

- [40] D. Mansour and B.H. Juang, "A family of distortion measures based upon projection operation for robust speech recognition," *Proc. ICASSP 88*, New York, NY, April 1988; also in *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-37 (11): 1659-1671, November 1989.
- [41] D. Mansour and B.H. Juang, "The short-time modified coherence representation and noisy speech recognition," *IEEE Trans. Acoustics, Speech, Signal Proc.* ASSP-37 (6): 795-804, June 1989.
- [42] O. Ghitza, "Auditory nerve representation as a front-end for speech recognition in a noisy environment," *Computer Speech and Language*, 1 (2): 109-130, December 1986.

## Chapter 6

# THEORY AND IMPLEMENTATION OF HIDDEN MARKOV MODELS

### 6.1 INTRODUCTION

In Chapters 4 and 5 we presented one major pattern-recognition approach to speech recognition, namely the template method. One key idea in the template method is to derive typical sequences of speech frames for a pattern (e.g., a word) via some averaging procedure, and to rely on the use of local spectral distance measures to compare patterns. Another key idea is to use some form of dynamic programming to temporally align patterns to account for differences in speaking rates across talkers as well as across repetitions of the word by the same talker. The methodology of the template approach is well developed and provides good recognition performance for a variety of practical applications.

The template approach, however, is not based on the ideas of statistical signal modeling in a strict sense. Even though statistical techniques have been widely used in clustering to create reference patterns, the template approach is best classified as a simplified, non-parametric method in which a multiplicity of reference tokens (sequences) are used to characterize the variation among different utterances. As such, statistical signal characterization inherent in the template representation is only implicit and often inadequate. Consider, for example, the use of a truncated cepstral distortion measure as the local distance for template matching. The Euclidean distance form of the cepstral distance measure suggests that the reference vector can be viewed as the *mean* of some assumed distribution.

Obviously, this simple form of the sufficient statistic<sup>1</sup> (use of only the mean reference vector) neglects the second-order statistics—i.e., *covariances*, which, as will be seen later, are of particular significance in statistical modeling. (Note that this distribution is used to account for variations of the cepstral coefficients at the frame level since time alignment is performed so as to match appropriate frames of the patterns being compared.) There is clearly a need to use a more elaborate and analytical statistical method for speech recognition.

In this chapter we will study one well-known and widely used statistical method of characterizing the spectral properties of the frames of a pattern, namely the hidden Markov model (HMM) approach. (These models are also referred to as Markov sources or probabilistic functions of Markov chains in the communications literature.) The underlying assumption of the HMM (or any other type of statistical model) is that the speech signal can be well characterized as a parametric random process, and that the parameters of the stochastic process can be determined (estimated) in a precise, well-defined manner. We will show that the HMM method provides a natural and highly reliable way of recognizing speech for a wide range of applications and integrates well into systems incorporating both task syntax and semantics.

The basic theory of hidden Markov models was published in a series of classic papers by Baum and his colleagues ([1]–[5]) in the late 1960s and early 1970s and was implemented for speech-processing applications by Baker [6] at CMU, and by Jelinek and his colleagues at IBM ([7]–[13]) in the 1970s.

We begin this chapter with a review of the theory of Markov chains and then extend the ideas to HMMs using several simple examples. Based on the now-classical approach of Jack Ferguson of IDA (Institute for Defense Analyses), as introduced in lectures and in writing [14], we will focus our attention on the three fundamental problems for HMM design; namely: the evaluation of the probability (or likelihood) of a sequence of observations given a specific HMM; the determination of a best sequence of model states; and the adjustment of model parameters so as to best account for the observed signal. We will show that once these three fundamental problems are solved, we can readily apply HMMs to selected problems in speech recognition.

## 6.2 DISCRETE-TIME MARKOV PROCESSES

Consider a system that may be described at any time as being in one of a set of  $N$  distinct states indexed by  $\{1, 2, \dots, N\}$  as illustrated in Figure 6.1 (where  $N = 5$  for simplicity). At regularly spaced, discrete times, the system undergoes a change of state (possibly back to the same state) according to a set of probabilities associated with the state. We denote the time instants associated with state changes as  $t = 1, 2, \dots$ , and we denote the actual

<sup>1</sup>Sufficient statistics are a set of measurements from a process which contain all the relevant information for estimating the parameters of that process.

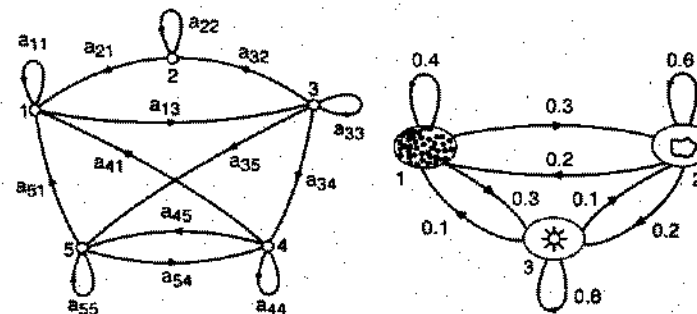


Figure 6.1 A Markov chain with five states (labeled 1 to 5) with selected state transitions.

Figure 6.2 Markov model of the weather.

state at time  $t$  as  $q_t$ . A full probabilistic description of the above system would, in general, require specification of the current state (at time  $t$ ), as well as all the predecessor states. For the special case of a discrete-time, first order, Markov chain, the probabilistic dependence is truncated to just the preceding state—that is,

$$P\{q_t = j | q_{t-1} = i, q_{t-2} = k, \dots\} = P\{q_t = j | q_{t-1} = i\}. \quad (6.1)$$

Furthermore, we consider only those processes in which the right-hand side of (6.1) is independent of time, thereby leading to the set of state-transition probabilities  $a_{ij}$  of the form

$$a_{ij} = P\{q_t = j | q_{t-1} = i\}, \quad 1 \leq i, j \leq N \quad (6.2)$$

with the following properties

$$a_{ij} \geq 0 \quad \forall j, i \quad (6.3a)$$

$$\sum_{j=1}^N a_{ij} = 1 \quad \forall i \quad (6.3b)$$

since they obey standard stochastic constraints.

The above stochastic process could be called an observable Markov model because the output of the process is the set of states at each instant of time, where each state corresponds to an observable event. To set ideas, consider a simple three-state Markov model of the weather as shown in Figure 6.2. We assume that once a day (e.g., at noon), the weather is observed as being one of the following:

**State 1:** precipitation (rain or snow)

**State 2:** cloudy

**State 3:** sunny.

We postulate that the weather on day  $t$  is characterized by a single one of the three states above, and that the matrix  $A$  of state-transition probabilities is

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

Given the model of Figure 6.2 we can now ask (and answer) several interesting questions about weather patterns over time. For example, we can pose the following simple problem:

#### Problem

What is the probability (according to the model) that the weather for eight consecutive days is "sun-sun-sun-rain-rain-sun-cloudy-sun"?

#### Solution

We define the observation sequence,  $O$ , as

$$\begin{array}{l} O = (\text{sunny, sunny, sunny, rain, rain, sunny, cloudy, sunny}) \\ = (3, 3, 3, 1, 1, 3, 2, 3) \\ \text{day} \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \end{array}$$

corresponding to the postulated set of weather conditions over the eight-day period and we want to calculate  $P(O|\text{Model})$ , the probability of the observation sequence  $O$ , given the model of Figure 6.2. We can directly determine  $P(O|\text{Model})$  as:

$$\begin{aligned} P(O|\text{Model}) &= P[3, 3, 3, 1, 1, 3, 2, 3|\text{Model}] \\ &= P[3]P[3|3]^2P[1|3]P[1|1] \\ &\quad P[3|1]P[2|3]P[3|2] \\ &= \pi_3 \cdot (a_{33})^2 a_{31} a_{11} a_{13} a_{32} a_{23} \\ &= (1.0)(0.8)^2(0.1)(0.4)(0.3)(0.1)(0.2) \\ &= 1.536 \times 10^{-4} \end{aligned}$$

where we use the notation:

$$\pi_i = P[q_1 = i], \quad 1 \leq i \leq N \quad (6.4)$$

to denote the initial state probabilities.

Another interesting question we can ask (and answer using the model) is:

#### Problem

Given that the system is in a known state, what is the probability that it stays in that state for exactly

#### Solution

This probability can be evaluated as the probability of the observation sequence

$$\begin{array}{l} O = (i, i, i, \dots, i, j \neq i) \\ \text{day} \quad 1 \quad 2 \quad 3 \quad \dots \quad d \quad d+1 \end{array}$$

given the model, which is

$$\begin{aligned} P(O|\text{Model}, q_1 = i) &= P(O, q_1 = i|\text{Model})/P(q_1 = i) \\ &= \pi_i (a_{ii})^{d-1} (1 - a_{ii}) / \pi_i \\ &= (a_{ii})^{d-1} (1 - a_{ii}) \\ &= p_i(d) \end{aligned} \quad (6.5)$$

The quantity  $p_i(d)$  is the probability distribution function of duration  $d$  in state  $i$ . This exponential distribution is characteristic of the state duration in a Markov chain. Based on  $p_i(d)$ , we can readily calculate the expected number of observations (duration) in a state, conditioned on starting in that state as

$$\bar{d}_i = \sum_{d=1}^{\infty} d p_i(d) \quad (6.6a)$$

$$= \sum_{d=1}^{\infty} d (a_{ii})^{d-1} (1 - a_{ii}) = \frac{1}{1 - a_{ii}} \quad (6.6b)$$

Thus the expected number of consecutive days of sunny weather, according to the model, is  $1/(0.2) = 5$ ; for cloudy it is 2.5; for rain it is 1.67.

#### Problem

Derive the expression for the mean of  $p_i(d)$ , i.e. Eq. (6.6b).

#### Solution

$$\begin{aligned} \bar{d}_i &= \sum_{d=1}^{\infty} d p_i(d) \\ &= \sum_{d=1}^{\infty} d (a_{ii})^{d-1} (1 - a_{ii}) \\ &= (1 - a_{ii}) \frac{\partial}{\partial a_{ii}} \left[ \sum_{d=1}^{\infty} a_{ii}^d \right] \\ &= (1 - a_{ii}) \frac{\partial}{\partial a_{ii}} \left( \frac{a_{ii}}{1 - a_{ii}} \right) \\ &= \frac{1}{1 - a_{ii}} \end{aligned}$$

## 6.3 EXTENSIONS TO HIDDEN MARKOV MODELS

So far we have considered Markov models in which each state corresponded to a deterministically observable event. Thus, the output of such sources in any given state is not random. This model is too restrictive to be applicable to many problems of interest. In this section we extend the concept of Markov models to include the case in which the observation is a probabilistic function of the state—that is, the resulting model (which is

called a hidden Markov model) is a doubly embedded stochastic process with an underlying stochastic process that is *not* directly observable (it is hidden) but can be observed only through another set of stochastic processes that produce the sequence of observations.

To illustrate the basic concepts of the hidden Markov model, we will use several simple examples including simple coin-tossing experiments. We begin with a review of some basic ideas of probability in the following exercise.

#### Exercise 6.1

Given a single fair coin, i.e.,  $P(\text{Heads}) = P(\text{Tails}) = 0.5$ , which you toss once and observe Tails,

1. What is the probability that the next 10 tosses will provide the sequence (HHTHTTHTTH)?
2. What is the probability that the next 10 tosses will produce the sequence (HHHHHHHHHH)?
3. What is the probability that 5 of the next 10 tosses will be tails? What is the expected number of tails over the next 10 tosses?

#### Solution 6.1

1. For a fair coin, with independent coin tosses, the probability of any specific observation sequence of length 10 (10 tosses) is  $(1/2)^{10}$  since there are  $2^{10}$  such sequences and all are equally probable. Thus:

$$P(\text{HHTHTTHTTH}) = \left(\frac{1}{2}\right)^{10}$$

2.

$$P(\text{HHHHHHHHHH}) = \left(\frac{1}{2}\right)^{10}$$

Thus a specified run of length 10 is as likely as a specified run of interlaced  $H$  and  $T$ .

3. The probability of 5 tails in the next 10 tosses is just the number of observation sequences with 5 tails and 5 heads (in any order) and this is

$$P(5H, 5T) = \binom{10}{5} \left(\frac{1}{2}\right)^{10} = \frac{252}{1024} \cong 0.25$$

since there are  $\binom{10}{5}$  ways of getting  $5H$  and  $5T$  in 10 tosses, and each sequence has probability of  $\left(\frac{1}{2}\right)^{10}$ . The expected number of tails in 10 tosses is

$$E(T \text{ in } 10 \text{ tosses}) = \sum_{d=0}^{10} d \binom{10}{d} \left(\frac{1}{2}\right)^{10} = 5.$$

Thus, on average, there will be  $5H$  and  $5T$  in 10 tosses, but the probability of exactly  $5H$  and  $5T$  is only 0.25.

### 6.3.1 Coin-Toss Models

Assume the following scenario. You are in a room with a barrier (e.g., a curtain) through

which you cannot see what is happening. On the other side of the barrier is another person who is performing a coin-tossing experiment (using one or more coins). The person will not tell you which coin he selects at any time; he will only tell you the result of each coin flip. Thus a sequence of *hidden* coin-tossing experiments is performed, with the observation sequence consisting of a series of heads and tails. A typical observation sequence would be

$$\begin{aligned} O &= (o_1 o_2 o_3 \dots o_T) \\ &= (HHTTTHTTH \dots H) \end{aligned}$$

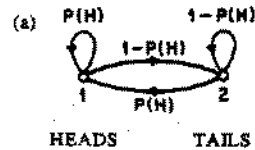
where  $H$  stands for heads and  $T$  stands for tails.

Given the above scenario, the question is, How do we build an HMM to explain (model) the observed sequence of heads and tails? The first problem we face is deciding what the states in the model correspond to, and then deciding how many states should be in the model. One possible choice would be to assume that only a single biased coin was being tossed. In this case, we could model the situation with a two-state model in which each state corresponds to the outcome of the previous toss (i.e., heads or tails). This model is depicted in Figure 6.3a. In this case, the Markov model is observable, and the only issue for complete specification of the model would be to decide on the best value for the single parameter of the model (i.e., the probability of, say, heads). Interestingly, an equivalent HMM to that of Figure 6.3a would be a degenerate one-state model in which the state corresponds to the single biased coin, and the unknown parameter is the bias of the coin.

A second HMM for explaining the observed sequence of coin toss outcomes is given in Figure 6.3b. In this case there are two states in the model, and each state corresponds to a different, biased coin being tossed. Each state is characterized by a probability distribution of heads and tails, and transitions between states are characterized by a state-transition matrix. The physical mechanism that accounts for how state transitions are selected could itself be a set of independent coin tosses or some other probabilistic event.

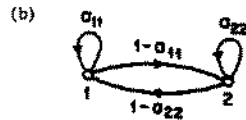
A third form of HMM for explaining the observed sequence of coin toss outcomes is given in Figure 6.3c. This model corresponds to using three biased coins, and choosing from among the three, based on some probabilistic event.

Given the choice among the three models shown in Figure 6.3 for explaining the observed sequence of heads and tails, a natural question would be which model best matches the actual observations. It should be clear that the simple one-coin model of Figure 6.3a has only one unknown parameter; the two-coin model of Figure 6.3b has four unknown parameters; and the three-coin model of Figure 6.3c has nine unknown parameters. Thus, with the greater degrees of freedom, the larger HMMs would seem to be inherently more capable of modeling a series of coin-tossing experiments than would equivalently smaller models. Although this is theoretically true, we will see later in this chapter that practical considerations impose some strong limitations on the size of models that we can consider. A fundamental question here is whether the observed head-tail sequence is long and rich enough to be able to specify a complex model. Also, it might just be the case that only a single coin is being tossed. Then using the three-coin model of Figure 6.3c would be inappropriate because we would be using an underspecified system.



1-COIN MODEL  
(OBSERVABLE MARKOV MODEL)

$O = HHTTHTHTHTTH \dots$   
 $S = 11221211221 \dots$

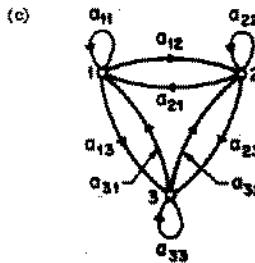


2-COINS MODEL  
(HIDDEN MARKOV MODEL)

$O = HHTTHTHTHTTH \dots$   
 $S = 21122212212 \dots$

$$P(H) = P_1 \quad P(H) = P_2$$

$$P(T) = 1 - P_1 \quad P(T) = 1 - P_2$$



3-COINS MODEL  
(HIDDEN MARKOV MODEL)

$O = HHTTHTHTHTTH \dots$   
 $S = 31233112313 \dots$

	1	2	3
$P(H)$	$P_1$	$P_2$	$P_3$
$P(T)$	$1 - P_1$	$1 - P_2$	$1 - P_3$

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.

### 6.3.2 The Urn-and-Ball Model

To extend the ideas of the HMM to a somewhat more complicated situation, consider the urn-and-ball system of Figure 6.4. We assume that there are  $N$  (large) glass urns in a room. Within each urn is a large quantity of colored balls. We assume there are  $M$  distinct colors of the balls. The physical process for obtaining observations is as follows. A genie is in the room, and, according to some random procedure, it chooses an initial urn. From this urn, a ball is chosen at random, and its color is recorded as the observation. The ball is

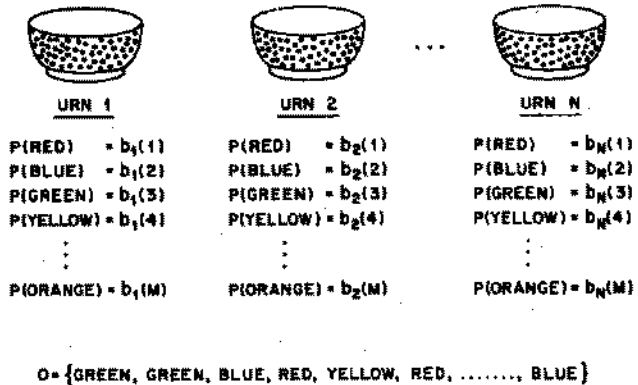


Figure 6.4 An  $N$ -state urn-and-ball model illustrating the general case of a discrete symbol HMM.

then replaced in the urn from which it was selected. A new urn is then selected according to the random selection procedure associated with the current urn, and the ball selection process is repeated. This entire process generates a finite observation sequence of colors, which we would like to model as the observable output of an HMM.

It should be obvious that the simplest HMM that corresponds to the urn-and-ball process is one in which each state corresponds to a specific urn, and for which a (ball) color probability is defined for each state. The choice of urns is dictated by the state-transition matrix of the HMM.

It should be noted that the ball colors in each urn may be the same, and the distinction among various urns is in the way the collection of colored balls is composed. Therefore, an isolated observation of a particular color ball does not immediately tell which urn it is drawn from.

### 6.3.3 Elements of an HMM

The above examples give us some idea of what an HMM is and how it can be applied to some simple scenarios. We now formally define the elements of an HMM.

An HMM for discrete symbol observations such as the above urn-and-ball model is characterized by the following:

1.  $N$ , the number of states in the model. Although the states are hidden, for many practical applications there is often some physical significance attached to the states or to sets of states of the model. Thus, in the coin-tossing experiments, each state corresponded to a distinct biased coin. In the urn-and-ball model, the states corresponded to the urns. Generally the states are interconnected in such a way that any state can be reached from any other state (i.e., an ergodic model); however, we will see later in this chapter that other possible interconnections of states are often

of interest and may better suit speech applications. We label the individual states as  $\{1, 2, \dots, N\}$ , and denote the state at time  $t$  as  $q_t$ .

- $M$ , 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  $V = \{v_1, v_2, \dots, v_M\}$ .
- The state-transition probability distribution  $A = \{a_{ij}\}$  where

$$a_{ij} = P[q_{t+1} = j | q_t = i], \quad 1 \leq i, j \leq N. \quad (6.7)$$

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.

- The observation symbol probability distribution,  $B = \{b_j(k)\}$ , in which

$$b_j(k) = P[o_t = v_k | q_t = j], \quad 1 \leq k \leq M, \quad (6.8)$$

defines the symbol distribution in state  $j, j = 1, 2, \dots, N$ .

- The initial state distribution  $\pi = \{\pi_i\}$  in which

$$\pi_i = P[q_1 = i], \quad 1 \leq i \leq N. \quad (6.9)$$

It can be seen from the above discussion that a complete specification of an HMM requires specification of two model parameters,  $N$  and  $M$ , specification of observation symbols, and the specification of the three sets of probability measures  $A, B$ , and  $\pi$ . For convenience, we use the compact notation

$$\lambda = (A, B, \pi) \quad (6.10)$$

to indicate the complete parameter set of the model. This parameter set, of course, defines a probability measure for  $O$ , i.e.  $P(O|\lambda)$ , which we discuss in the next section. We use the terminology HMM to indicate the parameter set  $\lambda$  and the associated probability measure interchangeably without ambiguity.

### 6.3.4 HMM Generator of Observations

Given appropriate values of  $N, M, A, B$ , and  $\pi$ , the HMM can be used as a generator to give an observation sequence

$$O = (o_1 o_2 \dots o_T) \quad (6.11)$$

(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:

- Choose an initial state  $q_1 = i$  according to the initial state distribution  $\pi$ .
- Set  $t = 1$ .
- Choose  $o_t = v_k$  according to the symbol probability distribution in state  $i$ , i.e.,  $b_j(k)$ .

- Transit to a new state  $q_{t+1} = j$  according to the state-transition probability distribution for state  $i$ , i.e.,  $a_{ij}$ .
- 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.

#### Exercise 6.2

Consider an HMM representation (parametrized by  $\lambda$ ) of a coin-tossing experiment. Assume a three-state model (corresponding to 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  $1/3$ . (Assume initial state probabilities of  $1/3$ .)

- You observe the sequence

$$O = (HHHHTHTTTT).$$

What state sequence is most likely? What is the probability of the observation sequence and this most likely state sequence?

- What is the probability that the observation sequence came entirely from state 1?
- Consider the observation sequence

$$\hat{O} = (HTTHTHTTHT).$$

How would your answers to parts a and b change?

- If the state-transition probabilities were

$$\begin{aligned} a_{11} &= 0.9 & a_{21} &= 0.45 & a_{31} &= 0.45 \\ a_{12} &= 0.05 & a_{22} &= 0.1 & a_{32} &= 0.45 \\ a_{13} &= 0.05 & a_{23} &= 0.45 & a_{33} &= 0.1 \end{aligned}$$

that is, a new model  $\lambda'$ , how would your answers to parts 1–3 change? What does this suggest about the type of sequences generated by the models?

#### Solution 6.2

- Given  $O = (HHHHTHTTTT)$  and that all state transitions are equiprobable, the most likely state sequence is the one for which the probability of each individual observation

is maximum. Thus for each  $H$ , the most likely state is 2 and for each  $T$  the most likely state is 3. Thus the most likely state sequence is

$$\mathbf{q} = (2222323333).$$

The probability of  $\mathbf{O}$  and  $\mathbf{q}$  (given the model) is

$$P(\mathbf{O}, \mathbf{q}|\lambda) = (0.75)^{10} \left(\frac{1}{3}\right)^{10}.$$

2. The probability of  $\mathbf{O}$  given that  $\hat{\mathbf{q}}$  is

$$\hat{\mathbf{q}} = (1111111111)$$

is

$$P(\mathbf{O}, \hat{\mathbf{q}}|\lambda) = (0.50)^{10} \left(\frac{1}{3}\right)^{10}.$$

The ratio of  $P(\mathbf{O}, \mathbf{q}|\lambda)$  to  $P(\mathbf{O}, \hat{\mathbf{q}}|\lambda)$  is:

$$R = \frac{P(\mathbf{O}, \mathbf{q}|\lambda)}{P(\mathbf{O}, \hat{\mathbf{q}}|\lambda)} = \left(\frac{3}{2}\right)^{10} = 57.67$$

which shows, as expected, that  $\mathbf{q}$  is more likely than  $\hat{\mathbf{q}}$ .

3. Given  $\hat{\mathbf{O}}$  which has the same number of  $H$ s and  $T$ s, the answers to parts 1 and 2 would remain the same, as the most likely states occur the same number of times in both cases.  
4. The new probability of  $\hat{\mathbf{O}}$  and  $\hat{\mathbf{q}}$  becomes

$$P(\hat{\mathbf{O}}, \hat{\mathbf{q}}|\lambda') = (0.75)^{10} \left(\frac{1}{3}\right)^6 (0.1)^6 (0.45)^3.$$

The new probability of  $\mathbf{O}$  and  $\hat{\mathbf{q}}$  becomes

$$P(\mathbf{O}, \hat{\mathbf{q}}|\lambda') = (0.50)^{10} \left(\frac{1}{3}\right)^6 (0.9)^9.$$

The ratio is

$$R = \left(\frac{3}{2}\right)^{10} \left(\frac{1}{9}\right)^6 \left(\frac{1}{2}\right)^3 = 1.36 \times 10^{-5}.$$

In other words, because of the nonuniform transition probabilities,  $\hat{\mathbf{q}}$  is more likely than  $\mathbf{q}$ . (The reader is encouraged to find the most likely state sequence in this case.) Now, the probability of  $\hat{\mathbf{O}}$  and  $\hat{\mathbf{q}}$  is not the same as the probability of  $\mathbf{O}$  and  $\mathbf{q}$ . We have

$$P(\hat{\mathbf{O}}, \mathbf{q}|\lambda') = \frac{1}{3} (0.1)^6 (0.45)^3 (0.25)^4 (0.75)^6$$

$$P(\hat{\mathbf{O}}, \hat{\mathbf{q}}|\lambda') = (0.50)^{10} \left(\frac{1}{3}\right)^6 (0.9)^9$$

with ratio

$$R = \left(\frac{1}{9}\right)^6 \left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^4 \left(\frac{3}{2}\right)^6 = 1.67 \times 10^{-7}.$$

Clearly, because  $a_{11} = 0.9$ ,  $\hat{\mathbf{q}}$  is more likely.

## 6.4 THE THREE BASIC PROBLEMS FOR HMMs

Given the form of HMM of the previous section, three basic problems of interest must be solved for the model to be useful in real-world applications. These problems are the following:

### Problem 1

Given the observation sequence  $\mathbf{O} = (o_1 o_2 \dots o_T)$ , and a model  $\lambda = (A, B, \pi)$ , how do we efficiently compute  $P(\mathbf{O}|\lambda)$ , the probability of the observation sequence, given the model?

### Problem 2

Given the observation sequence  $\mathbf{O} = (o_1 o_2 \dots o_T)$ , and the model  $\lambda$ , how do we choose a corresponding state sequence  $\mathbf{q} = (q_1 q_2 \dots q_T)$  that is optimal in some sense (i.e., best "explains" the observations)?

### Problem 3

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

Problem 1 is the evaluation problem; namely, given a model and a sequence of observations, how do we compute the probability that the observed sequence was produced by the model? We can also view the problem as one of scoring how well a given model matches a given observation sequence. The latter viewpoint is extremely useful. For example, if we consider the case in which we are trying to choose among several competing models, the solution to Problem 1 allows us to choose the model that best matches the observations.

Problem 2 is the one in which we attempt to uncover the hidden part of the model—that is, to find the "correct" state sequence. It should be clear that for all but the case of degenerate models, there is no "correct" state sequence to be found. Hence for practical situations, we usually use an optimality criterion to solve this problem as best as possible. As we will see, several reasonable optimality criteria can be imposed, and hence the choice of criterion is a strong function of the intended use for the uncovered state sequence. Typical uses might be to learn about the structure of the model, to find optimal state sequences for continuous speech recognition, or to get average statistics of individual states, etc.

Problem 3 is the one in which we attempt to optimize the model parameters to best describe how a given observation sequence comes about. The observation sequence used to adjust the model parameters is called a training sequence because it is used to "train" the HMM. The training problem is the crucial one for most applications of HMMs, because it allows us to optimally adapt model parameters to observed training data—i.e., to create best models for real phenomena.

To fix ideas, consider the following simple isolated-word speech recognizer. For each word of a  $W$  word vocabulary, we want to design a separate  $N$ -state HMM. We represent the speech signal of a given word as a time sequence of coded spectral vectors. We assume that the coding is done using a spectral codebook with  $M$  unique spectral vectors; hence each observation is the index of the spectral vector closest (in some spectral distortion sense) to the original speech signal. Thus, for each vocabulary word, we have a training sequence

consisting of a number of repetitions of sequences of codebook indices of the word (by one or more talkers). The first task is to build individual word models. This task is done by using the solution to Problem 3 to optimally estimate model parameters for each word model. To develop an understanding of the physical meaning of the model states, we use the solution to Problem 2 to segment each of the word training sequences into states, and then study the properties of the spectral vectors that lead to the observations occurring in each state. The goal here is to make refinements of the model (e.g., more states, different codebook size) to improve its capability of modeling the spoken word sequences. Finally, once the set of  $W$  HMMs has been designed and optimized, recognition of an unknown word is performed using the solution to Problem 1 to score each word model based upon the given test observation sequence, and select the word whose model score is highest (i.e., the highest likelihood).

In the next sections we present formal mathematical solutions to each fundamental problem for HMMs. We shall see that the three problems are tightly linked together under the probabilistic framework.

#### 6.4.1 Solution to Problem 1—Probability Evaluation

We wish to calculate the probability of the observation sequence,  $\mathbf{O} = (o_1 o_2 \dots o_T)$ , given the model  $\lambda$ , i.e.,  $P(\mathbf{O}|\lambda)$ . 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

$$\mathbf{q} = (q_1 q_2 \dots q_T) \quad (6.12)$$

where  $q_1$  is the initial state. The probability of the observation sequence  $\mathbf{O}$  given the state sequence of Eq. (6.12) is

$$P(\mathbf{O}|\mathbf{q}, \lambda) = \prod_{i=1}^T P(o_i|q_i, \lambda) \quad (6.13a)$$

where we have assumed statistical independence of observations. Thus we get

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

The probability of such a state sequence  $\mathbf{q}$  can be written as

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

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

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

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

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

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

The interpretation of the computation in the above equation is the following. Initially (at time  $t = 1$ ) we are in state  $q_1$  with probability  $\pi_{q_1}$ , and generate the symbol  $o_1$  (in this state) with probability  $b_{q_1}(o_1)$ . The clock changes from time  $t$  to  $t + 1$  (time = 2) and we make a transition to state  $q_2$  from state  $q_1$  with probability  $a_{q_1 q_2}$ , and generate symbol  $o_2$  with probability  $b_{q_2}(o_2)$ . This process continues in this manner until we make the last transition (at time  $T$ ) from state  $q_{T-1}$  to state  $q_T$  with probability  $a_{q_{T-1} q_T}$  and generate symbol  $o_T$  with probability  $b_{q_T}(o_T)$ .

A little thought should convince the reader that the calculation of  $P(\mathbf{O}|\lambda)$ , according to its direct definition (Eq. (6.17)) involves on the order of  $2T \cdot N^T$  calculations, since at every  $t = 1, 2, \dots, T$ , there are  $N$  possible states that can be reached (i.e., there are  $N^T$  possible state sequences), and for each such state sequence about  $2T$  calculations are required for each term in the sum of Eq. (6.17). (To be precise, we need  $(2T - 1)N^T$  multiplications, and  $N^T - 1$  additions.) This calculation is computationally infeasible, even for small values of  $N$  and  $T$ ; e.g., for  $N = 5$  (states),  $T = 100$  (observations), there are on the order of  $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$  computations! Clearly a more efficient procedure is required to solve problem 1. Fortunately such a procedure (called the forward procedure) exists.

##### 6.4.1.1 The Forward Procedure

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

$$\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = i | \lambda) \quad (6.18)$$

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

##### 1. Initialization

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N. \quad (6.19)$$

##### 2. Induction

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

##### 3. Termination

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

Step 1 initializes the forward probabilities as the joint probability of state  $i$  and initial observation  $o_1$ . The induction step, which is the heart of the forward calculation, is illustrated in Figure 6.5(a). This figure shows how state  $j$  can be reached at time  $t + 1$  from the  $N$  possible states,  $i$ ,  $1 \leq i \leq N$ , at time  $t$ . Since  $\alpha_t(i)$  is the probability of the joint event that  $o_1 o_2 \dots o_t$  are observed, and the state at time  $t$  is  $i$ , the product  $\alpha_t(i) a_{ij}$  is

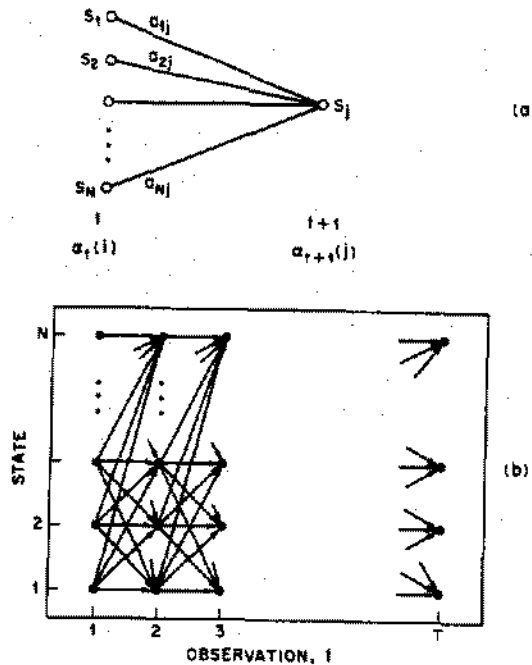


Figure 6.5 (a) Illustration of the sequence of operations required for the computation of the forward variable  $\alpha_{t+1}(j)$ . (b) Implementation of the computation of  $\alpha_t(i)$  in terms of a lattice of observations  $o_t$ , and states  $i$ .

then the probability of the joint event that  $o_1, o_2, \dots, o_t$  are observed, and state  $j$  is reached at time  $t+1$  via state  $i$  at time  $t$ . Summing this product over all the  $N$  possible states,  $i$ ,  $1 \leq i \leq N$  at time  $t$  results in the probability of  $j$  at time  $t+1$  with all the accompanying previous partial observations. Once this is done and  $j$  is known, it is easy to see that  $\alpha_{t+1}(j)$  is obtained by accounting for observation  $o_{t+1}$  in state  $j$ , i.e., by multiplying the summed quantity by the probability  $b_j(o_{t+1})$ . The computation of Eq. (6.20) is performed for all states  $j$ ,  $1 \leq j \leq N$ , for a given  $t$ ; the computation is then iterated for  $t = 1, 2, \dots, T-1$ . Finally, step 3 gives the desired calculation of  $P(\mathbf{O}|\lambda)$  as the sum of the terminal forward variables  $\alpha_T(i)$ . This is the case since, by definition,

$$\alpha_T(i) = P(o_1 o_2 \dots o_T, q_T = i | \lambda) \quad (6.22)$$

and hence  $P(\mathbf{O}|\lambda)$  is just the sum of the  $\alpha_T(i)$ 's.

If we examine the computation involved in the calculation of  $\alpha_t(j)$ ,  $1 \leq t \leq T$ ,  $1 \leq j \leq N$ , we see that it requires on the order of  $N^2 T$  calculations, rather than  $2TN^2$  as required by the direct calculation. (Again, to be precise, we need  $N(N+1)(T-1) + N$

multiplications and  $N(N-1)(T-1)$  additions.) For  $N = 5$ ,  $T = 100$ , we need about 3000 computations for the forward method, versus  $10^{72}$  computations for the direct calculation, a savings of about 69 orders of magnitude.

The forward probability calculation is, in effect, based upon the lattice (or trellis) structure shown in Figure 6.5(b). The key is that, because there are only  $N$  states (nodes at each time slot in the lattice), all the possible state sequences will remerge into these  $N$  nodes, no matter how long the observation sequence. At time  $t = 1$  (the first time slot in the lattice), we need to calculate values of  $\alpha_1(i)$ ,  $1 \leq i \leq N$ . At times  $t = 2, 3, \dots, T$ , we need only calculate values of  $\alpha_t(j)$ ,  $1 \leq j \leq N$ , where each calculation involves only the  $N$  previous values of  $\alpha_{t-1}(i)$  because each of the  $N$  grid points can be reached from only the  $N$  grid points at the previous time slot.

#### 6.4.1.2 The Backward Procedure

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

$$\beta_t(i) = P(o_{t+1} o_{t+2} \dots o_T | q_t = i, \lambda) \quad (6.23)$$

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

##### 1. Initialization

$$\beta_T(i) = 1, \quad 1 \leq i \leq N. \quad (6.24)$$

##### 2. Induction

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \dots, 1, \quad 1 \leq i \leq N. \quad (6.25)$$

The initialization step 1 arbitrarily defines  $\beta_T(i)$  to be 1 for all  $i$ . Step 2, which is illustrated in Figure 6.6, shows that in order to have been in state  $i$  at time  $t$ , and to account for the observation sequence from time  $t+1$  on, you have to consider all possible states  $j$  at time  $t+1$ , accounting for the transition from  $i$  to  $j$  (the  $a_{ij}$  term), as well as the observation  $o_{t+1}$  in state  $j$  (the  $b_j(o_{t+1})$  term), and then account for the remaining partial observation sequence from state  $j$  (the  $\beta_{t+1}(j)$  term). We will see later how the backward as well as the forward calculations are used to help solve fundamental Problems 2 and 3 of HMMs.

Again, the computation of  $\beta_t(i)$ ,  $1 \leq t \leq T$ ,  $1 \leq i \leq N$ , requires on the order of  $N^2 T$  calculations, and can be computed in a lattice structure similar to that of Figure 6.5(b).

#### 6.4.2 Solution to Problem 2—"Optimal" State Sequence

Unlike Problem 1, for which an exact solution can be given, there are several possible ways of solving Problem 2—namely, finding the "optimal" state sequence associated with the given observation sequence. The difficulty lies with the definition of the optimal state sequence—that is, there are several possible optimality criteria. For example, one possible

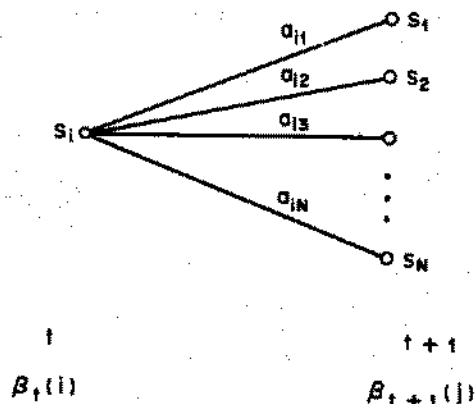


Figure 6.6 Sequence of operations required for the computation of the backward variable  $\beta_t(i)$ .

optimality criterion is to choose the states  $q_t$  that are *individually* most likely at each time  $t$ . This optimality criterion maximizes the expected number of correct individual states. To implement this solution to Problem 2, we can define the a posteriori probability variable

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

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

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

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

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

where we see that  $\alpha_t(i)$  accounts for the partial observation sequence  $\mathbf{o}_1 \mathbf{o}_2 \dots \mathbf{o}_t$  and state  $i$  at  $t$ , while  $\beta_t(i)$  accounts for the remainder of the observation sequence  $\mathbf{o}_{t+1} \mathbf{o}_{t+2} \dots \mathbf{o}_T$ , 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

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

Although Eq. (6.29) maximizes the expected number of correct states (by choosing the most likely state for each  $t$ ), there could be some problems with the resulting state sequence. For example, when the HMM has state transitions which have zero probability ( $a_{ij} = 0$  for some  $i$  and  $j$ ), the "optimal" state sequence may, in fact, not even be a valid state sequence. This is because the solution of Eq. (6.29) simply determines the most likely state at every instant, without regard to the probability of occurrence of sequences of states.

One possible solution to the above problem is to modify the optimality criterion. For example, one could solve for the state sequence that maximizes the expected number of correct pairs of states ( $q_t, q_{t+1}$ ), or triples of states ( $q_t, q_{t+1}, q_{t+2}$ ), etc. Although these criteria might be reasonable for some applications, the most widely used criterion is to find the *single* best state sequence (path)—that is, to maximize  $P(\mathbf{q} | \mathbf{O}, \lambda)$ , which is equivalent to maximizing  $P(\mathbf{q}, \mathbf{O} | \lambda)$ . A formal technique for finding this single best state sequence exists, based on dynamic programming methods, and is called the Viterbi algorithm [15, 16].

#### 6.4.2.1 The Viterbi Algorithm

To find the single best state sequence,  $\mathbf{q} = (q_1 q_2 \dots q_T)$ , for the given observation sequence  $\mathbf{O} = (\mathbf{o}_1 \mathbf{o}_2 \dots \mathbf{o}_T)$ , we need to define the quantity

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1 q_2 \dots q_{t-1}, q_t = i, \mathbf{o}_1 \mathbf{o}_2 \dots \mathbf{o}_t | \lambda] \quad (6.30)$$

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

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(\mathbf{o}_{t+1}). \quad (6.31)$$

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

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1), \quad 1 \leq i \leq N \quad (6.32a)$$

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

##### 2. Recursion

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(\mathbf{o}_t), \quad \begin{matrix} 2 \leq t \leq T \\ 1 \leq j \leq N \end{matrix} \quad (6.33a)$$

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

## 3. Termination

$$P^* = \max_{1 \leq j \leq N} [\delta_T(j)] \quad (6.34a)$$

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

## 4. Path (state sequence) backtracking

$$q_i^* = \psi_{i+1}(q_{i+1}^*), \quad i = T-1, T-2, \dots, 1. \quad (6.35)$$

It should be noted that the Viterbi algorithm is similar (except for the backtracking step) in implementation to the forward calculation of Eqs. (6.19)–(6.21). The major difference is the maximization in Eq. (6.33a) over previous states, which is used in place of the summing procedure in Eq. (6.20). It also should be clear that a lattice (or trellis) structure efficiently implements the computation of the Viterbi procedure.

## 6.4.2.2 Alternative Viterbi Implementation

By taking logarithms of the model parameters, the Viterbi algorithm of the preceding section can be implemented without the need for any multiplications. Thus:

## 0. Preprocessing

$$\begin{aligned} \bar{\pi}_i &= \log(\pi_i), & 1 \leq i \leq N \\ \bar{b}_i(\mathbf{o}_i) &= \log[b_i(\mathbf{o}_i)], & 1 \leq i \leq N, 1 \leq i \leq T \\ \bar{a}_{ij} &= \log(a_{ij}), & 1 \leq i, j \leq N \end{aligned}$$

## 1. Initialization

$$\begin{aligned} \bar{\delta}_1(i) &= \log(\delta_1(i)) = \bar{\pi}_i + \bar{b}_i(\mathbf{o}_1), & 1 \leq i \leq N \\ \psi_1(i) &= 0, & 1 \leq i \leq N \end{aligned}$$

## 2. Recursion

$$\begin{aligned} \bar{\delta}_t(j) &= \log(\delta_t(j)) = \max_{1 \leq i \leq N} [\bar{\delta}_{t-1}(i) + \bar{a}_{ij}] + \bar{b}_j(\mathbf{o}_t) \\ \psi_t(j) &= \arg \max_{1 \leq i \leq N} [\bar{\delta}_{t-1}(i) + \bar{a}_{ij}], & 2 \leq t \leq T, 1 \leq j \leq N \end{aligned}$$

## 3. Termination

$$\begin{aligned} \bar{P}^* &= \max_{1 \leq j \leq N} [\bar{\delta}_T(j)] \\ q_T^* &= \arg \max_{1 \leq i \leq N} [\bar{\delta}_T(i)] \end{aligned}$$

## 4. Backtracking

$$q_i^* = \psi_{i+1}(q_{i+1}^*), \quad i = T-1, T-2, \dots, 1$$

The calculation required for this alternative implementation is on the order of  $N^2T$  additions (plus the calculation for preprocessing). Because the preprocessing needs to be performed once and saved, its cost is negligible for most systems.

## Exercise 6.3

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

$$\mathbf{O} = (HHHHTHTTTT)$$

find the most likely path with the Viterbi algorithm.

## Solution 6.3

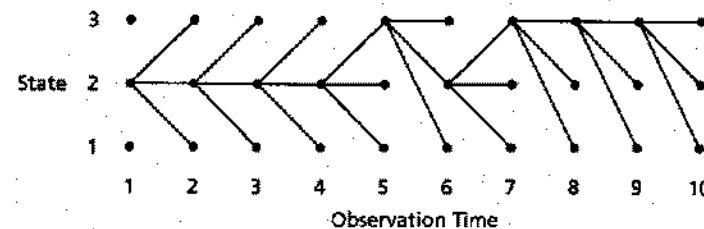
Since all  $a_{ij}$  terms are equal to  $1/3$ , we can omit these terms (as well as the initial state probability term), giving

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

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

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

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



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

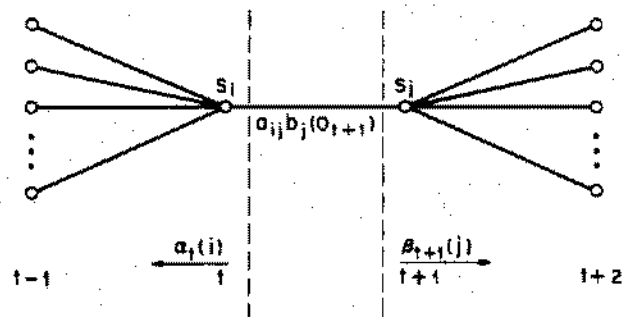


Figure 6.7 Illustration of the sequence of operations required for the computation of the joint event that the system is in state  $i$  at time  $t$  and state  $j$  at time  $t+1$ .

### 6.4.3 Solution to Problem 3—Parameter Estimation

The third, and by far the most difficult, problem of HMMs is to determine a method to adjust the model parameters  $(A, B, \pi)$  to satisfy a certain optimization criterion. There is no known way to analytically solve for the model parameter set that maximizes the probability of the observation sequence in a closed form. We can, however, choose  $\lambda = (A, B, \pi)$  such that its likelihood,  $P(\mathbf{O}|\lambda)$ , is locally maximized using an iterative procedure such as the Baum-Welch method (also known as the EM (expectation-maximization) method [17]), or using gradient techniques [18]. In this section we discuss one iterative procedure, based primarily on the classic work of Baum and his colleagues, for choosing the maximum likelihood (ML) model parameters.

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.

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

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

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

We have previously defined  $\gamma_t(i)$  as the probability of being in state  $i$  at time  $t$ , given

the entire observation sequence and the model; hence, we can relate  $\gamma_t(i)$  to  $\xi_t(i, j)$  by summing over  $j$ , giving

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j). \quad (6.38)$$

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,

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

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

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  $\pi$ ,  $A$ , and  $B$  is

$$\bar{\pi}_j = \frac{\text{expected frequency (number of times) in state } i \text{ at time } (t=1)}{\text{expected number of transitions from state } i \text{ to state } j} \quad (6.40a)$$

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

$$\begin{aligned} \bar{b}_j(k) &= \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j} \\ &= \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{k=1}^K \gamma_t(j)} \end{aligned} \quad (6.40c)$$

If we define the current model as  $\lambda = (A, B, \pi)$  and use that to compute the right-hand sides of Eqs. (6.40a)–(6.40c), and we define the reestimated model as  $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ , as determined from the left-hand sides of Eqs. (6.40a)–(6.40c), then it has been proven by Baum and his colleagues that either (1) the initial model  $\lambda$  defines a critical point of the

likelihood function, in which case  $\bar{\lambda} = \lambda$ ; or (2) model  $\bar{\lambda}$  is more likely than model  $\lambda$  in the sense that  $P(\mathbf{O}|\bar{\lambda}) > P(\mathbf{O}|\lambda)$ ; that is, we have found a new model  $\bar{\lambda}$  from which the observation sequence is more likely to have been produced.

Based on the above procedure, if we iteratively use  $\bar{\lambda}$  in place of  $\lambda$  and repeat the reestimation calculation, we then can improve the probability of  $\mathbf{O}$  being observed from the model until some limiting point is reached. The final result of this reestimation procedure is an ML estimate of the HMM. It should be pointed out that the forward-backward algorithm leads to local maxima only, and that in most problems of interest, the likelihood function is very complex and has many local maxima.

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

$$Q(\lambda', \lambda) = \sum_{\mathbf{q}} P(\mathbf{O}, \mathbf{q}|\lambda') \log P(\mathbf{O}, \mathbf{q}|\lambda) \quad (6.41)$$

over  $\lambda$ . Because

$$Q(\lambda', \lambda) \geq Q(\lambda', \lambda') \Rightarrow P(\mathbf{O}|\lambda) \geq P(\mathbf{O}|\lambda') \quad (6.42)$$

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

#### 6.4.3.1 Derivation of Reestimation Formulas from the $Q$ Function

The auxiliary function  $Q(\lambda', \lambda)$  was defined in Eq. (6.41) as

$$Q(\lambda', \lambda) = \sum_{\mathbf{q}} P(\mathbf{O}, \mathbf{q}|\lambda') \log P(\mathbf{O}, \mathbf{q}|\lambda)$$

in which we can express  $P$  and  $\log P$  (in terms of the HMM parameters) as

$$P(\mathbf{O}, \mathbf{q}|\lambda) = \pi_{q_0} \prod_{t=1}^T a_{q_{t-1}q_t} b_{q_t}(o_t)$$

$$\log P(\mathbf{O}, \mathbf{q}|\lambda) = \log \pi_{q_0} + \sum_{t=1}^T \log a_{q_{t-1}q_t} + \sum_{t=1}^T \log b_{q_t}(o_t)$$

(There is a slight difference between the above equations and the expression of Eq. (6.17) in which the first observation is associated with the initial state before any state transition is made. This difference is inconsequential and should not impede our understanding of the method.) Thus we can write  $Q(\lambda', \lambda)$  as

$$Q(\lambda', \lambda) = Q_{\pi}(\lambda', \pi) + \sum_{i=1}^N Q_{a_i}(\lambda', \mathbf{a}_i) + \sum_{i=1}^N Q_{b_i}(\lambda', \mathbf{b}_i)$$

where

$$\pi = [\pi_1, \pi_2, \dots, \pi_N],$$

$\mathbf{a}_i = [a_{i1}, a_{i2}, \dots, a_{iN}]$ ,  $\mathbf{b}_i$  is the parameter vector that defines  $b_i(\cdot)$

and

$$Q_{\pi}(\lambda', \pi) = \sum_{i=1}^N P(\mathbf{O}, q_0 = i|\lambda') \log \pi_i$$

$$Q_{a_i}(\lambda', \mathbf{a}_i) = \sum_{j=1}^N \sum_{t=1}^T P(\mathbf{O}, q_{t-1} = i, q_t = j|\lambda') \log a_{ij}$$

$$Q_{b_i}(\lambda', \mathbf{b}_i) = \sum_{t=1}^T P(\mathbf{O}, q_t = i|\lambda') \log b_i(o_t)$$

Because of the separability of  $Q(\lambda', \lambda)$  into three independent terms, we can maximize  $Q(\lambda', \lambda)$  over  $\lambda$  by maximizing the individual terms separately, subject to the stochastic constraints

$$\sum_{j=1}^N \pi_j = 1$$

$$\sum_{j=1}^N a_{ij} = 1, \quad \forall j$$

and (for discrete densities where  $b_i(o_t = v_k) = b_i(k)$ )

$$\sum_{k=1}^K b_i(k) = 1, \quad \forall i$$

Because the individual auxiliary functions all have the form

$$\sum_{j=1}^N w_j \log y_j$$

which, as a function of  $\{y_j\}_{j=1}^N$ , subject to the constraints  $\sum_{j=1}^N y_j = 1$ ,  $y_j \geq 0$ , attains a global maximum at the single point

$$y_j = \frac{w_j}{\sum_{i=1}^N w_i}, \quad j = 1, 2, \dots, N$$

then the maximization leads to the model reestimate  $\bar{\lambda} = [\bar{\pi}, \bar{\mathbf{A}}, \bar{\mathbf{B}}]$  where

$$\bar{\pi}_i = \frac{P(\mathbf{O}, q_0 = i|\lambda)}{P(\mathbf{O}|\lambda)}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^T P(\mathbf{O}, q_{t-1} = i, q_t = j | \lambda)}{\sum_{i=1}^N P(\mathbf{O}, q_{t-1} = i | \lambda)}$$

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T P(\mathbf{O}, q_t = j | \lambda) \delta(\mathbf{o}_t, \mathbf{v}_k)}{\sum_{i=1}^N P(\mathbf{O}, q_t = i | \lambda)}$$

where we have defined

$$\delta(\mathbf{o}_t, \mathbf{v}_k) = \begin{cases} 1 & \text{if } \mathbf{o}_t = \mathbf{v}_k \\ 0 & \text{otherwise.} \end{cases}$$

Using the definitions of the forward variable,  $\alpha_t(i) = P(\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_t = i | \lambda)$  and the backward variable,  $\beta_t(i) = P(\mathbf{o}_{t+1}, \dots, \mathbf{o}_T | q_t = i, \lambda)$ , the reestimation transformations can be easily calculated as

$$P(\mathbf{O}, q_t = i | \lambda) = \alpha_t(i) \beta_t(i)$$

$$P(\mathbf{O} | \lambda) = \sum_{i=1}^N \alpha_t(i) \beta_t(i) = \sum_{i=1}^N \alpha_T(i)$$

$$P(\mathbf{O}, q_{t-1} = i, q_t = j | \lambda) = \alpha_{t-1}(i) a_{ij} b_j(\mathbf{o}_t) \beta_t(j)$$

giving

$$\bar{\pi}_i = \frac{\alpha_0(i) \beta_0(i)}{\sum_{j=1}^N \alpha_T(j)} = \gamma_0(i)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^T \alpha_{t-1}(i) a_{ij} b_j(\mathbf{o}_t) \beta_t(j)}{\sum_{i=1}^N \alpha_{t-1}(i) \beta_{t-1}(i)} = \frac{\sum_{t=1}^T \xi_{t-1}(i, j)}{\sum_{i=1}^N \gamma_{t-1}(i)}$$

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \alpha_t(i) \beta_t(i) \delta(\mathbf{o}_t, \mathbf{v}_k)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)} = \frac{\sum_{\substack{t=1 \\ \mathbf{o}_t = \mathbf{v}_k}}^T \gamma_t(i)}{\sum_{i=1}^N \gamma_t(i)}$$

which are the formulas given in Eqs. (6.40a)–(6.40c).

#### 6.4.4 Notes on the Reestimation Procedure

The reestimation formulas can be readily interpreted as an implementation of the EM algorithm of statistics [17] in which the E (expectation) step is the calculation of the auxiliary function  $Q(\lambda', \lambda)$ , (which is the expectation of  $\log P(\mathbf{O}, \mathbf{q} | \lambda)$ ), and the M (maximization) step is the maximization of  $Q(\lambda', \lambda)$  over  $\lambda$  to obtain  $\bar{\lambda}$ . Thus the Baum-Welch reestimation equations are essentially identical to the EM steps for this particular problem.

An important property of the reestimation procedure is that the stochastic constraints of the HMM parameters, namely

$$\sum_{i=1}^N \bar{\pi}_i = 1 \quad (6.43a)$$

$$\sum_{j=1}^N \bar{a}_{ij} = 1, \quad 1 \leq i \leq N \quad (6.43b)$$

$$\sum_{k=1}^M \bar{b}_j(k) = 1, \quad 1 \leq j \leq N \quad (6.43c)$$

are automatically incorporated at each iteration. By looking at the parameter estimation problem as a constrained optimization of  $P(\mathbf{O} | \lambda)$  (subject to the constraints of Eq. (6.43)), we can formulate the solution procedure by use of the techniques of variational calculus to maximize  $P$  (we use the notation  $P = P(\mathbf{O} | \lambda)$  as shorthand in this section). Based on a standard Lagrange optimization setup using Lagrange multipliers, it can readily be shown that  $P$  is maximized when the following conditions are met:

$$\bar{\pi}_i = \frac{\pi_i \frac{\partial P}{\partial \pi_i}}{\sum_{k=1}^N \pi_k \frac{\partial P}{\partial \pi_k}} \quad (6.44a)$$

$$\bar{a}_{ij} = \frac{a_{ij} \frac{\partial P}{\partial a_{ij}}}{\sum_{k=1}^N a_{ik} \frac{\partial P}{\partial a_{ik}}} \quad (6.44b)$$

$$\bar{b}_j(k) = \frac{b_j(k) \frac{\partial P}{\partial b_j(k)}}{\sum_{\ell=1}^M b_j(\ell) \frac{\partial P}{\partial b_j(\ell)}} \quad (6.44c)$$

By appropriate manipulation of Eq. (6.44), the right-hand sides of each equation can be readily shown to be *identical* to the right-hand sides of each part of Eqs. (6.40a)–(6.40c), thereby showing that the reestimation formulas are indeed exactly correct at critical points of  $P$ . In fact, the form of Eq. (6.44) is essentially that of a reestimation formula in which

the left-hand side is the reestimate and the right-hand side is computed using the current values of the variables.

Finally, we note that since the entire problem can be set up as an optimization problem, standard gradient techniques can be used to solve for "optimal" values of the model parameters. Such procedures have been tried and have been shown to yield solutions comparable to those of the standard reestimation procedures [18]. One critical shortcoming of standard gradient technique, as applied to the maximization of  $P(O|\lambda)$ , is that the descent algorithms, which are critically dependent on taking a small step in the direction of the gradient, often do not produce *monotonic* improvement in the likelihood as the Baum-Welch reestimation is guaranteed by Eq. (6.42) to do.

## 6.5 TYPES OF HMMS

One way to classify types of HMMs is by the structure of the transition matrix,  $A$ , of the Markov chain. Until now, we have only considered the special case of ergodic or fully connected HMMs in which every state of the model could be reached (in a single step) from every other state of the model. (Strictly speaking, an ergodic model has the property that every state can be reached from every other state in a finite but aperiodic number of steps.) As shown in Figure 6.8(a), for an  $N = 4$  state model, this type of model has the property that every  $a_{ij}$  coefficient is positive. Hence for the example of Figure 6.8(a) we have

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

For some applications, particularly those to be discussed later in this chapter, other types of HMMs have been found to account for observed properties of the signal being modeled better than the standard ergodic model. One such model is shown in Figure 6.8(b). This model is called a left-right model or a Bakis model ([11], [10]) because the underlying state sequence associated with the model has the property that, as time increases, the state index increases (or stays the same)—that is, the system states proceed from left to right. Clearly the left-right type of HMM has the desirable property that it can readily model signals whose properties change over time in a successive manner—e.g., speech. The fundamental property of all left-right HMMs is that the state-transition coefficients have the property

$$a_{ij} = 0, \quad j < i \quad (6.45)$$

that is, no transitions are allowed to states whose indices are lower than that of the current state. Furthermore, the initial state probabilities have the property

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \quad (6.46)$$

because the state sequence must begin in state 1 (and end in state  $N$ ). Often, with left-right

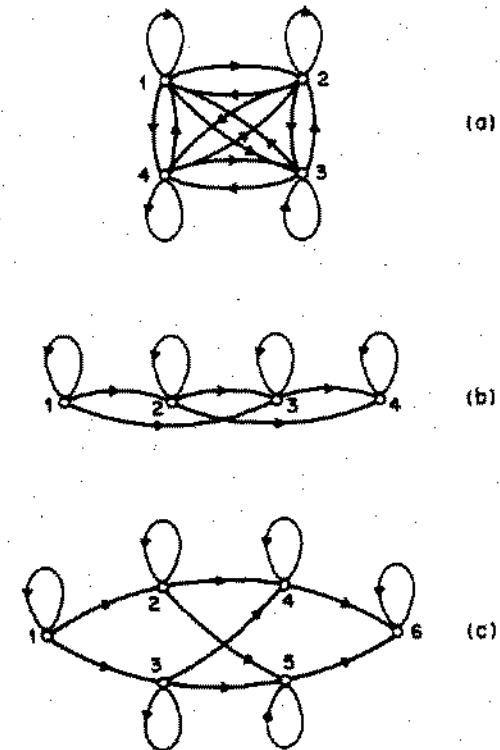


Figure 6.8 Illustration of three distinct types of HMMs. (a) A 4-state ergodic model. (b) A 4-state left-right model. (c) A 6-state parallel path left-right model.

models, additional constraints are placed on the state-transition coefficients to make sure that large changes in state indices do not occur; hence a constraint of the form

$$a_{ij} = 0, \quad j > i + \Delta i \quad (6.47)$$

is often used. In particular, for the example of Figure 6.8(b), the value of  $\Delta i$  is 2; that is, no jumps of more than two states are allowed. The form of the state-transition matrix for the example of Figure 6.8(b) is thus

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

It should be clear that, for the last state in a left-right model, the state-transition coefficients are specified as

$$a_{NN} = 1 \quad (6.48a)$$

$$a_{Ni} = 0, \quad i < N. \quad (6.48b)$$

Besides the above fully connected and left-right models, there are many other possible variations and combinations. By way of example, Figure 6.8(c) shows a cross-coupled connection of two parallel left-right HMMs. Strictly speaking, this model is a left-right model (it obeys all the  $a_{ij}$  constraints); however, it has certain flexibility not present in a strict left-right model (i.e., one without parallel paths).

It should be clear that the imposition of the constraints of the left-right model, or those of the constrained jump model, essentially have no effect on the reestimation procedure. This is the case because any HMM parameter set to zero initially will remain at zero throughout the reestimation procedure (see Eq. (6.44)).

## 6.6 CONTINUOUS OBSERVATION DENSITIES IN HMMS

All of our discussion to this point has considered only when the observations were characterized as discrete symbols chosen from a finite alphabet, and therefore we could use a discrete probability density within each state of this model ([19]–[21]). The problem with this approach, at least for some applications, is that the observations are often continuous signals (or vectors). Although it is possible to convert such continuous signal representations into a sequence of discrete symbols via vector quantization codebooks and other methods, there might be serious degradation associated with such discretization of the continuous signal. Hence it would be advantageous to be able to use HMMs with continuous observation densities to model continuous signal representations directly.

To use a continuous observation density, some restrictions must be placed on the form of the model probability density function (pdf) to ensure that the parameters of the pdf can be reestimated in a consistent way. The most general representation of the pdf, for which a reestimation procedure has been formulated, is a finite mixture of the form

$$b_j(\mathbf{o}) = \sum_{k=1}^M c_{jk} \mathcal{N}(\mathbf{o}; \mu_{jk}, \mathbf{U}_{jk}), \quad 1 \leq j \leq N \quad (6.49)$$

where  $\mathbf{o}$  is the observation vector being modeled,  $c_{jk}$  is the mixture coefficient for the  $k$ th mixture in state  $j$  and  $\mathcal{N}$  is any log-concave or elliptically symmetric density [18] (e.g., Gaussian). Without loss of generality, we assume that  $\mathcal{N}$  is Gaussian in Eq. (6.49) with mean vector  $\mu_{jk}$  and covariance matrix  $\mathbf{U}_{jk}$  for the  $k$ th mixture component in state  $j$ . The mixture gains  $c_{jk}$  satisfy the stochastic constraint

$$\sum_{k=1}^M c_{jk} = 1, \quad 1 \leq j \leq N \quad (6.50a)$$

$$c_{jk} \geq 0, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M \quad (6.50b)$$

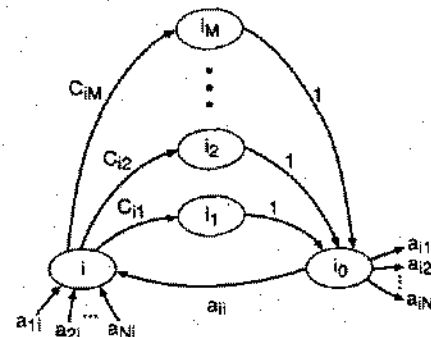


Figure 6.9 Equivalence of a state with a mixture density to a multistate single-density distribution (after Juang et al. [21]).

so that the pdf is properly normalized, i.e.,

$$\int_{-\infty}^{\infty} b_j(\mathbf{o}) d\mathbf{o} = 1, \quad 1 \leq j \leq N. \quad (6.51)$$

The pdf of Eq. (6.49) can be used to approximate, arbitrarily closely, any finite, continuous-density function. Hence it can be applied to a wide range of problems.

It has been shown that an HMM state with a mixture density is equivalent to a multistate single-mixture density model in the following way [21]. Consider a state  $i$  with an  $M$ -mixture Gaussian density. Because the mixture gain coefficients sum up to 1, they define a set of transition coefficients to substates  $i_1$  (with transition probability  $c_{i1}$ ),  $i_2$  (with transition probability  $c_{i2}$ ) through  $i_M$  (with transition probability  $c_{iM}$ ). Within each substate  $i_k$ , there is a single mixture with mean  $\mu_{ik}$  and variance  $\mathbf{U}_{ik}$  (see Figure 6.9 for a graphical interpretation). Each substate makes a transition to a wait state  $i_0$  with probability 1. The distribution of the composite set of substates (each with a single density) is mathematically equivalent to the composite mixture density within a single state.

It can be shown that the reestimation formulas for the coefficients of the mixture density, i.e.,  $c_{jk}$ ,  $\mu_{jk}$ , and  $\mathbf{U}_{jk}$ , are of the form

$$\bar{c}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(j, k)} \quad (6.52)$$

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) \cdot \mathbf{o}_t}{\sum_{t=1}^T \gamma_t(j, k)} \quad (6.53)$$