

Hidden Markov Models (HMMs)

(Discrete-time Discrete Observation Space Case)

[1] is a classical reference on this topic, available on [WEBCT](#) and [IEEE Xplore](#),

For a bioinformatics-related exposition, see Ch. 12 of

W.J. Ewens and G.R. Grant, *Statistical Methods in Bioinformatics: An Introduction*, 2nd ed. New York: Springer-Verlag, 2005.

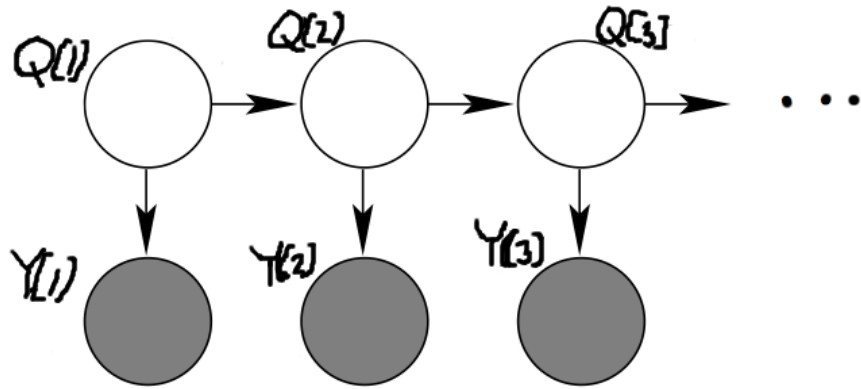
For an overview of HMM applications, see Ch. 1 of

O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models*. New York: Springer-Verlag, 2005.

For Viterbi algorithm and communications applications of HMMs, see Ch. 19 in Moon & Stirling.

For a general exposition on state space and hidden Markov models, see

H.R. Künsch, “State space and hidden Markov models,” in *Complex Stochastic Systems*, O.E. Barndorff-Nielsen, D.R. Cox, and C. Klüpelberg, Eds., London UK: Chapman & Hall, 2001, ch. 3, pp. 109–173.



A Review of Chain Rule

Consider

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_T \end{bmatrix}.$$

Then,

$$f_{\mathbf{X}}(\mathbf{x}) = f(x_1) f(x_2 | x_1) f(x_3 | x_1, x_2) \cdots f(x_T | \mathbf{x}_{1:T-1}). \quad (1)$$

Proof. Apply induction. The chain rule clearly holds for $N = 2$. Now, suppose that it is true for $T - 1$:

$$f(\mathbf{x}_{1:(T-1)}) = f(x_1) f(x_2 | x_1) f(x_3 | \mathbf{x}_{1:2}) \cdots f(x_{T-1} | \mathbf{x}_{1:T-2}). \quad (2)$$

To show that the chain rule holds for N , write

$$f_{\mathbf{X}}(\mathbf{x}) = f(\mathbf{x}_{1:(T-1)}) f(x^{[T]} | \mathbf{x}_{1:(T-1)}) \quad (3)$$

and substitute (2) into (3). \square

HMM Ingredients

- **State space:** $S = \{s_1, s_2, \dots, s_N\}$ containing N discrete values. The state at time t is $q[t] \in S$ and the vector of states from time 1 until time T is

$$\mathbf{q}_{1:T} = \begin{bmatrix} q[1] \\ q[2] \\ \vdots \\ q[T] \end{bmatrix}.$$

- **Observation space:** $\Upsilon = \{v_1, v_2, \dots, v_M\}$ containing M discrete values and the vector of observations from time 1 until time T is

$$\mathbf{y}_{1:T} = \begin{bmatrix} y[1] \\ y[2] \\ \vdots \\ y[T] \end{bmatrix}$$

where

$$y[t] \in \Upsilon.$$

- **Transition matrix A ,** whose (i, j) th element

$$a_{i,j} = \Pr\{Q[t+1] = s_j \mid Q[t] = s_i\} \quad i, j \in \{1, 2, \dots, N\}$$

describes a Markov chain; the transition probabilities $a_{i,j}$ are stationary (independent of t).

- **Likelihood** is stationary (independent of t):

$$l_j(v_m) = \Pr\{Y[t] = v_m \mid Q[t] = s_j\} \quad j = 1, \dots, N, m = 1, \dots, M.$$

Define an $N \times M$ matrix

$$\mathcal{L} = \{l_j(v_m)\} \quad j = 1, \dots, N, m = 1, \dots, M.$$

- **Initial distribution:**

$$\pi_i = \Pr\{Q[1] = s_i\} \quad i = 1, \dots, N.$$

Define

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_N \end{bmatrix}.$$

- $\boldsymbol{\theta}$ is the set of model parameters:

$$\boldsymbol{\theta} = \{A, \mathcal{L}, \boldsymbol{\pi}\}.$$

Reminder:

- N , number of possible hidden states,

- M , number of possible values that observations v_m can take,
- T , number of collected observations.

Notation: We use \mathbf{y} and $\mathbf{y}_{1:T}$, \mathbf{Y} and $\mathbf{Y}_{1:T}$, \mathbf{q} and $\mathbf{q}_{1:T}$, Q and $Q_{1:T}$ interchangeably.

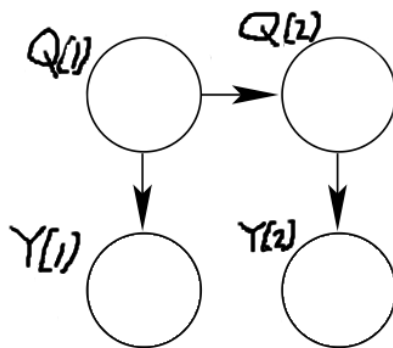
Complete-data Likelihood

$$\begin{aligned}
 p(\mathbf{y}, \mathbf{q} \mid \boldsymbol{\theta}) &= \overbrace{\pi_{q[1]} p(y[1] \mid q[1]) a_{q[1],q[2]}}^{\text{relevant for } q[1]} \\
 &\quad \cdot p(y[2] \mid q[2]) a_{q[2],q[3]} \cdots a_{q[T-1],q[T]} l_{q[T]}(y[T]) \\
 &= \pi_{q[1]} p(y[1] \mid q[1]) \overbrace{a_{q[1],q[2]} p(y[2] \mid q[2]) a_{q[2],q[3]}}^{\text{relevant for } q[2]} \\
 &\quad \cdots a_{q[T-1],q[T]} l_{q[T]}(y[T])
 \end{aligned}$$

etc. For example, for $T = 2$,

$$p(\mathbf{y}, \mathbf{q} \mid \boldsymbol{\theta}) = \underbrace{\pi_{q[1]}}_{p(q[1])} \underbrace{a_{q[1],q[2]}}_{p(q[2] \mid q[1])} \underbrace{l_{q[1]}(y[1])}_{p(y[1] \mid q[1])} \underbrace{l_{q[2]}(y[2])}_{p(y[2] \mid q[2])}$$

which follows from the HMM graph:

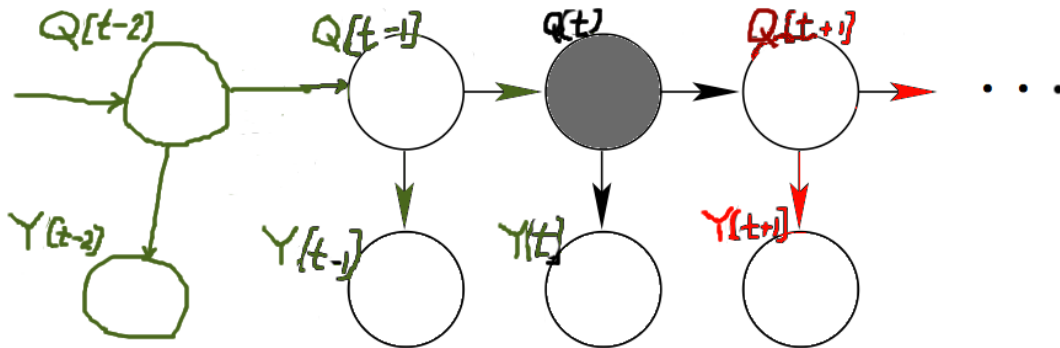


Dynamic programming is just a clever way of putting brackets and segmenting the above joint probability mass function (pmf).

Note the special conditional-independence structure

$$\{Y_{1:t}, Q_{1:(t-1)}\} \perp\!\!\!\perp \{Y_{(t+1):T}, Q_{(t+1):T}\} \mid Q[t]$$

depicted by the following graph:



Computing the Marginal Likelihood $p_{Y|\Theta}(\mathbf{y}|\boldsymbol{\theta})$

$$\Lambda_T(\boldsymbol{\theta}) = p(\underbrace{\mathbf{y}}_{\mathbf{y}_{1:T}} | \boldsymbol{\theta}) = \sum_{\mathbf{q} \in S^T} p(\mathbf{y}, \mathbf{q} | \boldsymbol{\theta}).$$

Example: Consider the simple case with

- $N = 2$ states, s_1 and s_2 , and hence $q[t] \in \{s_1, s_2\}$.
- $T = 2$ time points: $t = 1, 2$, and
- $M = 2$ possible observations v_1 and v_2 : $y[t] \in \{v_1, v_2\}$.

We wish to compute the observed-data likelihood $p(\mathbf{y}|\boldsymbol{\theta})$. Give names to $l_j(v_m)$:

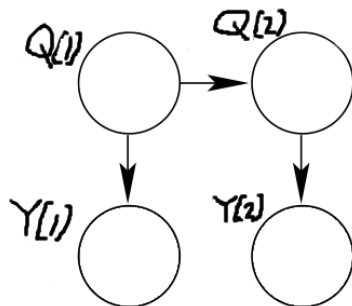
$$\begin{aligned} l_1(v_1) &= \Pr\{v_1 \text{ "observed" and } s_1 \text{ the true state}\} = p_1 \\ l_1(v_2) &= \Pr\{v_2 \text{ "observed" and } s_1 \text{ the true state}\} = 1 - p_1 \\ l_2(v_1) &= \Pr\{v_1 \text{ "observed" and } s_2 \text{ the true state}\} = 1 - p_2 \\ l_2(v_2) &= \Pr\{v_2 \text{ "observed" and } s_2 \text{ the true state}\} = p_2 \end{aligned}$$

implying $\mathcal{L} = \begin{matrix} \uparrow \\ \downarrow \end{matrix} \begin{bmatrix} p_1 & 1 - p_1 \\ 1 - p_2 & p_2 \end{bmatrix}$. Suppose that we have observed $\mathbf{y}' = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$. Define $\mathbf{Q} = \begin{bmatrix} Q[1] \\ Q[2] \end{bmatrix}$ and

$\mathbf{q} = \begin{bmatrix} q[1] \\ q[2] \end{bmatrix}$. Then, the observed-data likelihood function of $\boldsymbol{\theta}$ is

$$\begin{aligned}
 p_{\mathbf{Y}|\boldsymbol{\theta}}(\mathbf{y}'|\boldsymbol{\theta}) &= \sum_{\mathbf{q} \in \{s_1, s_2\}^2} p_{\mathbf{Y}, \mathbf{Q}|\boldsymbol{\theta}}(\mathbf{y}', \mathbf{q}|\boldsymbol{\theta}) \\
 &= p_{\mathbf{Y}, \mathbf{Q}|\boldsymbol{\theta}}\left(\mathbf{y}', \begin{bmatrix} s_1 \\ s_1 \end{bmatrix} \middle| \boldsymbol{\theta}\right) + p_{\mathbf{Y}, \mathbf{Q}|\boldsymbol{\theta}}\left(\mathbf{y}', \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \middle| \boldsymbol{\theta}\right) \\
 &\quad + p_{\mathbf{Y}, \mathbf{Q}|\boldsymbol{\theta}}\left(\mathbf{y}', \begin{bmatrix} s_2 \\ s_1 \end{bmatrix} \middle| \boldsymbol{\theta}\right) + p_{\mathbf{Y}, \mathbf{Q}|\boldsymbol{\theta}}\left(\mathbf{y}', \begin{bmatrix} s_2 \\ s_2 \end{bmatrix} \middle| \boldsymbol{\theta}\right) \\
 &= \pi_1 a_{1,1} \underbrace{p_1}_{l_1(v_1)} \underbrace{(1-p_1)}_{l_1(v_2)} + \pi_1 a_{1,2} p_1 p_2 \\
 &\quad + \pi_2 a_{2,1} (1-p_2) (1-p_1) + \pi_2 a_{2,2} (1-p_2) p_2
 \end{aligned}$$

where $p_{\mathbf{Y}, \mathbf{Q}}(\mathbf{y}', \mathbf{q}|\boldsymbol{\theta}) = \underbrace{\pi_{q[1]}}_{p(q[1])} \underbrace{a_{q[1], q[2]}}_{p(q[2]|q[1])} \underbrace{l_{q[1]}(y[1])}_{p(y[1]|q[1])} \underbrace{l_{q[2]}(y[2])}_{p(y[2]|q[2])}$
 which follows from the HMM graph:



Forward-summation Recursion

$$\begin{aligned}
 p(\mathbf{y} | \boldsymbol{\theta}) &= \sum_{\mathbf{q} \in S^T} p(\mathbf{y}, \mathbf{q} | \boldsymbol{\theta}) \\
 &= \sum_{\mathbf{q}} \pi_{q[1]} l_{q[1]}(y[1]) a_{q[1],q[2]} l_{q[2]}(y[2]) \cdots a_{q[T-1],q[T]} l_{q[T]}(y[T]) \\
 &= \sum_{q[T] \in S} \left[\sum_{q[T-1] \in S} \cdots \left[\sum_{q[3] \in S} \left[\sum_{q[2] \in S} \right. \right. \right. \\
 &\quad \left. \left. \left[\sum_{q[1] \in S} \pi_{q[1]} l_{q[1]}(y[1]) a_{q[1],q[2]} l_{q[2]}(y[2]) \right] \right. \right. \\
 &\quad \left. \left. a_{q[2],q[3]} l_{q[3]}(y[3]) \right] a_{q[3],q[4]} l_{q[4]}(y[4]) \right] \\
 &\quad \left. \cdots \right] a_{q[T-1],q[T]} l_{q[T]}(y[T]) \Big].
 \end{aligned}$$

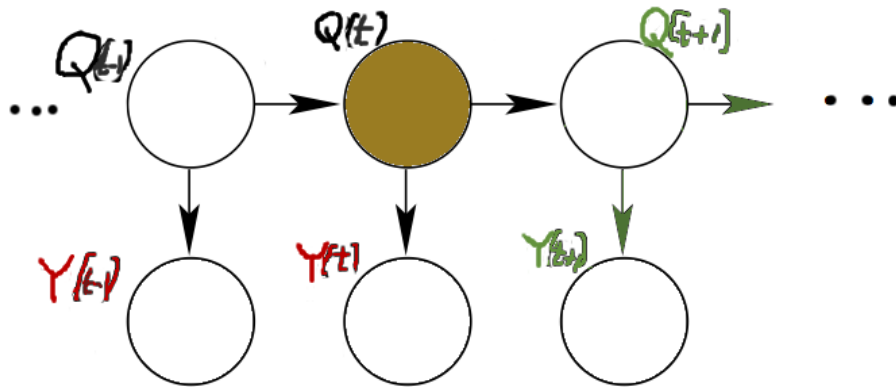
Define the joint likelihood contribution of $y[1], y[2], \dots, y[t]$ *and* the event $\{Q[t] = s_i\}$, *averaged over* $Q[1], Q[2], \dots, Q[t-1]$:

$$\begin{aligned}
 \alpha_t(i) &\triangleq p(\mathbf{y}_{1:t}, q[t] = s_i | \boldsymbol{\theta}) \\
 &= \sum_{\mathbf{q}_{1:t-1}} p(\mathbf{y}_{1:t}, \mathbf{q}_{1:t-1}, q[t] = s_i | \boldsymbol{\theta}). \quad (4)
 \end{aligned}$$

Then, $\alpha_1(i) = p(y[1], s_i | \boldsymbol{\theta}) = \pi_i l_i(y[1])$. We wish to derive a recursion for computing $\alpha_t(i)$ using induction. In particular, we now assume that $\alpha_t(j)$, $j = 1, 2, \dots, N$ are available and utilize this information to obtain $\alpha_{t+1}(i)$, $i = 1, 2, \dots, N$:

$$\begin{aligned}
 \alpha_{t+1}(i) &= p(\mathbf{y}_{1:t}, y[t+1], q[t+1] = s_i | \boldsymbol{\theta}) \\
 &= \sum_{j=1}^N p(\mathbf{y}_{1:t}, y[t+1], q[t] = s_j, q[t+1] = s_i | \boldsymbol{\theta}) \\
 &= \sum_{j=1}^N \underbrace{p(\mathbf{y}_{1:t}, q[t] = s_j | \boldsymbol{\theta})}_{\alpha_t(j)} \\
 &\quad \cdot \underbrace{p(y[t+1], q[t+1] = s_i | \mathbf{y}_{1:t}, q[t] = s_j, \boldsymbol{\theta})}_{p(y[t+1], q[t+1]=s_i | q[t]=s_j, \boldsymbol{\theta})} \\
 &= \sum_{j=1}^N \alpha_t(j) \cdot \underbrace{p(y[t+1] | q[t+1] = s_i, q[t] = s_j, \boldsymbol{\theta})}_{p(y[t+1] | q[t+1]=s_i, \boldsymbol{\theta})=l_i(y[t+1])} \\
 &\quad \cdot \underbrace{p(q[t+1] = s_i | q[t] = s_j, \boldsymbol{\theta})}_{a_{j,i}} \\
 &= \left[\sum_{j=1}^N \alpha_t(j) a_{j,i} \right] \cdot l_i(y[t+1])
 \end{aligned}$$

for $t = 1, 2, \dots, T - 1$ and $i = 1, 2, \dots, N$, see the two HMM graphs below, where we observe



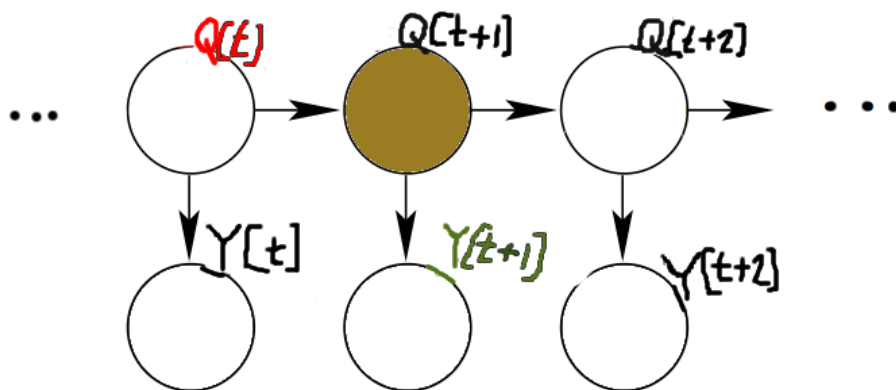
implying

$$Y[t+1], Q[t+1] \perp\!\!\!\perp \mathbf{Y}_{1:t} \mid \{Q[t], \Theta\}$$

and, therefore,

$$p(y[t+1], q[t+1] \mid \mathbf{y}_{1:t}, q[t], \theta) = p(y[t+1], q[t+1] \mid q[t], \theta)$$

and



implying

$$Y[t+1] \perp\!\!\!\perp Q[t] \mid \{Q[t+1], \Theta\}$$

and, therefore,

$$p(y[t + 1] \mid q[t + 1], q[t], \boldsymbol{\theta}) = p(y[t + 1] \mid q[t + 1], \boldsymbol{\theta}).$$

We summarize the above *forward-summation recursion*:

$$\alpha_1(i) = p(y[1], q[1] = s_i \mid \boldsymbol{\theta}) = \pi_i l_i(y[1]) \quad (5)$$

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^N \alpha_t(j) a_{j,i} \right] \cdot l_i(y[t + 1]) \quad (6)$$

for $t = 1, 2, \dots, T - 1$ and $i = 1, 2, \dots, N$.

We can implement the above recursion efficiently using matrix computations, e.g.

$$\boldsymbol{\alpha}_{t+1}^T = (\boldsymbol{\alpha}_t^T A) \odot \mathbf{l}^T(y[t + 1])$$

where “ \odot ” denotes the Hadamard (elementwise) product.

Note that

$$p(\underbrace{\mathbf{y}}_{\mathbf{y}_{1:T}} \mid \boldsymbol{\theta}) = \sum_{i=1}^N p(\mathbf{y}, q[T] = s_i \mid \boldsymbol{\theta}) \stackrel{\text{see (4)}}{=} \sum_{i=1}^N \alpha_T(i). \quad (7)$$

The forward-summation recursion (5)–(6) may be *unstable* because it calculates the likelihood rather than the log likelihood, i.e. the scale differences between $\alpha_t(i)$ s may be huge. We can solve this problem simply by normalizing $\alpha_t(i)$ at every step so that $\sum_i \alpha_t(i) = 1$. Here is an alternative normalization that has a nice probabilistic interpretation.

Normalized forward recursion. Define

$$\Lambda_t(\boldsymbol{\theta}) = \begin{cases} p(\mathbf{y}_{1:t} | \boldsymbol{\theta}), & t > 0 \\ 1, & t = 0 \end{cases} . \quad (8)$$

Then, the marginal log likelihood $\ln \Lambda_T(\boldsymbol{\theta})$ can be computed via the *chain rule* in (1):

$$\ln \Lambda_T(\boldsymbol{\theta}) = \sum_{t=1}^T \ln \omega_t(\boldsymbol{\theta}) = \ln \omega_T(\boldsymbol{\theta}) + \ln \Lambda_{T-1}(\boldsymbol{\theta})$$

for $t = 0, 1, 2, \dots, T - 1$

where

$$\omega_1(\boldsymbol{\theta}) \triangleq p(y[1] | \boldsymbol{\theta})$$

$$\omega_t(\boldsymbol{\theta}) \triangleq p(y[t] | \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta}) \quad t = 2, 3, \dots, T.$$

We compute $\omega_t(\boldsymbol{\theta})$ as follows: For $t = 1$, we have

$$\omega_1(\boldsymbol{\theta}) = \sum_{i=1}^N \underbrace{p(y[1], \mathbf{q}[1] = s_i | \boldsymbol{\theta})}_{l_i(y[1] | \boldsymbol{\theta}) \varphi_1(i | \boldsymbol{\theta})} = \sum_{i=1}^N l_i(y[1] | \boldsymbol{\theta}) \pi_i$$

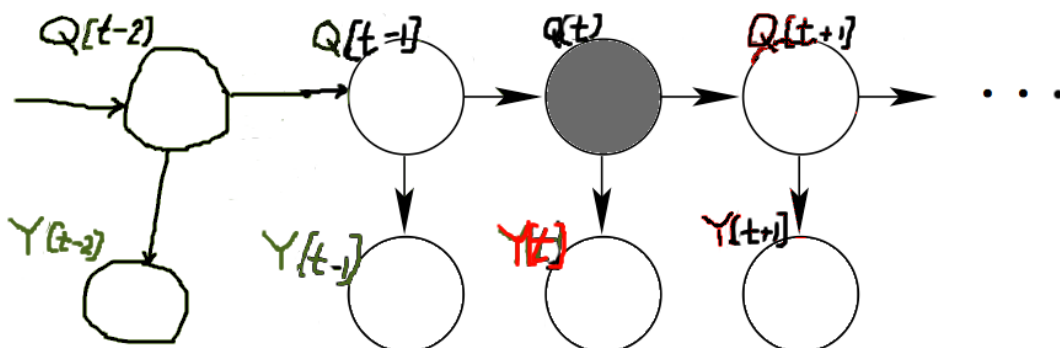
where

$$\varphi_1(i | \boldsymbol{\theta}) \triangleq p(\mathbf{q}[1] = s_i) = \pi_i \quad (9)$$

and, for $t = 2, 3, \dots, T$, we have

$$\begin{aligned} \omega_t(\boldsymbol{\theta}) &= p(y[t] | \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta}) = \sum_{i=1}^N p(y[t], q[t] = s_i | \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta}) \\ &= \sum_{i=1}^N \underbrace{p(y[t] | q[t] = s_i, \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta})}_{p(y[t] | q[t]=s_i)} \underbrace{p(q[t] = s_i | \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta})}_{\triangleq \varphi_t(i | \boldsymbol{\theta})} \\ &= \sum_{i=1}^N l_i(y[t] | \boldsymbol{\theta}) \varphi_t(i | \boldsymbol{\theta}) \end{aligned} \quad (10)$$

where the following graph:



implies

$$Y[t] \perp\!\!\!\perp \mathbf{Y}_{1:(t-1)} \mid Q[t]$$

and, therefore,

$$p(y[t] \mid \mathbf{y}_{1:(t-1)}, q[t], \boldsymbol{\theta}) = p(y[t] \mid q[t], \boldsymbol{\theta}).$$

Here,

$$\varphi_t(i \mid \boldsymbol{\theta}) \triangleq p(q[t] = s_i \mid \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta})$$

is the posterior-predictive pmf (and, hence, always taking values between zero and one) of the hidden state at time t ($q[t]$) given the past measurements $\mathbf{y}_{1:(t-1)}$. Observe that

$$p(y[t+1], q[t+1] = s_i \mid \mathbf{y}_{1:t}, \boldsymbol{\theta}) = \frac{\overbrace{p(y[t+1], q[t+1] = s_i, \mathbf{y}_{1:t} \mid \boldsymbol{\theta})}^{\alpha_{t+1}(i)}}{\underbrace{p(\mathbf{y}_{1:t} \mid \boldsymbol{\theta})}_{\Lambda_t(\boldsymbol{\theta})}}.$$

and use it to compute $\varphi_{t+1}(i \mid \boldsymbol{\theta})$:

$$\begin{aligned} \underbrace{p(q[t+1] = s_i \mid \mathbf{y}_{1:t}, \boldsymbol{\theta})}_{\varphi_{t+1}(i \mid \boldsymbol{\theta})} &= \frac{p(y[t+1], q[t+1] = s_i \mid \mathbf{y}_{1:t}, \boldsymbol{\theta})}{\underbrace{p(y[t+1] \mid q[t+1] = s_i, \mathbf{y}_{1:t}, \boldsymbol{\theta})}_{p(y[t+1] \mid q[t+1] = s_i, \boldsymbol{\theta}) = l_i(y[t+1] \mid \boldsymbol{\theta})}} \\ &= \frac{p(y[t+1], q[t+1] = s_i \mid \mathbf{y}_{1:t}, \boldsymbol{\theta})}{l_i(y[t+1] \mid \boldsymbol{\theta})} \\ &= \frac{\alpha_{t+1}(i \mid \boldsymbol{\theta})}{l_i(y[t+1] \mid \boldsymbol{\theta}) \Lambda_t(\boldsymbol{\theta})} \end{aligned} \quad (11)$$

for $t = 1, 2, \dots, T - 1$. Now, (11) implies

$$\varphi_t(j | \boldsymbol{\theta}) = \frac{\alpha_t(j | \boldsymbol{\theta})}{l_j(y[t] | \boldsymbol{\theta}) \Lambda_{t-1}(\boldsymbol{\theta})} \quad (12)$$

and, therefore,

$$l_j(y[t] | \boldsymbol{\theta}) \varphi_t(j | \boldsymbol{\theta}) = \frac{\alpha_t(j | \boldsymbol{\theta})}{\Lambda_{t-1}(\boldsymbol{\theta})}. \quad (13)$$

We now present a recursion for computing $\varphi_t(i | \boldsymbol{\theta})$. Start with (9) and continue as follows:

$$\begin{aligned} \varphi_{t+1}(i | \boldsymbol{\theta}) &\stackrel{\text{see (11)}}{=} \frac{\alpha_{t+1}(i | \boldsymbol{\theta})}{l_i(y[t+1] | \boldsymbol{\theta}) \Lambda_t(\boldsymbol{\theta})} \\ &\stackrel{\text{see (5)}}{=} \frac{\left[\sum_{j=1}^N \alpha_t(j | \boldsymbol{\theta}) a(j, i) \right] \cdot l_i(y[t+1] | \boldsymbol{\theta})}{l_i(y[t+1] | \boldsymbol{\theta}) \omega_t(\boldsymbol{\theta}) \Lambda_{t-1}(\boldsymbol{\theta})} \\ &\stackrel{\text{see (13)}}{=} \sum_{j=1}^N \frac{l_j(y[t] | \boldsymbol{\theta}) \varphi_t(j | \boldsymbol{\theta}) a(j, i)}{\omega_t(\boldsymbol{\theta})} \\ &= \sum_{j=1}^N z_t(j | \boldsymbol{\theta}) a(j, i) \end{aligned} \quad (14)$$

where we have used the chain rule

$$\Lambda_t(\boldsymbol{\theta}) = \omega_t(\boldsymbol{\theta}) \Lambda_{t-1}(\boldsymbol{\theta})$$

and

$$z_t(j | \boldsymbol{\theta}) \triangleq \frac{l_j(y[t] | \boldsymbol{\theta}) \varphi_t(j | \boldsymbol{\theta})}{\omega_t(\boldsymbol{\theta})} = \frac{l_j(y[t] | \boldsymbol{\theta}) \varphi_t(j | \boldsymbol{\theta})}{\sum_{i=1}^N l_i(y[t] | \boldsymbol{\theta}) \varphi_t(i | \boldsymbol{\theta})} \quad (15)$$

which is identical to the normalized forward recursion in [2, eqs. (5.15) and (5.14)] (obtained by substituting (5.14) into (5.15) into [2]).

We can stack $z_t(j | \boldsymbol{\theta})$ into a vector $\mathbf{z}_t(\boldsymbol{\theta})$, say. Then, we can implement (14) as

$$\boldsymbol{\varphi}_{t+1}^T(\boldsymbol{\theta}) = \mathbf{z}_{t+1}^T(\boldsymbol{\theta}) A$$

and (15) as

$$\mathbf{z}_t(\boldsymbol{\theta}) = \mathbf{l}(y[t] | \boldsymbol{\theta}) \odot \boldsymbol{\varphi}_t(\boldsymbol{\theta}) / \omega_t(\boldsymbol{\theta}).$$

Backward-summation Recursion

$$\begin{aligned} p(\mathbf{y} | \boldsymbol{\theta}) &= \sum_{\mathbf{q}} p(\mathbf{y}, \mathbf{q} | \boldsymbol{\theta}) \\ &= \sum_{q[1], \dots, q[T]} \pi_{q[1]} l_{q[1]}(y[1]) a_{q[1], q[2]} l_{q[2]}(y[2]) \\ &\quad \cdot a_{q[2], q[3]} \cdots a_{q[T-1], q[T]} l_{q[T]}(y[T]) \\ &= \sum_{q[1] \in S} \pi_{q[1]} l_{q[1]}(y[1]) \left\{ \sum_{q[2] \in S} a_{q[1], q[2]} l_{q[2]}(y[2]) \right. \\ &\quad \cdots \left[\sum_{q[T-1] \in S} a_{q[T-2], q[T-1]} l_{q[T-1]}(y[T-1]) \right. \\ &\quad \left. \left. \cdot \left[\sum_{q[T] \in S} a_{q[T-1], q[T]} l_{q[T]}(y[T]) \right] \right] \cdots \right\} \end{aligned}$$

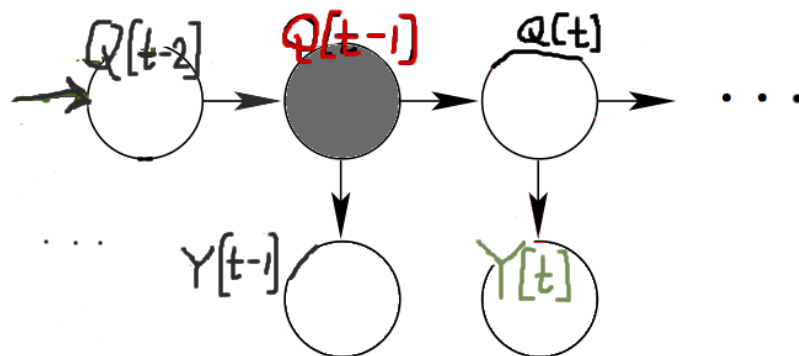
Define

$$\beta_t(i) = p(\mathbf{y}_{(t+1):T} \mid q[t] = s_i, \boldsymbol{\theta}). \quad (16)$$

Set $\beta_T(i) = 1$ and compute $\beta_{T-1}(i)$, $i = 1, 2, \dots, N$ as

$$\begin{aligned} \beta_{T-1}(i) &= p(y[T] \mid q[T-1] = s_i, \boldsymbol{\theta}) \\ &= \sum_{j \in \mathcal{S}} p(y[T], q[T] = s_j \mid q[T-1] = s_i, \boldsymbol{\theta}) \\ &= \underbrace{p(y[T] \mid q[T] = s_j, q[T-1] = s_i, \boldsymbol{\theta})}_{l_j(y[T])} \\ &\quad \cdot \underbrace{p(q[T] = s_j \mid q[T-1] = s_i, \boldsymbol{\theta})}_{a_{i,j}} \\ &= \sum_{j \in \mathcal{S}} a_{i,j} l_j(y[T]) = \left[\sum_{j \in \mathcal{S}} a_{i,j} l_j(y[T]) \right] \cdot \beta_T(i) \end{aligned}$$

where the following graph:



implies

$$Y[t] \perp\!\!\!\perp Q[t-1] \mid \{Q[t] = q[t], \Theta = \theta\}$$

and, therefore,

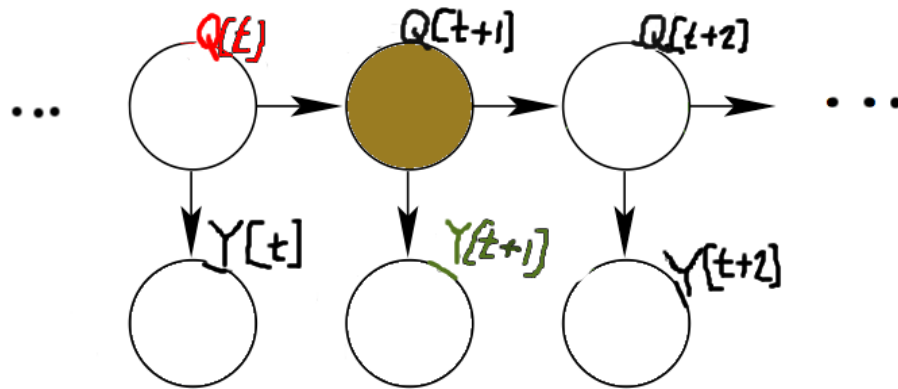
$$p(y[t] \mid q[t], q[t-1], \theta) = p(y[t] \mid q[t], \theta).$$

Assume that $\beta_{t+1}(j)$, $j = 1, 2, \dots, N$ are available and utilize

this information to compute $\beta_t(i)$, $i = 1, 2, \dots, N$:

$$\begin{aligned}
 \underbrace{\beta_t(i)}_{p(\mathbf{y}_{(t+1):T} | q[t]=s_i, \boldsymbol{\theta})} &= \sum_{j=1}^N p(\mathbf{y}_{(t+1):T}, q[t+1] = s_j | q[t] = s_i, \boldsymbol{\theta}) \\
 &= \sum_{j=1}^N p(y[t+1], \mathbf{y}_{(t+2):T}, q[t+1] = s_j | q[t] = s_i, \boldsymbol{\theta}) \\
 &= \sum_{j=1}^N \underbrace{p(y[t+1] | \mathbf{y}_{(t+2):T}, q[t+1] = s_j, q[t] = s_i, \boldsymbol{\theta})}_{l_j(y[t+1])} \\
 &\quad \cdot p(\mathbf{y}_{(t+2):T}, q[t+1] = s_j | q[t] = s_i, \boldsymbol{\theta}) \\
 &= \sum_{j=1}^N l_j(y[t+1]) \cdot \underbrace{p(\mathbf{y}_{(t+2):T} | q[t+1] = s_j, q[t] = s_i, \boldsymbol{\theta})}_{\beta_{t+1}(j)} \\
 &\quad \cdot \underbrace{p(q[t+1] = s_j | q[t] = s_i, \boldsymbol{\theta})}_{a_{i,j}} \\
 &= \sum_{j=1}^N a_{i,j} l_j(y[t+1]) \beta_{t+1}(j)
 \end{aligned}$$

for $t = T - 1, T - 2, \dots, 1$ and $i = 1, 2, \dots, N$, see the HMM graphs below where we observe:



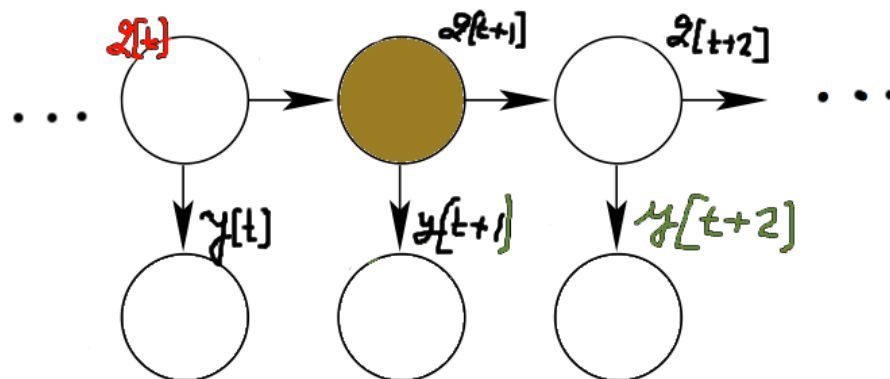
implying

$$Y[t + 1] \perp\!\!\!\perp \mathbf{Y}_{(t+2):T}, Q[t] \mid \{Q[t + 1], \Theta\}$$

and, therefore,

$$p(y[t + 1] \mid \mathbf{y}_{(t+2):T}, q[t], q[t + 1], \theta) = p(y[t + 1] \mid q[t + 1], \theta)$$

and



implying

$$\mathbf{Y}_{(t+2):T} \perp\!\!\!\perp Q[t] \mid \{Q[t + 1], \Theta\}$$

and, therefore,

$$p(\mathbf{y}_{(t+2):T} | q[t], q[t + 1], \boldsymbol{\theta}) = p(\mathbf{y}_{(t+2):T} | q[t + 1], \boldsymbol{\theta}).$$

We summarize the above backward recursion:

$$\begin{aligned}\beta_T(i) &= 1 \\ \beta_t(i) &= \sum_{j=1}^N a_{i,j} l_j(y[t + 1]) \beta_{t+1}(j)\end{aligned}$$

for $t = T - 1, T - 2, \dots, 1$ and $i = 1, 2, \dots, N$.

We should *stabilize* this recursion as well, using similar ideas as before. For example, we can simply normalize $\beta_t(i)$ at every step so that $\sum_{i=1}^N \beta_t(i) = 1$.

Implementation:

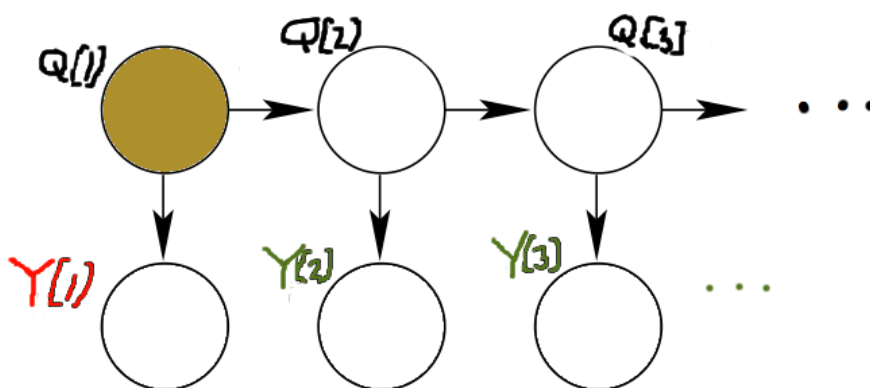
$$\boldsymbol{\beta}_t = A [\mathbf{l}(y[t + 1]) \odot \boldsymbol{\beta}_{t+1}]$$

and then normalize so that the elements of $\boldsymbol{\beta}_t$ sum to one.

Note that

$$\begin{aligned}
 p(\mathbf{y} | \boldsymbol{\theta}) &= \sum_{i=1}^N p(\mathbf{y}_{1:T}, q[1] = s_i | \boldsymbol{\theta}) \\
 &= \sum_{i=1}^N p(y[1], \mathbf{y}_{2:T}, q[1] = s_i | \boldsymbol{\theta}) \\
 &= \sum_{i=1}^N \underbrace{p(y[1], q[1] = s_i | \boldsymbol{\theta})}_{\pi_i l_i(y[1])} \cdot \underbrace{p(\mathbf{y}_{2:T} | y[1], q[1] = s_i, \boldsymbol{\theta})}_{p(\mathbf{y}_{2:T} | q[1]=s_i, \boldsymbol{\theta})} \\
 &= \sum_{i=1}^N \pi_i l_i(y[1]) \underbrace{p(\mathbf{y}_{2:T} | q[1] = s_i, \boldsymbol{\theta})}_{\beta_1(i), \text{ see (16)}} \\
 &= \sum_{i=1}^N \pi_i l_i(y[1]) \beta_1(i)
 \end{aligned}$$

see the HMM graph below, where we observe:



implying

$$\mathbf{Y}_{2:T} \perp\!\!\!\perp \mathbf{Y}[1] \mid \{Q[1] = q[1], \Theta = \theta\}$$

and, therefore,

$$p(\mathbf{y}_{2:T} \mid y[1], q[1], \theta) = p(\mathbf{y}_{2:T} \mid q[1], \theta).$$

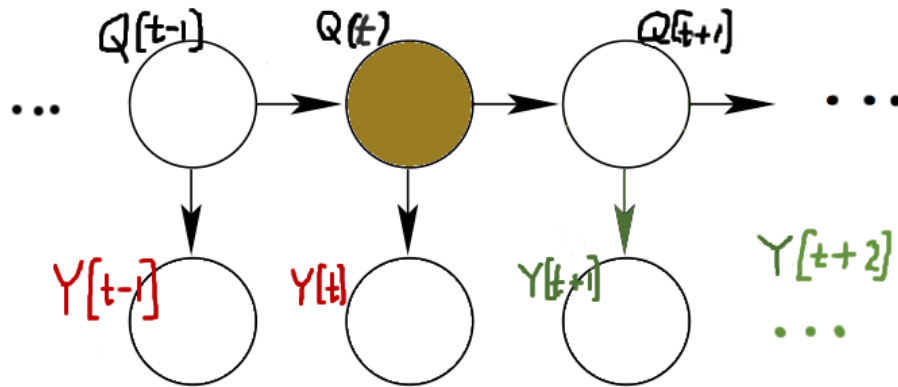
Assume that the model parameters θ are *known* — we will discuss their estimation later in this handout (pp. 39–44).

Marginal posterior pdf of the hidden state $q[t]$ given all measurements y :

$$\begin{aligned} \gamma_t(i) &\triangleq \Pr_{Q[t] \mid \mathbf{Y}, \Theta} \{Q[t] = s_i \mid \underbrace{\mathbf{y}}_{\mathbf{y}_{1:T}}, \theta\} \\ &\propto p(q[t] = s_i, \mathbf{y} \mid \theta) \\ &\propto p(\mathbf{y}_{1:t}, \mathbf{y}_{(t+1):T}, q[t] = s_i \mid \theta) \\ &\propto \underbrace{p(\mathbf{y}_{1:t}, q[t] = s_i \mid \theta)}_{\alpha_t(i), \text{ see (4)}} \cdot \underbrace{p(\mathbf{y}_{(t+1):T} \mid \mathbf{y}_{1:t}, q[t] = s_i, \theta)}_{p(\mathbf{y}_{t+1:T} \mid q[t]=s_i, \theta) = \beta_t(i) \text{ see (16)}} \\ &\propto \alpha_t(i) \beta_t(i) \end{aligned} \tag{17}$$

$$\stackrel{\text{see (12)}}{\propto} \varphi(i \mid \theta) l_i(y[t] \mid \theta) \beta_t(i) \tag{18}$$

see the HMM graph below where we observe:



implying

$$\mathbf{Y}_{(t+1):T} \perp\!\!\!\perp \mathbf{Y}_{1:t} \mid \{Q[t], \boldsymbol{\Theta}\}$$

and, therefore,

$$p(\mathbf{y}_{(t+1):T} \mid \mathbf{y}_{1:t}, q[t], \boldsymbol{\theta}) = p(\mathbf{y}_{(t+1):T} \mid q[t], \boldsymbol{\theta}).$$

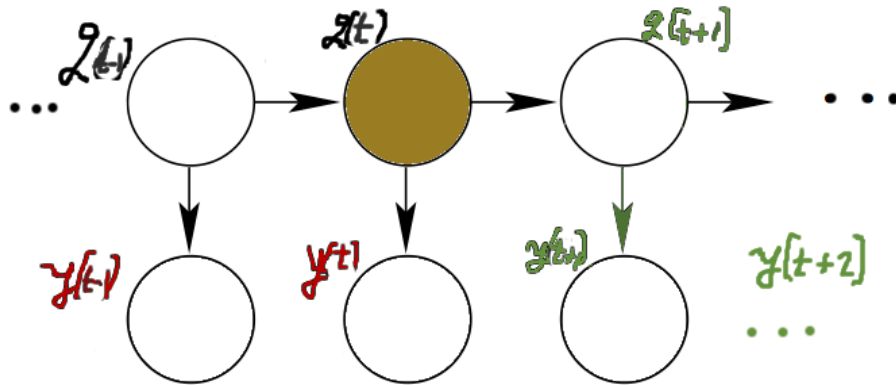
Normalize (17) and (18) to obtain the exact expression for the marginal posterior pmf $\gamma_t(i)$:

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)} = \frac{l_i(y[t] \mid \boldsymbol{\theta}) \varphi_t(i \mid \boldsymbol{\theta}) \beta_t(i)}{\sum_{j=0}^{N-1} l_j(y[t] \mid \boldsymbol{\theta}) \varphi_t(j \mid \boldsymbol{\theta}) \beta_t(j)}$$

for $t = 1, 2, \dots, T$. Furthermore,

$$\begin{aligned}
 \xi_t(i, j) &\triangleq \Pr_{Q[t], Q[t+1] | \mathbf{Y}, \Theta} \{Q[t] = s_i, Q[t+1] = s_j | \mathbf{y}, \boldsymbol{\theta}\} \\
 &\propto p(q[t] = s_i, q[t+1] = s_j, \mathbf{y} | \boldsymbol{\theta}) \\
 &= \underbrace{p(\mathbf{y}_{1:t}, q[t] = s_i | \boldsymbol{\theta})}_{\alpha_t(i)} \\
 &\quad \cdot \underbrace{p(q[t+1] = s_j, \mathbf{y}_{(t+1):T} | \mathbf{y}_{1:t}, q[t] = s_i, \boldsymbol{\theta})}_{p(q[t+1]=s_j, \mathbf{y}_{(t+1):T} | q[t]=s_i, \boldsymbol{\theta})} \\
 &= \alpha_t(i) p(\mathbf{y}_{(t+1):T}, q[t+1] = s_j | q[t] = s_i, \boldsymbol{\theta}) \\
 &= \alpha_t(i) \underbrace{p(\mathbf{y}_{(t+1):T} | q[t+1] = s_j, q[t] = s_i, \boldsymbol{\theta})}_{p(\mathbf{y}_{(t+1):T} | q[t+1]=s_j, \boldsymbol{\theta})} \\
 &\quad \cdot \underbrace{p(q[t+1] = s_j | q[t] = s_i, \boldsymbol{\theta})}_{a_{i,j}} \\
 &= \alpha_t(i) a_{i,j} p(y[t+1], \dots, y[T] | q[t+1] = s_j, \boldsymbol{\theta}) \\
 &= \alpha_t(i) a_{i,j} l_j(y[t+1]) \underbrace{p(\mathbf{y}_{(t+2):T} | q[t+1] = s_j, \boldsymbol{\theta})}_{\beta_{t+1}(j)} \\
 &= \alpha_t(i) a_{i,j} l_j(y[t+1]) \beta_{t+1}(j)
 \end{aligned}$$

see the HMM graph below where we observe:



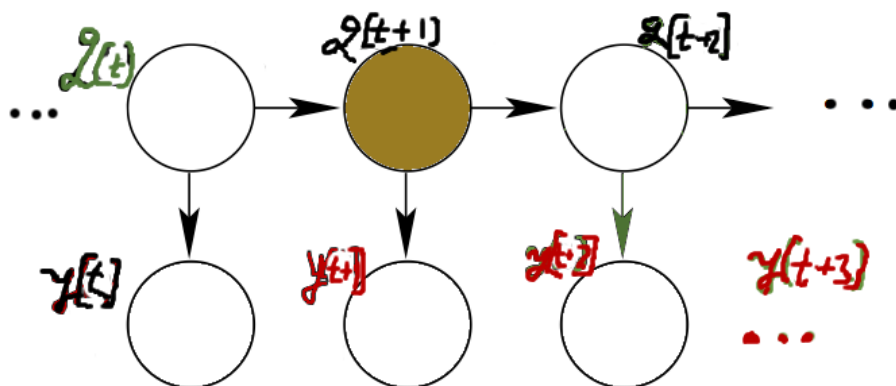
implying

$$Q[t+1], \mathbf{Y}_{(t+1):T} \perp\!\!\!\perp \mathbf{Y}_{1:t} \mid Q[t], \Theta$$

and, therefore,

$$p(q[t+1], y[t+1], y[t+2], \dots, y[T] \mid y[1], y[2], \dots, y[t], q[t], \theta)$$

and



implying

$$y[t+1], y[t+2], \dots \perp\!\!\!\perp q[t] \mid q[t+1], \theta$$

and, therefore,

$$\begin{aligned} p(y[t + 1], \dots, y[T] \mid q[t + 1] = s_j, q[t] = s_i, \boldsymbol{\theta}) \\ = p(y[t + 1], \dots, y[T] \mid q[t + 1] = s_j, \boldsymbol{\theta}). \end{aligned}$$

Hence, the posterior probability $\xi_t(i, j)$ that the HMM system was in state i at time t *and* transitioned to state j at time $t + 1$ is

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{i,j} l_j(y[t + 1]) \beta_{t+1}(j)}{\sum_{i'=1}^N \sum_{j'=1}^N \alpha_t(i') a_{i',j'} l_{j'}(y[t + 1]) \beta_{t+1}(j')}$$

for $t = 1, 2, \dots, T - 1$. Observe that, due to total probability,

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j).$$

Optimal State-path and Individual-state Estimation

We assume that the model parameters θ are known and focus on estimation of q .

- For squared-error loss, the minimum mean-square error (MMSE) estimate of Q is optimal and given by

$$E_{Q|Y,\theta}(Q | \mathbf{y}, \theta)$$

where the expectation is with respect to the marginal posterior pdf:

$$\gamma_t(i) = p_{Q[t]|Y,\theta}\{s_i | \mathbf{y}, \theta\} = \Pr_{Q[t]|Y,\theta}\{Q[t] = s_i | \mathbf{y}, \theta\}.$$

But, the obtained state path will be impossible in general.

- If we use the 0-1 loss, the maximum *a posteriori* (MAP) estimate is optimal. It is obtained by maximizing

$$p(\mathbf{q} | \mathbf{y}, \theta) \propto p(\mathbf{y}, \mathbf{q} | \theta).$$

Note: We can also decide to make inference on individual states (rather than the entire path); this inference should be

based on the posterior marginal pmf

$$\gamma_t(i) = \Pr_{Q|Y,\Theta}\{Q[t] = s_i | \mathbf{y}, \boldsymbol{\theta}\} = p_{Q[t]|Y,\Theta}(s_i | \mathbf{y}, \boldsymbol{\theta}).$$

Then, we can obtain an estimate of $Q[t]$ as the mode of $P_{Q|Y,\Theta}\{q[t] = s_i | \mathbf{y}, \boldsymbol{\theta}\}$ [i.e. the *marginal posterior mode* (MPM)]:

$$q^{\text{MPM}}[t] = s_{\arg \max_i \gamma_t(i)}$$

known as the **BCJR** algorithm in digital communications [4]; the name BCJR comes from the first letters of the last names of the authors of [4]: L. **B**ahl, J. **C**ocke, F. **J**elinek, and J. **R**aviv.

Comments:

- Ideally, we should utilize the marginal posterior pmf $\gamma_t(i)$ for inference on an individual symbol $q[t]$, because it fully takes into account *all sources of uncertainty* in the system if $\boldsymbol{\theta}$ is known, due to *both*
 - the measurement-error model and
 - the fact that the other states $q[1], q[2], \dots, q[t-1], q[t+1], \dots, q[T]$ (i.e. other than $q[t]$ about which we wish to make a decision) are *unknown*.
- The entire marginal pmf $\gamma_t(i)$ provides information about reliability of our inference for each hidden state $Q[t]$. Utilizing this reliability information has been a popular research topic in digital communications.

Viterbi Algorithm

Let us go back to estimating the optimal state path, i.e. maximizing

$$p(\mathbf{y}, \mathbf{q} | \boldsymbol{\theta}).$$

with respect to \mathbf{q} .

We apply dynamic programming again, now known as the *Viterbi algorithm*. Note that

$$\begin{aligned} & \max_{\mathbf{q}} p(\mathbf{y}, \mathbf{q} | \boldsymbol{\theta}) \\ &= \max_{q[1], \dots, q[T]} \pi_{q[1]} l_{q[1]}(y[1]) a_{q[1], q[2]} l_{q[2]}(y[2]) \\ & \quad \cdot a_{q[2], q[3]} \cdots a_{q[T-1], q[T]} l_{q[T]}(y[T]) \\ &= \max_{q[T] \in S} \left[\max_{q[T-1] \in S} \cdots \left[\max_{q[3] \in S} \left[\max_{q[2] \in S} \right. \right. \right. \\ & \quad \left. \left. \left[\max_{q[1] \in S} \pi_{q[1]} l_{q[1]}(y[1]) a_{q[1], q[2]} l_{q[2]}(y[2]) \right] \right. \right. \\ & \quad \left. \left. a_{q[2], q[3]} l_{q[3]}(y[3]) \right] a_{q[3], q[4]} l_{q[4]}(y[4]) \right] \\ & \quad \left. \cdots \right] a_{q[T-1], q[T]} l_{q[T]}(y[T]) \right]. \end{aligned}$$

This scheme is very similar to the forward summation: simply replace \sum with \max . As before, it is just a clever way of putting brackets to segment the likelihood expression (for *complete data*, in this case).

Define

$$\delta_t(i) \triangleq \max_{\mathbf{q}_{1:(t-1)}} p(\mathbf{y}_{1:t}, \mathbf{q}_{1:(t-1)}, q[t] = s_i \mid \boldsymbol{\theta})$$

the maximum probability of all ways to end in state s_i at time t , having observed $\mathbf{y}_{1:t}$.

Then, by analogy with the forward summation, we obtain

$$\delta_{t+1}(i) = \max_j \{ \delta_t(j) a_{j,i} \} l_i(y[t+1]) \quad (19)$$

for $t = 1, 2, \dots, T-1$ and $i = 1, 2, \dots, N$. Compare (19) with (6): it follows by replacing \sum with \max and α with δ . Now,

$$\delta_1(i) = \pi_i l_i(y[1])$$

the same as $\alpha_1(i)$. To summarize, here is our recursion, analogous to (5)–(6):

$$\delta_1(i) = \pi_i l_i(\mathbf{y}[1]) \quad (20)$$

$$\delta_{t+1}(i) = \max_j \{ \delta_t(j) a_{j,i} \} l_i(\mathbf{y}[t+1]) \quad (21)$$

for $t = 1, 2, \dots, T-1$ and $i = 1, 2, \dots, N$.

Applying the above recursion gives us an $N \times T$ matrix of delta values $\{\delta_t(i), i = 1, 2, \dots, N, t = 1, 2, \dots, T\}$. This matrix should be utilized to obtain the path $\mathbf{q} = \mathbf{q}^{\text{opt}} = [q^{\text{opt}}[1], q^{\text{opt}}[2], \dots, q^{\text{opt}}[T]]^T$ that maximizes $p(\mathbf{y}, \mathbf{q} | \boldsymbol{\theta})$. For this purpose, we apply the following strategy:

- Calculate $\delta_t(i)$, $i = 1, 2, \dots, N$, $t = 1, 2, \dots, T$ using the above recursion;
- Then, backtrack and recover the optimal sequence \mathbf{q}^{opt} that gives $\max_{\mathbf{q}} p(\mathbf{y}, \mathbf{q} | \boldsymbol{\theta}) = \max_{i=1,2,\dots,N} \delta_T(i)$.

Define

$$\psi_T \triangleq \arg \max_{i=1,2,\dots,N} \delta_T(i)$$

and set

$$q^{\text{opt}}[T] = s_{\psi_T}.$$

Clearly, $q^{\text{opt}}[T]$ is the final state in the optimal sequence and

$$\begin{aligned} \max_{\mathbf{q}} p(\mathbf{y}, \mathbf{q} | \boldsymbol{\theta}) &= \delta_T(\psi_T) \\ &= \max_j \{ \delta_{T-1}(j) a_{j, \psi_T} \} l_{\psi_T}(y[t+1]) \end{aligned}$$

which implies that we can obtain $q^{\text{opt}}[T-1]$ as

$$q^{\text{opt}}[T-1] = s_{\psi_{T-1}}$$

where

$$\psi_{T-1} = \arg \max_{j=1,2,\dots,N} \{ \delta_{T-1}(j) a_{j, \psi_T} \}.$$

To facilitate this insight and define the general recursion (for all times t), let us introduce the following notation:

$$\psi_t = \arg \max_{j=1,2,\dots,N} \{ \delta_t(j) a_{j, \psi_{t+1}} \}$$

for $t = T-1, T-2, \dots, 1$. Then,

$$q^{\text{opt}}[t] = s_{\psi_t} \quad t = T-1, T-2, \dots, 1.$$

If the $\arg \max$ operation in the above scheme is not unique and if our goal is to find one optimal path, then arbitrarily take one value of j giving the maximum. (Otherwise, if we wish to find all optimal sequences, we need to explore all values of j giving the maximum.)

Comments: Viterbi algorithm

- requires only a forward-type recursion and is therefore simpler than the forward-backward approach;

but it

- *does not* provide marginal (state-level) posterior probability distributions — this requires the full forward-backward recursion;
- is inferior to the MPM (BCJR) approach if the signal-to-noise ratio (SNR) is low.

Estimating the Model Parameters θ

We obtain the (marginal) ML estimates of θ by maximizing

$$p(\theta | \mathbf{y})$$

with respect to θ via the EM algorithm. This type of an algorithm for HMM model-parameter estimation was first developed by Baum and Welch in the late 1960s but never published. (It was apparently classified; at that time, this type of work was funded by NSA).

For an application of the EM algorithm (known as Baum-Welch algorithm in the HMM context) to channel estimation in digital communications, see [5].

E Step

In the E step, we compute

$$\begin{aligned} Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}_p) &= \mathbb{E}_{\mathbf{q} \mid \mathbf{y}}[\ln p(\mathbf{q}, \mathbf{y} \mid \boldsymbol{\theta}); \boldsymbol{\theta}_p] \\ &= \mathbb{E}_{\mathbf{q} \mid \mathbf{y}} \left\{ \ln [\pi_{q[1]} l_{q[1]}(y[1]) a_{q[1],q[2]} l_{q[2]}(y[2]) a_{q[2],q[3]} \right. \\ &\quad \left. \cdots a_{q[T-1],q[T]} l_{q[T]}(y[T])] ; \boldsymbol{\theta}_p \right\} \\ &= \mathbb{E}_{\mathbf{q} \mid \mathbf{y}} \left\{ \ln \left[\pi_{q[1]} l_{q[1]}(y[1]) \prod_{t=2}^T a_{q[t-1],q[t]} l_{q[t]}(y[t]) \right] ; \boldsymbol{\theta}_p \right\} \\ &= \mathbb{E}_{\mathbf{q} \mid \mathbf{y}} \{ \ln \pi_{q[1]} ; \boldsymbol{\theta}_p \} + \sum_{t=2}^T \mathbb{E}_{\mathbf{q} \mid \mathbf{y}} \{ \ln a_{q[t-1],q[t]} ; \boldsymbol{\theta}_p \} \\ &\quad + \sum_{t=1}^T \mathbb{E}_{\mathbf{q} \mid \mathbf{y}} \{ \ln l_{q[t]}(y[t]) ; \boldsymbol{\theta}_p \}. \end{aligned}$$

We now explicitly compute the terms in the above summation:

$$\mathbb{E}_{\mathbf{q}|\mathbf{y}}\{\ln \pi_{q[1]}; \boldsymbol{\theta}_p\} = \sum_{i=1}^N \ln \pi_i \cdot \underbrace{p(q[1] = s_i | \mathbf{y}; \boldsymbol{\theta}_p)}_{\gamma_{1,p}(i)}$$

$$\begin{aligned} \mathbb{E}_{\mathbf{q}|\mathbf{y}}\{\ln a_{q[t-1],q[t]}; \boldsymbol{\theta}_p\} \\ = \sum_{i=1}^N \sum_{j=1}^N \ln a_{i,j} \cdot \underbrace{p(q[t-1] = s_i, q[t] = s_j | \mathbf{y}; \boldsymbol{\theta}_p)}_{\xi_{t-1,p}(i,j)} \end{aligned}$$

$$\mathbb{E}_{\mathbf{q}|\mathbf{y}}\{\ln l_{q[t]}(y[t]); \boldsymbol{\theta}_p\} = \sum_{i=1}^N \ln l_i(y[t]) \cdot \underbrace{p(q[t] = s_i | \mathbf{y}; \boldsymbol{\theta}_p)}_{\gamma_{t,p}(i)}.$$

M Step

Maximize

$$\begin{aligned} Q(\boldsymbol{\theta} | \boldsymbol{\theta}_p) &= \left[\sum_{i=1}^N \gamma_{1,p}(i) \ln \pi_i \right] \\ &+ \left[\sum_{t=1}^T \sum_{i=1}^N \gamma_{t,p}(i) \ln l_i(y[t]) \right] \\ &+ \sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N \xi_{t,p}(i, j) \ln a_{i,j} \end{aligned}$$

subject to

$$\begin{aligned} \sum_{i=1}^N \pi_i &= 1 \\ \sum_{j=1}^N a_{i,j} &= 1, \quad i = 1, 2, \dots, N \\ \sum_{m=1}^M l_i(v_m) &= 1, \quad i = 1, 2, \dots, N. \end{aligned}$$

Using Lagrange multipliers, we obtain the following

unconstrained optimization problem: maximize

$$\begin{aligned}
 & \left[\sum_{i=1}^N \gamma_{1,p}(i) \ln \pi_i \right] + \left[\sum_{t=1}^T \sum_{i=1}^N \gamma_{t,p}(i) \ln l_i(y[t]) \right] \\
 & + \left[\sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N \xi_{t,p}(i, j) \ln a_{i,j} \right] - \lambda_1 \left(\sum_{i=1}^N \pi_i - 1 \right) \\
 & - \left[\sum_{i=1}^N \lambda_{2,i} \left(\sum_{j=1}^N a_{i,j} - 1 \right) \right] - \left\{ \sum_{i=1}^N \lambda_{3,i} [l_i(v_m) - 1] \right\}
 \end{aligned}$$

which leads to

$$\begin{aligned}
 (\pi_i)_{p+1} &= \gamma_{1,p}(i) \\
 (a_{i,j})_{p+1} &= \frac{\sum_{t=2}^T \xi_{t-1,p}(i, j)}{\sum_{t=2}^T \underbrace{\sum_{\ell=1}^N \xi_{t-1,p}(i, \ell)}_{\gamma_{t-1,p}(i)}} = \frac{\sum_{t=1}^{T-1} \xi_{t,p}(i, j)}{\sum_{t=1}^{T-1} \gamma_{t,p}(i)} \\
 [l_i(v_m)]_{p+1} &= \frac{\sum_{t=1}^T \mathbf{s.t. } y[t]=v_m \gamma_{t,p}(i)}{\sum_{\mu=1}^M \sum_{t=1}^T \mathbf{s.t. } y[t]=v_\mu \gamma_{t,p}(i)} \\
 &= \left[\sum_{\substack{t=1 \\ y[t]=v_m}}^T \gamma_{t,p}(i) \right] / \left[\sum_{t=1}^T \gamma_{t,p}(i) \right].
 \end{aligned}$$

To summarize, in the $(p + 1)$ st step, we first compute $\gamma_{t,p}(i)$ and $\xi_{t,p}(i, j)$ for $i, j = 1, 2, \dots, N$ and $t = 1, 2, \dots, T$ using the model parameters $\boldsymbol{\theta} = \boldsymbol{\theta}_p$ (which are the latest parameter estimates updated in the p th step). Then, we update the model parameters as follows:

$$\begin{aligned}
 (\pi_i)_{p+1} &= \gamma_{1,p}(i) \\
 (a_{i,j})_{p+1} &= \frac{\sum_{t=1}^{T-1} \xi_{t,p}(i, j)}{\sum_{t=1}^{T-1} \gamma_{t,p}(i)} \\
 [l_i(v_m)]_{p+1} &= \left[\sum_{\substack{t=1 \\ y[t] = v_m}}^T \gamma_{t,p}(i) \right] / \left[\sum_{t=1}^T \gamma_{t,p}(i) \right].
 \end{aligned}$$

References

- [1] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–286, Feb. 1989. [\(document\)](#)
- [2] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1518–1569, Jun. 2002. [\(document\)](#)

- [3] O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models*. New York: Springer-Verlag, 2005.
- [4] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inform. Theory*, vol. 49, pp. 284–287, Mar. 1974. (document)
- [5] H.A. Cirpan and M.K. Tsatsanis, “Stochastic maximum likelihood methods for semi-blind channel estimation,” *IEEE Signal Processing Lett.*, vol. 5, pp. 21–24, Jan. 1998. (document)