The Backward Algorithm

Of the HMM algorithms we currently know, the Forward algorithm finds the probability of a sequence $P(x)$ and the Viterbi algorithm finds the most probable path that generated sequence x.

However, we may need to do further inference on the sequence. For example, we may wish to know the probability that observation x_i in the sequence came from state k i.e. $P(\pi_i = k \mid x)$. This is the posterior probability of state k at time step i when the emitted sequence is known.

The approach to obtaining this posterior probability is a bit indirect. We first calculate the join probability of observing the sequence and having state $\pi_i = k$:

$$\begin{aligned} P(x,\ \pi_i = k) &= P(x_1 \dots x_i,\ \pi_i = k)\ P(x_{i+1} \dots x_L \mid x_1 \dots x_i,\ \pi_i = k) \\ &= P(x_1 \dots x_i,\ \pi_i = k)\ P(x_{i+1} \dots x_L \mid \pi_i = k) \end{aligned}$$

The first term in the product is the probability of observing the sequence up to the i^{th} symbol where the i^{th} symbol is generated from state k. We see that this term is simply a value $f_k(i)$ computed in the forward algorithm. The second term, we call $b_k(i)$, is the second term and it is computed in the **Backward algorithm**:

$$b_k(i) = P(x_{i+1} \dots x_L \mid \pi_i = k)$$

Thus, we calculate $P(x,\ \pi_i = k)$ as follows:

$$P(x,\ \pi_i = k) = P(\pi_i \mid x)P(x) = f_k(i)b_k(i)$$

And therefore,

$$P(\pi_i \mid x) = \frac{f_k(i)b_k(i)}{P(x)}$$

$$= \frac{f_k(i)b_k(i)}{f_N(L)}$$

where L is the length of sequence x and N is the end state in the HMM.

Description

We build a dynamic programming matrix such that the (k, i)th value of the matrix is defined as:

$$b_k(i) = P(x_{i+1} \dots x_L \mid \pi_i = k) \tag{1}$$

That is, $b_k(i)$ stores the probability of observing the rest of the sequence after time step i given that at time step i we are in state k in the HMM.

We terminate when we compute $b_0(0)$ which is the probability of observing the entire sequence given that the first state is the begin state. In practice, we would usually not run the algorithm to completion because the forward algorithm is used to find the full probability of the sequence. Nonetheless, we see that $P(x)$ is the value of $b_0(0)$ in the Backward algorithm.

We compute the values in the matrix from the right most column (i.e. the L^{th} column) which corresponds to the probabilities of moving to the end state after observing the L^{th} element in the sequence from each state. Thus, we perform the following initialization:

$$b_k(L) = a_{k,0} \text{ for all } k$$

Each element is then calculated using the following recurrence:

$$b_k(i) = \sum_l a_{kl}\, e_l(x_{i+1})\, b_l(i+1) \tag{2}$$

That is, we sum over all states l where each term in the sum is the probability of transition from k to the next state l, denoted a_{kl}, times the probability of emitting the next character in the next state $e_l(x_i + 1)$ all multiplied the backward probability calculated from that next entry in the matrix $b_l(i + 1)$.

Example

Assume the following sequence was generated from the example HMM:

$$x = \text{TAGA} \tag{3}$$

We wish to compute the full matrix which is the posterior $P(\pi_i \mid x)$ for every state at every time step. We start by initializing the right-most column in the matrix:

State, l \ Time Step, t	0	1 (T)	2 (A)	3 (G)	4 (A)
γ_0	-	-	-	-	**0**
γ_1	-	-	-	-	**0**
γ_2	-	-	-	-	**0**
γ_3	-	-	-	-	**0.6**
γ_4	-	-	-	-	**0.9**
γ_5	-	-	-	-	**0**

We have fully filled in the probabilities at time step for the last symbol $t = 4$. The entries in this column denote the probability moving to the end state from each state after generating the entire sequence.

We now work backward from the left column to the right column filling in the matrix. We show the calculations for filling in the first four entries of the column corresponding to $t = 3$:

$$b_{\gamma_1}(t = 3) = a_{\gamma_1\gamma_3}\ e_{\gamma_3}(\text{A})\ b_{\gamma_3}(4) + a_{\gamma_1\gamma_1}\ e_{\gamma_1}(\text{A})\ b_{\gamma_1}(4)$$

$$= (0.8 \times 0.2 \times 0.6) + (0.2 \times 0.4 \times 0)$$

$$= 0.096$$

Notice that in the sum $\sum_l a_{kl}\ e_l(x_{i+1})\ b_l(i + 1)$, we only need to sum over the states l where $a_{kl} \neq 0$. Since the only transitions from state γ_1 are to $gamma_3$ and itself, we only include these states in the summation because the transition probabilities to the other states from γ_1 are 0.

We continue with our calculations:

$$b_{\gamma_2}(t = 3) = a_{\gamma_2\gamma_4}\ e_{\gamma_4}(\text{A})\ b_{\gamma_4}(4) + a_{\gamma_2\gamma_2}\ e_{\gamma_2}(\text{A})\ b_{\gamma_2}(4)$$

$$= (0.2 \times 0.1 \times 0.9) + (0.8 \times 0.4 \times 0)$$

$$= 0.018$$

$$b_{\gamma_3}(t=3) = a_{\gamma_3\gamma_5}\ e_{\gamma_3}(\text{A})\ b_{\gamma_5}(4) + a_{\gamma_3\gamma_3}\ e_{\gamma_3}(\text{A})\ b_{\gamma_3}(4)$$

$$= (0.6 \times 0 \times 0) + (0.4 \times 0.2 \times 0.6)$$

$$= 0.048$$

We insert these values in the matrix:

State, l \ Time Step, t	0	1 (T)	2 (A)	3 (G)	4 (A)
γ_0	-	-	-	**0**	0
γ_1	-	-	-	**0.096**	0
γ_2	-	-	-	**0.018**	0
γ_3	-	-	-	**0.048**	0.6
γ_4	-	-	-	-	0.9
γ_5	-	-	-	-	0

We continue these computations and fill in the entire matrix. The full probability of the sequence under the model will be the value $b_{\gamma_0}(t=0)$ (i.e. the tom-most-left entry) in the dynamic programming matrix.