Graphical models - MVA 2017/2018
Homework 3

Rémi Lespinet

1 HMM Implementation

The HMM model that we use in this homework is represented in the figure 1.

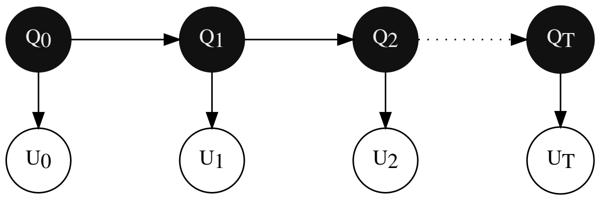


Figure 1: Representation of the HMM model

1. Recurence formulas

With these notations, the alpha-beta recursion formulas are given by

$$\begin{cases} \alpha_0(q_0) & = p(u_0 | q_0) p(q_0) \\ \alpha_{t+1}(q_{t+1}) & = p(\overline{u_{t+1}} | q_{t+1}) \sum_{q_t} p(q_{t+1} | q_t) \alpha_t(q_t) \end{cases}$$

and

$$\begin{cases} \beta_T(q_T) & = 1 \\ \beta_t(q_t) & = \sum_{q_{t+1}} p(q_{t+1} | q_t) p(\overline{u_{t+1}} | q_{t+1}) \beta_{t+1}(q_{t+1}) \end{cases}$$

Log-coding

In the computation, we use the log coding which is given by the formulas

$$\log \alpha_{t+1}(q_{t+1}) = \log p(\overline{u_{t+1}} | q_{t+1}) + \log \sum_{q_t} \exp \Big(\log p(q_{t+1} | q_t) + \log \alpha_t(q_t) \Big)$$

and

$$\log \beta_t(q_t) = \log \sum_{q_{t+1}} \exp \Big(\log p(q_{t+1} | q_t) + \log p(\overline{u_{t+1}} | q_{t+1}) + \log \beta_{t+1}(q_{t+1}) \Big)$$

And we use that

$$\log \sum_{i=1}^{n} \exp(x_i) = a + \log \sum_{i=1}^{n} \exp(x_i - a)$$

with

$$a = \max_{i=1,\dots,n} x_i$$

(This is done in the `log_sum_exp` function)

We can then compute

$$\log p(u_0 \dots u_T) = \log \sum_{q_t} \exp\left(\log \alpha_t(q_t) + \log \beta_t(q_t)\right)$$

and

$$\begin{aligned}\log \gamma_t(q_t) &= \log p(q_t \,|\, u_0, \dots u_T)\\ &= \log\left(\frac{\alpha_t(q_t)\beta_t(q_t)}{p(u_0 \dots u_T)}\right)\\ &= \log \alpha_t(q_t) + \log \beta_t(q_t) - \log p(u_0 \dots u_T)\end{aligned}$$

finally,

$$\begin{aligned}\log \xi_{t,t+1}(q_t) &= \log p(q_{t+1}, q_t \,|\, u_0, \dots u_T)\\ &= \log\left(\frac{\alpha_t(q_t)\beta_{t+1}(q_{t+1})p(q_{t+1}\,|\,q_t)p(u_{t+1}\,|\,q_{t+1})}{p(u_0 \dots u_T)}\right)\\ &= \log \alpha_t(q_t) + \log \beta_{t+1}(q_{t+1}) + \log p(q_{t+1}\,|\,q_t)\\ &+ \log p(u_{t+1}\,|\,q_{t+1}) - \log p(u_0 \dots u_T)\end{aligned}$$

2. In this question, we are using the same parameters for the means and covariances of the Gaussians as the one obtained Homework 2 i.e. The table 1 summarize the values of the parameters. These parameters are also represented in figure 2.

	Gaussian 0	Gaussian 1	Gaussian 2	Gaussian 3
means	$\mu_0 = \begin{pmatrix} 3.80 \\ -3.80 \end{pmatrix}$	$\mu_1 = \begin{pmatrix} 3.98 \\ 3.77 \end{pmatrix}$	$\mu_2 = \begin{pmatrix} 3.06 \\ -3.53 \end{pmatrix}$	$\mu_3 = \begin{pmatrix} 2.03 \\ 4.17 \end{pmatrix}$
cov.	$\Sigma_0 = \begin{pmatrix} 0.92 & 0.06 \\ 0.06 & 1.87 \end{pmatrix}$	$\Sigma_1 = \begin{pmatrix} 0.21 & 0.29 \\ 0.29 & 12.24 \end{pmatrix}$	$\Sigma_2 = \begin{pmatrix} 6.24 & 6.05 \\ 6.05 & 6.18 \end{pmatrix}$	$\Sigma_3 = \begin{pmatrix} 2.90 & 0.21 \\ 0.21 & 2.76 \end{pmatrix}$

Table 1: Parameters learnt by Expectation Maximization on Gaussian mixtures in previous homework for the same dataset

The first 100 points of $p(q_t \,|\, u_0 \dots u_T)$ are represented in figure 3. Even if have taken a relatively uniform transition probability matrix (the probabilities, $p(q_{t+1} = i \,|\, q_t = j)$ are equals given $i \neq j$), we see some patterns in the data, which make us believe that the data has probably been generated with a HMM (in which case we see that the probability of staying in state 2 seems to be high, the probability of reaching state 3 from state 1 also seem to be high (and conversely))

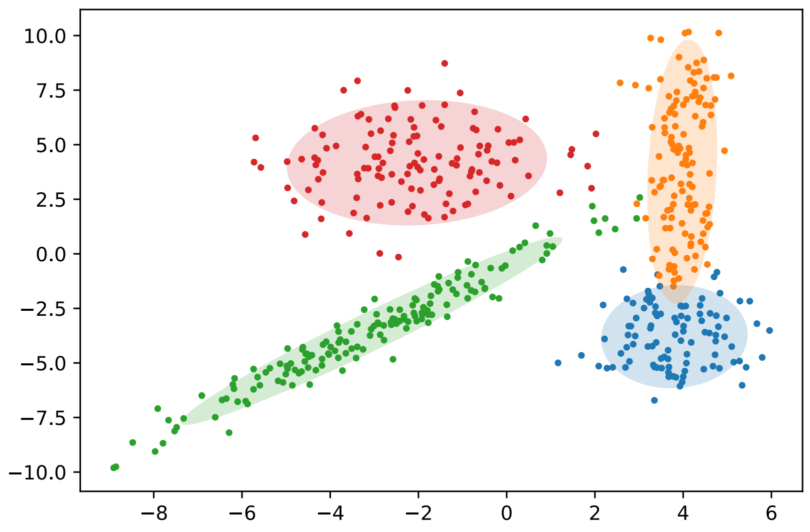


Figure 2: Representation of the means and covariances learnt by Expectation Maximization on a Gaussian mixture (4 kernels) in the previous homework for the same dataset (Ellipses contain 95% of the distributions)

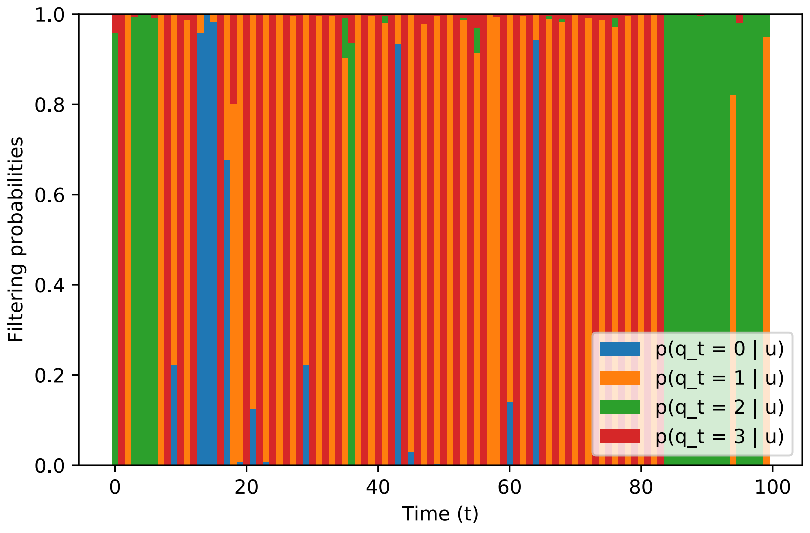


Figure 3: Representation of the filtering probabilities $p(q_t \mid u_0, \dots u_T)$ for each state for the HMM with chosen parameters (not trained)

3. Let us consider the case where each q_t take M values. The parameters of the model are

$$\theta = (\pi, A, \mu_0, \Sigma_0, \dots \mu_M, \Sigma_M)$$

We have

$$p_\theta(q, y) = p_\theta(q_0) \prod_{t=0}^{T-1} p_\theta(q_{t+1}|q_t) \prod_{t=0}^{T} p_\theta(u_t|q_t)$$

The complete log likelihood in the case of this HMM is expressed as

$$\mathcal{L}(\theta, q, y) = \log p_\theta(q_0) + \sum_{t=0}^{T-1} \log p_\theta(q_{t+1}|q_t) + \sum_{t=0}^{T} \log p_\theta(u_t|q_t)$$

Using the one hot encoding

$$q_t^i = \begin{cases} 1 & \text{if } q_t = i \\ 0 & \text{otherwise} \end{cases}$$

We can rewrite the first term as

$$\log p_\theta(q_0) = \log \pi(q_0) = \log \prod_{i=1}^{M} \pi_i^{q_0^i} = \sum_{i=1}^{M} q_0^i \log \pi_i$$

The second term can be simplified as

$$\log p_\theta(q_{t+1}|q_t) = \log a_{q_t,q_{t+1}} = \log \prod_{i,j=1}^{M} a_{i,j}^{q_t^i q_{t+1}^j} = \sum_{i,j=1}^{M} q_t^i q_{t+1}^j \log a_{i,j}$$

Finally, we have

$$\log p_\theta(u_t|q_t) = -\frac{1}{2}\log(2\pi)^d - \frac{1}{2}\log\det\Sigma_{q_t} - \frac{1}{2}(u_t - \mu_{q_t})^T \Sigma_{q_t}^{-1}(u_t - \mu_{q_t})^T$$

Which can also be rewriten using the one hot encoding as

$$\log p_\theta(u_t|q_t) = -\frac{1}{2}\log(2\pi)^d - \frac{1}{2}\sum_{i=1}^{M} q_t^i \log\det\Sigma_i - \frac{1}{2}\sum_{i=1}^{M} q_t^i (u_t - \mu_i)^T \Sigma_i^{-1}(u_t - \mu_i)^T \qquad (1)$$

Hence, the complete log likelihood is given by

$$\boxed{\begin{aligned} \mathcal{L}(\theta, q, y) = &\sum_{i=1}^{M} q_0^i \log \pi_i + \sum_{t=0}^{T-1}\sum_{i,j=1}^{M} q_t^i q_{t+1}^j \log a_{i,j} \\ &- \frac{1}{2}\log(2\pi)^d - \frac{1}{2}\sum_{t=0}^{T}\sum_{i=1}^{M} q_t^i \log\det\Sigma_i - \frac{1}{2}\sum_{t=0}^{T}\sum_{i=1}^{M} q_t^i (u_t - \mu_i)^T \Sigma_i^{-1}(u_t - \mu_i)^T \end{aligned}}$$

E step

Taking the same notation as in the polycopié, we write

$$\gamma_t^i \triangleq p(q_t = i | u_0, \dots u_T)$$

and

$$\xi_{t,t+1}^{i,j} \triangleq p(q_t = i, q_{t+1} = j | u_0, \dots u_T)$$

We then compute the following expectations

$$\begin{aligned} \mathbb{E}_{q_0,\dots q_T}\left[q_t^i q_{t+1}^j | u_0, \dots u_T\right] &= \sum_{q_t, q_{t+1}} q_t^i q_{t+1}^j p(q_t, q_{t+1} | u_0, \dots u_T) \\ &= p(q_t = i, q_{t+1} = j | u_0, \dots, u_T) \\ &= \xi_{t,t+1}^{i,j} \end{aligned}$$

And

$$\begin{aligned}
\mathbb{E}_{q_0,\dots q_T}\left[q_t^i | u_0,\dots u_T\right] &= \sum_{q_t} q_t^i p(q_t | u_0,\dots u_T) \\
&= p(q_t = i | u_0,\dots,u_T) \\
&= \gamma_t^i
\end{aligned}$$

We can then write the expected complete log likelihood

$$\begin{aligned}
\mathbb{E}_{q_0,\dots q_T}\left[\mathcal{L}(\theta; q, y) | u_0,\dots u_T\right] &= \sum_{i=1}^{M} \gamma_0^i \log \pi_i + \sum_{t=0}^{T-1}\sum_{i,j=1}^{M} \xi_{t,t+1}^{i,j} \log a_{i,j} \\
&- \frac{1}{2}\log (2\pi)^d - \frac{1}{2}\sum_{t=0}^{T}\sum_{i=1}^{M} \gamma_t^i \log\det\Sigma_i \\
&- \frac{1}{2}\sum_{t=0}^{T}\sum_{i=1}^{M} \gamma_t^i (u_t - \mu_i)^T \Sigma_i^{-1}(u_t - \mu_i)
\end{aligned}$$

M step

Looking at the expression of the expected complete log-likelihood, we see that we can minimize separately over π, $(a_{i,j})$ and (u_i, Σ_i)

Maximizing over π

We want to maximize

$$\begin{aligned}
&\text{maximize} \quad \sum_{i=1}^{M} \gamma_0^i \log \pi_i \\
&\text{subject to} \quad \pi_1 + \cdots + \pi_M = 1
\end{aligned}$$

The Lagrangian is

$$\mathcal{L}(\pi, \lambda) = \sum_{i=1}^{M} \gamma_0^i \log \pi_i - \lambda\left(\sum_{i=1}^{M} \pi_i - 1\right)$$

This is a concave problem (as a non negative weighted sum of concave functions), we have feasibility (there is no inequatity constraints), hence by Slater constraint qualification strong duality holds. The KKT conditions are

$$\begin{cases}
\forall i \in \{1 \dots M\}, \ \dfrac{\gamma_0^i}{\pi_i} - \lambda = 0 \\
\pi_0 + \cdots + \pi_M = 1
\end{cases}$$

Posing

$$\hat{\pi}_i = \frac{\gamma_0^i}{\sum_{i=1}^{M} \gamma_0^i}$$

and

$$\hat{\lambda} = \sum_{i=1}^{M} \gamma_0^i$$

we see that the couple $(\hat{\pi}, \hat{\lambda})$ verify the KKT conditions and is thus primal-dual optimal. Since

$$\sum_{i=1}^{M} \gamma_0^i = \sum_{i=1}^{M} p(q_t = i | u_0, \dots u_T) = 1$$

We finally obtain

$$\boxed{\hat{\pi}_i = \gamma_0^i}$$

Maximizing over A

We want to maximize

$$\begin{aligned} &\text{maximize} && \sum_{t=0}^{T-1} \sum_{i,j=1}^{M} \xi_{t,t+1}^{i,j} \log a_{i,j} \\ &\text{subject to} && a_{i,1} + \dots + a_{i,M} = 1, \ i = 1 \dots M \end{aligned}$$

The lagrangian is

$$\mathcal{L}(\pi, \lambda) = \sum_{t=0}^{T-1} \sum_{i,j=1}^{M} \xi_{t,t+1}^{i,j} \log a_{i,j} - \sum_{i=1}^{M} \lambda_i \left(\sum_{j=1}^{M} a_{i,j} - 1 \right)$$

This is also a concave problem (for the same reasons), it is also feasible, hence by Slater constraint qualification strong duality holds. The KKT conditions are

$$\begin{cases} \forall i, j \in \{1 \dots M\}, \ \sum_{t=0}^{T-1} \dfrac{\xi_{t,t+1}^{i,j}}{a_{i,j}} - \lambda_i = 0 \\ \forall i \in \{1 \dots M\}, \ a_{i,1} + \dots + a_{i,M} = 1 \end{cases}$$

Summing the first condition on the row, we obtain

$$\sum_{t=0}^{T-1} \sum_{j=1}^{M} \xi_{t,t+1}^{i,j} - \sum_{j=1}^{M} a_{i,j} \lambda_i = 0$$

Using the second condition, we have

$$\lambda_i = \sum_{t=0}^{T-1} \sum_{j=1}^{M} \xi_{t,t+1}^{i,j}$$

Thus,

$$\boxed{a_{i,j} = \frac{\sum_{t=0}^{T-1} \xi_{t,t+1}^{i,j}}{\sum_{j=1}^{M} \sum_{t=0}^{T-1} \xi_{t,t+1}^{i,j}}}$$

Maximizing over (μ_i, Σ_i)

The optimization problem that we want to solve is

$$\text{maximize} \quad -\sum_{t=0}^{T}\sum_{i=1}^{M}\gamma_t^i \log\det\Sigma_i - \sum_{t=0}^{T}\sum_{i=1}^{M}\gamma_t^i (u_t-\mu_i)^T\Sigma_i^{-1}(u_t-\mu_i)$$

Let $\Omega = \Sigma^{-1}$, because

$$\begin{array}{ccc} \mathcal{S}_n^{++} & \rightarrow & \mathcal{S}_n^{++} \\ X & \rightarrow & X^{-1} \end{array}$$

is one-to-one with image covering $\mathbb{S}_n^{++}$ (set of positive semi-definite matrices), we can solve the following optimization problem

$$\text{maximize} \quad \sum_{t=0}^{T}\sum_{i=1}^{M}\gamma_t^i \log\det\Omega_i - \sum_{t=0}^{T}\sum_{i=1}^{M}\gamma_t^i (u_t-\mu_i)^T\Omega_i(u_t-\mu_i)$$

and a solution (μ_i, Ω_i) to this problem obtains a solution to the original problem : (μ_i, Ω_i^{-1}). With this change of variable the objective function is now concave ($X \rightarrow \log\det(X)$ is concave) and $X \rightarrow a^T Xb$ is linear.

We can minimize w.r.t μ_i first, this is a concave function of μ_i (covariance matrix is symmetric positive semidefinite, and so is it's inverse) taking the gradient leads to

$$-2\sum_{t=0}^{T}\gamma_t^i\Omega_i(u_t-\mu_i) = 0$$

Hence, assuming that the covariance matrix is definite,

$$\sum_{t=0}^{T}\gamma_t^i\mu_i = \sum_{t=0}^{T}\gamma_t^i u_t$$

and

$$\boxed{\mu_i = \frac{\sum_{t=0}^{T}\gamma_t^i u_t}{\sum_{t=0}^{T}\gamma_t^i}}$$

Which is simply the average of the observations u_t such that $q_t = i$. We can now derive with respect to Ω_i. First, we notice that

$$\begin{aligned}\sum_{t=0}^{T}\sum_{i=1}^{M}\gamma_t^i (u_t-\mu_i)^T\Omega_i(u_t-\mu_i) &= \sum_{t=0}^{T}\sum_{i=1}^{M}\gamma_t^i \mathrm{Tr}\left((u_t-\mu_i)^T\Omega_i(u_t-\mu_i)\right) \\ &= \sum_{t=0}^{T}\sum_{i=1}^{M}\gamma_t^i \mathrm{Tr}\left(\Omega_i(u_t-\mu_i)(u_t-\mu_i)^T\right)\end{aligned}$$

Hence setting the gradient to 0,

$$\sum_{t=0}^{T}\gamma_t^i\Omega_i^{-1} - \sum_{t=0}^{T}\gamma_t^i(u_t-\mu_i)(u_t-\mu_i)^T = 0$$

Hence,

$$\Omega_i^{-1} = \frac{\sum_{t=0}^T \gamma_t^i (u_t - \mu_i)(u_t - \mu_i)^T}{\sum_{t=0}^T \gamma_t^i}$$

And

$$\boxed{\Sigma_i = \frac{\sum_{t=0}^T \gamma_t^i (u_t - \mu_i)(u_t - \mu_i)^T}{\sum_{t=0}^T \gamma_t^i}}$$

4. The initial transition matrix and initial probability distribution are uniform, the parameters of the gaussian emission kernels have been initialized with the values obtained in the previous homework. The table 2 summarize the values of parameters (μ_i, Σ_i) learned by the EM algorithm. The table 3 represents the obtained transition matrix. As we expected in our discussion on figure 3, we see that the probability $A_{1,3}$, $A_{3,1}$ and $A_{2,2}$ (indicing the from 0) are very high which means that their corresponding probabilities $p(q_{t+1} = 1 \,|\, q_t = 3)$, $p(q_{t+1} = 3 \,|\, q_t = 1)$, $p(q_{t+1} = 2 \,|\, q_t = 2)$ are high. This means that the HMM successfuly learned the patterns that we observed (which correspond to the transition probabilities of the HMM that generated the data).

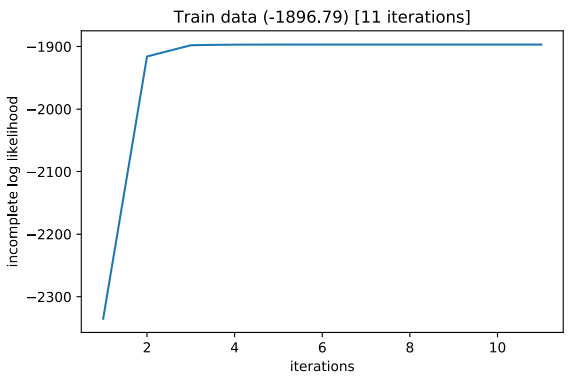
	Gaussian 0	Gaussian 1	Gaussian 2	Gaussian 3
means	$\mu_0 = \begin{pmatrix} 3.79 \\ -3.97 \end{pmatrix}$	$\mu_1 = \begin{pmatrix} 3.99 \\ 3.63 \end{pmatrix}$	$\mu_2 = \begin{pmatrix} -2.97 \\ -3.45 \end{pmatrix}$	$\mu_3 = \begin{pmatrix} -1.95 \\ 4.19 \end{pmatrix}$
cov.	$\Sigma_0 = \begin{pmatrix} 0.94 & 0.06 \\ 0.06 & 1.55 \end{pmatrix}$	$\Sigma_1 = \begin{pmatrix} 0.20 & 0.26 \\ 0.26 & 12.34 \end{pmatrix}$	$\Sigma_2 = \begin{pmatrix} 6.81 & 6.59 \\ 6.59 & 6.69 \end{pmatrix}$	$\Sigma_3 = \begin{pmatrix} 3.28 & 0.30 \\ 0.30 & 2.83 \end{pmatrix}$

Table 2: Parameters learned by Expectation Maximization on the HMM (on the train dataset) initialized with the values obtained in the previous homework

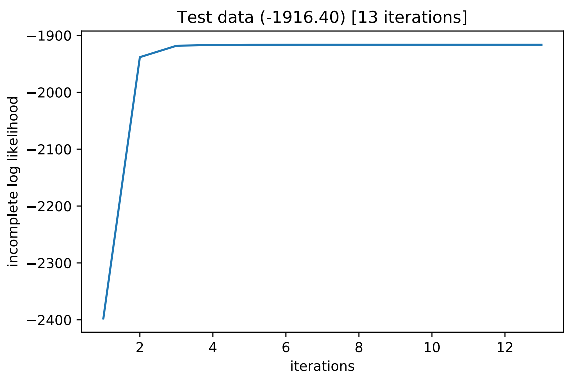
$$A = \begin{pmatrix} 0.88 & 0.05 & 0.03 & 0.04 \\ 0.01 & 0.02 & 0.03 & 0.93 \\ 0.02 & 0.07 & 0.90 & 0.01 \\ 0.05 & 0.87 & 0.06 & 0.02 \end{pmatrix}$$

Table 3: Transition matrix learned by Expectation Maximization on the HMM (on the train dataset)

5. The incomplete log likelihood $p_\theta(u_0, \dots u_T)$ for the test and train data as a function of the iterations of the EM algorithm are given in the figure 4. The algorithm stops when the variation of the incomplete log likelihood is below a threshold ϵ (in this case $\epsilon = 10^{-6}$). We notice that the algorithm converges in about 10 iterations in both cases (the initialization was very good). We also notice that the parameters learned are almost the same in both cases and that the likelihood obtained is very close. This means that they have probably been generated by the same HMM.



(a) Train data (500 points)



(b) Test data (500 points)

Figure 4: Evolution of the incomplete log likelihood for the train and test data with the number of iteration of the EM algorithm

6. The table 4 compares the incomplete log likelihood obtained on the train and test dataset for the HMM (after EM parameter learning) with the different models of the previous homework (Isotropic and full GMM after EM learning).

It makes sense to compare these models, because they all are expressing the incomplete log likelihood which can be viewed as the probability of observing these data given the model.

As already explained in the previous homework, it makes sense that the likelihood of the Full GMM is higher than the Isotropic GMM, because the Isotropic GMM is a particular case of the Full GMM (where the covariances matrices can be written $\sigma_i I$) and hence by learning on the Full GMM, we are maximizing on a larger set.

The exact same argument applies for the HMM and the Full GMM, the Full GMM is the particular case of a HMM whose transition matrix can be written as

$$A = \begin{pmatrix} \pi_0 & \pi_1 & \pi_2 & \pi_3 \\ \pi_0 & \pi_1 & \pi_2 & \pi_3 \\ \pi_0 & \pi_1 & \pi_2 & \pi_3 \\ \pi_0 & \pi_1 & \pi_2 & \pi_3 \end{pmatrix} \tag{$\star$}$$

Which means that the probability of the next state does not depend on the probability of the current state.

	Train	**Test**
Isotropic GMM	-2652.89	-2654.39
Full GMM	-2332.29	-2417.30
HMM	-1896.79	-1916.40

Table 4: Incomplete log likelihood of the train and test dataset obtained for different models

7. For the studied HMM, the Viterbi algorithm aims to compute

$$\max_{q_0 \cdots q_T} p(q_0 \dots q_T | u_0, \dots u_T)$$

This is the same as maximizing

$$\max_{q_0 \cdots q_T} p(q_0 \dots q_T, \overline{u_0}, \dots \overline{u_T})$$

This can be rewriten as

$$\max_{q_0 \cdots q_T} \left(p(q_0) \prod_{t=0}^{T} p(\overline{u_t} | q_t) \prod_{t=0}^{T-1} p(q_{t+1} | q_t) \right)$$

Algorithm derived from the book

This can be expanded to

$$\max_{q_T} \dots \max_{q_1} \Big(p(q_2 | q_1) p(\overline{u_1} | q_1) \max_{q_0} \Big(p(q_0) p(\overline{u_0} | q_0) p(q_1 | q_0) \Big) \Big)$$

Let

$$m_1(q_1) = \max_{q_0} \Big(p(q_0) p(\overline{u_0} | q_0) p(q_1 | q_0) \Big)$$

We can then rewrite the previous formula :

$$\max_{q_T} \dots \max_{q_1} \Big(p(q_2 | q_1) p(\overline{u_1} | q_1) m_1(q_1) \Big)$$

This leads to the following recurrence formula

$$\boxed{\begin{cases} m_0(q_0) = \pi(q_0) \\ m_{t+1}(q_{t+1}) = \max_{q_t} \Big(p(\overline{u_t} | q_t) p(q_{t+1} | q_t) m_t(q_t) \Big) \end{cases}}$$

At the end of the recurrence, we obtain

$$\max_{q_0 \cdots q_T} p(q_0 \dots q_T, \overline{u_0}, \dots \overline{u_T}) = \max_{q_T} \Big(m_{T-1}(q_T) p(\overline{u_T} | q_T) \Big)$$

Now, we want to compute the combination $(\hat{q}_0, \dots \hat{q}_T)$ that realizes this maximum. We can easily get a value for $\hat{q}_T$ by taking

$$\hat{q}_T = \arg\max_{q_T} \Big(m_{T-1}(q_T) p(\overline{u_T} | q_T) \Big)$$

now that we have the value $\hat{q}_T$, we would like to know the value of all the states $\hat{q}_t$ that have been chosen to obtain the maximum value of $\max_{q_0 \dots q_T} p(q_0 \dots q_T, \overline{u_0}, \dots \overline{u_T})$. This can be easily computed at the forward pass using the following recurrence formula :

$$s_t(q_{t+1}) = \arg\max_{q_t} \Big(p(\overline{u_t} | q_t) p(q_{t+1} | q_t) m_t(q_t) \Big)$$

We can then determine $\hat{q}_t$ by simply computing $s_t(\hat{q}_{t+1})$. The algorithm 1 resumes these steps and traduce it for the case of a HMM with Gaussian emissions (for computational issues, we compute sums of logarithms instead of the products directly)

Algorithm 1 Viterbi 1

```
1:  for $k = 0..K$ do
2:      $m[k] \leftarrow \log(\pi[k])$
3:  for $t = 0..(T-1)$ do
4:      for $q = 0..K$ do
5:          $new\_m[q] \leftarrow \max_{k=0}^{K}(m[k] + \log A_{k,q} + \log \mathcal{N}_{\mu_k,\Sigma_k}(observation[t]))$
6:          $s[t,q] \leftarrow \arg\max_{k=0}^{K}(m[k] + \log A_{k,q} + \log \mathcal{N}_{\mu_k,\Sigma_k}(observation[t]))$
7:      $m \leftarrow new\_m$
8:  $ids[T] \leftarrow \arg\max_k m + \log \mathcal{N}_{\mu_k,\Sigma_k}(observation[T]))$
9:  for $t = (T-1)..0$ do
10:     $ids[t] \leftarrow s[t, ids[t+1]]$
11: return $ids$
```

Other version of the algorithm

We see that for all $t \in \{1 \dots T\}$, $(q_0, \dots q_{t-1})$ and $(u_0, \dots u_{t-1})$ also form a Hidden markov chain, and we have

$$p(q_0 \dots q_t, \overline{u_0}, \dots \overline{u_t}) = p(q_0) \prod_{k=0}^{t} p(\overline{u_k}|q_k) \prod_{k=0}^{t-1} p(q_{k+1}|q_k)$$

Hence,

$$p(q_0 \dots q_t, \overline{u_0}, \dots \overline{u_t}) = p(q_0 \dots q_{t-1}, \overline{u_0}, \dots \overline{u_{t-1}}) p(\overline{u_t}|q_t) p(q_t|q_{t-1})$$

If we note

$$T_t(i) = \max_{q_0, \dots q_{t-1}} p(q_0 \dots q_t = i, \overline{u_0}, \dots \overline{u_t})$$

We obtain

$$T_t(i) = \max_{q_{t-1}} \Big(p(\overline{u_t}|q_t = i) p(q_t = i|q_{t-1}) T_{t-1}(q_{t-1}) \Big)$$

Which leads to the following recurence formula

$$\boxed{\begin{cases} T_0(i) = \pi(q_0 = i) p(\overline{u_0}|q_0 = i) \\ T_t(i) = \max_{q_{t-1}} \Big(p(\overline{u_t}|q_t = i) p(q_t = i|q_{t-1}) T_{t-1}(q_{t-1}) \Big) \end{cases}}$$

Now, we want to compute the combination $(\hat{q}_0, \dots \hat{q}_T)$ that realizes this maximum. We can keep track of which values of each state has been chosen by storing at each step of the computation

$$\begin{cases} S_0(i) = 0 \\ S_t(i) = \arg\max_{q_{t-1}} \Big(p(\overline{u_t}|q_t = i) p(q_t = i|q_{t-1}) T_{t-1}(q_{t-1}) \Big) \end{cases}$$

The algorithm 2 sumarize this version of the algorithm. It is more efficient, as the max is only done on 2 terms.

Algorithm 2 Viterbi 2

```
1: for $k = 0..K$ do
2:     $m[k] \leftarrow \log(\pi[k]) + \log \mathcal{N}_{\mu_k,\Sigma_k}(observation[0])$
3: for $t = 0..(T-1)$ do
4:     for $q = 0..K$ do
5:         $new\_m[q] \leftarrow \log \mathcal{N}_{\mu_q,\Sigma_q}(observation[t+1]) + \max_{k=0}^{K}(m[k] + \log A_{k,q})$
6:         $s[t,q] \leftarrow \arg\max_{k=0}^{K}(m[k] + \log A_{k,q})$
7:     $m \leftarrow new\_m$
8: $ids[T] \leftarrow \arg\max_k m[k]$
9: for $t = (T-1)..0$ do
10:     $ids[t] \leftarrow s[t, ids[t+1]]$
11: return $ids$
```

8. I've implemented both version (`Viterbi_1` and `Viterbi_2`). The most likely sequence of state is represented in figure 5. We see some points are assigned the yellow cluster (class 1) even though $p(u_t \,|\, q_t = 1)$ are very small (they are far from the cluster center). This traduces the fact that the transition matrix plays a very important role in this case (one coefficient of each line has very high probability).

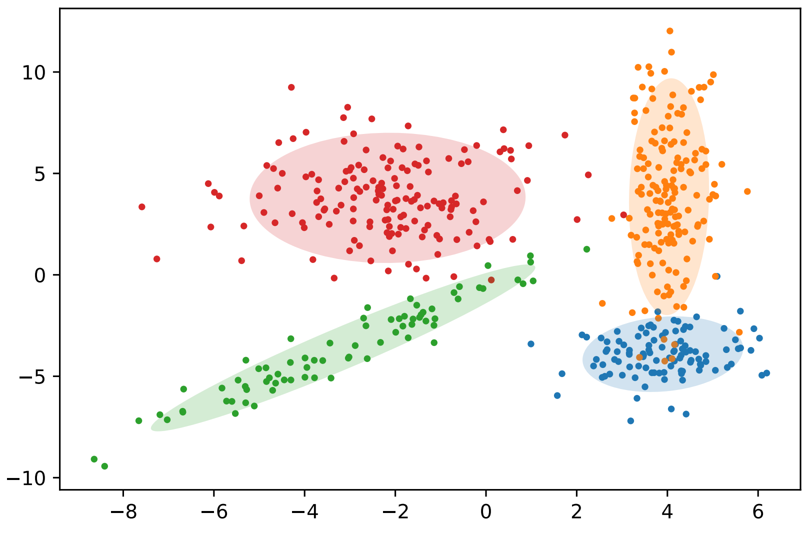


Figure 5: Most likely sequence of state returned by the Viterbi algorithm on the test data

9. The marginal probabilities $p(q_t = i \,|\, u_0, \dots u_T)$ for each state i in $\{0 \dots 3\}$ are given in the figure 6.

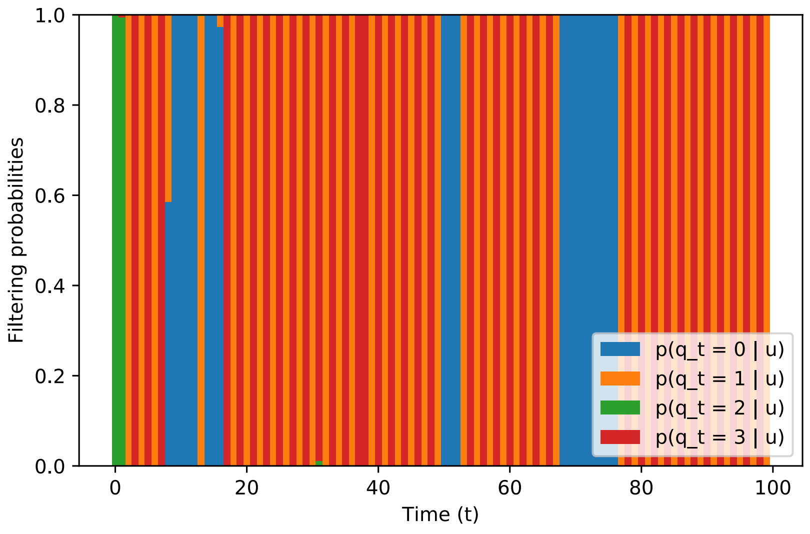


Figure 6: Representation of the filtering probabilities $p(q_t \mid u_0, \ldots u_T)$ for each state for the HMM trained on the test dataset

10. The most likely state of each datapoint according to the marginal probability $p(q_t | u_0, \ldots u_T)$ of state is represented in figure 7.

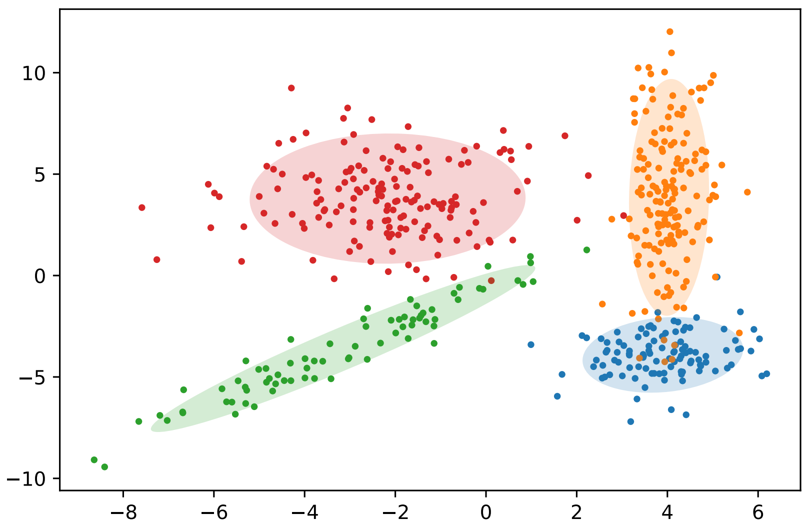
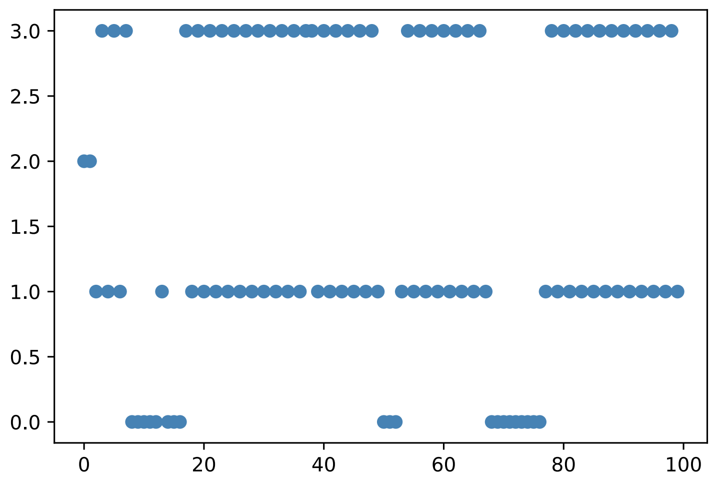


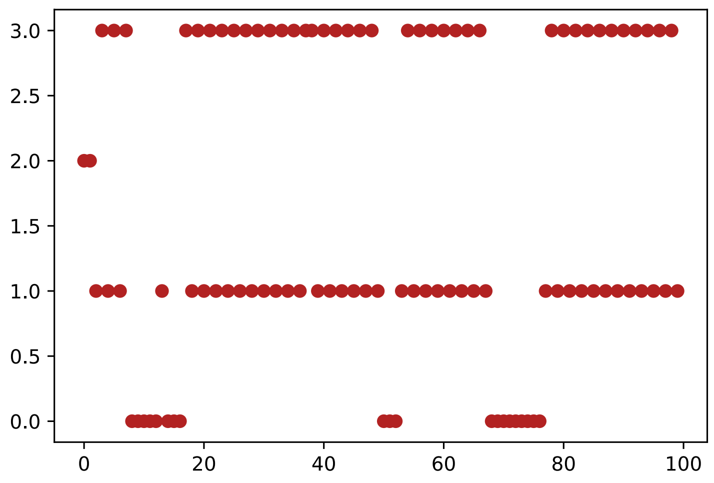
Figure 7: Representation of the most likely state of each datapoint according to the marginal $p(q_t | u_0, \ldots u_T)$

11. The figure 8 compares the most likely state for each datapoint according to both criterion (Viterbi and marginal probabilities). We see that there is no difference in this case (every point is assigned the same class). This means that this assignation of the point maximize both $p(q_t | u_0, \ldots u_T)$ for all t and $p(q_0, \ldots, q_T | u_0, \ldots u_T)$, put in word, the assignation of points corresponding to the most probable sequence of states also maximises the probability of each individual state. This shows that the trained model fits the data very well.

12. As we increase the number of state, the likelihood will increase, we would like to maximize the likelihood but at the same time, we don't want the number of states to be too high, because



(a) Most likely state for each datapoint according to the Viterbi criterion



(b) Most likely state for each datapoint accordint to the marginal probability criterion

Figure 8: Representation of the most likely states for each datapoint according to the Viterbi and the marginal probability criterion

it would lead to overfitting, and also impact the performances (the alpha beta recursion which is necessary for EM has a complexity of $O(S^2T)$ where S is the number of states and T the number of points, and the Viterbi algorithm also has the same complexity). As suggested in the lesson for K-means, we can have optimize a function of the number of states that recompense the likelihood of the model and penalize the number of states. There exist several model selection criterion based on this such as the Bayesian information criterion, which aims to

$$\underset{S}{\text{minimize}} \quad \ln(T)S - 2\ln(\mathcal{L})$$

To choose the number of state, one could increase the number of state up to the point where the gain in the likelihood factor do not compensate the cost of the number of state.