APPENDIX A: BAUM-WELCH ALGORITHM FOR THE TWO HIDDEN STATE HMM

For the HMM with two hidden states, $R_{1:n}$ and $S_{1:n}$ are the hidden states, see Figure 1. Each state in the hidden process $S_{1:n}$ can take on one of m_1 possible values, while each state in the hidden process $R_{1:n}$ can take on one of m_2 possible values. The length of both series is still n. We define the following parameters:

$$\begin{aligned} C_{ij} &= P(R_t = j | R_{t-1} = i) \\ D_{j,k,l} &= P(S_t = l | R_t = j, S_{t-1} = k) \\ A_{ik,jl} &= C_{ij} D_{jkl} = P(R_t = j, S_t = l | R_{t-1} = i, S_{t-1} = k) \\ Z_t &= (R_t, S_t) \end{aligned}$$

The constraints are $\sum_j C_{ij} = 1, \sum_l D_{jkl} = 1$.

Let $\theta = (\pi, A, B, C, D)$ be the model parameters, where π and B are the initial state distribution and emission distribution, respectively, as defined for the first order HMM. Let $\theta^{(t)}$ be the current values of these parameters at time t in the Baum-Welch Algorithm. Define c to be a constant. Then, the auxiliary function for the E step of the update Baum-Welch Algorithm for the HMM with two hidden states can be written as:

$$\begin{aligned} Q(\theta, \theta^{(t)}) &= \mathbb{E}_{\theta^{(t)}}(\log p_\theta(X_{1:n}, Z_{1:n} | X_{1:n} = x_{1:n})) \\ &= c + \sum_{t=2}^{n} \sum_{i,k} \sum_{j,l} P_{\theta_k}(R_{t-1} = i, S_{t-1} = k, R_t = j, S_t = l | X_{1:n}) \log C_{ik,jl} \end{aligned}$$

Let

$$D_{t,ik,jl} = P_{\theta^{(t)}}(R_{t-1} = i, S_{t-1} = k, R_t = j, S_t = l | X_{1:n}).$$

We have that

$$\log A_{ik,jl} = \log C_{ij} + \log D_{j,k,l}.$$

Then, we can find the value of θ to maximize $Q(\theta, \theta^{(t)})$ (where ν is a Lagrange multiplier to handle the constraints placed on C and D):

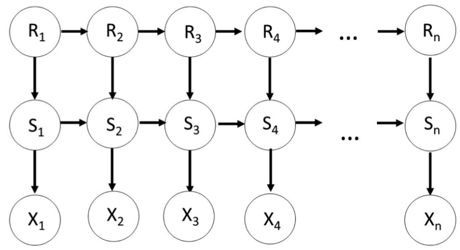

Fig 1: Directed graph of the HMM with two hidden states. Both the $R_{1:n}$ and the $S_{1:n}$ are hidden states.

$$
\begin{aligned}
0 &= \frac{\partial}{\partial C_{ij}} \left(Q(\theta, \theta^{(t)}) - \nu \sum_j C_{ij} \right) \\
0 &= \left(\sum_{t=2}^{n} \sum_k \sum_l D_{t,ik,jl} \frac{1}{C_{ij}} \right) - \nu \\
\nu C_{ij} &= \sum_{t=2}^{n} \sum_k \sum_l D_{t,ik,jl} \\
\nu &= \sum_j \sum_{t=2}^{n} \sum_k \sum_l D_{t,ik,jl} \\
C_{ij} &\propto \sum_{t=2}^{n} \sum_{k,l} D_{t,ik,jl} \quad \forall 1 \le i, j \le m_2
\end{aligned}
$$

Likewise,

$$
\begin{aligned}
0 &= \frac{\partial}{\partial D_{j,k,l}} \left(Q(\theta, \theta^{(t)}) - \nu \sum_l D_{j,k,l} \right) \\
&= \sum_{t=2}^{n} \sum_i D_{t,ik,jl} \frac{1}{D_{j,k,l}} - \nu \\
\nu D_{j,k,l} &= \sum_{t=2}^{n} \sum_i D_{t,ik,jl} \\
D_{j,k,l} &\propto \sum_{t=2}^{n} \sum_i D_{t,ik,jl} \quad \forall 1 \le l, k \le m_1, 1 \le j \le m_2
\end{aligned}
$$

The Forward-Backward Algorithm is exactly the same as in the first order

HMM case, where A as defined above is the transition matrix used. π and B are updated exactly the same way as in the Baum-Welch Algorithm for the first order HMM.