Birth and Death Process

Let $X(t)$ represent the number of individuals alive at time t in a population. Two types of events can occur-one representing birth (i.e. arrival), which increase the population and the other representing death (i.e. departure), which decreases the population. Then the discrete random process $\{X(t)\}$ is called the birth and death process if the following postulates are satisfied.

1. $P(1 \text{ birth in } (t, t+\Delta t)) = \lambda_n \Delta t + O(\Delta t)$

2. $P(0 \text{ birth in } (t, t+\Delta t)) = 1 - \lambda_n \Delta t + O(\Delta t)$

3. $P(2 \text{ or more birth in } (t, t+\Delta t)) = O(\Delta t)$

4. Births occurring in $(t, t+\Delta t)$ are independent of time since last birth.

5. $P(1 \text{ death in } (t, t+\Delta t)) = \mu_n \Delta t + O(\Delta t)$

6. $P(0 \text{ death in } (t, t+\Delta t)) = 1 - \mu_n \Delta t + O(\Delta t)$

7. $P(2 \text{ or more death in } (t, t+\Delta t)) = O(\Delta t)$

8. Deaths occurring in $(t, t+\Delta t)$ are independent of time since last death.

9. Birth and death occur independently of each other at any time.

Probability distribution of $X(t)$

Let

$P_n(t) = P(X(t) = n)$

$P_n(t) = P(\text{size of the population is n at time t})$

Let

$P_n(t+\Delta t) = P(X(t+\Delta t) = n)$

$P_n(t) = P(\text{size of the population is n at time t} + \Delta t)$

Then the event $X(t+\Delta t) = n$ can happen in any one of the following mutually exclusive ways.

1. $X(t) = n$, no birth, no death in $(t, t+\Delta t)$

2. $X(t) = n-1$, 1 birth, no death in $(t, t+\Delta t)$

3. $X(t) = n+1$, no birth, 1 death in $(t, t+\Delta t)$

4. $X(t) = n$, 1 birth, 1 death in $(t, t+\Delta t)$

Then

$P_n(t, t+\Delta t) = P(1) + P(2) + P(3) + P(4)$

$$= P_n(t)(1-\lambda_n)\,\Delta t(1-\mu_n)\,\Delta t + P_{n-1}(t)(\lambda_{n-1}\,\Delta t)(1-\lambda_{n-1}\,\Delta t) + P_{n+1}(t)(1-\lambda_{n+1}\,\Delta t)\mu_{n+1}\,\Delta t + P_n(t)\lambda_n\,\Delta t\mu_n\,\Delta t$$

$$P_n(t)(t+\Delta t) = P_n(t) - P_n(t)\lambda_n\,\Delta t - P_n(t)\mu_n\,\Delta t + P_{n-1}(t)\lambda_{n-1}\,\Delta t + P_{n+1}(t)\mu_{n+1}\,\Delta t$$

(Omitting terms with higher powers of Δt)

$$P_n(t)(t+\Delta t) - P_n(t) = -P_n(t)\lambda_n \Delta t - P_n(t)\mu_n \Delta t + P_{n-1}(t)\lambda_{n-1} \Delta t + P_{n+1}(t)\mu_{n+1} \Delta t$$

$$\frac{P_n(t)(t+\Delta t) - P_n(t)}{\Delta t} = -P_n(t)\lambda_n - P_n(t)\mu_n + P_{n-1}(t)\lambda_{n-1} + P_{n+1}(t)\mu_{n+1}$$

Taking limit as $\Delta t \to 0$, we get

$$P'_n(t) = P_{n-1}(t)\lambda_{n-1} - (\lambda_n + \mu_n)P_n(t) + P_{n+1}(t)\mu_{n+1} \qquad (1)$$

This differential difference equation holds for $n \geq 1$

When $n = 0$

$$P_0(t)(t+\Delta t) = P_0(t)(1-\lambda_0 \Delta t) + P_1(t)(1-\lambda_1 \Delta t)\mu_1 \Delta t$$

$$P_0(t)(t+\Delta t) = P_0(t) - P_0(t)\lambda_0 \Delta t + P_1(t)\mu_1 \Delta t$$

$$P_0(t)(t+\Delta t) - P_0(t) = -P_0(t)\lambda_0 \Delta t + P_1(t)\mu_1 \Delta t$$

$$\frac{P_0(t)(t+\Delta t) - P_0(t)}{\Delta t} = -P_0(t)\lambda_0 + P_1(t)\mu_1$$

As $\Delta t \to 0$

$$P'_n(t) = -P_0(t)\lambda_0 + P_1(t)\mu_1 \qquad (2)$$

Equation (1) and (2) re the equations of the birth and death process.

Solving these , we get $P_n(t) = P(X(t) = n)$, which is the probability distribution of $X(t)$.

Estimation of distribution algorithms (EDAs), sometimes called *probabilistic model-building genetic algorithms* (PMBGAs), are Stochastic Optimization methods that guide the search for the optimum by building and sampling explicit probabilistic models of promising candidate solutions. Optimization is viewed as a series of incremental updates of a probabilistic model, starting with the model encoding the uniform distribution over admissible solutions and ending with the model that generates only the global optima.

EDAs belong to the class of evolutionary algorithims. The main difference between EDAs and most conventional evolutionary algorithms is that evolutionary algorithms generate new candidate solutions using an *implicit* distribution defined by one or more variation operators, whereas EDAs use an *explicit* probability distribution encoded by a Bayesian network , a multivariate normal distribution , or another model class. Similarly as other evolutionary algorithms, EDAs can be used to solve optimization problems defined over a number of representations from vectors to LISP style S expressions, and the quality of candidate solutions is often evaluated using one or more objective functions.

The general procedure of an EDA is outlined in the following:

1. t = 0
2. initialize model M(0) to represent uniform distribution over admissible solutions
3. while (termination criteria not met)

    1. P = generate N>0 candidate solutions by sampling M(t)
    2. F = evaluate all candidate solutions in P
    3. M(t+1) = adjust_model(P,F,M(t))
    4. t = t + 1

Using explicit probabilistic models in optimization allowed EDAs to feasibly solve optimization problems that were notoriously difficult for most conventional evolutionary algorithms and traditional optimization techniques, such as problems with high levels of epistasis. Nonetheless, the advantage of EDAs is also that these algorithms provide an optimization practitioner with a series of probabilistic models that reveal a lot of information about the problem being solved. This information can in turn be used to design problem-specific neighborhood operators for local search, to bias future runs of EDAs on a similar problem, or to create an efficient computational model of the problem.

Hidden Markov Models and the Viterbi algorithm:

An HMM H=(p_{ij},$e_i(a)$,w_i) is understood to have N hidden Markov states labeled by $i(1 \le i \le N)$, and M possible observables for each state, labeled by $a(1 \le a \le N)$. The state transition probabilistic are $p_{ij} = p(q_{t+1} = j|q_t = i)$, $1 \le i,j \le N$ (where q_t is then hidden state at time t), the emission probability for the observable a from state i is $e_i(a) = p(O_t = a|q_t = i)$ (where O_t is the observation at time t), and the initial state probabilities are $w_i = p(q_1 = i)$.

Given a sequence of observations $O = O_1, O_2 \dots O_T$ and an HMM H=(p_{ij},$e_i(a)$,w_i), we wish to find the maximum probability state path $Q = q_1, q_2 \dots q_T$. This can be done recursively using the Viterbi algorithm.

Let $v_i(t)$ be the probability of the most probable path ending in state i at time t

(i.e)

$$v_i(t) = \max_{q_1,q_2\dots q_{t-1}} P(q_1, q_2 \dots q_{t-1}, iO_1, O_2 \dots O_t|H),$$

and let w_i be the initial probabilities to the states i at time t=1. (Note this notation avoids the frightening greek letters δ, π and λ used used in the Rabiner notes, using instead v for Viterbi, w for weights and H for hidden Markov model. The correspondence with the notation used in the Rabiner notes is $v_i(t) \leftrightarrow \delta_i(t), e_i(a) \leftrightarrow b_i(a), p_{ij} \leftrightarrow a_{ij}, w_i \leftrightarrow \pi_i$ $H \leftrightarrow$

Then $v_j(t)$ can b calculated recursively using

$$v_j(t) = \max_{1 \le i \le N} \left[v_i(t-1)p_{ij}\right] e_j(O_t)$$

together with initiazation

$$v_i(1) = w_i e_j(O_1) \quad 1 \le i \le N\text{n}$$

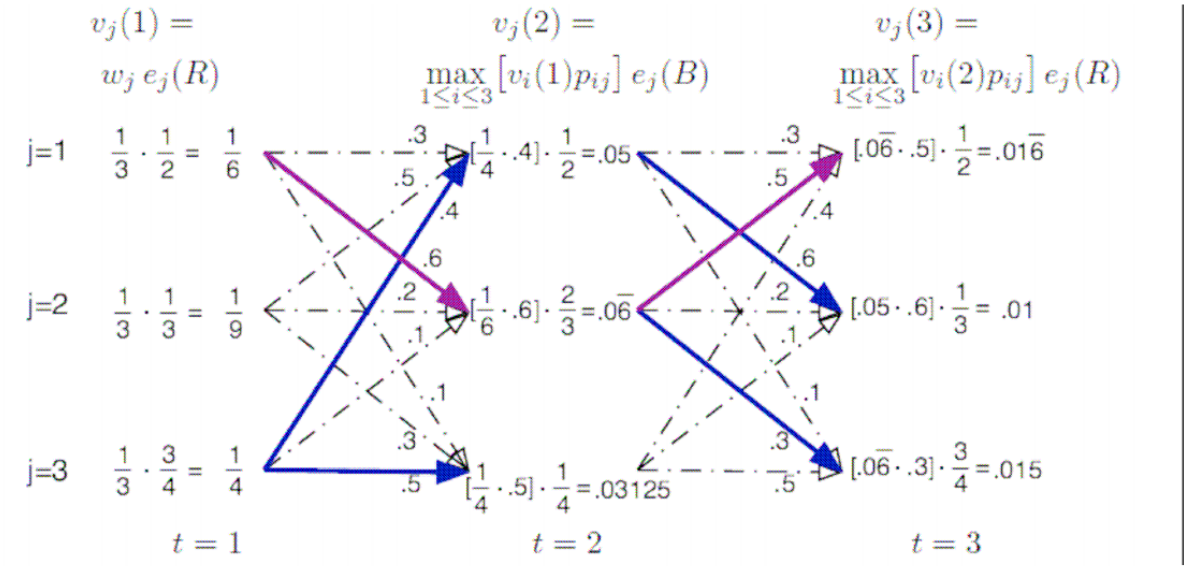
and terminati

$$P^* = \max_{1 \le i \le N} v_i(T)$$

(i.e) at the end we choose the highest probability endpoint and then we backtrack from there to find the highest probability path)

Note that the maximally likely path is not the only possible optimality criterion, for example choosing the most likely state at any given time requires a different alogorithm and can give a slightly different result. But the overall most likely path provided by the Viterbi algorithm provides an optimal state sequence for many purposes.

To illustrate this, consider a three state HMM, with R or B emitted by each state (e.g., three urns each with red or blues balls) with emission probabilities $e_1(R) = \frac{1}{2}, e_2(R) = \frac{1}{3}$ and $e_3(R) = 3/4$ (and correspondingly $e_1(B) = \frac{1}{2}, e_2(B) = \frac{2}{3}$ and $e_3(B) = 1/4$), state transition matrix $p_{ij} = \begin{pmatrix} .3 & .6 & .1 \\ .5 & .2 & .3 \\ .4 & .1 & .5 \end{pmatrix}$, and initial state probabilities $w_i = 1/3$. Suppose we observe the sequence RBR, the we can find the optimal state sequence to explain this sequence of observations by running the Viterbi algorithm by hand:



In the first step, we initialize the probabilities at $t = 1\ to\ v_j(t = 1) = w_j e_j(R)$ for each j=1,2,3. These are given in the first column to the left, as $\frac{1}{6}, \frac{1}{9}, \frac{1}{4}$ respectively.

In the second step t=2, we determine first $v_1(t = 2)$ by considering the three quantities $v_i(1)p_{i1}$ for i = 1,2,3. They are respectively (1/6).3,(1/9).5 and

(1/4).4. The third one is the largest, so according to the algorithm we set $v_1(2) = \left[\left(\frac{1}{4}\right).4\right].\left(\frac{1}{2}\right) = .05$ and remember that the maximum probability path to state j=1 at time t=2 came from state j=3 at time t=1(blue line). Similarly to determine $v_2(2)$ we consider the three quantities $v_i(1)p_{i2}$ for i=1,2,3 respectively (1/6).6,(1/9).2 and (1/4).1 and the first is the large, so we set $v_2(2) = \left[\left(\frac{1}{6}\right).6\right].\left(\frac{2}{3}\right) = .06$. Finally, to determine $v_3(2)$ we consider the three quantities $v_i(1)p_{i3}$ for i=1,2,3 respectively (1/6).1,(1/9).3 and (1/4).5 and the third is the largest, so we set $v_3(2) = \left[\left(\frac{1}{4}\right).5\right].\left(\frac{1}{4}\right) = .03125$.

In the third step t=3, we determine first $v_1(t=3)$ by considering the three quantities $v_i(2)p_{i1}$ for i = 1,2,3. They are respectively .05.3.0$\bar{6}$ .5and.03125.4. The second is the largest, so according to the algorithm we set $v_1(3) = [0\bar{6}.5].\left(\frac{1}{2}\right) = .01\bar{6}$ and remember that the maximum probability path to state j=1 at time t=3 came from state j=2 at time t=2(blue line). Similarly to determine $v_2(3)$ we consider the three quantities $v_i(2)p_{i2}$ for i=1,2,3 respectively .05.6.0$\bar{6}$.2.03125.1and the first is the largest, so we set $v_3(3) = [.05.6].\left(\frac{1}{3}\right) = .01$. Finally to determine $v_3(2)$ we consider the three quantities $v_i(1)p_{i2}$ for i=1,2,3, respectively .05.1.0$\bar{6}$.3.03125.5 and the second is the largest, so we set $v_3(3) = [.0\bar{6}.3].\left(\frac{3}{4}\right) = .015$.

Since there are only three observations, we can now use the termination step to determine that the maximum probability for the observations O=RBR is $P^*.01\bar{6}$ with state path Q=1,2,1.

HMMs and the forward-backward algorithm

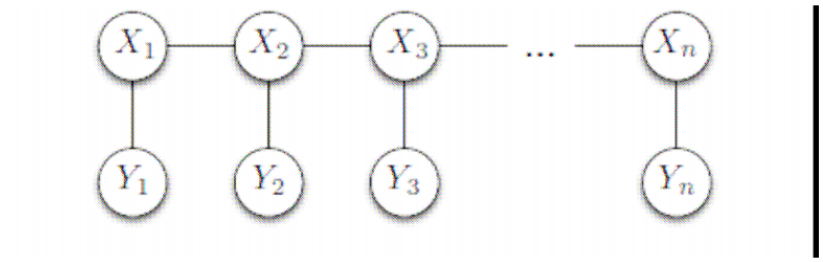
These notes give a short review of Hidden Markov Models (HMMs) and the forward-backward algorithm. They are written assuming familiarity with the sum product belief propagation algorithm, but should be accessible to any one who's seen the fundamentals of HMMs before

The notation here is borrowed from Introduction to Probability by Bertekas and Tsitsiklis random variable are represented with capital letters, values they take

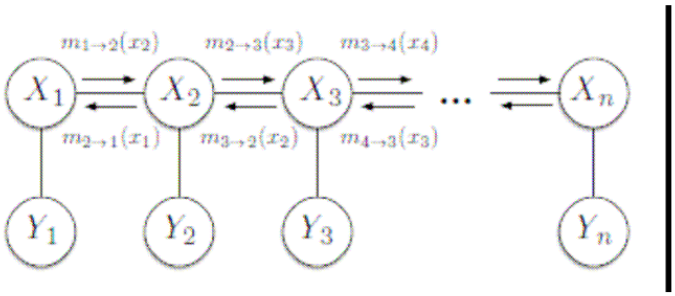
are presented with lowercase letters, p_X presents a probability distribution for random variable X and $p_X(x)$ represents the probability of value x (according to p_X).

Hidden Markov Models

Figure shows the (undirected) graphical model for HMMs. Here's a quick recap of the important facts.



- We observe Y_1 through Y_n which we model as being observed from hidden states X_1 through X_n
- Any particular state variable X_k depends only on X_{k-1} (what came before it). X_{k+1} (what comes after it) and Y_k (the observation associated with it)
- The goal of the forward-backward algorithm is to find the conditional distribution over hidden states given the data.
- In order to specify an HMM we need three pieces.



A transition distribution, $pX_{k+1}|X_k(x_{k+1}|x_k) = W(x_{k+1}|x_k)$ which describes the distribution for the next state given the current state. This is often represented as a matrix that we will call A. Rows of A correspond to the current state, columns correspond to the next state, and each entry corresponds to the

transition probability. So, the entry at row i and column j. A_{ij} is $pX_{k+1}|X_k(\frac{j}{i})$ or equivalently W(j/i).

An observation distribution (also called an ''emission distribution'') $pY_k|X_k(y_k|x_k) = pY/X(y_k|x_k)^2$ which describes the distribution for the output given the current state. We will represent this with matrix B. Here rows correspond to the current state and column correspond to the observation. So $B_{ij} = pY_k|X_k(j|i)$ the probability of observing output j from state i is B_{ij}. Since the number of possible observation isn't necessarily the same as the number of possible states, B won't necessarily be square.

An initial state distribution pX_1 which describes the starting distribution over states. We will represent this with a vector called π_0 where item i in the vector represents $pX_1(i)$.

- The forward backward algorithm computes forward and backward messages as follows:

$$m_{(k-1)\to k}(x_k) = \sum_{x_{k-1}} \overbrace{m_{(k-2)\to(k-1)}(x_{k-1})}^{\text{prev. message}} \overbrace{p_{Y|X}(y_{k-1}|x_{k-1})}^{\text{observation term}} \overbrace{W(x_{k-1}|x_k)}^{\text{transition term}}$$

$$m_{(k+1)\to k}(x_k) = \sum_{x_{k+1}} \underbrace{m_{(k+2)\to(k+1)}(x_{k+1})}_{\text{prev. message}} \underbrace{p_{Y|X}(y_{k+1}|x_{k+1})}_{\text{observation term}} \underbrace{W(x_k|x_{k+1})}_{\text{transition term}}$$

These messages are illustrated in figure 2. The first forward message $m_{0\to 1}(x_1)$ is initialized to $\pi_0(x_1) = pX_1(x_1)$. The first backward message $m_{(n+1)\to n}(x_n)$ is initialized to uniform (this is equivalent to not including it at all)

Figure 3 illustrates the computation of one forward message $m_{2\to 3}(x_3)$.

- To obtain a marginal distribution for a particular state given all the observations, $pX_k/Y_1.....Y_n$ we simply multiply the incoming messages together with the observation term, and then normalize:

$$p_{X_k|Y_1,\ldots,Y_n}(x_k|y_1,\ldots,y_n) \propto m_{(k-1)\to k}(x_k)m_{(k+1)\to k}(x_k)p_{Y|X}(y_k|x_k)$$

Here the symbol ∞ means " is proportional to" and indicates that we have to normalize at the end so that the answer sums to 1.

- Traditionally, the forward backward algorithm computes a slightly different set of messages. The forward message $\boldsymbol{\alpha_k}$ represents a message from k-1 to k that includes $\boldsymbol{pY|X(y_k|x_k)}$ and the backward message $\boldsymbol{\beta_k}$ represents message from k+1 to k identical $\boldsymbol{m_{(K+1)\to k}}$ above

$$\alpha_k(x_k) = \overbrace{p_{Y|X}(y_k|x_k)}^{\text{observation term}} \sum_{x_{k-1}} \overbrace{\alpha_{k-1}(x_{k-1})}^{\text{prev. message}} \overbrace{W(x_{k-1}|x_k)}^{\text{transition term}}$$

$$\beta_k(x_k) = \sum_{x_{k+1}} \underbrace{\beta_{k+1}(x_{k+1})}_{\text{prev. message}} \underbrace{p_{Y|X}(y_{k+1}|x_{k+1})}_{\text{observation term}} \underbrace{W(x_k|x_{k+1})}_{\text{transition term}}$$

These message have a particularly nice interpretation as probabilities.

$$\alpha_k(x_k) = p_{Y_1,Y_2,\ldots,Y_k,X_k}(y_1, y_2, \ldots, y_k, x_k)$$
$$\beta_k(x_k) = p_{Y_{k+1},Y_{k+2},\ldots,Y_n|X_k}(y_{k+1}, y_{k+2}, \ldots, y_n|x_k)$$

The initial forward $\boldsymbol{\alpha}$ message is initialized $\alpha_1(x_1) = p_{X_1}(x_1)p_{Y|X}(y_1|x_1)$.

To obtain a marginal distribution we simply multiply the message together and normalize:

$$p_{X_k|Y_1,\ldots,Y_n}(x_k|y_1, \ldots, y_n) \propto \alpha_k(x_k)\beta_k(x_k)$$

Example

Suppose you send a robot to Mars. Unfortunately, it gets stuck in a canyon while landing and most of its sensors break. You know the canyon has 3 areas. Areas 1 and 3 are sunny and hot, while Area 2 is cold. You decide to plan a rescue mission for the robot from Area 3 knowing the following things about the robot:

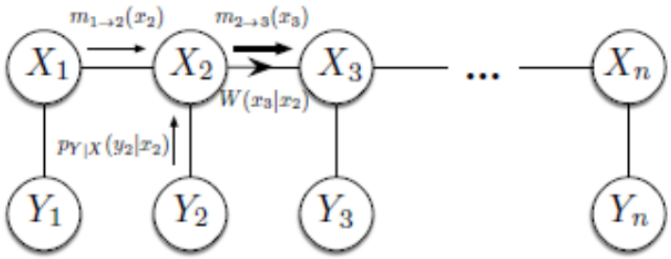


Figure 3

Figure 3: An illustration of how to compute $m_{2\to3}(x_3)$. In order for node 2 to summarize its belief about X_3, it must incorporate the pervious message $m_{1\to2}(x_2)$, its observation $pY|X(y_2|x_2)$, and the relationship $W(y_2|x_2)$ between X_2 and X_3.

- Every hour, it tries to move forward by one area (i.e. from Area 1 to Area 2, or Area 2 to Area 3). It succeeds with probability 0.75 and fails with probability 0.25. If it fails, it stays where it is. If it is in Area 3, it always stays where it is. If it is an Area 3, it always stays there (and waits to be rescued).
- The temperature sensor still works. Every hour, we get a binary reading telling us whether the robot's current environment is hot or cold.
- We have no idea where the robot initially got stuck.

Solution

a) Construct an HMM for this problem: define a transition matrix A, an observation matrix B, and an initial state distribution π_0.

b) Suppose we observe the sequence (hot, cold, hot). First before doing any computation, determine the sequence of locations. Then, compute the forward and backward messages, and determine the distribution for the second state using the message. Do your answers match up?

c) We will start with the transition matrix. Remember that each row corresponds to the current state, and each column corresponds to the next state. We will use 3 states, each corresponding to an area.

- If the robot is in Area 1, it stays, where it is with probability 0.25, moves to Area 2 with probability 0.75 and can't move to Area 3.
- Similarly if the robot is in Area 2, it stays where it is with probability 0.25, can't move back to Area 1 and moves to Area 3 with probability 0.75.
- If the robot is in Area 3, it always stays in Area 3.

Each item above gives us one row of A. Putting it all together, we obtain

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0.25 & 0.75 & 0 \\ 0 & 0.25 & 0.75 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

Next let's look at the observation matrix. There are two possible observations, hot and cold. Area 1 and 3 always produce "hot" reading s while Area 2 always produces a "cold" reading:

$$B = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cc} \text{hot} & \text{cold} \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}$$

Last but not least, since we have no idea where the robot starts, out initial state distribution will be uniform:

$$\pi_0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \\ \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} \end{array}$$

Before doing any computation, we see that the sequence (hot, cold,hot) could only have been observed from the hidden state sequence (1,2,3). Make sure you convince yourself this is true before continuting.

We will start with the forward message.

$$m_{1\to 2} = \sum_{x_1} \underbrace{m_{0\to 1}(x_1) p_{Y|X}(y_1|x_1)}_{\text{depends only on } x_1 \text{ and } y_1} \psi(x_1, x_2)$$

The output message should have three different possibilities, one for each value of x_2. We can therefore represent it as a vector indexed by x_2:

$$\begin{pmatrix} \cdot \\ \cdot \\ \cdot \end{pmatrix} \begin{matrix} \text{value for } x_2 = 1 \\ \text{value for } x_2 = 2 \\ \text{value for } x_2 = 3 \end{matrix}$$

For each term in the sum(i.e. each possible value of x_1):

- $m_{0\to 1}$ comes from the initial distribution . Normally it would come from the previous message, but our first forward message is always set to initial state distribution.
- $pY|X(y_1|x_1)$ comes from the column of B corresponding to our observation y_1=hot.
- ψ comes from row of A: we are fixing x_1 and asking about possible values for x_2, which corresponding exactly to the transition distributions given in the rows of A(remember that the rows of A correspond to the current state and the columns correspond to the next state)

So, we obtain

$$m_{1\to 2} = \overbrace{\frac{1}{3} \cdot 1 \cdot \begin{pmatrix} .25 \\ .75 \\ 0 \end{pmatrix}}^{x_1=1} + \overbrace{\frac{1}{3} \cdot 0 \cdot \begin{pmatrix} 0 \\ .25 \\ .75 \end{pmatrix}}^{x_1=2} + \overbrace{\frac{1}{3} \cdot 1 \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}}^{x_1=3}$$
$$\propto \begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix}$$

Since our probabilities are eventually computed by multiplying message and normalizing, we can arbitrary renormalize at any step to make the computation easier.

For the second message, we perform a similar computation:

$$
\begin{aligned}
m_{2\to 3} &= \sum_{x_2} m_{1\to 2}(x_2)\tilde{\phi}(x_2)\psi(x_2, x_3) \\
&= \overbrace{1 \cdot 0 \cdot \begin{pmatrix} .25 \\ .75 \\ 0 \end{pmatrix}}^{x_2=1} + \overbrace{3 \cdot 1 \cdot \begin{pmatrix} 0 \\ .25 \\ .75 \end{pmatrix}}^{x_2=2} + \overbrace{4 \cdot 0 \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}}^{x_2=3} \\
&\propto \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix}
\end{aligned}
$$

The backward message are computed using a similar formula:

$$
m_{3\to 2} = \sum_{x_3} \underbrace{m_{4\to 3}(x_3)\tilde{\phi}(x_3)}_{\text{depends only on } x_3} \psi(x_2, x_3)
$$

The first backwards message, $m_{4\to 3}(x_2)$ is always initialized to uniform since we have no information about what the last state should be. Note that this is equivalent to not including that term at all.

For each value of x_3 the transition term $\psi(x_2, x_3)$ is now drawn from a column of A, since we are interested in the probability of arriving at x_3 from each possible state for x_2. We compute the messages as:

$$
\begin{aligned}
m_{3\to 2} &= \overbrace{1 \cdot \begin{pmatrix} .25 \\ 0 \\ 0 \end{pmatrix}}^{x_3=1} + \overbrace{0 \cdot \begin{pmatrix} .75 \\ .25 \\ 0 \end{pmatrix}}^{x_3=2} + \overbrace{1 \cdot \begin{pmatrix} 0 \\ .75 \\ 1 \end{pmatrix}}^{x_3=3} \\
&\propto \begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix}
\end{aligned}
$$

Similarly, the second backwards message is:

$$m_{2\to 1} = \overbrace{1\cdot 0\cdot \begin{pmatrix}.25\\0\\0\end{pmatrix}}^{x_2=1} + \overbrace{3\cdot 1\cdot \begin{pmatrix}.75\\.25\\0\end{pmatrix}}^{x_2=2} + \overbrace{4\cdot 0\cdot \begin{pmatrix}0\\.75\\1\end{pmatrix}}^{x_2=3}$$
$$\propto \begin{pmatrix}3\\1\\0\end{pmatrix}$$

Notice from the symmetry of the problem that our forwards messages and backwards messages were the same.

To compute the marginal distribution for $\boldsymbol{X_2}$ given the data, we multiply the messages and the observation:

$$p_{X_2|Y_1,\ldots,Y_n}(x_2|y_1,\ldots,y_n) \propto m_{1\to 2}(x_2)m_{3\to 2}(x_2)\tilde{\phi}(x_2)$$
$$\propto \begin{pmatrix}1\\3\\4\end{pmatrix}\cdot\begin{pmatrix}1\\3\\4\end{pmatrix}\cdot\begin{pmatrix}0\\1\\0\end{pmatrix}$$
$$= \begin{pmatrix}0\\1\\0\end{pmatrix}$$

Notice that in this case , because of our simplified observation model, the observation “cold” allowed us to determine the state. This matches up with our earlier conclusion that the robot must have been in Area 2 during the second hour.

If we were to compute $\boldsymbol{\alpha}$ messages, we would start with our initial messages $\boldsymbol{\alpha_1}$:

$$\alpha_1(x_1) = p_{X_1}(x_1)p_{Y|X}(y_1|x_1) = \begin{pmatrix}1/3\\0\\1/3\end{pmatrix}$$

The first real message is computed as follows:

$$\alpha_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \cdot \left(\overbrace{\cdot 1/3 \cdot \begin{pmatrix} .25 \\ .75 \\ 0 \end{pmatrix}}^{x_1=1} + \overbrace{\cdot 0 \cdot \begin{pmatrix} 0 \\ .25 \\ .75 \end{pmatrix}}^{x_1=2} + \overbrace{\cdot 1/3 \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}}^{x_1=3} \right)$$

$$\propto \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

The second message is similar:

$$\alpha_3 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \left(\overbrace{\cdot 0 \cdot \begin{pmatrix} .25 \\ .75 \\ 0 \end{pmatrix}}^{x_1=1} + \overbrace{\cdot 1 \cdot \begin{pmatrix} 0 \\ .25 \\ .75 \end{pmatrix}}^{x_1=2} + \overbrace{\cdot 0 \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}}^{x_1=3} \right)$$

$$\propto \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

The $\boldsymbol{\beta}$ messages would be identical to our backwards message computed earlier.

Reference

The Classification of Birth and Death Processes by Samuel Karlin and James McGregor

Probability and Random Processes 3rd Edition by Geoffrey R. Grimmett, David R. Stirzaker

Probability, Statistics, and Random Processes for Engineers (4th Edition) by Henry Stark, John Woods

Introduction to Probability, Statistics, and Random Processes – August 24, 2014 by Hossein Pishro-Nik

Statistics of Random Processes II 2nd rev. and exp. ed. 2001 Edition by Robert S. Liptser, Albert N. Shiryaev