normalized importance weights. From an implementation point of view, this requires some rewriting. Calculating $\tilde{w_i}(x_i)$ requires computing :

$$\frac{\nu(x)}{q_{\theta^{(t)}}(x)} = \frac{1}{K}\frac{\Phi(x_1, x_2 + b(x_1^2 - \sigma_1^2), x_3, ..., x_d)}{\sum_{j=1}^{p}\alpha_{jt}\mathcal{N}(x;\mu_{jt},\Sigma_{jt})} = \frac{1}{K}c(x) \tag{25}$$

where K is the normalization constant of Φ.

For the same risk of underflow or overflow when computing the (τ_{ij}), we compute this quantity as a log :

$$\log(w_i) = -\log(K)+\log(\Phi(x_1, x_2+b(x_1^2-\sigma_1^2), x_3, ..., x_d))-\log\left(\sum_{j=1}^{p}\alpha_{jt}\mathcal{N}(x;\mu_{jt},\Sigma_{jt})\right) = -\log(K)+\log(c_i(x)) \tag{26}$$

$\log\left(\sum_{j=1}^{p}\alpha_{jt}\mathcal{N}(x;\mu_{jt},\Sigma_{jt})\right)$ is computed using the same log-sum-exp trick as for the (τ_{ij}) calculations. And we have

$$\begin{aligned}\log(\tilde{w_i}) &= \log(w_i) + \log(n) - \log\left(\sum_{j=1}^{n} w_j\right)\\ &= -\log(K) + \log(c_i) + \log(n) + \log(K) - \log\left(\sum_{j=1}^{n} c_j\right)\\ &= \log(c_i) + \log(n) - \log\left(\sum_{j=1}^{n}\exp(\log c_j)\right)\\ &= \log(c_i) + \log(n) - B - \log\left(\sum_{j=1}^{n}\exp(b_j - B)\right)\end{aligned} \tag{27}$$

again using the log-sum-exp trick, $b_j = \log c_j$ and $B = \max b_j$

We display the results in Figure 12 with the projection in the two first dimensions of the distribution. We choose $b = 1$ to accentuate the curvature of the "banana" shape.

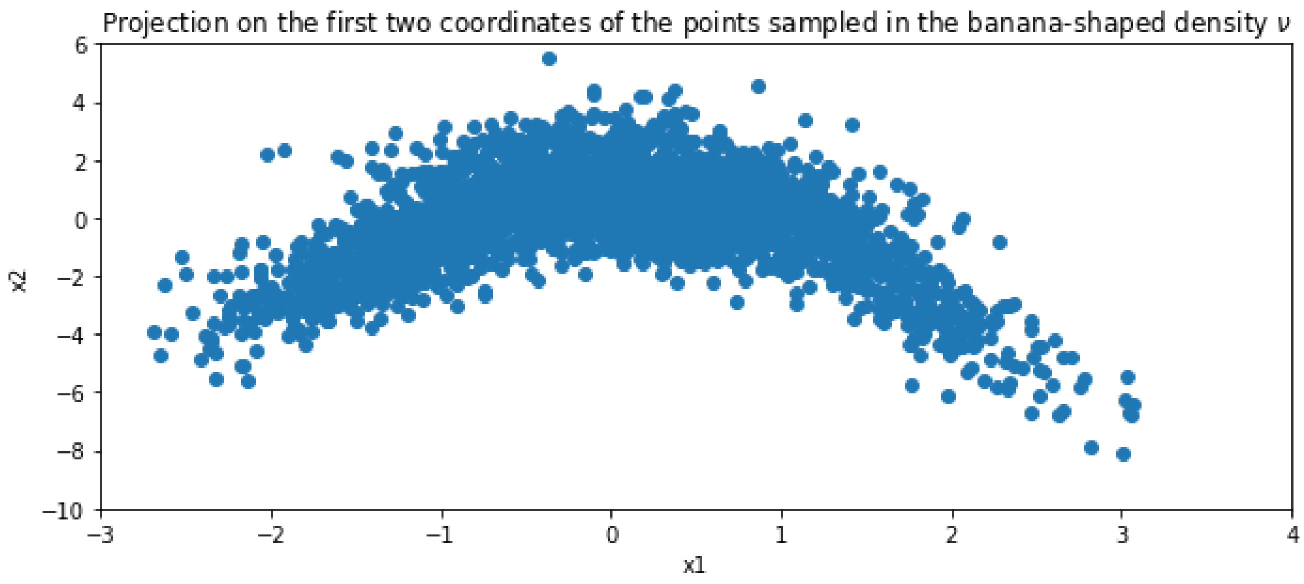


FIGURE 12 – Samples drawn from the density ν thanks to the Population Monte Carlo Algorithm with Gaussian mixture. Samples are projected on the first two dimensions. We have chosen $n = 3000$, $b = 1$ and $\sigma_1^2 = 1$