Formalities

The *discrete observation* HMM is restricted to the production of a finite set of discrete observations (or sequences). The output distribution at any state is given by:

$$b(k, i) \equiv P(\underline{y}(t) = k \,|\, \underline{x}(t) = i)$$

The observation probabilities are assumed to be independent of time. We can write the probability of observing a particular observation, $\underline{y}(t)$, as:

$$b(y(t) \,|\, i) \equiv P(\underline{y}(t) = y(t) \,|\, \underline{x}(t) = i)$$

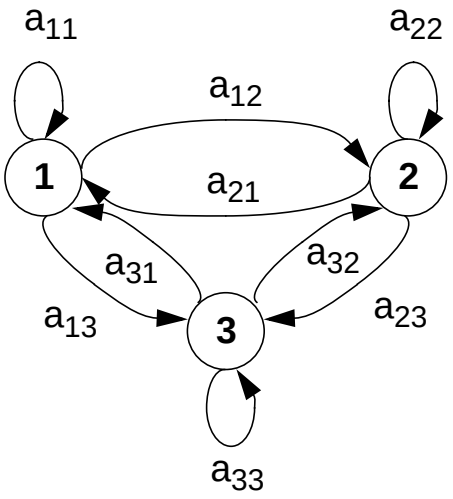
The observation probability distribution can be represented as a matrix whose dimension is K rows x S states.
We can define the observation probability vector as:

$$\boldsymbol{p}(t) = \begin{bmatrix} P(\underline{y}(t) = 1) \\ P(\underline{y}(t) = 2) \\ \cdots \\ P(\underline{y}(t) = K) \end{bmatrix}, \quad \text{or,} \quad \boldsymbol{p}(t) = \boldsymbol{B}\pi(t) = \boldsymbol{B}\boldsymbol{A}^{t-1}\pi(1)$$

The mathematical specification of an HMM can be summarized as:

$$\boldsymbol{M} = \{S, \pi(1), \boldsymbol{A}, \boldsymbol{B}, \{\boldsymbol{y}_k, 1 \le k \le K\}\}$$

For example, reviewing our coin-toss model:



P(H): P_1 P_2 P_3
P(T): $1\text{-}P_1$ $1\text{-}P_2$ $1\text{-}P_3$

$$S = 3$$

$$\pi(1) = \begin{Bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{Bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} P_1 & P_2 & P_3 \\ 1-P_1 & 1-P_2 & 1-P_3 \end{bmatrix}$$

Recognition Using Discrete HMMs

Denote any partial sequence of observations in time by:

$$y_{t_1}^{t_2} \equiv \{y(t_1), y(t_1+1), y(t_1+2), \ldots, y(t_2)\}$$

The forward partial sequence of observations at time t is

$$y_1^t \equiv \{y(1), y(2), \ldots, y(t)\}$$

The backward partial sequence of observations at time t is

$$y_{t+1}^T \equiv \{y(t+1), y(t+2), \ldots, y(T)\}$$

A complete set of observations of length T is denoted as $y \equiv y_1^T$.

What is the likelihood of an HMM?

We would like to calculate $P(M|\underline{y} = y)$ — however, we can't. We can (see the introductory notes) calculate $P(\underline{y} = y|M)$. Consider the brute force method of computing this. Let $\vartheta = \{i_1, i_2, \ldots, i_T\}$ denote a specific state sequence. The probability of a given observation sequence being produced by this state sequence is:

$$P(y|\vartheta, M) = b(y(1)|i_1)b(y(2)|i_2)\ldots b(y(T)|i_T)$$

The probability of the state sequence is

$$P(\vartheta|M) = P(\underline{x}(1) = i_1)a(i_2|i_1)a(i_3|i_2)\ldots a(i_T|i_{T-1})$$

Therefore,

$$P(y,(\vartheta|M)) = P(\underline{x}(1) = i_1)a(i_2|i_1)a(i_3|i_2)\ldots a(i_T|i_{T-1})$$
$$x \; b(y(1)|i_1)b(y(2)|i_2)\ldots b(y(T)|i_T)$$

To find $P(y|M)$, we must sum over all possible paths:

$$P(y|M) = \sum_{\forall \vartheta} P(y,(\vartheta|M))$$

This requires $O(2TS^T)$ flops. For $S = 5$ and $T = 100$, this gives about 1.6×10^{72} computations per HMM!

The "Any Path" Method (Forward-Backward, Baum-Welch)

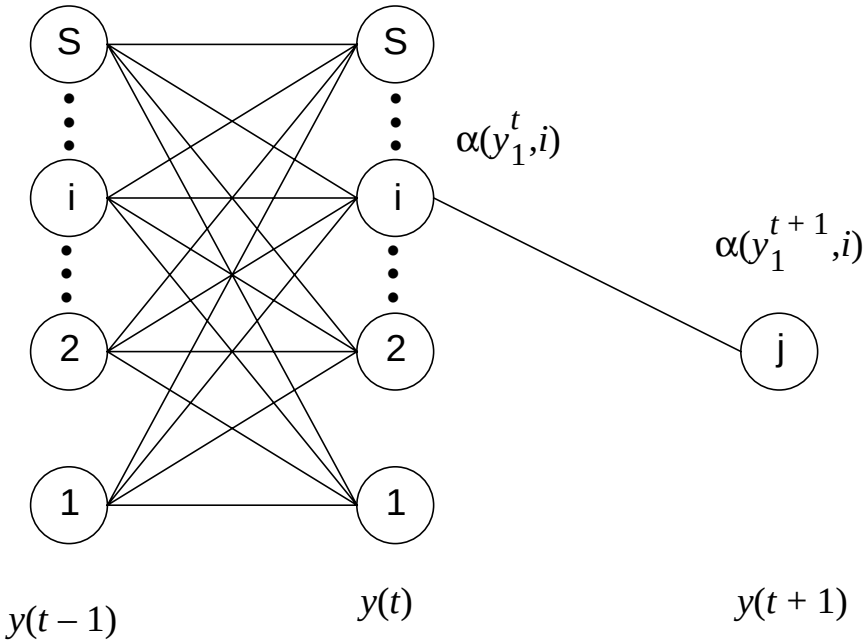
The *forward-backward* (F-B) *algorithm* begins by defining a "forward-going" probability sequence:

$$\alpha(y_1^t) \equiv P(\underline{y}_1^t = y_1^t, \underline{x}(t) = i \,|\, M)$$

and a "backward-going" probability sequence:

$$\beta(y_{t+1}^T \,|\, i) \equiv P(\underline{y}_{t+1}^T = y_{t+1}^T \,|\, \underline{x}(t) = i, M)$$

Let us next consider the contribution to the overall sequence probability made by a single transition:



$$\begin{aligned}\alpha(y_1^{t+1}, j) &= \alpha(y_1^t, i) P(\underline{x}(t+1) = j \,|\, \underline{x}(t) = i) \times \\ &\quad P(\underline{y}(t+1) = y(t+1) \,|\, \underline{x}(t+1) = j) \\ &= \alpha(y_1^t, i) a(j \,|\, i) b(y(t+1) \,|\, j)\end{aligned}$$

Summing over all possibilities for reaching state "j":

$$\alpha(y_1^{t+1}, j) = \sum_{i=1}^{S} \alpha(y_1^t, i) a(j \,|\, i) b(y(t+1) \,|\, j)$$

Baum-Welch (Continued)

The recursion is initiated by setting:

$$\alpha(y_1^t, j) = P(\underline{x}(1) = j) b(y(1) | j)$$

Similarly, we can derive an expression for β:

$$\beta(y_{i+1}^T \Big| i) = \sum_{j=1}^{S} \beta(y_{t+2}^T \Big| j) a(j|i) b(y(t+1)|j)$$

This recursion is initialized by:

$$\beta(y_{T+1}^T \Big| i) \equiv \begin{cases} 1, & \text{if } i \text{ is a legal final state} \\ 0, & \text{otherwise} \end{cases}$$

We still need to find $P(y|M)$:

$$P(y, \underline{x}(t) = i | M) = \alpha(y_1^t, i) \beta(y_{t+1}^T \Big| i)$$

for any state i. Therefore,

$$P(y|M) = \sum_{i=1}^{S} \alpha(y_1^t, i) \beta(y_{t+1}^T \Big| i)$$

But we also note that we should be able to compute this probability using only the forward direction. By considering $t = T$, we can write:

$$P(y|M) = \sum_{i=1}^{S} \alpha(y_1^T, i)$$

These equations suggest a recursion in which, for each value of t we iterate over ALL states and update $\alpha(y_1^t, j)$. When $t = T$, $P(y|M)$ is computed by summing over ALL states.

The complexity of this algorithm is $O(S^2 T)$, or for $S = 5$ and $T = 100$, approximately 2500 flops are required (compared to 10^{72} flops for the exhaustive search).

The Viterbi Algorithm

Instead of allowing any path to produce the output sequence, and hence, creating the need to sum over all paths, we can simply assume only one path produced the output. We would like to find the single most likely path that could have produced the output. Calculation of this path and probability is straightforward, using the dynamic programming algorithm previously discussed:

$$D(t, i) = a(i, j^*)b(k|i)D(t-1, j^*)$$

where

$$j^* = \arg \max_{valid\ j} \{D(t-1, j)\}$$

(in other words, the predecessor node with the best score). Often, probabilities are replaced with the logarithm of the probability, which converts multiplications to summations. In this case, the HMM looks remarkably similar to our familiar DP systems.

Beam Search

In the context of the best path method, it is easy to see that we can employ a beam search similar to what we used in DP systems:

$$D_{min}(t, i) \geq D_{min}(t, i^*_t) - \delta(t)$$

In other words, for a path to survive, its score must be within a range of the best current score. This can be viewed as a time-synchronous beam search. It has the advantage that, since all hypotheses are at the same point in time, their scores can be compared directly. This is due to the fact that each hypothesis accounts for the same amount of time (same number of frames).