Hidden Markov Models (HMM)

Jul 02

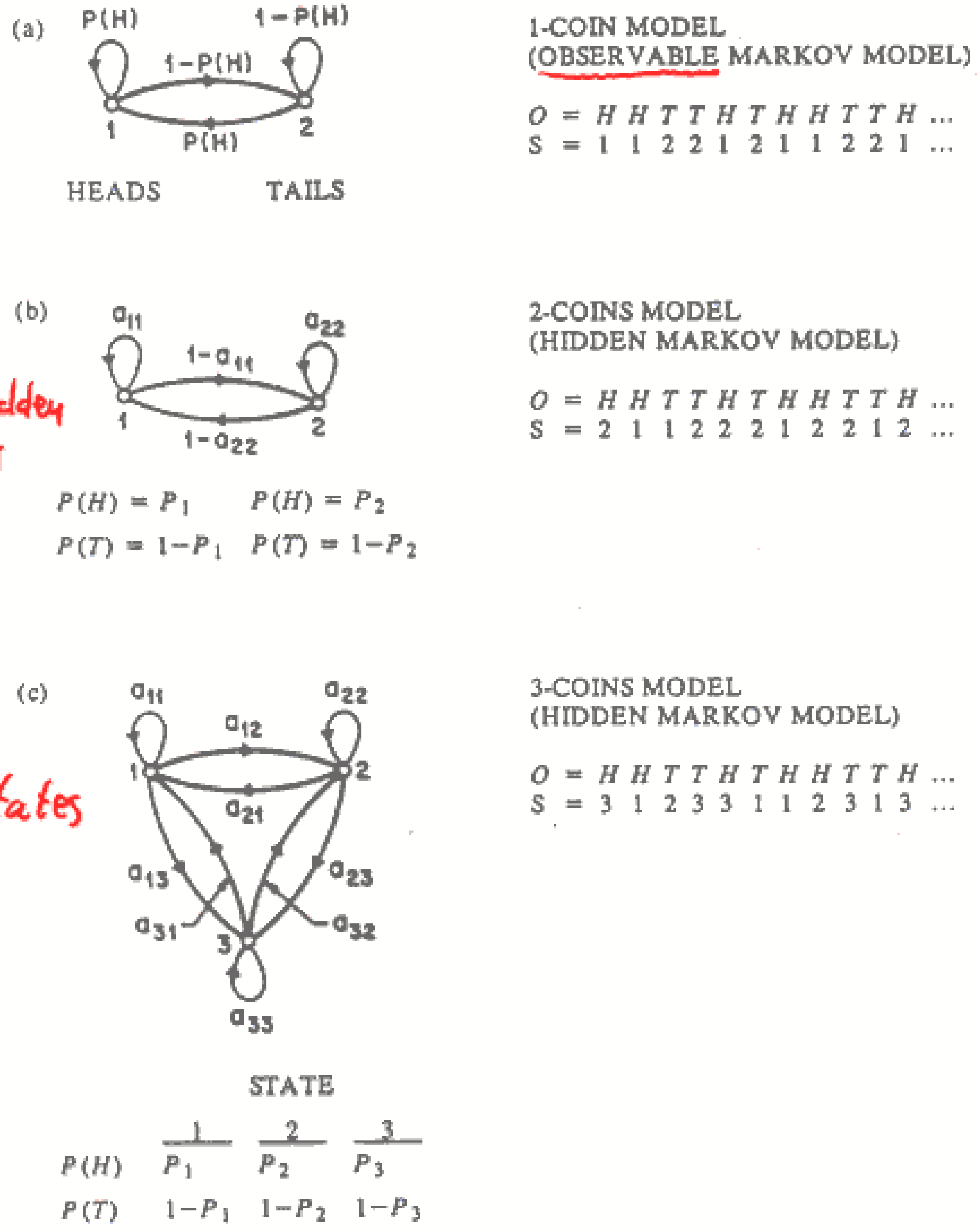
example: coin-tossing (two possible observations)

(not hidden)

extension to hidden MM

two hidden states

three hidden states



(a)
P(H)
1 − P(H)
1−P(H)
P(H)
1
2
HEADS
TAILS
1-COIN MODEL
(OBSERVABLE MARKOV MODEL)
O = H H T T H T H H T T H ...
S = 1 1 2 2 1 2 1 1 2 2 1 ...
(b)
a11
a22
1−a11
1−a22
1
2
2-COINS MODEL
(HIDDEN MARKOV MODEL)
O = H H T T H T H H T T H ...
S = 2 1 1 2 2 2 1 2 2 1 2 ...
P(H) = P1   P(H) = P2
P(T) = 1−P1   P(T) = 1−P2
(c)
a11
a22
a12
a21
a13
a23
a31
a32
a33
1
2
3
3-COINS MODEL
(HIDDEN MARKOV MODEL)
O = H H T T H T H H T T H ...
S = 3 1 2 3 3 1 1 2 3 1 3 ...
STATE
1   2   3
P(H)   P1   P2   P3
P(T)   1−P1   1−P2   1−P3

Figure 6.3 Three possible Markov models that can account for the results of hidden coin-tossing experiments. (a) one-coin model, (b) two-coins model, (c) three-coins model.

example: urn-and-ball model (M possible observations)

N hidden states

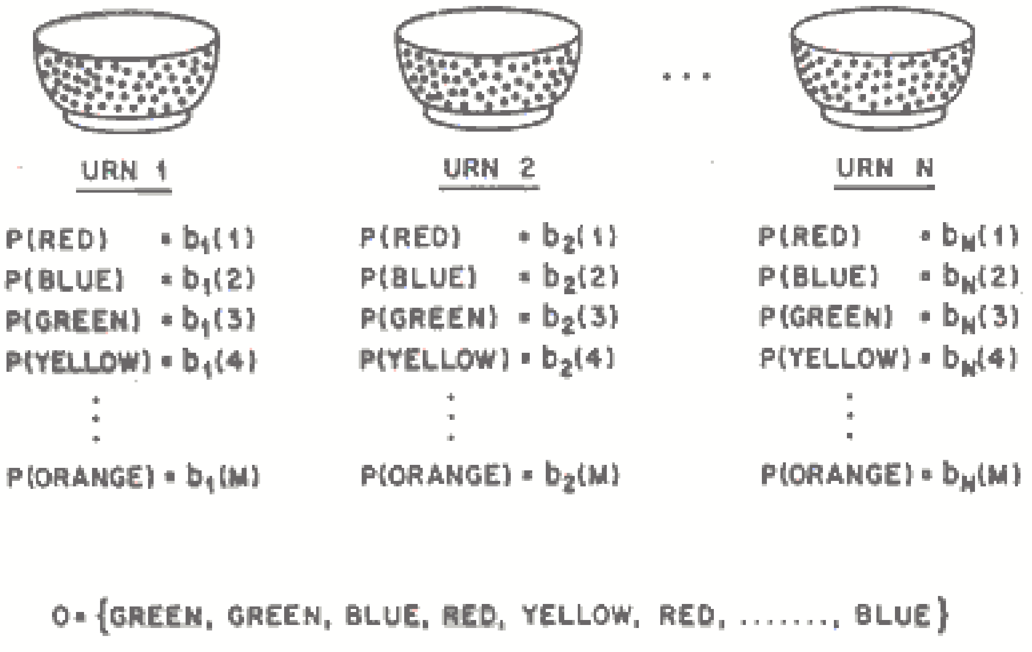


Figure 6.4 An N-state urn-and-ball model illustrating the general case of a discrete symbol HMM.

Elements of an HMM

1. N, the number of states in the model (states are hidden) of interest and may better suit speech applications. We label the individual states as $\{1, 2, \ldots, N\}$, and denote the state at time t as q_t.
2. M, the number of distinct observation symbols per state—i.e., the discrete alphabet size. The observation symbols correspond to the physical output of the system being modeled. For the coin-toss experiments the observation symbols were simply heads or tails; for the ball-and-urn model they were the colors of the balls selected from the urns. We denote the individual symbols as $V = \{\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_M\}$.
3. The state-transition probability distribution $A = \{a_{ij}\}$ where
$$a_{ij} = P[q_{t+1} = j | q_t = i], \qquad 1 \leq i, j \leq N. \tag{6.7}$$
For the special case in which any state can reach any other state in a single step, we have $a_{ij} > 0$ for all i, j. For other types of HMMs, we would have $a_{ij} = 0$ for one or more (i, j) pairs.
4. The observation symbol probability distribution, $B = \{b_j(k)\}$, in which
$$b_j(k) = P[\mathrm{o}_t = \mathrm{v}_k | q_t = j], \qquad 1 \leq k \leq M, \tag{6.8}$$
defines the symbol distribution in state j, $j = 1, 2, \ldots, N$.
5. The initial state distribution $\pi = \{\pi_i\}$ in which
$$\pi_i = P[q_1 = i], \qquad 1 \leq i \leq N. \tag{6.9}$$

⇒ compact notation : $\lambda = (A, B, \pi)$

→ complete specification of an HMM

HMM Generator of Observations

Given appropriate values of N, M, A, B, and π, the HMM can be used as a generator to give an observation sequence
$$\mathrm{O} = (\mathrm{o}_1 \mathrm{o}_2 \ldots \mathrm{o}_T) \tag{6.11}$$
(in which each observation o_t is one of the symbols from V, and T is the number of observations in the sequence) as follows:

1. Choose an initial state $q_1 = i$ according to the initial state distribution π.
2. Set $t = 1$.
3. Choose $\mathrm{o}_t = \mathrm{v}_k$ according to the symbol probability distribution in state i, i.e., $b_i(k)$.
4. Transit to a new state $q_{t+1} = j$ according to the state-transition probability distribution for state i, i.e., a_{ij}.
5. Set $t = t + 1$; return to step 3 if $t < T$; otherwise, terminate the procedure.

The following table shows the sequence of states and observations generated by the above procedure:

time, t	1	2	3	4	5	6	...	T
state	q_1	q_2	q_3	q_4	q_5	q_6	...	q_T
observation	o_1	o_2	o_3	o_4	o_5	o_6	...	o_T

The above procedure can be used as both a generator of observations and as a model to simulate how a given observation sequence was generated by an appropriate HMM.

The three basic problems for HMMs

Problem 1

Given the observation sequence $O = (o_1 o \ldots o_T)$, and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model?

→ evaluation problem
How well a model matches an observation?

Problem 2

Given the observation sequence $O = (o_1 o \ldots o_T)$, and the model λ, how do we choose a corresponding state sequence $q = (q_1 q_2 \ldots q_T)$ that is optimal in some sense (i.e., best "explains" the observations)?

→ uncover the hidden part

Problem 3

How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$?

→ training problem

- Problems 1 and 2 → analysis problems
  Problem 3 → synthesis problem

example: single word recognition (one HMM per word):

- build individual word models → Prob. 3
- understanding model states → e.g. change no. of states → Prob. 2
- recognition of unknown word → Prob. 1

Solution to Problem 1—Probability Evaluation

We wish to calculate the probability of the observation sequence, $O = (o_1 o \ldots o_T)$, given the model λ, i.e., $P(O|\lambda)$. The most straightforward way of doing this is through enumerating every possible state sequence of length T (the number of observations). There are N^T such state sequences. Consider one such fixed-state sequence

$$q = (q_1 q_2 \cdots q_T) \tag{6.12}$$

where q_1 is the initial state. The probability of the observation sequence O given the state sequence of Eq. (6.12) is

$$P(O|q, \lambda) = \prod_{t=1}^{T} P(o_t|q_t, \lambda) \tag{6.13a}$$

where we have assumed statistical indeper dence of observations. Thus we get

$$P(O|q, \lambda) = \ell_{q_1}(o_1) \cdot b_{q_2}(o_2) \ldots b_{q_T}(o_T). \tag{6.13b}$$

The probability of such a state sequence q can be written as

$$P(q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}. \tag{6.14}$$

The joint probability of O and q, i.e., the probability that O and q occur simultaneously, is simply the product of the above two terms, i.e.,

$$P(O, q|\lambda) = P(O|q, \lambda) P(q|\lambda). \tag{6.15}$$

The probability of O (given the model) is obtained by summing this joint probability over all possible state sequences q, giving

$$P(O|\lambda) = \sum_{\text{all } q} P(O|q, \lambda) P(q|\lambda) \tag{6.16}$$

$$= \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \ldots a_{q_{T-1} q_T} b_{q_T}(o_T). \tag{6.17}$$

⇒ about $2 \cdot T \cdot N^T$ calculations needed → infeasible
(e.g. $N = 5$ $T = 100$ ⇒ $\approx 10^{72}$ computations)

→ a more efficient algorithm is required to solve problem 1

→ The Forward Procedure

Consider the forward variable $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(o_1 o_2 \ldots o_t, q_t = i|\lambda) \tag{6.18}$$

that is, the probability of the partial observation sequence, $o_1\, o_2 \ldots o_t$, (until time t) and state i at time t, given the model λ. We can solve for $\alpha_t(i)$ inductively, as follows:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(o_1), \qquad 1 \leq i \leq N. \tag{6.19}$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}), \qquad \begin{array}{l} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{array} \tag{6.20}$$

3. Termination

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{6.21}$$

⇒ only about N^2T calculations needed

(e.g., $N=5$, $T=100$ ⇒ ≈3000, 69 orders of magnitude less then direct calculation)

induction step :

lattice (or trellis) structure :

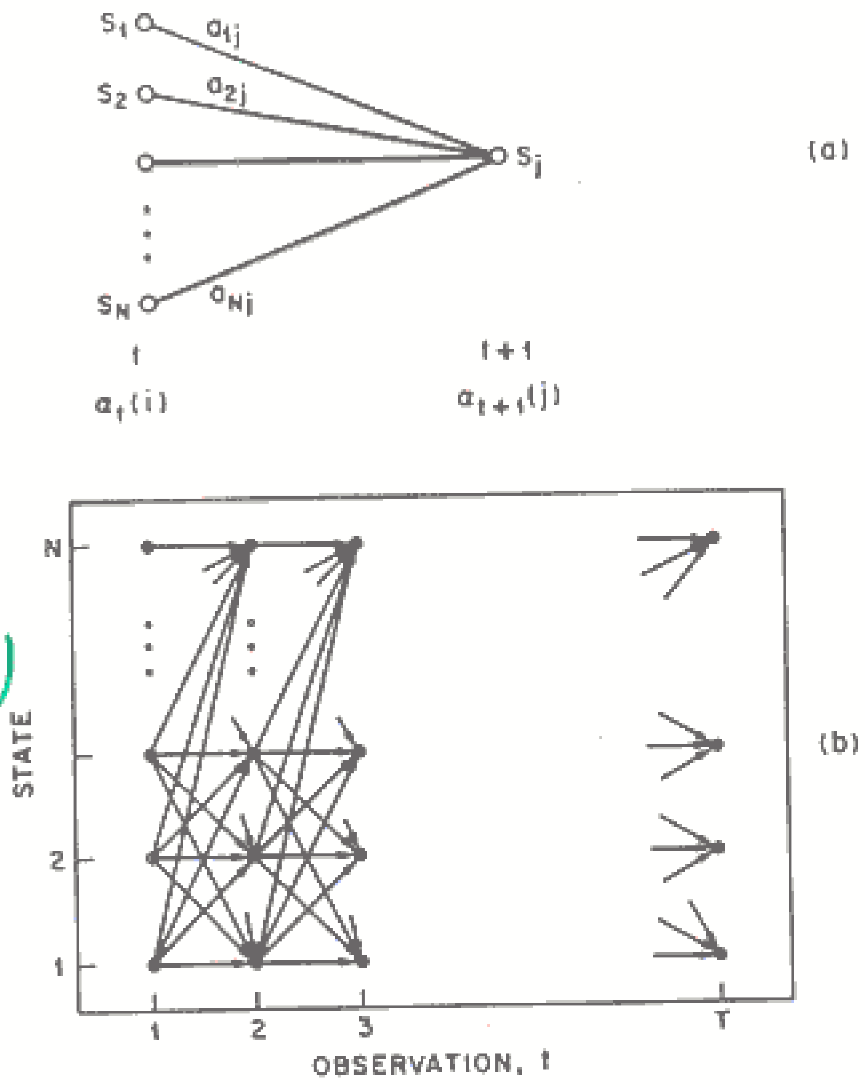


Figure 6.5 (a) Illustration of the sequence of operations required for the computation of the forward variable $\alpha_{t+1}(j)$. (b) Implementation of the computation of $\alpha_t(i)$ in terms of a lattice of observations t, and states i.

The Backward Procedure

In a similar manner, we can consider a backward variable $\beta_t(i)$ defined as

$$\beta_t(i) = P(\mathbf{o}_{t+1}\mathbf{o}_{t+2}\dots\mathbf{o}_T|q_t = i, \lambda) \tag{6.23}$$

that is, the probability of the partial observation sequence from $t+1$ to the end, given state i at time t and the model λ. Again we can solve for $\beta_t(i)$ inductively, as follows:

1. Initialization

$$\beta_T(i) = 1, \qquad 1 \leq i \leq N. \tag{6.24}$$

2. Induction

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j),$$

$$t = T-1, T-2, \dots, 1, \qquad 1 \leq i \leq N. \tag{6.25}$$

3. Termination $P(O|\lambda) = \sum_{i=1}^{N} \pi_i\, b_i(o_1)\, \beta_1(i)$

→ just an other method to solve problem 1

backward and forward are needed for solving problem 2 and 3

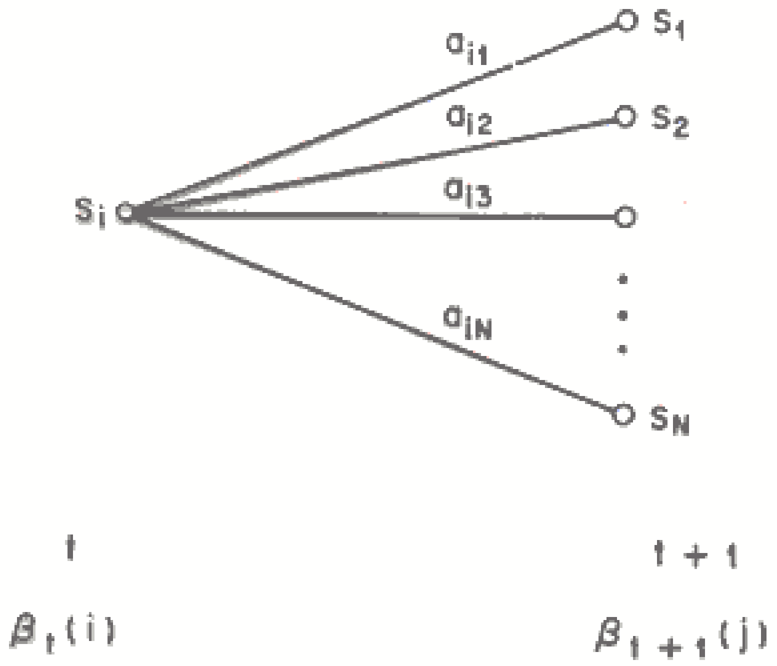


Figure 6.6 Sequence of operations required for the computation of the backward variable $\beta_t(i)$.

Solution to problem 2 ~ "optimal" state sequence

what is "optimal"? → there are several possible criteria

(1) choose the states q_t that are individually most likely at each time t

we can define the a posteriori probability variable

$$\gamma_t(i) = P(q_t = i | O, \lambda) \tag{6.26}$$

that is, the probability of being in state i at time t, given the observation sequence O, and the model λ. We can express $\gamma_t(i)$ in several forms, including

$$\begin{aligned}\gamma_t(i) &= P(q_t = i \mid O, \lambda) \\ &= \frac{P(O, q_t = i \mid \lambda)}{P(O \mid \lambda)} \\ &= \frac{P(O, q_t = i \mid \lambda)}{\sum_{i=1}^{N} P(O, q_t = i \mid \lambda)}.\end{aligned} \tag{6.27}$$

Since $P(O, q_t = i \mid \lambda)$ is equal to $\alpha_t(i)\beta_t(i)$, we can write $\gamma_t(i)$ as

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{6.28}$$

where we see that $\alpha_t(i)$ accounts for the partial observation sequence $o_1 o_2 \ldots o_t$ and state i at t, while $\beta_t(i)$ accounts for the remainder of the observation sequence $o_{t+1} o_{t+2} \ldots o_T$, given state $q_t = i$ at t.

Using $\gamma_t(i)$, we can solve for the individually most likely state q_t^* at time t, as

$$q_t^* = \arg \min_{1 \le i \le N} [\gamma_t(i)], \qquad 1 \le t \le T. \tag{6.29}$$

→ problem with this criterion

given $a_{ij} = 0$ for some i and j

→ we may get an invalid state sequence

better optimality criterion :

2. to find the single best state sequence
(most widely used criterion)

→ The Viterbi Algorithm

To find the single best state sequence, $\mathbf{q} = (q_1 q_2 \ldots q_T)$, for the given observation sequence $\mathbf{O} = (o_1 o_2 \ldots o_T)$, we need to define the quantity

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \ldots q_{t-1},\ q_t = i,\ o_1 o_2 \ldots o_t|\lambda] \qquad (6.30)$$

that is, $\delta_t(i)$ is the best score (highest probability) along a single path, at time t, which accounts for the first t observations and ends in state i. By induction we have

$$\delta_{t+1}(j) = [\max_i \delta_t(i)\, a_{ij}] \cdot b_j(o_{t+1}). \qquad (6.31)$$

To actually retrieve the state sequence, we need to keep track of the argument that maximized Eq. (6.31), for each t and j. We do this via the array $\psi_t(j)$. The complete procedure for finding the best state sequence can now be stated as follows:

1. Initialization

$$\delta_1(i) = \pi_i b_i(o_1), \qquad 1 \le i \le N \qquad (6.32a)$$

$$\psi_1(i) = 0. \qquad (6.32b)$$

2. Recursion

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i)\, a_{ij}] b_j(o_t), \qquad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N \end{matrix} \qquad (6.33a)$$

$$\psi_t(j) = \arg \max_{1 \le i \le N} [\delta_{t-1}(i)\, a_{ij}], \qquad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N. \end{matrix} \qquad (6.33b)$$

3. Termination

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \qquad (6.34a)$$

$$q_T^* = \arg \max_{1 \le i \le N} [\delta_T(i)]. \qquad (6.34b)$$

4. Path (state sequence) backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \ldots, 1. \qquad (6.35)$$

- algorithm maximizes $P(O, q \mid \lambda)$ for given O and λ
- a lattice (or trellis) structure efficiently implements the computation
- about $N^2 T$ calculations are needed

Exercise 2

[redacted]

Given the model of the coin-toss experiment used in Exercise 6.2 (i.e., three different coins) with probabilities

	State 1	State 2	State 3
$P(H)$	0.5	0.75	0.25
$P(T)$	0.5	0.25	0.75

and with all state transition probabilities equal to 1/3, and with initial probabilities equal to 1/3, for the observation sequence

$$O = (H\,H\,H\,H\,T\,H\,T\,T\,T\,T)$$

find the most likely path with the Viterbi algorithm.

Solution 6.3

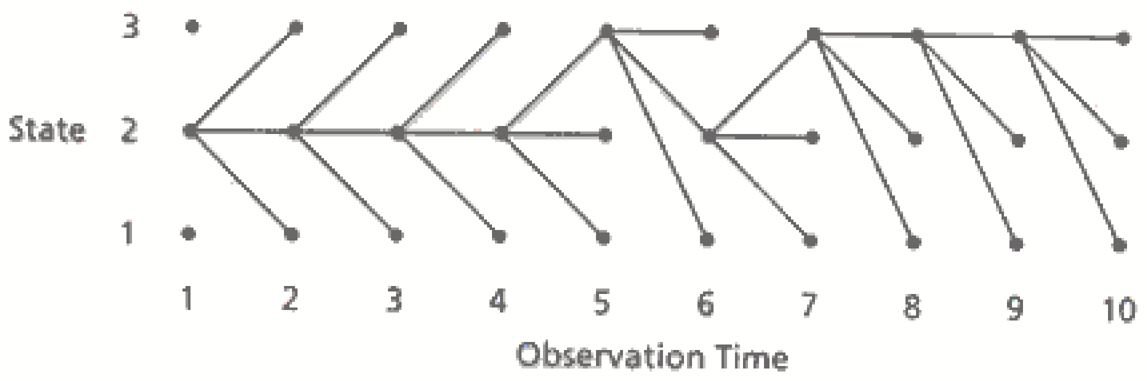
Since all a_{ij} terms are equal to 1/3, we can omit these terms (as well as the initial state probability term), giving

$$\delta_1(1) = 0.5, \quad \delta_1(2) = 0.75, \quad \delta_1(3) = 0.25.$$

The recursion for $\delta_t(j)$ gives ($2 \le t \le 10$)

$$\begin{array}{lll}
\delta_2(1) = (0.75)(0.5), & \delta_2(2) = (0.75)^2, & \delta_2(3) = (0.75)(0.25) \\
\delta_3(1) = (0.75)^2(0.5), & \delta_3(2) = (0.75)^3, & \delta_3(3) = (0.75)^2(0.25) \\
\delta_4(1) = (0.75)^3(0.5), & \delta_4(2) = (0.75)^4, & \delta_4(3) = (0.75)^3(0.25) \\
\delta_5(1) = (0.75)^4(0.5), & \delta_5(2) = (0.75)^4(0.25), & \delta_5(3) = (0.75)^5 \\
\delta_6(1) = (0.75)^5(0.5), & \delta_6(2) = (0.75)^6, & \delta_6(3) = (0.75)^5(0.25) \\
\delta_7(1) = (0.75)^6(0.5), & \delta_7(2) = (0.75)^6(0.25), & \delta_7(3) = (0.75)^7 \\
\delta_8(1) = (0.75)^7(0.5), & \delta_8(2) = (0.75)^7(0.25), & \delta_8(3) = (0.75)^8 \\
\delta_9(1) = (0.75)^8(0.5), & \delta_9(2) = (0.75)^8(0.25), & \delta_9(3) = (0.75)^9 \\
\delta_{10}(1) = (0.75)^9(0.5), & \delta_{10}(2) = (0.75)^9(0.25), & \delta_{10}(3) = (0.75)^{10}
\end{array}$$

This leads to a diagram (trellis) of the form:



Hence, the most likely state sequence is $\{2, 2, 2, 2, 3, 2, 3, 3, 3, 3\}$.

Solution to problem 3 - Parameter estimation

maximize $P(O|\lambda)$ for given O → find optimal $\lambda = (A, B, \pi)$

(unknown how to do this)

but: choose λ such that $P(O|\lambda)$ is locally maximized

→ Baum-Welch method (iterative procedure)

To describe the procedure for reestimation (iterative update and improvement) of HMM parameters, we first define $\xi_t(i,j)$, the probability of being in state i at time t, and state j at time $t+1$, given the model and the observation sequence, i.e.

$$\xi_t(i,j) = P(q_t = i,\ q_{t+1} = j | \mathbf{O}, \lambda). \tag{6.36}$$

The paths that satisfy the conditions required by Eq. (6.36) are illustrated in Figure 6.7. From the definitions of the forward and backward variables, we can write $\xi_t(i,j)$ in the form

$$\begin{aligned} \xi_t(i,j) &= \frac{P(q_t = i,\ q_{t+1} = j,\ \mathbf{O} \mid \lambda)}{P(\mathbf{O} \mid \lambda)} \\ &= \frac{\alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{P(\mathbf{O}|\lambda)} \\ &= \frac{\alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}. \end{aligned} \tag{6.37}$$

with $\alpha_t(i) = P(O_1 O_2 \dots O_t,\ q_t = i / \lambda)$

$\beta_t(i) = P(O_{t+1} O_{t+2} \dots O_T / q_t = i,\ \lambda)$

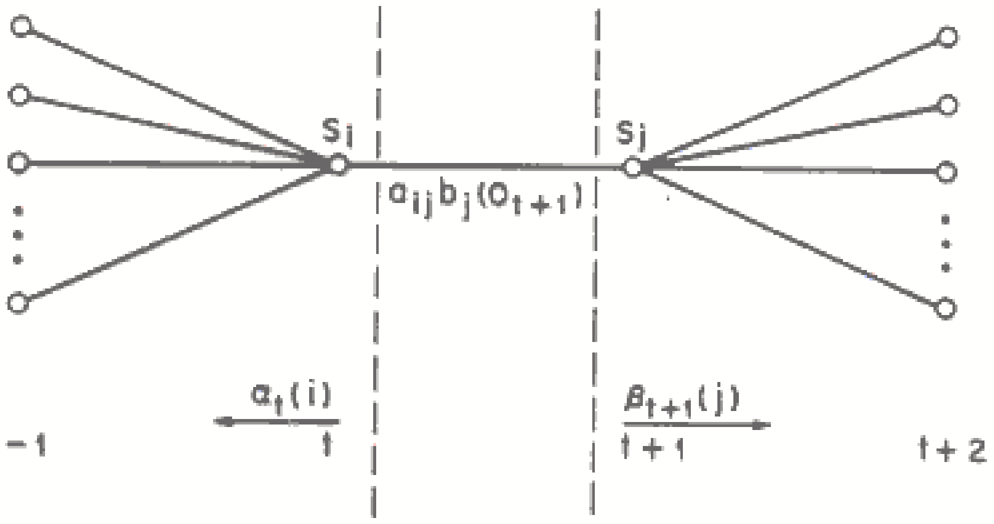


Figure 6.7 Illustration of the sequence of operations required for the computation of the joint event that the system is in state i at time t and state j at time $t+1$.

$\gamma_t(i)$ – prob. of being in state i at time t given O, λ

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j). = P(q_t = i \mid O, \lambda) \qquad (6.38)$$

If we sum $\gamma_t(i)$ over the time index t, we get a quantity that can be interpreted as the expected (over time) number of times that state i is visited, or equivalently, the expected number of transitions made from state i (if we exclude the time slot $t = T$ from the summation). Similarly, summation of $\xi_t(i,j)$ over t (from $t = 1$ to $t = T - 1$) can be interpreted as the expected number of transitions from state i to state j. That is,

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from state } i \text{ in O} \qquad (6.39a)$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{expected number of transitions from state } i \text{ to state } j \text{ in O}. \qquad (6.39b)$$

Using the above formulas (and the concept of counting event occurrences), we can give a method for reestimation of the parameters of an HMM. A set of reasonable reestimation formulas for π, A, and B is

$$\bar{\pi}_j = \text{expected frequency (number of times) in state } i \text{ at time } (t = 1) = \gamma_1(i) \qquad (6.40a)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad (6.40b)$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } \mathbf{v}_k}{\text{expected number of times in state } j} = \frac{\sum\limits_{\substack{t=1 \\ \text{s.t. } O_t = v_k}}^{T} \gamma_t(j)}{\sum\limits_{t=1}^{T} \gamma_t(j)}. \qquad (6.40c)$$

i.e., given $\lambda = (A, B, \pi)$ we get a new $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ with

$P(O \mid \bar{\lambda}) > P(O \mid \lambda)$ that is, $\bar{\lambda}$ is better

- repeat procedure till convergence

The reestimation formulas of Eqs. (6.40a)–(6.40c) can be derived directly by maximizing (using standard constrained optimization techniques) Baum's auxiliary function

$$Q(\lambda', \lambda) = \sum_{\mathbf{q}} P(\mathrm{O}, \mathbf{q} \mid \lambda') \log P(\mathrm{O}, \mathbf{q} \mid \lambda) \qquad (6.41)$$

over λ. Because

$$Q(\lambda', \lambda) \geq Q(\lambda', \lambda') \Rightarrow P(\mathrm{O} \mid \lambda) \geq P(\mathrm{O} \mid \lambda') \qquad (6.42)$$

we can maximize the function $Q(\lambda', \lambda)$ over λ to improve λ' in the sense of increasing the likelihood $P(\mathrm{O} \mid \lambda)$. Eventually the likelihood function converges to a critical point if we iterate the procedure.

- Note: stochastic constraints of the HMM parameters λ are automatically incorporated at each iteration