21.1 Two HMM problems

In this chapter, we focus on two types of problems that HMMs can solve.

1. **The evaluation problem**
   - **Given:** $\sigma, \lambda$.
   - **Compute:** $p(\sigma|\lambda)$, the probability of observing the sequence $\sigma$ in the model $\lambda$.

2. **The decoding problem**
   - **Given:** $\sigma, \lambda$.
   - **Compute:** The sequence of hidden states $Q = q_1q_2\ldots q_T$ that optimally “explains” the sequence of observations.

21.1.1 Solving the evaluation problem

We solve the evaluation problem via the forward algorithm, which makes use of the forward variable

$$\alpha_t(i) = p(\sigma_1\sigma_2\ldots\sigma_t, q_t = s_i).$$

The forward algorithm for finding $\alpha_t(i)$ and $p(\sigma|\lambda)$ is as follows.

1. **Initialization**
   
   $\alpha_1(i) = \pi_ib_i(\sigma_1)$
   
   for $1 \leq i \leq N$. This expression gives the joint probability of the first observed symbol and some state $i$, $p(\sigma_1, s_i)$.

2. **Recurrence**
   
   $$\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i)a_{ij}b_j(\sigma_{t+1})$$
   
   for $1 \leq i, j \leq N$ and $1 \leq t \leq T - 1$. The sum in the above expression accounts for the fact that we can reach state $s_j$ at time $t + 1$ by coming from any state $s_1, s_2, \ldots, s_N$ at time $t$. Observe that $\alpha_t(i)$ is the probability of the joint event of observing the sequence $\sigma_1\sigma_2\ldots\sigma_t$ and ending up at state $s_i$ at time $t$. Thus, $\alpha_t(i)a_{ij}$ is the probability of the joint event of observing $\sigma_1\sigma_2\ldots\sigma_t$ and going from state $s_i$ at time $t$ to state $s_j$ at time $t + 1$. If we let $i$ range over the $N$ states and sum up, we obtain the joint probability of observing $\sigma_1\sigma_2\ldots\sigma_t$ and ending up at state $s_j$ at time $t + 1$.
3. Termination

\[ p(\sigma|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \]

### 21.1.2 Solving the decoding problem

To solve the decoding problem, we need to define what we mean by finding the “best” sequence of hidden states \( Q \). We will proceed by maximizing the probability of the hidden state sequence \( Q \) given the observed sequence \( \sigma \), \( p(Q|\sigma) \), over all possible choices of \( Q \). The famous Viterbi algorithm carries out this maximization. Define \( \delta_t(i) \) by

\[ \delta_t(i) = \max_{q_1 \ldots q_{t-1}} p(q_1 \ldots q_{t-1}, q_t = s_i \mid \sigma_1 \ldots \sigma_t). \]

In words, \( \delta_t(i) \) is the best score (i.e. highest probability) along the path defined by \( q_1 q_2 \ldots q_{t-1} q_t \) that accounts for the first \( t \) observation symbols and ends up at the state \( s_i \). The key recurrence is

\[ \delta_{t+1}(j) = \max_i \delta_t(i) a_{ij} b_j(\sigma_{t+1}). \]

Here is the algorithm.

1. **Initialization**

\[ \delta_1(i) = \pi_i b_i(\sigma_1), \]
\[ \psi_1(i) = 0, \]

for \( 1 \leq i \leq N \).

2. **Recurrence**

\[ \delta_t(j) = \max_{1 \leq i \leq N} \delta_{t-1}(i) a_{ij} b_j(\sigma_t), \]
\[ \psi_t(j) = \text{argmax}_{1 \leq i \leq N} \delta_{t-1}(i) a_{ij}, \]

for \( 2 \leq t \leq T \) and \( 1 \leq j \leq N \).

3. **Termination**

\[ p^* = \max_{1 \leq i \leq N} \delta_T(i), \]
\[ q_T^* = \text{argmax}_{1 \leq i \leq N} \delta_T(i). \]

4. **Back tracking**

We recover the optimal hidden state path by the following recurrence for \( q_t^* \).

\[ q_t^* = \psi_{t+1}(q_{t+1}^*), \]

for \( t = T - 1, T - 2, \ldots, 1 \).