

osl-dynamics: HMM Cost Function

C. Gohil, and R. Huang

OHBA, Department of Psychiatry, Warneford Hospital, Oxford, OX3 7JX

(February 15, 2026)

Abstract

In `osl-dynamics`, the Hidden Markov Model (HMM) is trained using the Expectation-Maximisation (EM) algorithm. Here, we describe the calculation of the cost function used to update the observation model parameters (state means and covariances) in the M-step. The E-step is calculated using the Baum-Welch algorithm. We also describe the calculation of the variational free energy.

1 Variational Free Energy

In variational Bayesian inference we learn a posterior distribution for model parameters, $q(\cdot)$, by minimising the *variational free energy*, \mathcal{F} , given some data we have observed, \mathbf{x}_t . For the HMM, our model parameters are:

- The hidden state at each time point, s_t .
- The state transition probability matrix, \mathbf{A} , where the elements of this matrix are the transition probabilities, $A_{ij} = P(s_t = j | s_{t-1} = i)$.
- The initial state probabilities, $\boldsymbol{\pi}_1$.
- The observation model parameters, θ_{obs} .

If we were being Bayesian on all of these model parameters, we would minimise the following variational free energy¹ [1]

$$\mathcal{F} = \iiint q(s_{1:T})q(\mathbf{A})q(\boldsymbol{\pi}_1)q(\theta_{\text{obs}}) \log \left[\frac{q(s_{1:T})q(\mathbf{A})q(\boldsymbol{\pi}_1)q(\theta_{\text{obs}})}{p(\mathbf{x}_{1:T}, s_{1:T}, \mathbf{A}, \boldsymbol{\pi}_1, \theta_{\text{obs}})} \right] ds_{1:T} d\mathbf{A} d\boldsymbol{\pi}_1 d\theta_{\text{obs}}, \quad (1)$$

where $s_{1:T}$ and $\mathbf{x}_{1:T}$ denote s_1, \dots, s_T and $\mathbf{x}_1, \dots, \mathbf{x}_T$ respectively. However, in the `osl-dynamics` implementation of an HMM, we will only be Bayesian on the hidden states, $s_{1:T}$. We will learn point estimates for all the other parameters: θ_{obs} , \mathbf{A} and $\boldsymbol{\pi}_1$. We learn all of our model parameters by minimising the following variational free energy,

$$\mathcal{F} = \int q(s_{1:T}) \log \left[\frac{q(s_{1:T})}{p(\mathbf{x}_{1:T}, s_{1:T})} \right] ds_{1:T}. \quad (2)$$

We will show that Eq. (2) implicitly depends on the point estimates for θ_{obs} below.

¹We have used the mean field approximation.

2 Generative Model

The denominator in the log function, $p(\cdot)$, is determined by our generative model. For the HMM, if we were being fully Bayesian this would be [1]

$$p(\mathbf{x}_{1:T}, s_{1:T}, \mathbf{A}, \boldsymbol{\pi}_1, \theta_{\text{obs}}) = p(\mathbf{x}_1 | s_1, \theta_{\text{obs}}) p(s_1 | \boldsymbol{\pi}_1) p(\boldsymbol{\pi}_1) p(\theta_{\text{obs}}) \prod_{t=2}^T p(\mathbf{x}_t | s_t, \theta_{\text{obs}}) p(s_t | s_{t-1}, \mathbf{A}) p(\mathbf{A}). \quad (3)$$

However, because we are learning point estimates for most of these parameters ($\theta_{\text{obs}}, \mathbf{A}, \boldsymbol{\pi}_1$) their prior distributions disappear. We will use the following generative model,

$$p(\mathbf{x}_{1:T}, s_{1:T}) = p(\mathbf{x}_1 | s_1, \theta_{\text{obs}}) p(s_1) \prod_{t=2}^T p(\mathbf{x}_t | s_t, \theta_{\text{obs}}) p(s_t | s_{t-1}), \quad (4)$$

where θ_{obs} is a point estimate. We assume a multivariate normal distribution for the observed data,

$$p(\mathbf{x}_t | s_t = k, \theta_{\text{obs}}) = \mathcal{N}(\mathbf{m}_k, \mathbf{C}_k), \quad (5)$$

where \mathbf{m}_k and \mathbf{C}_k are the mean and covariance for state k respectively. Our observation model parameters θ_{obs} are the set of state means and covariances, $\theta_{\text{obs}} = \{\mathbf{m}_k, \mathbf{C}_k\}_{k=1}^K$.

3 The EM Algorithm

Note that although we are optimising the variational free energy, this does not constitute ‘‘variational Bayes’’. We treat all model parameters (apart from the latent states) as point estimates and perform maximum likelihood estimation via the *EM algorithm*.

The posterior for the latent states is computed exactly in the E-step and used to update parameter estimates in the M-step. EM can be viewed as a special case of variational inference in which the variational family contains the true posterior over latent variables and the posterior over parameters is restricted to a delta function. Also note, in our approach the M-step is implemented via stochastic gradient descent on the variational free energy.

E-step. Here, we calculate an exact posterior $q(s_{1:T})$ using the Baum-Welch algorithm [2] using the current parameter estimates $\{\mathbf{A}, \boldsymbol{\pi}_1, \theta_{\text{obs}}\}$.

M-step. Using the exact posterior $q(s_{1:T})$ from the E-step, we update $\{\mathbf{A}, \boldsymbol{\pi}_1\}$ and we update θ_{obs} to minimise the variational free energy (Eq. (2)) using stochastic gradient descent.

4 Cost Function for Learning $\theta_{\text{obs}} = \{m_k, C_k\}$

Here, we derive the cost function for updating θ_{obs} in the M-step. We update our point estimate for θ_{obs} by minimising Eq. (2). We separate Eq. (2) into the following terms²

$$\mathcal{F} = - \int q(s_{1:T}) \log [p(\mathbf{x}_{1:T}, s_{1:T})] ds_{1:T} + \int q(s_{1:T}) \log [q(s_{1:T})] ds_{1:T}. \quad (6)$$

Only the first term depends on θ_{obs} so the second term can be ignored. Substituting Eq. (4) into the first term, we have

$$\mathcal{F} \propto - \int q(s_{1:T}) \log \left[p(\mathbf{x}_1 | s_1, \theta_{\text{obs}}) p(s_1) \prod_{t=2}^T p(\mathbf{x}_t | s_t, \theta_{\text{obs}}) p(s_t | s_{t-1}) \right] ds_{1:T}. \quad (7)$$

²We have used $\int q(\xi) d\xi = 1$ to evaluate some of the integrals.

Again, only retaining the factors that depend on θ_{obs} , we have

$$\begin{aligned}\mathcal{F} &\propto - \int q(s_{1:T}) \log \left[\prod_{t=1}^T p(\mathbf{x}_t | s_t, \theta_{\text{obs}}) \right] ds_{1:T} \\ &\propto - \sum_{t=1}^T \int q(s_{1:T}) \log [p(\mathbf{x}_t | s_t, \theta_{\text{obs}})] ds_{1:T}\end{aligned}\tag{8}$$

To evaluate this, we rewrite the posterior as

$$q(s_{1:T}) = q(s_t, s_\tau),\tag{9}$$

where τ denotes the all of the time points excluding t . Now we can marginalise s_τ ,

$$\begin{aligned}\mathcal{F} &\propto - \sum_{t=1}^T \iint q(s_t, s_\tau) \log [p(\mathbf{x}_t | s_t, \theta_{\text{obs}})] ds_t ds_\tau \\ &\propto - \sum_{t=1}^T \int q(s_t) \log [p(\mathbf{x}_t | s_t, \theta_{\text{obs}})] ds_t = \mathcal{L}.\end{aligned}\tag{10}$$

Here, we have defined the negative log-likelihood loss, \mathcal{L} , which is minimised via stochastic gradient descent to learn the parameters θ_{obs} . $q(s_t)$ is the marginal posterior calculated using the Baum-Welch algorithm [2], commonly denoted using the symbol $\gamma(t)$. As $q(s_t)$ is a discrete probability distribution for the state, we can evaluate the integral as

$$\begin{aligned}\mathcal{L} &= - \sum_{t=1}^T \sum_{k=1}^K q(s_t = k) \log [p(\mathbf{x}_t | s_t = k, \theta_{\text{obs}})] \\ &= - \sum_{t=1}^T \sum_{k=1}^K \gamma_k(t) \log [p(\mathbf{x}_t | s_t = k, \theta_{\text{obs}})],\end{aligned}\tag{11}$$

where K is the number of states and $q(s_t = k) = \gamma_k(t)$ are the elements of the vector $\gamma(t)$, which denote the probability of state k at time t . Substituting Eq. (5) into this we have

$$\mathcal{L} = - \sum_{t=1}^T \sum_{k=1}^K \gamma_k(t) \log [\mathcal{N}(\mathbf{x}_t | \mathbf{m}_k, \mathbf{C}_k)],\tag{12}$$

which is the log-likelihood loss function implemented in `osl-dynamics` for inferring the point estimates for the observation model parameters $\theta_{\text{obs}} = \{\mathbf{m}_k, \mathbf{C}_k\}$.

5 Calculation of the Variational Free Energy

Once we have trained an HMM we may want to evaluate the variational free energy, i.e. Eq. (2). This can be done with the `free_energy` method of the `hmm.Model` class. The method calculates

Eq. (2) by first splitting it into three terms:

$$\begin{aligned}
\mathcal{F} &= \int q(s_{1:T}) \log \left[\frac{q(s_{1:T})}{p(\mathbf{x}_{1:T}, s_{1:T})} \right] ds_{1:T}, \\
&= \int q(s_{1:T}) \log \left[\frac{q(s_{1:T})}{p(\mathbf{x}_1|s_1)p(s_1) \prod_{t=2}^T p(\mathbf{x}_t|s_t)p(s_t|s_{t-1})} \right] ds_{1:T}, \\
&= - \int q(s_{1:T}) \log \left[\prod_{t=1}^T p(\mathbf{x}_t|s_t) \right] ds_{1:T} + \int q(s_{1:T}) \log \left[\frac{q(s_{1:T})}{p(s_1) \prod_{t=2}^T p(s_t|s_{t-1})} \right] ds_{1:T}, \\
&= - \int q(s_{1:T}) \log \left[\prod_{t=1}^T p(\mathbf{x}_t|s_t) \right] ds_{1:T} + \int q(s_{1:T}) \log [q(s_{1:T})] ds_{1:T} \\
&\quad - \int q(s_{1:T}) \log \left[p(s_1) \prod_{t=2}^T p(s_t|s_{t-1}) \right] ds_{1:T}, \\
&= -LL + E - P,
\end{aligned} \tag{13}$$

where LL is the posterior expected log-likelihood (same as Eq. (12)), E is the posterior entropy and P is the posterior expected prior probability. To evaluate the terms in the above equation we factorise the posterior as

$$q(s_{1:T}) = q(s_1) \prod_{t=2}^T q(s_t|s_{t-1}) = q(s_1) \prod_{t=2}^T \frac{q(s_{t-1}, s_t)}{q(s_{t-1})} = q(s_1) \prod_{t=1}^{T-1} \frac{q(s_t, s_{t+1})}{q(s_t)}. \tag{14}$$

The above factorisation is an assumption of the Baum-Welch algorithm. Let's first look at the entropy term,

$$\begin{aligned}
E &= \int q(s_{1:T}) \log [q(s_{1:T})] ds_{1:T}, \\
&= \int q(s_{1:T}) \log \left[q(s_1) \prod_{t=1}^{T-1} \frac{q(s_t, s_{t+1})}{q(s_t)} \right] ds_{1:T}, \\
&= \int q(s_{1:T}) \log \left[\frac{\prod_{t=1}^{T-1} q(s_t, s_{t+1})}{\prod_{t=2}^{T-1} q(s_t)} \right] ds_{1:T}, \\
&= \sum_{t=1}^{T-1} \int q(s_{1:T}) \log q(s_t, s_{t+1}) ds_{1:T} - \sum_{t=2}^{T-1} \int q(s_{1:T}) \log q(s_t) ds_{1:T}.
\end{aligned} \tag{15}$$

To evaluate the integral we marginalise out the state at times that do not appear inside the log function,

$$\begin{aligned}
E &= \sum_{t=1}^{T-1} \int q(s_t, s_{t+1}, s_\tau) \log q(s_t, s_{t+1}) ds_t ds_{t+1} ds_\tau - \sum_{t=2}^{T-1} \int q(s_t, s_\tau) \log q(s_t) ds_t ds_\tau. \\
&= \sum_{t=1}^{T-1} \int q(s_t, s_{t+1}) \log q(s_t, s_{t+1}) ds_t ds_{t+1} - \sum_{t=2}^{T-1} \int q(s_t) \log q(s_t) ds_t.
\end{aligned} \tag{16}$$

This can be calculated using the marginal posterior, $\gamma(t) = q(s_t)$, and joint posterior, $\xi(t) = q(s_t, s_{t+1})$, provided by the Baum-Welch algorithm:

$$E = \sum_{t=1}^{T-1} \sum_{i,j=1}^K \xi_{ij}(t) \log \xi_{ij}(t) - \sum_{t=2}^{T-1} \sum_{i=1}^K \gamma_i(t) \log \gamma_i(t), \tag{17}$$

where $\xi_{ij}(t) = P(s_t = i, s_{t+1} = j)$. Finally, we calculate the posterior expected prior probability as³

$$\begin{aligned}
P &= \int q(s_{1:T}) \log \left[p(s_1) \prod_{t=2}^T p(s_t | s_{t-1}) \right] ds_{1:T}, \\
&= \int q(s_1) \prod_{\tau=1}^{T-1} \frac{q(s_\tau, s_{\tau+1})}{q(s_\tau)} \log \left[p(s_1) \prod_{t=2}^T p(s_t | s_{t-1}) \right] ds_{1:T}, \\
&= \int q(s_1) \prod_{\tau=1}^{T-1} \frac{q(s_\tau, s_{\tau+1})}{q(s_\tau)} \log p(s_1) ds_{1:T} + \int q(s_1) \prod_{\tau=1}^{T-1} \frac{q(s_\tau, s_{\tau+1})}{q(s_\tau)} \log \left[\prod_{t=2}^T p(s_t | s_{t-1}) \right] ds_{1:T}, \\
&= \int \dots \int q(s_1) \prod_{\tau=1}^{T-2} \frac{q(s_\tau, s_{\tau+1})}{q(s_\tau)} \log p(s_1) ds_1 \dots ds_{T-1} \int \frac{q(s_{T-1}, s_T)}{q(s_{T-1})} ds_T \\
&\quad + \int q(s_1) \prod_{\tau=1}^{T-1} \frac{q(s_\tau, s_{\tau+1})}{q(s_\tau)} \log \left[\prod_{t=2}^T p(s_t | s_{t-1}) \right] ds_{1:T}, \\
&= \int q(s_1) \log p(s_1) ds_1 + \int q(s_1) \prod_{\tau=1}^{T-1} \frac{q(s_\tau, s_{\tau+1})}{q(s_\tau)} \log \left[\prod_{t=2}^T p(s_t | s_{t-1}) \right] ds_{1:T}, \\
&= \int q(s_1) \log p(s_1) ds_1 + \int q(s_1) \prod_{\tau=1}^{T-1} \frac{q(s_\tau, s_{\tau+1})}{q(s_\tau)} \left\{ \sum_{t=2}^T \log p(s_t | s_{t-1}) \right\} ds_{1:T}, \\
&= \int q(s_1) \log p(s_1) ds_1 + \int q(s_1) \prod_{\tau=1}^{T-1} \frac{q(s_\tau, s_{\tau+1})}{q(s_\tau)} \left\{ \sum_{t=1}^{T-1} \log p(s_{t+1} | s_t) \right\} ds_{1:T}, \\
&= \int q(s_1) \log p(s_1) ds_1 + \sum_{t=1}^{T-1} \int q(s_1) \prod_{\tau=1}^{T-1} \frac{q(s_\tau, s_{\tau+1})}{q(s_\tau)} \log p(s_{t+1} | s_t) ds_{1:T}, \\
&= \int q(s_1) \log p(s_1) ds_1 + \sum_{t=1}^{T-1} \int \dots \int q(s_1) \frac{q(s_1, s_2)}{q(s_1)} \dots \frac{q(s_{T-1}, s_T)}{q(s_T)} \log p(s_{t+1} | s_t) ds_1 \dots ds_T, \\
&= \int q(s_1) \log p(s_1) ds_1 + \sum_{t=1}^{T-1} \int \dots \int \left\{ \int q(s_1, s_2) ds_1 \right\} \frac{q(s_2, s_3)}{q(s_2)} \dots \frac{q(s_{T-1}, s_T)}{q(s_T)} \log p(s_{t+1} | s_t) ds_2 \dots ds_T, \\
&= \int q(s_1) \log p(s_1) ds_1 + \sum_{t=1}^{T-1} \int \dots \int q(s_2) \frac{q(s_2, s_3)}{q(s_2)} \dots \frac{q(s_{T-1}, s_T)}{q(s_T)} \log p(s_{t+1} | s_t) ds_2 \dots ds_T, \\
&= \int q(s_1) \log p(s_1) ds_1 + \sum_{t=1}^{T-1} \iint q(s_t, s_{t+1}) \log p(s_{t+1} | s_t) ds_t ds_{t+1}.
\end{aligned} \tag{18}$$

Using the marginal and joint posterior provided by the Baum-Welch algorithm and the point estimates for the initial probabilities, $\boldsymbol{\pi}_1$ and transition probability matrix, \mathbf{A} , this is evaluated as

$$P = \sum_{i=1}^K \gamma_i(1) \log \pi_{1,i} + \sum_{t=1}^{T-1} \sum_{i,j=1}^K \xi_{ij}(t) \log A_{ij}. \tag{19}$$

³We use $\int \frac{p(x,y)}{p(x)} dy = \frac{1}{p(x)} \int p(x,y) dy = \frac{1}{p(x)} p(x) = 1$ to evaluate some of the integrals.

References

- [1] I. Rezek and S. Roberts, Ensemble hidden Markov models with extended observation densities for biosignal analysis. Probabilistic modeling in bioinformatics and medical informatics. Springer, London, 419-450 (2005).
- [2] https://en.wikipedia.org/wiki/Baum-Welch_algorithm