12

Detection of Markov Chains with Known Parameters

12.1 Introduction

Although the Gaussian signal model used in the second part of the book is applicable to a wide class of radar and sonar detection problems, it has become increasingly ill-adapted to the study of modern digital communications systems. Specifically, it assumes that over each observation interval, the received signal is free of intersymbol interference (ISI). Although for a fixed channel and a matched filter receiver, ISI can be theoretically avoided by proper design of a signaling pulse, many channels such as wireless channels change rapidly, so for such channels, ISI is unavoidable and an equalizer needs to be implemented. Another implicit assumption of the detection problems we have considered up to this point is that channel coding and modulation are two separate operations, so that signal demodulation and decoding can be handled separately. While this viewpoint was appropriate in the 1960s and 1970s when signal detection was first developed as a coherent theory, subsequent developments have made this picture obsolete. Specifically, the introduction of trellis-coded modulation [1] in order to transmit data reliably over bandlimited channels has made it impossible to keep the demodulation and decoding operations separate. More recently, turbo equalizers [2, 3] have also demonstrated the benefit of cooperation between the equalization and decoding blocks of communication receivers. For all the above mentioned examples, the received signal can no longer be modelled as a deterministic signal (possibly with unknown parameters) in WGN. Instead, it needs to be viewed as a function of the state of a Markov chain or of its state transitions, possibly with unknown parameters, observed in WGN. We refer the reader to [4, 5] for recent presentations of detection problems of digital communications from this perspective.

In addition to digital communications, Markov chain (MC) detection problems arise in the study of speech recognition and segmentation problems [6, 7]. So in this chapter we examine detection problems where the observed signal is a function of the states or transitions of a Markov chain. The chapter is

B.C. Levy, *Principles of Signal Detection and Parameter Estimation*,
DOI: 10.1007/978-0-387-76544-0_12,

organized as follows: In Section 12.2, we examine the binary hypothesis testing problem for exactly observed Markov chains. It is found that the empirical probability distribution for two successive states of the Markov chain forms a sufficient statistic, and by analogy with the minimum Euclidean distance rule for testing Gaussian signals in WGN, the optimum test can be expressed as a mimimum discrimination function test, where the discrimination function is a measure of proximity of joint probability distributions for pairs of discrete random variables. The asymptotic rate of decay of the two types of errors for the optimum test as the number of observations increases is evaluated by using the theory of large deviations. The detection of partially observed Markov chains is examined in Section 12.3. It turns out that, given noisy observations of an MC over a fixed time interval, we can consider either an MAP sequence detection problem or a pointwise MAP detection problem. The MAP sequence detection problem seeks to maximize the a-posteriori probability of the entire state trajectory, whereas pointwise MAP detection maximizes the a-posteriori probability of the state value at each instant in time. The calculation of the MAP sequence is performed by the Viterbi algorithm, which is an instance of the method of dynamic programming. The evaluation of the pointwise MAP state probabilities is accomplished recursively by the forward-backward algorithm. Although the Viterbi algorithm is slightly more efficient than the forward-backward algorithm, the latter algorithm evaluates the a-posteriori probability of each state at each time. This soft information makes it ideally suited for use in iterative/turbo decoding algorithms of the type discussed in [4, 5]. To illustrate the detection of partially observed Markov chains, Section 12.4 examines the MAP equalization of ISI signals in noise.

12.2 Detection of Completely Observed Markov Chains

12.2.1 Notation and Background

For implementation reasons, the Markov chains arising in engineering systems have usually a finite number of states. So throughout this chapter, we restrict our attention to DT Markov chains $X(t)$ with $t \geq 1$ where the state $X(t)$ takes values in a finite set $\mathcal{S} = \{1, 2, \ldots, k, \ldots, K\}$. We also assume that the Markov chain is homogeneous, i.e., time-invariant. Then the Markov chain is specified by the row vector

$$\boldsymbol{\pi}(1) = \big[\, \pi_1(1)\ \pi_2(1) \ldots \pi_k(1) \ldots \pi_K(1) \,\big]$$

representing its initial probability distribution

$$\pi_k(1) = P[X(1) = k]\,, \tag{12.1}$$

and by its one-step transition probability matrix $\mathbf{P}$. This matrix has dimension $K \times K$ and its (k, ℓ)-th element is given by

$$p_{k\ell} = P[X(t+1) = \ell | X(t) = k] \,. \tag{12.2}$$

The set of allowed state transitions is denoted as

$$\mathcal{T} = \{(k, \ell) \in \mathcal{S}^2 \,:\, p_{k\ell} > 0\} \,. \tag{12.3}$$

To this set, we can associate a directed graph $\mathcal{G}$ which is called the transition diagram of the Markov chain (MC). This graph has for vertices the states of $\mathcal{S}$ and for edges the elements of $\mathcal{T}$, i.e., an edge goes from vertex k to vertex ℓ if $p_{k\ell} > 0$. The graph $\mathcal{G}$ is used to determine several important properties of the MC. Specifically, the MC is irreducible if there exists a directed path from any vertex k to any other vertex ℓ. In other words, if an MC is irreducible, all its states are connected to each other. In this case, independently of how the MC is initialized, all the states are visited an infinite number of times, and the MC has a single recurrent class. The period d of an MC is the greatest common divisor of the lengths of the cycles of its graph. The MC is said to be aperiodic if $d = 1$. To fix ideas, we consider a simple example.

Example 12.1: Transition diagram of a Markov chain

Consider a Markov chain with one-step transition probability matrix

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \end{bmatrix} .$$

This MC can be used to model the phase states of a continuous-phase frequency-shift-keyed (CPFSK) signal with modulation index $h = 1/2$ [8, p. 193]. Its state transition diagram is shown in Fig. 12.1.

By inspection it can be verified that all states are connected, so the MC is irreducible. The MC has several elementary cycles of length 2, such as the cycle 1-3-1. Recall that a cycle is called elementary if only the first and last

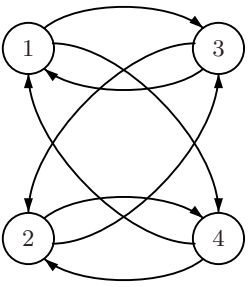

Fig. 12.1. Transition diagram $\mathcal{G}$ of the MC with transition matrix $\mathbf{P}$.

vertex are repeated. Similarly, it has several elementary cycles of length 4, such as 1-4-2-3-1. Accordingly, the period of the MC is $d = 2$. Note that this is easy to see both from the block structure of $\mathbf{P}$ and from the transition diagram $\mathcal{G}$. Specifically, the set $\mathcal{S} = \{1, 2, 3, 4\}$ of states can be partitioned into two groups: $\mathcal{S}_1 = \{1, 2\}$ and $\mathcal{S}_2 = \{3, 4\}$, and if the state $X(t)$ belongs to one of the two groups at time t, it jumps to a member of the other group at time $t + 1$. In the block partition

$$\mathbf{P} = \begin{bmatrix} \mathbf{0} & \mathbf{P}_T \\ \mathbf{P}_T & \mathbf{0} \end{bmatrix}$$

of $\mathbf{P}$, with

$$\mathbf{P}_T = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} ,$$

the zero block diagonal matrices clearly indicate that there is no transition internal to either $\mathcal{S}_1$ or $\mathcal{S}_2$. Similarly, the graph $\mathcal{G}$ has a bipartite structure, where all edges originating from states in $\mathcal{S}_1$ terminate in $\mathcal{S}_2$, and vice versa. □

In this section it is assumed that the MCs we consider are irreducible and aperiodic, so that all states form a single recurrent class. Then, as a consequence of the Perron-Frobenius theorem for matrices with nonnegative entries [9, Chap. 8], [10, Sec. 4.4], $\lambda = 1$ is the eigenvalue of $\mathbf{P}$ with the largest magnitude, it has multiplicity one, its corresponding right eigenvector is

$$\mathbf{e} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T \tag{12.4}$$

up to a scale factor, and when the sum of its entries is normalized to one, its corresponding left eigenvector is a probability distrbution

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_1 & \dots & \pi_k & \dots & \pi_K \end{bmatrix} \tag{12.5}$$

with $\pi_k > 0$ for all $1 \le k \le K$. This implies that the t-step probability transition matrix P^t satisfies

$$\lim_{t \to \infty} P^t = \mathbf{e}\boldsymbol{\pi} \, . \tag{12.6}$$

Let also

$$\boldsymbol{\pi}(t) = \begin{bmatrix} \pi_1(t) & \dots & \pi_k(t) & \dots & \pi_K(t) \end{bmatrix} \tag{12.7}$$

be the row vector representing the state probability distribution

$$\pi_k(t) = P[X(t) = k]$$

of the Markov chain at time t. Since it obeys

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(1)\mathbf{P}^{t-1} \, , \tag{12.8}$$

we conclude from (12.6) that

$$\lim_{t\to\infty} \boldsymbol{\pi}(t) = \boldsymbol{\pi} \,. \tag{12.9}$$

In other words, $\boldsymbol{\pi}$ is the steady-state probability distribution of the Markov chain, independently of the choice of initial probability distribution $\boldsymbol{\pi}(1)$.

Next, suppose that we observe the Markov chain $X(t)$ over a time interval $1 \le t \le N+1$. The observed state at time t is $x_t \in \mathcal{S}$. Let $N_{k\ell}(\mathbf{x})$ denote the number of transitions from state k to state ℓ observed in the sequence

$$\mathbf{x} = (x_1, \ldots, x_t, \ldots, x_{N+1}) \,.$$

Then the number

$$N_k(\mathbf{x}) = \sum_{\ell=1}^{K} N_{k\ell}(\mathbf{x}) \tag{12.10}$$

of transitions out of state k corresponds also to the number of times the state k appears in the truncated state sequence

$$\mathbf{x}_T = (x_1, \ldots, x_t, \ldots, x_N)$$

obtained by removing the last observation from the sequence $\mathbf{x}$. We define the empirical successive state probability distribution of the Markov chain as the function

$$q_{k\ell}(\mathbf{x}) = \frac{N_{k\ell}(\mathbf{x})}{N} \tag{12.11}$$

defined for $1 \le k, \ell \le K$. By using the strong law of large numbers, it is easy to verify that as $N \to \infty$

$$q_{k\ell}(\mathbf{X}) \xrightarrow{\text{a.s.}} \pi_k p_{k\ell} \tag{12.12}$$

and

$$\sum_{\ell=1}^{K} q_{k\,\ell}(\mathbf{X}) = \frac{N_k(\mathbf{X})}{N} \xrightarrow{\text{a.s.}} \pi_k \,, \tag{12.13}$$

so both the one-step transition probability matrix $\mathbf{P}$ and the steady-state probability distribution $\boldsymbol{\pi}$ can be recovered asymptotically from the empirical distribution q.

The empirical probability distribution q belongs to the class $\mathcal{Q}$ of probability mass distributions over $\mathcal{S}^2$. This class is formed by functions $\alpha_{k\ell}$ such that $\alpha_{k\ell} \ge 0$ and

$$\sum_{k\ell} \alpha_{k\ell} = 1.$$

For a distribution α of $\mathcal{Q}$, we define the marginal distribution with respect to the first variable as

$$\alpha_k = \sum_{\ell} \alpha_{k\ell} \,, \tag{12.14}$$

and the conditional distribution of the second variable given the first as

$$\alpha_{\ell|\,k} = \alpha_{k\ell}/\alpha_k. \tag{12.15}$$

Equivalently, if we consider the $K \times K$ matrix with entries $\alpha_{k,\ell}$, α_k represents the sum of the entries of the k-th row of this matrix and $\alpha_{\cdot|\,k}$ is the row obtained after normalizing the k-th row by α_k. Thus $\alpha_{\cdot|\,k}$ is itself a probability distribution, i.e.,

$$\sum_{\ell=1}^{K} \alpha_{\ell|\,k} = 1\,.$$

Then, given two distributions α and β of $\mathcal{Q}$, the information discrimination measure between α and β is defined as

$$\begin{aligned} J(\alpha,\beta) &= \sum_{k\,\ell} \alpha_{k\,\ell} \ln\left(\frac{\alpha_{\ell|\,k}}{\beta_{\ell|\,k}}\right) \\ &= \sum_{k=1}^{K} \alpha_k D(\alpha_{\cdot|\,k}|\beta_{\cdot|\,k}). \end{aligned} \tag{12.16}$$

In other words, $J(\alpha,\beta)$ is a weighted sum of the Kullback-Leibler divergences of the normalized rows of the matrices representing α and β, where the weights are given by the marginal distribution of α. Since $\alpha_k \geq 0$ and the KL divergence is nonnegative, we deduce that $J(\alpha,\beta) \geq 0$. It is of particular interest to examine the situation where α and β are the steady-state joint probability distributions for two successive states $X(t)$ and $X(t+1)$ of Markov chains with one-step transition probability matrices $\mathbf{P}^0$ and $\mathbf{P}^1$. We assume both chains are irreducible and aperiodic with the same set $\mathcal{T}$ of allowed transitions, and thus the same transition diagram $\mathcal{G}$. Let $\boldsymbol{\pi}^0$ and $\boldsymbol{\pi}^1$ be the steady-state marginal probability distributions of state $X(t)$ corresponding to $\mathbf{P}^0$ and $\mathbf{P}^1$. Note that, since they are the left eigenvectors of $\mathbf{P}^0$ and $\mathbf{P}^1$ corresponding to $\lambda = 1$, they are uniquely specified by the one-step transition probability matrices. Then the steady state probability distributions of two successive states for the two Markov chains are given by

$$\alpha_{k\ell} = \pi^0_k p^0_{k\ell} \quad , \quad \beta_{k\ell} = \pi^1_k p^1_{k\ell} \tag{12.17}$$

and the information discrimination measure can be expressed as

$$J(\alpha,\beta) = \sum_{k=1}^{K} \pi^0_k D(p^0_{k\cdot}|p^1_{k\cdot})\,. \tag{12.18}$$

Note that the requirement that both chains have the same set $\mathcal{T}$ of allowed transitions ensures that $J(\alpha,\beta)$ is finite. If the allowed transitions $\mathcal{T}^0$ and $\mathcal{T}^1$ of both chains are different, $J(\alpha,\beta) = \infty$. Then, since $\pi^0_k > 0$ for all k, we conclude that $J(\alpha,\beta) = 0$ if and only if $\mathbf{P}^0 = \mathbf{P}^1$, i.e., if and only if the Markov

chains are identical. Thus the function $J(\alpha, \beta)$ is well adapted to the task of discriminating between two irreducible aperiodic Markov chains. Of course, like the Kullback-Leibler divergence and the Itakura-Saito spectral distortion measure considered earlier, it is not a true metric, since it is not symmetric, i.e, $J(\alpha, \beta) \neq J(\beta, \alpha)$, and it does not satisfy the triangle inequality. Since the one-step transition probability matrix of an irreducible aperiodic Markov chain specifies uniquely its steady-state probability distribution, whenever the probability distribution α corresponds to the steady-state distribution of two consecutive states of a Markov chain with one-step transition probability matrix $\mathbf{P}$, by some abuse of notation, we will replace α by $\mathbf{P}$ as an argument of the discrimination function J. For example, if $\alpha = q$ is the empirical probability distribution of two consecutive states, and β is the steady-state probability distribution of two consecutive states specified by $\mathbf{P}$, the discrimination function is denoted as $J(q, \mathbf{P})$. Similarly, if α and β are the steady-state probability distributions (12.17) specified by the Markov chains with transition matrices $\mathbf{P}^0$ and $\mathbf{P}^1$, their discrimination function is denoted as $J(\mathbf{P}^0, \mathbf{P}^1)$.

12.2.2 Binary Hypothesis Testing

Consider now the binary hypothesis testing problem where under hypothesis H_j with $j = 0,\ 1$, we observe $N+1$ consecutive states $Y(t)$ with $1 \leq t \leq N+1$ of a Markov chain with one-step transition probability matrix $\mathbf{P}^j$. We assume that the two Markov chains have K states, are irreducible and aperiodic, and that the set $\mathcal{T}$ of allowed transitions is the same for both chains. The motivation for this assumption is that otherwise, the detection problem is singular. Specifically, since the Markov chain is recurrent under both hypotheses, each possible transition occurs sooner or later. So, by waiting long enough, we are certain that a transition allowed by one chain, and not by the other, will take place, which allows a perfect decision. So we remove this trivial case by assuming that both chains have the same set of allowable transitions. To simplify our analysis, we also assume that the probability distribution $\boldsymbol{\pi}(1)$ of the initial state is the same under both hypotheses.

Then, let $Y(t) = y_t$ with $1 \leq t \leq N+1$ be the observed sequence of states with $y_t \in \mathcal{S}$. The log-likelihood ratio for the binary hypothesis testing problem can be expressed as

$$\ln\big(L(\mathbf{y})\big) = \sum_{t=1}^{N} \ln\left(\frac{p^1_{y_t y_{t+1}}}{p^0_{y_t y_{t+1}}}\right) \tag{12.19}$$

$$= \sum_{(k,\,\ell)\in\mathcal{T}} N_{k\ell}(\mathbf{y}) \ln\left(\frac{p^1_{k\ell}}{p^0_{k\ell}}\right)$$

$$= N\big[J(q(\mathbf{y}), \mathbf{P}^0) - J(q(\mathbf{y}), \mathbf{P}^1)\big]\,, \tag{12.20}$$

where $q(\mathbf{y})$ denotes the empirical probability distribution for two consecutive states specified by the sequence $\mathbf{y}$. From expression (12.20) we see that

$$S_N = J(q(\mathbf{y}), \mathbf{P}^0) - J(q(\mathbf{y}), \mathbf{P}^1) \tag{12.21}$$

is a sufficient statistic for the test, and the LRT takes the form

$$S_N \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_N = \frac{\ln(\tau)}{N} \tag{12.22}$$

where τ denotes the Bayesian threshold specified by (2.19). Note that if the costs C_{ij} are function of the number N of observations, τ will depend on N.

For the case when the costs are symmetric and the two hypotheses are equally likely, the threshold $\gamma_N = 0$, so the test (12.21)–(12.22) takes the geometrically appealing form

$$J(q(\mathbf{y}), \mathbf{P}^0) \underset{H_0}{\overset{H_1}{\gtrless}} J(q(\mathbf{y}), \mathbf{P}^1)\,. \tag{12.23}$$

Thus, hypothesis H_j is selected if the observed empirical distribution q is closest to $\mathbf{P}^j$ in the sense of the discrimination function J. In other words, J plays the same role for Markov chain detection as the Euclidean distance for the detection of known signals in WGN.

12.2.3 Asymptotic Performance

The empirical probability distribution $q_{k\ell}(\mathbf{y})$ has the property that under hypothesis H_j, it converges almost surely to the steady state probability distribution $\pi_k^j p_{k\ell}^j$ for two consecutive states of the Markov chain specified by $\mathbf{P}^j$. This implies that as N tends to infinity

$$S_N \xrightarrow{\text{a.s.}} J(\mathbf{P}^1, \mathbf{P}^0) \tag{12.24}$$

under H_1, and

$$S_N \xrightarrow{\text{a.s.}} -J(\mathbf{P}^0, \mathbf{P}^1) \tag{12.25}$$

under H_0. If we assume that τ is either constant or tends to a finite constant as N tends to infinity, the threshold γ_N in (12.23) tends to zero as N goes to infinity. Assuming that $\mathbf{P}^1 \neq \mathbf{P}^0$, the limits (12.24) and (12.25) indicate that it is possible to make decisions with vanishingly small probability of error as N increases.

The objective of this section is to show that for a test of the form (12.22), where the threshold γ on the right-hand side is a constant independent of N, the probabilities of false alarm and of a miss decay exponentially with the number N of observations. Note that for finite N, the probability of false alarm and of a miss do not admit simple closed form expressions. So the expressions obtained below for large N will provide some rough estimates of the test performance. The approach that we use to analyze the asymptotic behavior of the test is based again on the theory of large deviations. Specifically, observe

that for $N+1$ observations $Y(t)$ with $1 \le t \le N+1$, the sufficient statistic S_N can be written as

$$S_N = \frac{1}{N} \sum_{t=1}^{N} Z_t \tag{12.26}$$

with

$$Z_t = \ln(p^1_{Y(t)Y(t+1)} / p^0_{Y(t)Y(t+1)}) \,. \tag{12.27}$$

In this expression, the random variables Z_t are not independent, so Cramér's theorem is not applicable. However, as will be shown below, the sequence S_N admits an asymptotic log-generating function under H_j for $j = 0,\ 1$, so the Gärtner-Ellis theorem can be employed to evaluate the asymptotic probabilities of false alarm and of a miss of the test as N tends to infinity.

Consider the generating functions

$$G^j_N(u) = E[\exp(uS_N)|H_j] \tag{12.28}$$

for $j = 0,\ 1$. We have

$$G^j_N(Nu) = E\big[\prod_{t=1}^{N} \left(\frac{p^1_{Y(t)Y(t+1)}}{p^0_{Y(t)Y(t+1)}} \right)^u \mid H_j \big] \,. \tag{12.29}$$

Then, at stage t, assume that the expectations with respect to $Y(t+1), \ldots, Y(N+1)$ have already been taken, conditioned on the knowledge of $Y(1), \ldots, Y(t)$. If we now take the expectation with respect to $Y(t)$ conditioned on the knowledge of $Y(1), \ldots, Y(t-1)$, and proceed recursively in this manner until $t = 1$, we obtain

$$G^0_N(Nu) = \boldsymbol{\pi}(1)\mathbf{M}(u)^N \mathbf{e} \tag{12.30}$$

where $\mathbf{M}(u)$ denotes the $K \times K$ matrix with (k,ℓ)-th element

$$m_{k\ell}(u) = (p^1_{k\ell})^u (p^0_{k\ell})^{1-u} \,, \tag{12.31}$$

$\boldsymbol{\pi}(1)$ is the row vector representing the probability distribution of $Y(1)$, and $\mathbf{e}$ is the vector (12.4) whose entries are all equal to one. The matrix $\mathbf{M}(u)$ can be expressed as a Hadamard product [11, Chap. 5]

$$\mathbf{M}(u) = (\mathbf{P}^1)^{(u)} \circ (\mathbf{P}^0)^{(1-u)} \tag{12.32}$$

where $\mathbf{P}^{(v)} = (p^v_{k\ell}, 1 \le k,\ell \le K)$ denotes the v-th Hadamard power of a matrix $\mathbf{P}$. Although $\mathbf{M}(u)$ is not a stochastic matrix, since its row sums are not necessarily equal to one for all u, it is a nonegative matrix since $m_{k\ell}(u) \ge 0$ for all $1 \le k,\ell \le K$. It admits a directed graph $\mathcal{G}$ constructed in the same way as for a stochastic matrix. Its vertices are the elements of $\mathcal{S}$, and a directed edge links vertices k and ℓ if $m_{k\ell}(u) > 0$. Obviously $\mathcal{G}$ does not depend on u and coincides with the transition diagram of $\mathbf{P}^0$ and $\mathbf{P}^1$, which is the same since both chains have the same set of allowed transitions. Since $\mathbf{P}^0$ and $\mathbf{P}^1$

were assumed irreducible and aperiodic, and since the irreducibility and period of a nonnegative matrix depend only on its graph, this implies that the matrix $\mathbf{M}(u)$ is irreducible and aperiodic.

Accordingly, the Perron-Frobenius theorem is applicable to $\mathbf{M}(u)$, so it has a positive real eigenvalue $\lambda(u)$ which is greater than the magnitude of all the other eigenvalues. The multiplicity of $\lambda(u)$ is one, and if we consider the left and right eigenvectors

$$\begin{aligned}\mathbf{a}^T(u) &= \big[\, a_1(u)\; a_2(u)\; \ldots\; a_K(u) \,\big] \\ \mathbf{b}^T(u) &= \big[\, b_1(u)\; b_2(u)\; \ldots\; b_K(u) \,\big]\end{aligned}$$

of $\mathbf{M}(u)$ corresponding to $\lambda(u)$, i.e.,

$$\begin{aligned}\mathbf{a}^T(u)\mathbf{M}(u) &= \lambda(u)\mathbf{a}^T(u) \\ \mathbf{M}(u)\mathbf{b}(u) &= \lambda(u)\mathbf{b}(u)\end{aligned} \tag{12.33}$$

all the entries $a_k(u)$ and $b_k(u)$ are real and positive for $1 \le k \le K$. Since the eigenvectors $\mathbf{a}(u)$ and $\mathbf{b}(u)$ can be scaled arbitrarily, we can always ensure that

$$\mathbf{a}^T(u)\mathbf{b}(u) = \sum_{k=1}^{K} a_k(u) b_k(u) = 1\,. \tag{12.34}$$

In addition, since $\mathbf{M}(0) = \mathbf{P}^0$ and $\mathbf{M}(1) = \mathbf{P}^1$, we have

$$\lambda(0) = \lambda(1) = 1\,, \tag{12.35}$$

and we can always select

$$\mathbf{a}^T(0) = \boldsymbol{\pi}^0 \quad , \mathbf{a}^T(1) = \boldsymbol{\pi}^1 \tag{12.36}$$

$$\mathbf{b}(0) = \mathbf{b}(1) = \mathbf{e} \tag{12.37}$$

where $\boldsymbol{\pi}^j$ denotes the steady-state probability distribution of the Markov chain with transition matrix $\mathbf{P}^j$ for $j = 0,\ 1$.

Then as $N \to \infty$, we have

$$\frac{G_N^0(Nu)}{\lambda^N(u)} \to \boldsymbol{\pi}(1)\mathbf{b}(u)\mathbf{a}^T(u)\mathbf{e} \tag{12.38}$$

where the inner products $\boldsymbol{\pi}(1)\mathbf{b}(u)$ and $\mathbf{a}^T(u)\mathbf{e}$ are both nonzero. Accordingly, under H_0 the sequence S_N admits the asymptotic log-generating function

$$\Lambda_0(u) \stackrel{\triangle}{=} \lim_{N\to\infty} \frac{1}{N} \ln G_N^0(Nu) = \ln \lambda(u)\,, \tag{12.39}$$

i.e., the log-generating function is the logarithm of the spectral radius of $\mathbf{M}(u)$. Similarly, since the generating function of S_N under H_1 can be expressed as

$$G_N^1(u) = G_N^0(u+1)\,,$$

we find that under H_1, S_N admits the log-generating function

$$\Lambda_1(u) \stackrel{\triangle}{=} \lim_{N\to\infty} \frac{1}{N} \ln G_N^1(Nu) = \ln \lambda(u+1)\,. \tag{12.40}$$

To verify that the function $\ln \lambda(u)$ is convex, we employ the following property of Hadamard products of nonegative matrices. Let $\mathbf{M}_1$ and $\mathbf{M}_2$ be two $K \times K$ matrices with nonnegative entries. It is shown in [11, p. 361–362] that for $0 \le \alpha \le 1$

$$\rho(\mathbf{M}_1^{(\alpha)} \circ \mathbf{M}_2^{(1-\alpha)}) \le \rho(\mathbf{M}_1)^\alpha \rho(\mathbf{M}_2)^{1-\alpha} \tag{12.41}$$

where $\rho(\mathbf{M})$ denotes the spectral radius of an arbitrary matrix $\mathbf{M}$. Then, for $u_1,\ u_2 \in \mathbb{R}$ and $0 \le \alpha \le 1$, by observing that

$$\mathbf{M}(\alpha u_1 + (1-\alpha)u_2) = \mathbf{M}(u_1)^{(\alpha)} \circ \mathbf{M}(u_2)^{(1-\alpha)} \tag{12.42}$$

we deduce

$$\begin{aligned} \lambda(\alpha u_1 + (1-\alpha)u_2) &= \rho(\mathbf{M}(\alpha u_1 + (1-\alpha)u_2) \\ &\le \rho(\mathbf{M}(u_1))^\alpha \rho(\mathbf{M}(u_2))^{1-\alpha} = \lambda(u_1)^\alpha \lambda(u_2)^{1-\alpha}\,, \end{aligned} \tag{12.43}$$

so $\ln \lambda(u)$ is convex.

By multiplying $\mathbf{M}(u)$ on the left by $\mathbf{a}^T(u)$ and on the right by $\mathbf{b}(u)$, and taking into account the normalization (12.34) we obtain the identity

$$\lambda(u) = \mathbf{a}^T(u)\mathbf{M}(u)\mathbf{b}(u)\,. \tag{12.44}$$

Differentiating with respect to u gives

$$\frac{d\lambda}{du} = \mathbf{a}^T(u)\frac{d\mathbf{M}}{du}(u)\mathbf{b}(u) = \sum_{k\ell} a_k(u)(p_{k\ell}^1)^u (p_{k\ell}^0)^{1-u} b_\ell(u) \ln\left(\frac{p_{k\,\ell}^1}{p_{k\,\ell}^0}\right), \tag{12.45}$$

where we have used the fact that $\mathbf{a}^T(u)$ and $\mathbf{b}(u)$ are left and right eigenvectors of $\mathbf{M}(u)$, as well as

$$\frac{d}{du}\mathbf{a}^T(u)\mathbf{b}(u) = 0\,.$$

Then, setting $u = 0$ and $u = 1$ inside (12.45) and taking into account (12.35)–(12.37), we find

$$\begin{aligned} \frac{d}{du}\Lambda_0(0) &= \frac{d\lambda}{du}(0) = -J(\mathbf{P}^0, \mathbf{P}^1) \\ \frac{d}{du}\Lambda_0(1) &= \frac{d\lambda}{du}(1) = J(\mathbf{P}^1, \mathbf{P}^0)\,. \end{aligned} \tag{12.46}$$

Accordingly, the asymptotic log-generating function $\Lambda_0(u)$ has a graph of the form shown in Fig. 10.4, where the only difference is that the slopes e_0 and e_1 at $u = 0$ and $u = 1$ are given now by $e_0 = -J(\mathbf{P}^0, \mathbf{P}^1)$ and $e_1 = J(\mathbf{P}^1, \mathbf{P}^0)$.

Consider now the Legendre transform

$$I_j(z) = \sup_{u \in \mathbb{R}} (zu - \Lambda_j(u)) \tag{12.47}$$

for $j = 0,\ 1$. The identity

$$\Lambda_1(u) = \Lambda_0(u+1) \tag{12.48}$$

implies

$$I_1(z) = I_0(z) - z\,. \tag{12.49}$$

Then, let $P_F(N) = P[S_N \geq \gamma | H_0]$ and $P_M(N) = P[S_N < \gamma | H_1]$ denote respectively the probability of false alarm, and the probability of a miss, for the test (12.22) based on the observations $Y(t)$ for $1 \leq t \leq N+1$. By applying the Gärtner-Ellis theorem we obtain

$$\begin{aligned} \lim_{N\to\infty} \frac{1}{N} \ln P_F(N) &= -I_0(\gamma) \\ \lim_{N\to\infty} \frac{1}{N} \ln P_M(N) &= -I_1(\gamma) = -[I_0(\gamma) - \gamma]\,. \end{aligned} \tag{12.50}$$

We can again consider several special cases for the threshold γ. For an NP test of type I, the probability P_M of a miss is minimized while keeping the probability P_F of false alarm below a bound α. Since we know that under H_0 the test satistic S_N converges almost surely to e_0, to guarantee that the probability of false alarm stays below α, we can select as threshold

$$\gamma_{\text{NP}}^{I} = e_0 + \epsilon$$

with $\epsilon > 0$ arbitrarily small. In this case $I_0(e_0) = 0$, since the line with slope e_0 is tangent to $\Lambda_0(u)$ at the origin. In the light of (12.50), this implies $I_1(e_0) = -e_0$, so for an NP test of type I, the probability of a miss decays exponentially according to

$$P_M(N) = K_M(N) \exp(-J(\mathbf{P}^1, \mathbf{P}^0)N) \tag{12.51}$$

where the constant $K_M(N)$ varies with N at a sub-exponential rate, i.e.,

$$\lim_{N\to\infty} \frac{1}{N} \ln(K_M(N)) = 0\,.$$

Similarly, for an NP test of type II, the probability of false alarm P_F is minimized while requiring that the probability P_M of a miss should stay below a constant β. Since S_N converges almost surely to $e_1 = J(\mathbf{P}^1, \mathbf{P}^0)$ under H_1, the test threshold can be selected as

$$\gamma_{\text{NP}}^{II} = e_1 - \epsilon$$

with $\epsilon > 0$ arbitrary small. In this case, $I_1(e_1) = 0$ and $I_0(e_1) = e_1$, so for an NP test of type II, the probability of false alarm decays exponentially according to

$$P_F(N) = K_F(N) \exp(-J(\mathbf{P}^0, \mathbf{P}^1)N) \tag{12.52}$$

where the constant $K_F(N)$ varies with N at a sub-exponential rate.

For a test with threshold γ, identities (12.50) imply that for large N

$$\begin{aligned} P_F(N) &\approx K_F(N) \exp(-I_0(\gamma)N) \\ P_M(N) &\approx K_M(N) \exp(-I_1(\gamma)N) \end{aligned} \tag{12.53}$$

where the coefficients $K_F(N)$ and $K_M(N)$ vary with N at a sub-exponential rate. If π_0 denotes the a-priori probability of H_0, the probability of error for the test (12.22) can be expressed as

$$P_E(N) \approx K_F(N)\pi_0 \exp(-I_0(\gamma)N) + K_M(N)(1-\pi_0) \exp(-I_1(\gamma)N) \tag{12.54}$$

The threshold γ that maximizes the overall rate of decay of $P_E(N)$ is again achieved by equalizing the rates of decay for the two types of errors and setting $\gamma = 0$. With this choice, the rate of decay of the probability of error is given by

$$I_0(0) = -\min_{0 \le u \le 1} \Lambda_0(u) = -\min_{0 \le u \le 1} \ln \lambda(u)\,. \tag{12.55}$$

Although the Legendre transforms $I_0(z)$ and $I_1(z)$ cannot usually be evaluated in closed form, they admit an interesting geometric interpretation in terms of the information discrimination measure J. For $0 \le u \le 1$, consider the family of Markov chains with state space $\mathcal{S}$ and transition diagram $\mathcal{G}$ specified by the one-step transition matrix

$$\mathbf{P}(u) = \frac{1}{\lambda(u)} \mathbf{T}^{-1}(u) \mathbf{M}(u) \mathbf{T}(u)\,, \tag{12.56}$$

where

$$\mathbf{T}(u) \stackrel{\triangle}{=} \text{diag}\,\{b_k(u),\, 1 \le k \le K\} \tag{12.57}$$

denotes the diagonal similarity transformation specified by the entries of the right eigenvector of $\mathbf{M}(u)$ corresponding to $\lambda(u)$. The (k,ℓ)-th element of $\mathbf{P}(u)$ is given by

$$p_{k\ell}(u) = \frac{1}{\lambda(u)} (p^1_{k\ell})^u (p^0_{k\ell})^{1-u} \frac{b_\ell(u)}{b_k(u)}\,. \tag{12.58}$$

Clearly $\mathbf{P}(u)$ is a stochastic matrix with the same graph $\mathcal{G}$ as $\mathbf{P}^0$, $\mathbf{P}^1$ and $\mathbf{M}(u)$. Its right and left eigenvectors corresponding to $\lambda = 1$ are given respectively by $\mathbf{e}$ and

$$\begin{aligned} \boldsymbol{\pi}(u) &= \mathbf{a}^T(u)\mathbf{T}(u) \\ &= \left[\, a_1(u)b_1(u)\; a_2(u)b_2(u)\; \ldots\; a_K(u)b_K(u) \,\right]\,. \end{aligned} \tag{12.59}$$

The family $\mathbf{P}(u)$ forms the geodesic linking the Markov chain with transition matrix $\mathbf{P}^0$ to the one with transition matrix $\mathbf{P}^1$ as u varies from 0 to 1. By direct evaluation, we find

$$\begin{aligned} J(\mathbf{P}(u), \mathbf{P}^0) &= \sum_{k\,\ell} a_k(u) b_k(u) p_{k\,\ell}(u) \ln\left(\frac{p_{k\ell}(u)}{p^0_{k\ell}}\right) \\ &= \frac{1}{\lambda(u)} \sum_{k\,\ell} a_k(u) (p^1_{k\ell})^u (p^0_{k\ell})^{1-u} b_\ell(u) \ln\left(\left(\frac{p^1_{k\ell}}{p^0_{k\ell}}\right)^u \frac{b_k(u)}{\lambda(u) b_\ell(u)}\right) \\ &= -\ln\lambda(u) + u\frac{d}{du}\ln\lambda(u)\,. \end{aligned} \tag{12.60}$$

Then, if we refer to the definition (12.47) of the Legendre transform $I_0(z)$ of $\Lambda_0(u)$, and to the Fig. 10.5 showing its geometric construction, for a line with slope z and vertical intercept $-I_0(z)$, the horizontal coordinate $u(z)$ of the tangency point of the line to $\Lambda_0(u)$ satisfies the equation

$$z = \frac{d}{du}\Lambda_0(u) = \frac{d}{du}\ln\lambda(u)\,. \tag{12.61}$$

Substituting this identity inside (12.60) yields

$$I_0(z) = J(\mathbf{P}(u(z)), \mathbf{P}^0)\,, \tag{12.62}$$

so the Legendre transform represents the information discrimination measure existing between the Markov chain with transition matrix $\mathbf{P}(u(z))$ and the one with transition matrix $\mathbf{P}^1$. We can show in a similar manner that

$$I_1(z) = J(\mathbf{P}(u(z)), \mathbf{P}^1)\,. \tag{12.63}$$

When $z = 0$, we have $I_0(0) = I_1(0)$, and in this case the transition matrix $\mathbf{P}(u(0))$ satisfies

$$J(\mathbf{P}(u(0)), \mathbf{P}^1) = J(\mathbf{P}(u(0)), \mathbf{P}^0)\,, \tag{12.64}$$

so in terms of the information discrimination measure J, $\mathbf{P}(u(0))$ is located at the midway point on the geodesic linking $\mathbf{P}^0$ and $\mathbf{P}^1$ (see [12] for a discussion of the differential geometry of statistical models).

Example 12.2: Discrimination of symmetric 2-state MCs

Consider a Markov chain with 2 states, which under hypotheses H_0 and H_1 admits the transition matrices

$$\mathbf{P}^0 = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix} \quad , \quad \mathbf{P}^1 = \begin{bmatrix} p & q \\ q & p \end{bmatrix}$$

with $q = 1 - p$, $0 < p < 1$, and $p \neq 1/2$. Under H_0 and H_1, the MC has the feature that it remains invariant under a permutation of states 1 and 2, so both states play a symmetrical role. Both transition matrices have for steady-state probability distribution

$$\boldsymbol{\pi} = \left[1/2 \; 1/2 \right] .$$

This implies that the frequency of occurrence of each state does not provide any information about the correct hypothesis. Instead, the optimum test depends on the transition rates between states. Specifically, if we consider the empirical probability distribution $q_{k\ell}$ with $k, \ell = 1,\ 2$ of successive states of the Markov chain, by taking into account the normalization

$$q_{11} + q_{12} + q_{21} + q_{22} = 1$$

we find that the sufficient test statistic

$$\begin{aligned} S_N &= (q_{11} + q_{22}) \ln(2p) + (q_{12} + q_{21}) \ln(2q) \\ &= (q_{11} + q_{22})(\ln(2p) - \ln(2q)) + \ln(2q) \end{aligned}$$

depends only on the fraction $q_{11}+q_{22}$ of successive states which are unchanged. By direct evaluation, we find that the information discrimination functions of $\mathbf{P}^0$ and $\mathbf{P}^1$ are given by

$$\begin{aligned} J(\mathbf{P}^0, \mathbf{P}^1) &= -\ln(2(pq)^{1/2}) \\ J(\mathbf{P}^1, \mathbf{P}^0) &= p \ln(2p) + q \ln(2q) . \end{aligned}$$

It is also easy to check that the spectral radius of

$$\mathbf{M}(u) = \frac{1}{2} \begin{bmatrix} (2p)^u & (2q)^u \\ (2q)^u & (2p)^u \end{bmatrix}$$

is given by

$$\lambda(u) = \frac{1}{2}[(2p)^u + (2q)^u]$$

and its corresponding left and right eigenvectors $\mathbf{a}^T(u)$ and $\mathbf{b}(u)$ are given by

$$\mathbf{a}(u) = \mathbf{b}(u) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} .$$

To illustrate the convexity of the log-generating function $\Lambda_0(u) = \ln \lambda(u)$, it is plotted in Fig. 12.2 for $p = 3/4$, $q = 1/4$ and $-0.5 \leq u \leq 1.5$.

For this example, the Legendre transform can be evaluated in closed form. By solving

$$\frac{d}{du}(zu - \ln \lambda(u)) = z - \frac{d}{du}\lambda(u) = 0$$

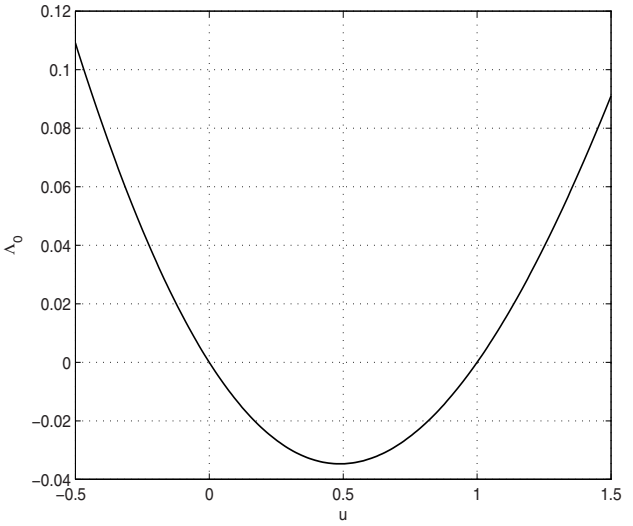

Fig. 12.2. Asymptotic log-generating function $\Lambda_0(u)$ for the LRT between $\mathbf{P}^1$ and $\mathbf{P}^0$.

with

$$\frac{d}{du}\lambda(u) = \frac{(2p)^u \ln(2p) + (2q)^u \ln(2q)}{(2p)^u + (2q)^u} ,$$

we obtain

$$u(z) = \frac{\ln[(\ln(2q) - z)/(z - \ln(2p))]}{\ln(p/q)} .$$

The corresponding rate functions

$$I_0(z) = zu(z) - \ln \lambda(u(z))$$

and

$$I_1(z) = I_0(z) - z$$

are plotted below in Fig. 12.3 for $p = 3/4$ and $q = 1/4$ and $-0.2 \leq z \leq 0.2$. Note that for the above values of p and q, we have $e_0 = -J(\mathbf{P}^0, \mathbf{P}^1) = -0.1438$ and $e_1 = J(\mathbf{P}^1, \mathbf{P}^0) = 0.1308$, so the parts of the plot that are relevant for hypothesis testing purposes extend only between e_0 and e_1, for which $I_0(e_0) = 0$ and $I_1(e_1) = 0$, as expected. □

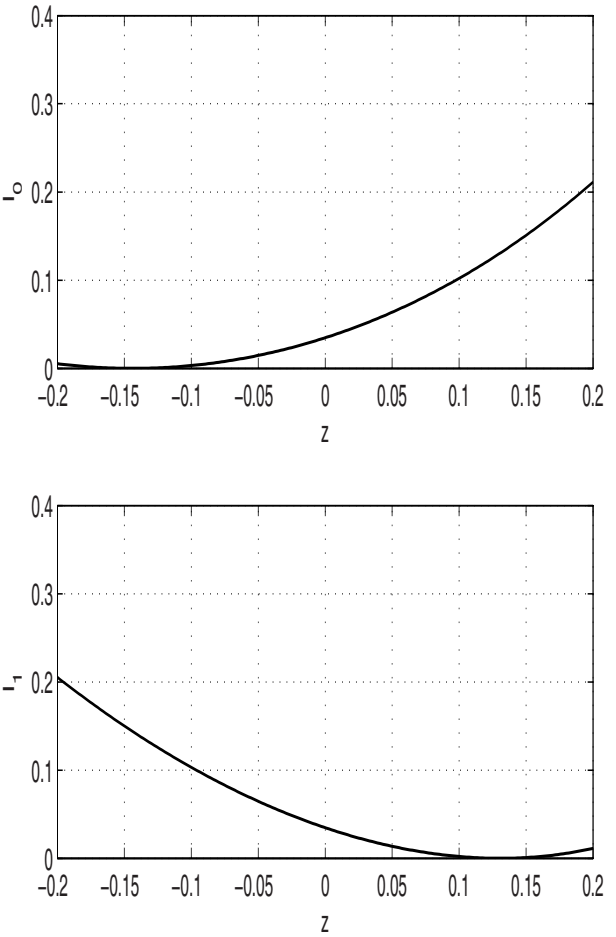

Fig. 12.3. Rate functions $I_0(z)$ and $I_1(z)$ for the asymptotic LRT between $\mathbf{P}^1$ and $\mathbf{P}^0$.

12.3 Detection of Partially Observed Markov Chains

Consider now the situation where instead of observing directly the Markov chain state $X(t)$, we observe

$$Y(t) = h(X(t), X(t+1)) + V(t) \tag{12.65}$$

for $1 \leq t \leq N$, where $h(X(t), X(t+1))$ is a function of two successive states, and $V(t)$ is a zero-mean WGN with variance σ^2. The model (12.65) can be employed either when the observed quantity is a function of the current state $X(t)$ only, or when it depends on the transition between $X(t)$ to $X(t+1)$. The MC has for initial probability distribution $\boldsymbol{\pi}(1)$ and we assume it is homogeneous with one-step transition probability matrix $\mathbf{P}$. The MC is assumed to be irreducible, but not necessarily aperiodic, since MCs arising in digital communications applications, such as the one considered in Example 12.1, are often periodic.

Given observations $Y(t)$ with $1 \leq t \leq N$, the detection problem consists of finding the state sequence $X(t)$ with $1 \leq t \leq N+1$ corresponding to it. To formulate this problem as a standard M-ary hypothesis testing problem of the type considered in Chapter 2, it is convenient to introduce the state trellis associated to an MC. For an MC defined over $1 \leq t \leq N+1$, the trellis is obtained by unfolding in time the transition diagram $\mathcal{G}$ of the MC. Specifically, if we consider the transition diagram $\mathcal{G}$, an arbitrary edge leaves a node representing the value of the MC at time t and arrives at a node representing its value at time $t+1$. Instead of using the same set of nodes to represent the state values at two consecutive times, the state trellis assigns a node to each state value at each time. So for an MC with K states and $N+1$ times, the trellis has $K(N+1)$ nodes, and a directed edge connects the vertex corresponding to state k at time t to one corresponding to state ℓ at time $t+1$ if $p_{k\ell} > 0$. The trellis has K source nodes describing the different possible state values at time $t = 1$, and it has K sink nodes representing the state values at time $t = N+1$. A directed path linking a source node to a sink node corresponds to a complete state trajectory over interval $1 \leq t \leq N+1$. The set of all state trajectories is denoted as $\mathcal{U}$. Obviously, the cardinality U (the number of elements) of $\mathcal{U}$ is huge. If we assume that all states at time $t = 0$ are allowed, i.e., that $\pi_k(1) > 0$ for all k, and if we assume that the trellis is regular in the sense that the same number T of transitions is allowed out of each state (note that $T \leq K$), then $U = KT^N$. Thus, the total number U of trajectories grows exponentially with the length N of the observation interval. To illustrate the above discussion, we consider an example.

Example 12.1, continued

Consider the MC of Example 12.1. After time-unfolding the state transition diagram of Fig. 12.1, we obtain the state trellis shown in Fig. 12.4. The trellis is regular, since $P = 2$ transitions are allowed out of each state, so the total number of trellis trajectories is $U = 4\,2^N$. Since all state transitions have probability 1/2, if the initial states are equally likely, i.e.,

$$\boldsymbol{\pi}(1) = \frac{1}{K} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} ,$$

then all state trajectories are equally likely, a situation which is actually rather common in digital communications applications.

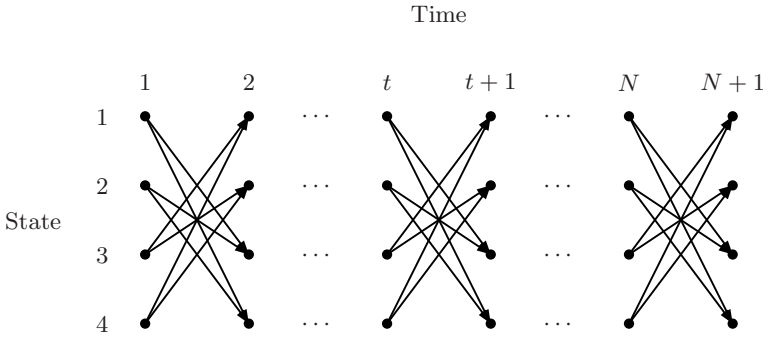

Fig. 12.4. State trellis for the Markov chain of Example 12.1.

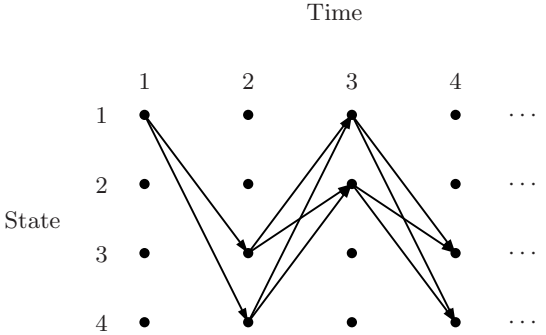

Fig. 12.5. Pruned state trellis obtained by removing impossible states/transitions.

Another situation that occurs frequently is when the initial state or the terminal state, or both, are known. For example, in wireless communications, blocks of transmitted data are divided into training data and payload subblocks. So if the received signal needs to be equalized, the fact that the training block is known has the effect of fixing the initial state of the MC representing the symbols stored by the channel. When the initial state is known, the trellis needs to be pruned to remove impossible states and/or transitions. For example, for the trellis of Fig. 12.4, suppose the initial state is $X(1) = 1$. Then the pruned trellis obtained by removing trajectories starting from states 2, 3 and 4 is shown in Fig. 12.5. Note that because the MC has period $d = 2$ and the initial state is known, for $t \geq 2$, $X(t)$ must belong to $\mathcal{S}_2 = \{3, 4\}$ for t even, and to $\mathcal{S}_1 = \{1, 2\}$ for t odd. It is worth observing that after all trajectories starting from initial states other than 1 have been removed, all the remaining trajectories are equally likely. When the final state $X(N+1)$ is known, a similar pruning operation can be performed to remove trajectories terminating in forbidden final states. □

12.3.1 MAP Sequence Detection

The MC detection problem corresponding to observations (12.65) can be formulated as follows: To each possible trajectory

$$\mathbf{x}_u = (x_t^u,\, 1 \le t \le N+1) \tag{12.66}$$

of the MC trellis with $1 \le u \le U$ corresponds a known signal

$$s_u(t) = h(x_t^u, x_{t+1}^u) \tag{12.67}$$

with $1 \le t \le N$. We assume that different trajectories $\mathbf{x}_u$ and $\mathbf{x}_v$ with $u \ne v$ give rise to different signals $s_u(t)$ and $s_v(t)$. Otherwise the detection problem is ill-posed, since it is impossible to discriminate between state trajectories corresponding to the same signal $s_u(t)$. Then the MC detection problem is a standard U-ary detection problem, where we need to decide between hypotheses

$$H_u \;:\; Y(t) = s_u(t) + V(t) \tag{12.68}$$

with $1 \le u \le U$, where the signal $s_u(t)$ is known. The a-priori probability of hypothesis H_u is the probability of the MC trajectory $\mathbf{x}_u$, i.e.,

$$p_u = P[H_u] = \pi_{x_1^u}(1) \prod_{t=1}^{N} p_{x_t^u x_{t+1}^u}\,. \tag{12.69}$$

Recall that because all impossible trajectories have been removed from the state trellis, $p_u > 0$ for all $1 \le u \le U$. Then, as indicated by (2.140) to find the MAP state trajectory, we need to apply the decision rule

$$\delta_{\mathrm{MAP}}(\mathbf{y}) = \arg \max_{1 \le u \le U} [\ln(f(\mathbf{y}|H_u)) + \ln(p_u)]\,, \tag{12.70}$$

where according to (12.69), $\ln(p_u)$ can be expressed as

$$\ln(p_u) = \ln \pi_{x_1^u}(1) + \sum_{t=1}^{N} \ln p_{x_t^u x_{t+1}^u}\,. \tag{12.71}$$

In this expression $\mathbf{y} \in \mathbb{R}^N$ is the observed value of the random vector

$$\mathbf{Y} = \big[\, Y(1) \cdots\; Y(t) \cdots\; Y(N) \,\big]^T \tag{12.72}$$

and if

$$\mathbf{s}_u = \big[\, s_u(1) \cdots\; s_u(t) \cdots\; s_u(N) \,\big]^T \tag{12.73}$$

denotes the N-dimensional vector representing the signal $s_u(t)$, since $X(t)$ is a zero-mean WGN with variance σ^2, the random vector $\mathbf{Y}$ is $N(\mathbf{s}_u, \sigma^2 \mathbf{I}_N)$ distributed under H_u, i.e.,

$$f(\mathbf{y}|H_u) = \frac{1}{(2\pi\sigma^2)^N} \exp\Big(-\frac{1}{2\sigma^2} ||\mathbf{y} - \mathbf{s}_u||_2^2 \Big) . \tag{12.74}$$

Multiplying the quantity to be maximized by $-2\sigma^2$, replacing the maximization by a minimization, and dropping terms that do not depend on u, the MAP decision rule (12.70) can be expressed as

$$\delta(\mathbf{y}) = \arg \min_{1 \le u \le U} [||\mathbf{y} - \mathbf{s}_u||_2^2 - 2\sigma^2 \ln(p_u)] . \tag{12.75}$$

For the special case when all trajectories are equally likely, the MAP decision rule (12.75) reduces to the minimum distance rule

$$\delta(\mathbf{y}) = \arg \min_{1 \le u \le U} ||\mathbf{y} - \mathbf{s}_u||_2^2 . \tag{12.76}$$

Up to this point, there is nothing to distinguish the MC detection problem from any conventional U-ary detection problem, except that the total number U of trellis trajectories grows exponentially with the length N of the observation block. Since U is a huge number, it is impossible to examine the trajectories one by one to find the one that maximizes the maximum a-posteriori probability. So clever search strategies must be devised to explore the state trellis in such a way that only the most promising trajectories are considered. In 1967, Viterbi [13] proposed a search technique which is optimal in the sense that it always yields the MAP trellis trajectory. This method has since become known as the Viterbi algorithm. Its key idea is, for each possible state value at time t, to find the MAP trellis path between time $t = 1$ and this state. Thus the Viterbi algorithm can be viewed as a breadth-first trellis search since it examines as many paths as there are possible state values at time t, i.e., the search is performed over an entire cross-section of the trellis. Furthermore, all paths under consideration are extended simultaneously. By comparison, a depth-first strategy would focus on one promising trellis path and extend it until it reveals to be less promising than initially thought, at which point backtracking would be required prior to a forward thrust in another search direction, and so on. Although depth-first strategies typically have a low computational cost, they can never guarantee optimality. The Viterbi algorithm was originally proposed for decoding convolutional codes [13]. It was later extended by Forney [14] to the ML equalization of ISI signals. Then in [15], Forney observed that the method used by Viterbi to extend MAP paths from time t to time $t + 1$ was just an instance of the optimality principle employed by Bellman [16, 17] to develop the theory of of dynamic programming for solving optimization problems with an additive cost structure. It was also shown in [15] that the Viterbi algorithm is applicable to the MAP sequence detection problem for a partially observed MC, which opened the way to its use in speech recognition [6] and computational biology.

Before describing the Viterbi algorithm in detail, it is useful to highlight the specific property of the MAP trellis search problem (12.75) that it exploits. Observe that the squared Euclidean distance between $\mathbf{y}$ and $\mathbf{s}_u$ can be

expressed as

$$||\mathbf{y} - \mathbf{s}_u||_2^2 = \sum_{t=1}^{N} (y_t - h(x_t^u, x_{t+1}^u))^2 \,. \tag{12.77}$$

Then, if we drop the index u identifying the trellis trajectory, by combining (12.71) and (12.77), we find that the MAP trellis path $\hat{\mathbf{x}}_{\mathrm{MAP}}(\mathbf{y})$ can be expressed as the solution of the minimization problem

$$\hat{\mathbf{x}}_{\mathrm{MAP}}(\mathbf{y}) = \arg \min_{\mathbf{x} \in \mathcal{U}} J(\mathbf{x}) \tag{12.78}$$

with

$$J(\mathbf{x}) = -2\sigma^2 \ln p_{x_1} + \sum_{t=1}^{N} c(x_t, x_{t+1}, t) \,, \tag{12.79}$$

where each incremental cost

$$c(x_t, x_{t+1}, t) = (y_t - h(x_t, x_{t+1}))^2 - 2\sigma^2 \ln p_{x_t x_{t+1}} \tag{12.80}$$

depends only on the path segment (x_t, x_{t+1}). For the MAP sequence detection problem, the cost $c(x_t, x_{t+1}, t)$ is known as the *branch metric* and $J(\mathbf{x})$ is called the *path metric*. For applications where all allowed transitions $(k, \ell) \in \mathcal{T}$ are equally likely, the term $\ln p_{x_t x_{t+1}}$ is constant, so it can be dropped from the cost since it is the same for all branches. In this case

$$c(x_t, x_{t+1}, t) = (y_t - h(x_t, x_{t+1}))^2 \tag{12.81}$$

is the Euclidean metric. Two properties allow the application of Bellman's optimality principle to the MAP sequence detection problem. First, the decomposition (12.79) of the cost function $J(\mathbf{x})$ is *additive*. Second, each incremental cost $c(x_t, x_{t+1}, t)$ is *local* since it depends only on the segment (x_t, x_{t+1}) of the trajectory. These two features characterize optimization problems that can be handled by dynamic programming.

The optimality principle can be stated as follows [17, p. 83]: "An optimal policy has the property that whatever the initial state and the initial decision, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." Since this is a very general prescription, let us describe how it can be applied to the problem at hand.

Remark: To keep our analysis as simple as possible, it has been assumed in observation model (12.65) that the observation $Y(t)$ and function $h(X(t), X(t+1)$ are scalar. However, for applications such as decoding of convolutional codes with rate $1/M$, or MAP fractional equalization of modulated signals sampled at rate M/T, where T denotes the baud interval, $Y(t)$ and $h(X(t), X(t+1))$ are vectors of dimension M. For such problems, the only change required consists of replacing the scalar quadratic term $(y_t - h(x_t, x_{t+1}))^2$ by the quadratic vector norm $||\mathbf{Y}_t - \mathbf{h}(x_t, x_{t+1})||_2^2$ in the definition (12.80) of the branch metric $c(x_t, x_{t+1}, t)$. This change does not affect the foregoing derivation of the Viterbi algorithm.

Viterbi algorithm: Let $\mathcal{U}(k,t)$ denote the set of all trellis trajectories starting from an arbitrary state at time $t = 1$ and terminating in the state k at time t. Consider also the function

$$J_t(\mathbf{x}) = -2\sigma^2 \ln \pi_{x_1}(1) + \sum_{s=1}^{t-1} c(x_s, x_{s+1}, s) \tag{12.82}$$

representing the accumulated cost for the portion of trellis trajectories extending from time 1 to time t. Note that if there are t states on a partial trajectory, there are $t-1$ transitions between them. Then at time t, and for each state k with $1 \le k \le K$, let

$$\hat{\mathbf{x}}(k,t) = \arg \min_{\mathbf{x} \in \mathcal{U}(k,t)} J_t(\mathbf{x}) \tag{12.83}$$

denote the MAP trajectory terminating in state k. The minimum value of $J_t(\mathbf{x})$ achieved by this trajectory is called the value function

$$V(k,t) \stackrel{\triangle}{=} J_t(\hat{\mathbf{x}}(k,t)) \,. \tag{12.84}$$

In the context of MAP sequence detection, this function is called the accumulated path metric, and the MAP trajectories $\hat{\mathbf{x}}(t,k)$ with $1 \le k \le K$ are called the survivor trajectories. For the case when there is no path terminating in state k at time t, the set $\mathcal{U}(k,t)$ is empty. If we consider the pruned trellis of Fig. 12.5, we see for example that no path terminates in states 1 and 2 for t even, or in states 3 and 4 for t odd. When this situation arises we set the value function

$$V(k,t) = \infty \,.$$

So suppose that at time t, the MAP trajectories $\hat{\mathbf{x}}(k,t)$ and value function $V(k,t)$ are known for $1 \le k \le K$, and we seek to find the MAP trajectories and value function at time $t+1$. According to the optimality principle, the MAP trajectory terminating in state ℓ at time $t+1$ must be formed by an MAP trajectory terminating in one of the K states, say k, at time t, to which we concatenate an allowed transition $(k,\ell) \in \mathcal{T}$. Then for all possible transitions from a state k to state ℓ, we select the one with the smallest accumulated path metric, i.e.,

$$V(\ell, t+1) = \min_{k:\,(k,\ell)\in\mathcal{T}} (V(k,t) + c(k,\ell,t)) \,. \tag{12.85}$$

The argument

$$b(\ell, t+1) = \arg \min_{k:\,(k,\ell)\in\mathcal{T}} (V(k,t) + c(k,\ell,t)) \tag{12.86}$$

of the minimization (12.85) forms what is called the backpointer function. Specifically, the state $b(\ell, t+1) \in \mathcal{S}$ represents the predecessor state at time

t on the MAP path terminating in state ℓ at time $t+1$, and the MAP path terminating in ℓ at time $t+1$ can be expressed as

$$\hat{\mathbf{x}}(\ell, t+1) = (\hat{\mathbf{x}}(b(\ell, t+1), t), \ell) \,. \tag{12.87}$$

In other words, it is obtained by concatenating the segment $(b(\ell, t+1), \ell)$ to the MAP path terminating in $b(\ell, t+1)$ at time t.

The recursion (12.85)–(12.87) is implemented until the end of the trellis is reached, corresponding to time $t = N+1$. At this point the survivor paths $\hat{\mathbf{x}}(k, N+1)$ and the value function $V(k, N+1)$ are available. These paths represent the final candidates for selecting the overall MAP path. The optimal terminal state

$$k_0 = \arg \min_{k \in \mathcal{S}} V(k, N+1) \tag{12.88}$$

is selected by minimizing the value function, and once this terminal state has been identified, the MAP path is the corresponding survivor path, i.e.

$$\hat{\mathbf{x}}_{\text{MAP}} = \hat{\mathbf{x}}(k_0, N+1) \,. \tag{12.89}$$

The associated path metric

$$V(k_0, N+1) = J(\hat{\mathbf{x}}_{MAP}) \tag{12.90}$$

represents the minimum of the function $J(\mathbf{x})$ over the entire set $\mathcal{U}$ of trellis paths.

Viterbi algorithm implementation: From the above presentation of the Viterbi algorithm, we see that the key steps are recursions (12.85)–(12.87). Equation (12.85) is usually implemented by what is called an add, compare and select (ACS) unit. Observe for a given state ℓ, equation (12.85) adds to the value function $V(k, t)$ for each possible predecessor state the transition cost $c(k, \ell, t)$. Then the sums $V(k, t) + c(k, \ell, t)$ are compared and the smallest is selected, so the acronym ACS captures precisely all the steps required to implement (12.85).

As for equations (12.86) and (12.87), they are employed by the survivor memory unit (SMU) to keep track of the survivor paths. Two different approaches can be used to accomplish this task. The register exchange method employs a $K \times N+1$ register array to store the MAP survivor paths. At stage t, the k-th row of the array stores the survivor path $\hat{\mathbf{x}}(k, t)$. Since this path has length t, this means that at stage t, only the first t columns of the array are filled, and the remaining columns are zero. Then at stage $t+1$, as indicated by equation (12.87), the MAP survivor path $\hat{\mathbf{x}}(\ell, t+1)$ which is stored in row ℓ of the array is obtained by taking the MAP path stored in row $b(\ell, t+1)$ at stage t and appending the state ℓ in column $t+1$ as the last trajectory element. Since this scheme involves swapping the contents of different array rows as we go from stage t to stage $t+1$, it is called the register exchange algorithm. Note that after the new survivor trajectories have been stored at stage

$t+1$, all the stage t survivor paths can be discarded since they are no longer needed. Right at the end, at stage $N+1$, the array has been filled. Then, after the terminal state k_0 of the MAP path has been identified by performing the minimization (12.88), the entire MAP trajectory is obtained instantaneously by reading off the k_0-th row of the array. So the register exchange method has a high memory management overhead since it requires moving information continuously from one set of registers to another, but it has no latency.

The second approach that can be used to evaluate the MAP trajectory is based on an observation of Rader [18], and is called the trace-back method [19,20]. It employs a register array of size K by N, but in this case the contents of the array do not need to be moved around. Specifically, the element of row ℓ and column t is the backpointer $b(\ell, t+1)$, which indicates the predecessor state of ℓ on the survivor trajectory terminating in ℓ at time $t+1$. So as stage $t+1$, the t-th column of the array is evaluated, and as the Viterbi recursion (12.85)–(12.86) advances, the array is filled one column at a time. At stage $N+1$, the array is full, and after the MAP terminal state k_0 has been identified, we need to identify the MAP trajectory

$$\hat{\mathbf{x}}_{\text{MAP}} = (\hat{x}_t^{\text{MAP}},\, 1 \le t \le N+1)\,.$$

This trajectory is identified by applying the trace-back recursion

$$\hat{x}_t^{\text{MAP}} = b(\hat{x}_{t+1}^{\text{MAP}}, t+1) \tag{12.91}$$

with initial condition

$$\hat{x}_{N+1}^{\text{MAP}} = k_0\,. \tag{12.92}$$

So at stage $t+1$, the recursion involves reading the element in row $\hat{x}_{t+1}^{\text{MAP}}$ and column t of the array to determine the state of the MAP trajectory at time t. Proceeding sequentially, we see that the entire MAP trajectory is traced back, starting from the terminal state k_0 at time $N+1$. From the above discussion, we see that the trace-back method has no memory management overhead, but it incurs some latency due to the need to trace-back the entire trajectory. So depending on the power consumption and latency requirements of the application one considers, either the register exchange method or the trace-back technique may be preferred.

As explained in [21], a variety of additional computation tricks, such as parallelization or pipelining, can be applied to the implementation of the Viterbi algorithm.

Example 12.3: Viterbi decoding of AMI line code

To illustrate the Viterbi algorithm, consider a two-state Markov chain with transition matrix

$$\mathbf{P} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\,.$$

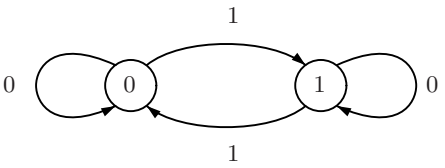

Fig. 12.6. State transition diagram for the AMI MC, with transitions labeled by the corresponding input bits.

Its state-transition diagram is shown in Fig. 12.6. Note that to simplify the following discussion, the state labels are $\{0, 1\}$. The signal

$$s(t) = X(t+1) - X(t) \tag{12.93}$$

takes on the ternary values $\{1, 0, -1\}$ and corresponds to the signal generated by an alternate mark inversion (AMI) line encoder. In this context, if $\{U(t),\, t \geq 1\}$ denotes the bit sequence at the input of the AMI encoder, the state of the MC satisfies

$$X(t+1) = X(t) \oplus U(t)\,, \tag{12.94}$$

where $\oplus$ denotes the modulo two addition, so that $X(t)$ can be interpreted as the running digital sum of all binary inputs prior to time t. The state transitions of Fig. 12.6 are labeled by the value of the input $U(t)$ giving rise to the transition. As can be seen from the diagram, a 0 input leaves the state unchanged, whereas a 1 forces a transition to the other state. In setting up the MC model, it is assumed that input bits $U(t)$ are independent and take values 0 and 1 with probability 1/2.

Given observations

$$Y(t) = s(t) + V(t)\,, \tag{12.95}$$

we seek to reconstruct the state sequence $X(t)$. Note that once the sequence $X(t)$ is known, the input $U(t)$ is given by

$$U(t) = X(t+1) \oplus X(t)\,. \tag{12.96}$$

The standard AMI decoder consists of a ternary slicer followed by a memoryless mapper [22, pp. 564–568]. As indicated by the input-output characteristic shown in Fig. 12.7, the slicer applies a minimum distance rule to observation $Y(t)$ to select the value 1, 0, or -1 of the signal estimate $\hat{s}(t)$. Then the mapper associates points $\hat{s}(t) = 1$ or -1 to the estimated input $\hat{U}(t) = 1$, and point $\hat{s}(t) = 0$ to $\hat{U}(t) = 0$. However, this detector is not optimal, and it was shown in [23] that Viterbi decoding yields a gain of up to 3dB over this decoder.

We focus here on the mechanics of the Viterbi algorithm. The initial state of the MC is assumed to be $X(1) = 0$. The observed sequence is

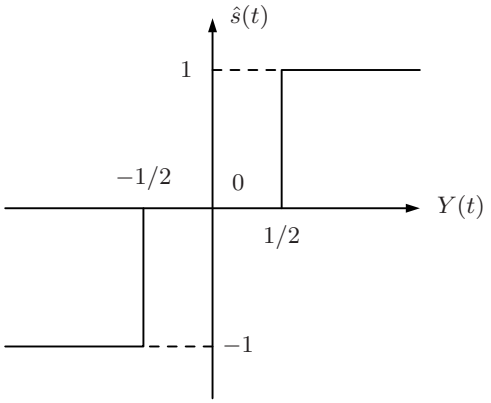

Fig. 12.7. Input-output characteristic of the ternary slicer used to map the observation $Y(t)$ into a signal estimate $\hat{s}(t)$.

$$\begin{aligned}\mathbf{y} &= \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \end{bmatrix}^T \\ &= \begin{bmatrix} 0.1 & 0.8 & -1.1 & 0.4 & -0.2 & -0.9 & 0.2 & 0.1 \end{bmatrix}^T .\end{aligned}$$

Since all trellis paths are equally likely, we can use

$$c(x_{t+1}, x_t, t) = (y_t - x_{t+1} + x_t)^2 \tag{12.97}$$

as the metric for the trellis branches. Then the survivor paths obtained at successive stages of the Viterbi algorithm are shown in Figs. 12.8 and 12.9. In this figure, each branch of the survivor paths is labeled by its Euclidean metric (12.97).

When the end of the trellis has been reached, we see from Fig. 12.9 that the survivor terminating in state 0 has a smaller metric than the survivor terminating in 1, so the MAP state sequence is given by

$$\begin{aligned}\hat{\mathbf{x}}_{\text{MAP}} &= \begin{bmatrix} \hat{x}_1^{\text{MAP}} & \hat{x}_2^{\text{MAP}} & \hat{x}_3^{\text{MAP}} & \hat{x}_4^{\text{MAP}} & \hat{x}_5^{\text{MAP}} & \hat{x}_6^{\text{MAP}} & \hat{x}_7^{\text{MAP}} & \hat{x}_8^{\text{MAP}} & \hat{x}_9^{\text{MAP}} \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T . \end{aligned} \tag{12.98}$$

Applying the decoding rule (12.96), we find that the corresponding input sequence is given by

$$\begin{aligned}\hat{\mathbf{u}}_{\text{MAP}} &= \begin{bmatrix} \hat{u}_1^{\text{MAP}} & \hat{u}_2^{\text{MAP}} & \hat{u}_3^{\text{MAP}} & \hat{u}_4^{\text{MAP}} & \hat{u}_5^{\text{MAP}} & \hat{u}_6^{\text{MAP}} & \hat{u}_7^{\text{MAP}} & \hat{u}_8^{\text{MAP}} \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}^T .\end{aligned}$$

Note that that for $t = 4$, this sequence differs from the one produced by the conventional AMI line decoder, which yields

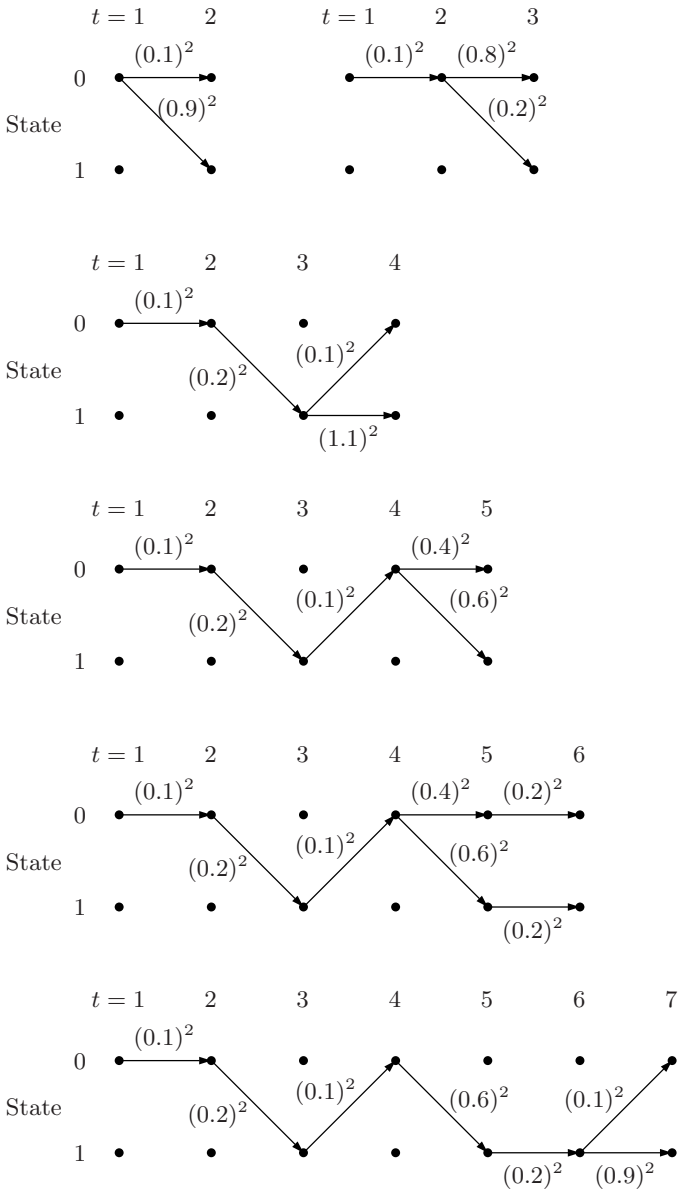

Fig. 12.8. Survivor paths at the first six stages of the Viterbi algorithm for Example 12.3.

$$\hat{u}_4^{\mathrm{AMI}} = 0\,.$$

An important feature of survivor paths which is in evidence in Figs. 12.8 and 12.9 is that as the Viterbi algorithm progresses, the survivors merge, i.e.,

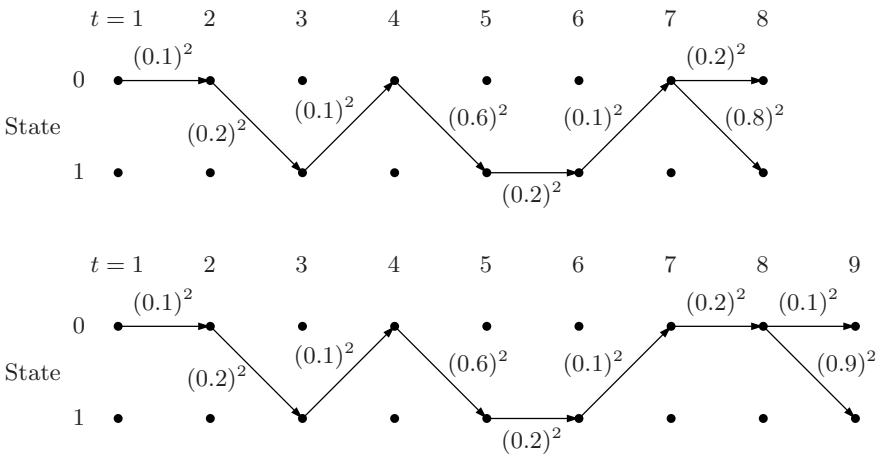

Fig. 12.9. Survivor paths at the last two stages of the Viterbi algorithm for Example 12.3.

while there may be some disagreement about recent states among different survivors, all survivors tend to have a common remote past. This allows the online implementation of the Viterbi algorithm, where, as the Viterbi algorithm processes the current observation $Y(t)$, decisions about the value of state $X(t-D)$ can be made. The value of the delay D is based on computer simulations or on an analysis of error events for the MC. What constitutes an "error event" is discussed later in this section. For the AMI example considered here, a delay $D = 2$ suffices.

Whereas Figs. 12.8 and 12.9 implement the register exchange version of the Viterbi algorithm, a trace-back implementation would store the back-pointer function $b(t,k)$ shown in Table 12.1, where the columns represent the time index t, and the rows the state index k. Recall that $b(t,k)$ specifies the state at time $t-1$ of the survivor path terminating in state k at time t.

Then, if we observe that the terminal survivor path metric is minimized by $k_0 = 0$, the MAP path (12.98) can be generated by using the entries of Table 12.1 to implement the trace-back recursion (12.91). □

Probability of error: We have seen above that the MAP sequence detection problem for Markov chains can be viewed as a U-ary hypothesis testing problem among the different signals $s_u(t)$ with $1 \le u \le U$ corresponding to the

Table 12.1. Backpointer $b(t,k)$ as a function of time t and terminal state k.

$b(t,k)$	2	3	4	5	6	7	8	9
0	0	0	1	0	0	1	0	0
1	0	0	1	0	1	1	0	0

paths of the MC trellis. Except for the fact that this U-ary detection problem has a very large cardinality, the performance of the MAP path detector can therefore be analyzed by the approach described in Section 2.7.3. However, closer examination suggests that the path error may not necessarily be the right way to evaluate the performance of the MAP sequence detector. Specifically, as the block length N tends to infinity, the probability of path error tends to 1. To see why this is the case, assume for simplicity that all trellis paths are equally likely. To interpret the MAP sequence detection problem, we adopt a signal space viewpoint where the signal $s_u(t)$, $1 \le t \le N$ associated to a trellis path is represented by a N-dimensional vector $\mathbf{s}_u$, and observations $Y(t)$, $1 \le t \le N$ are represented by observation vector $\mathbf{Y}$. Then the MAP decision rule selects the vector $\mathbf{s}_u$ closest to $\mathbf{Y}$ in terms of the Euclidean distance. From the error event analysis described below, we find that as N increases, the Euclidean distance h between a vector $\mathbf{s}_u$ and its nearest neighbors remains *constant*, but the number of neighbors grows linearly with N. Under H_u, the observation vector $\mathbf{Y}$ can be expressed as

$$\mathbf{Y} = \mathbf{s}_u + \mathbf{V} \tag{12.99}$$

where $\mathbf{V} \sim N(0, \sigma^2 \mathbf{I}_N)$ is the noise vector with entries $V(t)$ for $1 \le t \le N$. Then as N becomes large, the probability of a correct decision can be roughly approximated as

$$P[C|H_u] \approx P[\mathbf{V} \in \mathcal{B}] \tag{12.100}$$

where

$$\mathcal{B} = \{\mathbf{v} : |v_t| < h/2 \text{ for } 1 \le t \le N\}$$

denotes a square box centered at the origin with width h in each direction. The probability that the noise vector $\mathbf{V}$ falls within this box is given by

$$\begin{aligned} P[\mathbf{V} \in \mathcal{B}] &= \prod_{k=1}^{N} P[|V(t)| < h/2] \\ &= [1 - 2Q(\frac{h}{2\sigma})]^N \,, \end{aligned} \tag{12.101}$$

which tends to zero as $N \to \infty$. This shows that $P[C|H_u] \to 0$ as $N \to \infty$.

Thus the probability that the MAP sequence detector selects the correct sequence is zero as the block length N goes to infinity. While this result may appear discouraging, it is worth noting that a path may be viewed as incorrect even though it contains only a small number of times t such that $\hat{X}_t^{\text{MAP}} \neq X(t)$. So instead of focusing on the probability that the entire MAP sequence is incorrect, we ought to to focus on the probability that any one of its components, say $\hat{X}_t^{\text{MAP}}$ differs from the state $X(t)$. While the pointwise error probability

$$P[E] = P[\hat{X}_t^{\text{MAP}} \neq X(t)]$$

cannot be evaluated in closed form, it was shown by Forney [14] that it admits an upper bound constructed by using the improved union bound method of Section 2.7.4.

First, note that by conditioning with respect to the entire state sequence

$$\mathbf{X} = (X(t),\, 1 \le t \le N+1)\,,$$

the probability of detecting incorrectly the state at time t can be expressed as

$$P[E] = \sum_{u=1}^{U} P[\hat{X}_t^{\mathrm{MAP}} \neq x_t^u | \mathbf{X} = \mathbf{x}_u] p_u\,, \tag{12.102}$$

where

$$p_u = P[\mathbf{X} = \mathbf{x}_u]$$

is given by (12.69). In our analysis, we assume that the time t we consider is located far from both ends of the trellis, so that edge effects do not come into play. Then for a fixed trellis path $\mathbf{x}_u$ and a fixed time t, we introduce the concept of error event. From the definition (12.78) of the MAP path, we see that given that the actual path is $\mathbf{x}_u$, a pairwise error event occurs at time t if there exists a path $\mathbf{x}_v$ such that $J(\mathbf{x}_v) < J(\mathbf{x}_u)$ and $x_t^v \neq x_t^u$. The set of paths of $\mathcal{U}$ satisfying these conditions can be denoted as $\mathcal{E}(t, \mathbf{x}_u)$. Among pairwise error events of $\mathcal{E}(t, \mathbf{x}_u)$, if we refer to the improved union bound (2.166), in order to obtain a tight bound, it is of interest to identify a sufficient set, i.e., a set of paths of $\mathcal{E}(t, \mathbf{x}_u)$ that are neighbors of $\mathbf{x}_u$ and which cover all circumstances under which the MAP path differs from $\mathbf{x}_u$ at time t. A sufficient set is provided by the *simple error events* $\mathcal{S}(t, \mathbf{x}_u)$ introduced in [24]. These are the paths $\mathbf{x}_v$ of $\mathcal{E}(t, \mathbf{x}_u)$ that differ from $\mathbf{x}_u$ in a connected pattern. For such a path, there exists times t_L snd t_R such that $t_L < t < t_R$ and

$$\begin{aligned} &x_r^v = x_r^u \; r \le t_L \\ &x_r^v \neq x_r^u \; t_L < r < t_R \\ &x_r^v = x_r^u \; t_R \le r\,. \end{aligned} \tag{12.103}$$

In other words, the paths $\mathbf{x}_v$ and $\mathbf{x}_u$ merge at a time t_L to the left of t and remain the same thereafter. Similarly, $\mathbf{x}_v$ and $\mathbf{x}_u$ merge at a time t_R to the right of t and remain the same thereafter. Accordingly, the value $x_t^v \neq x_t^u$ is part of a single excursion of the path $\mathbf{x}_v$ away from the true path $\mathbf{x}_u$. For example, if we consider a Markov chain with two states $\{0, 1\}$ and if we assume that the true path $\mathbf{x}_u$ is the all zero state path, a simple error event is shown in Fig. 12.10. For this example, we see that the left and right merging times are $t_L = t - 2$ and $t_R = t + 1$, respectively.

To explain why the set $\mathcal{S}(t, \mathbf{x}_u)$ of simple error events forms a sufficient set of pairwise error events covering all cases where the MAP path $\hat{\mathbf{x}}_{\mathrm{MAP}}$ differs from $\mathbf{x}_u$ at time t, we use the optimality principle. For a Markov chain with three states $\{0, 1, 2\}$ and where the true path $\mathbf{x}_u$ is the all zero state path, suppose the MAP path is shown in Fig. 12.11. It contains several excursions

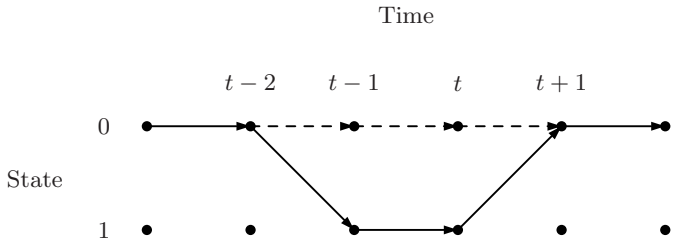

Fig. 12.10. Simple error event $\mathbf{x}_v$ with respect to the all-zero path in a two-state MC.

away from the true path, two of which are depicted in the figure. Then consider a time t at which the MAP path differs from $\mathbf{x}_u$, and let $\mathbf{x}_v$ be the path obtained by retaining only the local excursion of $\hat{\mathbf{x}}_{\text{MAP}}$ surrounding time t, and denote by t_L and t_R the left and right merge times for this excursion. Since $\hat{\mathbf{x}}_{\text{MAP}} \neq \mathbf{x}_u$, we know that $J(\hat{\mathbf{x}}_{\text{MAP}}) < J(\mathbf{x}_u)$. Then observe that between times t_L and t_R, the paths $\hat{\mathbf{x}}_{\text{MAP}}$ and $\mathbf{x}_v$ coincide. By the optimality principle of the MAP path, $\mathbf{x}_v$ must have a lower metric than $\mathbf{x}_u$, i.e. $J(\mathbf{x}_v) < J(\mathbf{x}_u)$, since otherwise the $\mathbf{x}_u$ segment would have been preferred over $\mathbf{x}_v$ between times t_L and t_R. So we have shown that if the MAP path differs from the

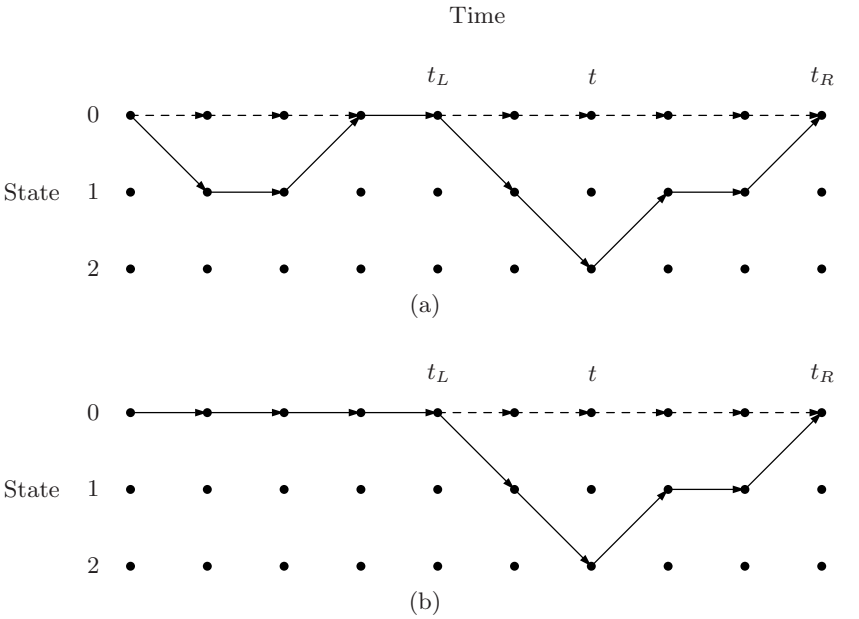

Fig. 12.11. (a) MAP path $\hat{\mathbf{x}}_{\text{MAP}}$ differing from the all zero path at time t, and (b) simple error event $\mathbf{x}_v$ obtained by retaining only the local excursion away from the all zero path.

true path at time t, the local excursion $\mathbf{x}_v$ of the MAP path surrounding t is a simple error event. Consequently, simple error events form a sufficient set of pairwise error events for the decision $\hat{x}_t^{\text{MAP}} \neq x_t^u$.

Then, according to the improved union bound (2.166), the conditional probability of error

$$P[\hat{X}_t^{\text{MAP}} \neq x_t^u | \mathbf{X} = \mathbf{x}_u] \leq \sum_{\mathbf{x}_v \in \mathcal{S}(t,\mathbf{x}_u)} P[J(\mathbf{x}_v) < J(\mathbf{x}_u) | \mathbf{X} = \mathbf{x}_u]$$
$$= \sum_{\mathbf{x}_v \in \mathcal{S}(t,\mathbf{x}_u)} Q\left(\frac{d_{uv}}{2} - \frac{\ln(p_v/p_u)}{d_{uv}} \right), \quad (12.104)$$

where

$$d_{uv} = \frac{||\mathbf{s}_u - \mathbf{s}_v||_2}{\sigma} \quad (12.105)$$

is the Euclidean distance between signals $\mathbf{s}_u$ and $\mathbf{s}_v$, scaled by the standard deviation σ of the noise, i.e., d_{uv} represents the Euclidean distance of the two signals measured in units of the standard deviation. To go from the first to the second line of (12.104), consider the random variable

$$Z \triangleq \frac{J(\mathbf{x}_v) - J(\mathbf{x}_u)}{2\sigma^2 d_{uv}}$$
$$= \frac{1}{2\sigma^2 d_{uv}} \left(||\mathbf{Y} - \mathbf{s}_v||_2^2 - ||\mathbf{Y} - \mathbf{s}_u||_2^2 \right) - \frac{\ln(p_v/p_u)}{d_{uv}} . \quad (12.106)$$

Under H_u, by substituting (12.99) inside (12.106), we find

$$Z = \frac{\mathbf{V}^T(\mathbf{s}_u - \mathbf{s}_v)}{\sigma^2 d_{uv}} + \frac{d_{uv}}{2} - \frac{\ln(p_v/p_u)}{d_{uv}} . \quad (12.107)$$

Since $\mathbf{V} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_N)$, this implies that under H_u

$$Z \sim N(\frac{d_{uv}}{2} - \frac{\ln(p_v/p_u)}{d_{uv}}, 1) . \quad (12.108)$$

But $\mathbf{x}_v$ is preferred to $\mathbf{x}_u$ whenever $Z < 0$, and from the distribution (12.108) we find

$$P[Z < 0 | H_u] = Q\left(\frac{d_{uv}}{2} - \frac{\ln(p_v/p_u)}{d_{uv}} \right), \quad (12.109)$$

which proves the second line of (12.104).

In theory, to obtain an upper bound for the probability of error $P[E]$ given by (12.103), we need to characterize the simple error events $\mathcal{S}(t, \mathbf{x}_u)$ for *each* trellis path $\mathbf{x}_u$ and average the bound (12.104) for path $\mathbf{x}_u$ by weighting it with the path probability p_u. However, in some applications, such as decoding of convolutional codes, there exists a transformation that maps the set of simple error events for one path into the set of simple error events for any other path while, at the same time, preserving the distance d_{uv}. In such a case, if we

assume also that all trellis paths are equally likely, the probability of error admits the upper bound

$$P[E] \leq \sum_{\mathbf{x}_v \in \mathcal{S}(t,\mathbf{x}_u)} Q(d_{uv}/2)\,, \tag{12.110}$$

where $\mathbf{x}_u$ is an arbitrary reference path, which is usually selected as the all zero path for decoding of convolutional codes.

At this point it is worth noting that the set $\mathcal{S}(t,\mathbf{x}_u)$ contains paths that are translates of each other. For example, if we consider the simple error event $\mathbf{x}_v$ of Fig. 12.10, the path $\mathbf{x}_w$ obtained in Fig. 12.12 by shifting $\mathbf{x}_v$ to the right by one time unit belongs also to $\mathcal{S}(t,\mathbf{x}_u)$. But, since the mapping $h(x_t, x_{t+1})$ used to generate the output signal from a state transition (x_t, x_{t+1}) is time-invariant, time-shifting error events do not affect their distance to the reference path $\mathbf{x}_u$, i.e., $d_{uw} = d_{uv}$. So, once we have considered a simple error event of $\mathcal{S}(t,\mathbf{x}_u)$, it is unnecessary to examine its translates. Let $\mathbf{S}_r(t,\mathbf{x}_u)$ be the reduced set of error events obtained under translation equivalence. If $\mathbf{x}_v$ is an element of $\mathbf{S}_r(t,\mathbf{x}_u)$, let

$$l_{uv} = t_R - t_L - 1 \tag{12.111}$$

denote the length of time for which $\mathbf{x}_s^v \neq \mathbf{x}_s^u$. Then the upper bound (12.110) can be written as

$$P[E] \leq \sum_{\mathbf{x}_v \in \mathcal{S}_r(t,\mathbf{x}_u)} l_{uv} Q(d_{uv}/2)\,. \tag{12.112}$$

For high SNR, the union bound sum is dominated by the terms corresponding to simple error events $\mathbf{x}_v$ for which the normalized distance d_{uv} between $\mathbf{s}_v$ and $\mathbf{s}_u$ is minimum. These events form the *nearest neighbors* of $\mathbf{x}_u$. Let $d_{\min}$ denote the minimal normalized distance between $\mathbf{s}_u$ and signals $\mathbf{s}_v$ corresponding to simple error events $\mathbf{x}_v \in \mathcal{S}_r(t,\mathbf{x}_u)$. Then for high SNR (high $d_{\min}$), the upper bound (12.112) reduces to

$$P[E] \leq C_U Q(d_{\min}/2)\,. \tag{12.113}$$

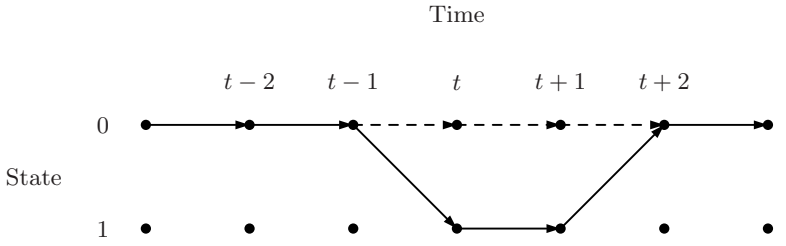

Fig. 12.12. Error event $\mathbf{x}_w$ obtained by shifting the event $\mathbf{x}_v$ of Fig. 12.10 to the right by one time unit.

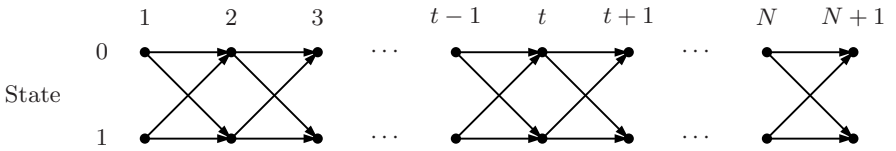

Fig. 12.13. Trellis for the MC representing the AMI line encoder.

where the constant C_U represents the sum of the lengths l_{uv} of all different nearest neighbors of $\mathbf{x}_u$ in $\mathcal{S}_r(t, \mathbf{x}_u)$.

Example 12.3, continued

Consider the trellis for the AMI line code shown in Fig. 12.13. For this example, there does not exist a transformation that maps an arbitrary trellis path $\mathbf{x}_u$ into the all zero state path in such a way that the scaled Euclidean distance d_{uv} between signal $\mathbf{s}_u$ corresponding to $\mathbf{x}_u$ and the signal $\mathbf{s}_v$ corresponding to another path $\mathbf{x}_v$ is preserved by the transformation. This is due to the fact that, unlike error-correcting codes where the observed signal $s(t)$ is generated by binary arithmetic, the AMI signal $s(t)$ given by (12.93) takes ternary values $\{1, 0, -1\}$. This means that the set $\mathcal{S}(t, \mathbf{x}_u)$ of simple error events and the subset of nearest neighbors of $\mathbf{x}_u$ depends on the reference path $\mathbf{x}_u$.

Specifically, let $\mathbf{x}_u$ be the dashed path in Fig. 12.14. For this path, the paths $\mathbf{x}_v$, $\mathbf{x}_w$ and $\mathbf{x}_q$ represented by solid trajectories in parts (a) to (c) of Fig. 12.14 correspond all to simple error events in $\mathcal{S}(t, \mathbf{x}_u)$. Over subinterval $[t, t+3]$, the signals corresponding to $\mathbf{x}_u$ and paths $\mathbf{x}_v$, $\mathbf{x}_w$ and $\mathbf{x}_q$ are given respectively by

$$\begin{bmatrix} s_u(t) \\ s_u(t+1) \\ s_u(t+2) \\ s_u(t+3) \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \\ -1 \end{bmatrix} \quad , \quad \begin{bmatrix} s_v(t) \\ s_v(t+1) \\ s_v(t+2) \\ s_v(t+3) \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} s_w(t) \\ s_w(t+1) \\ s_w(t+2) \\ s_w(t+3) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \quad , \quad \begin{bmatrix} s_q(t) \\ s_q(t+1) \\ s_q(t+2) \\ s_q(t+3) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} ,$$

and

$$s_u(r) = s_v(r) = s_w(r) = s_q(r)$$

for all values of r not in $[t, t+3]$. We find therefore that

$$d_{\min} = d_{uv} = d_{uw} = \frac{\sqrt{2}}{\sigma}$$

and

$$d_{uq} = \frac{\sqrt{6}}{\sigma} ,$$

so that both paths $\mathbf{x}_v$ and $\mathbf{x}_w$ are nearest neighbors of $\mathbf{x}_u$, but $\mathbf{x}_q$ is not a nearest neighbor even though it belongs to the set $\mathcal{S}(t, \mathbf{x}_u)$ of simple error events.

More generally, for an arbitrary path $\mathbf{x}_u$, the path $\mathbf{x}_v$ such that $x_r^u = x_r^v$ for $r \neq t$ and $x_t^v = 1 - x_t^u$ is a nearest neighbor at distance $d_{\min} = \sqrt{2}/\sigma$, but it is not necessarily the only nearest neighbor. For example, for the path $\mathbf{x}_u$

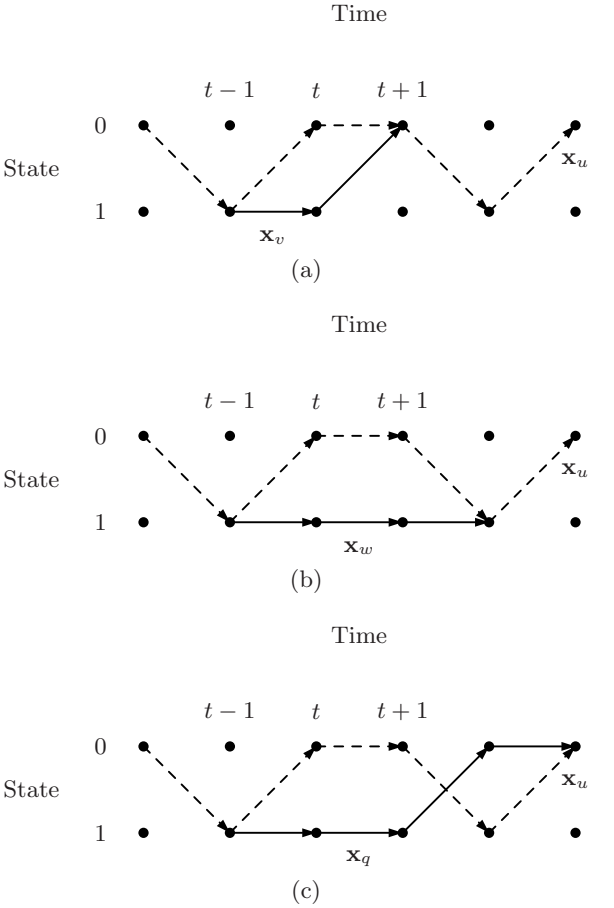

Fig. 12.14. Reference path $\mathbf{x}_u$ and (a) nearest neighbor path $\mathbf{x}_v$, (b) nearest neighbor path $\mathbf{x}_w$, and (c) simple error event $\mathbf{x}_q$.

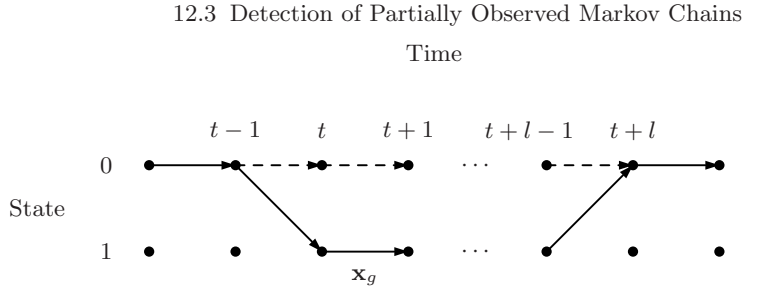

Fig. 12.15. Nearest neighbor path $\mathbf{x}_g$ of the zero path $\mathbf{0}$ specified by (12.114).

of Fig. 12.14, the only two nearest neighbors are $\mathbf{x}_v$ and $\mathbf{x}_w$. However, if the true path is the all zero path $\mathbf{0}$, then the path

$$x_r^g = \begin{cases} 0 & r \le t_L = t-1 \\ 1 & t \le r \le t+l-1 \\ 0 & t+l \le r \,, \end{cases} \tag{12.114}$$

which is depicted in Fig. 12.15 is a nearest neighbor of $\mathbf{0}$. Observe indeed that the signal corresponding to $\mathbf{x}_g$ is given by

$$s_g(r) = \begin{cases} 1 & r = t-1 \\ -1 & r = t+l-1 \\ 0 & \text{otherwise} \,, \end{cases}$$

so the normalized distance between $\mathbf{s}_g$ and the vector $\mathbf{0}$ representing the all zero path is $d_{\min} = \sqrt{2}/\sigma$. Note that in the specification (12.114), the merge time $t+l$ is such that $l > 1$ is arbitrary. Furthermore, all translates of $\mathbf{x}_g$ to the left by less than l steps are also nearest neighbors.

So for the AMI line code, the number of nearest neighbors in $\mathcal{S}(t, \mathbf{x}_u)$ depends on the reference path $\mathbf{x}_u$, but each path $\mathbf{x}_u$ admits at least one nearest neighbor at distance $d_{\min} = \sqrt{2}/\sigma$. This implies that at high SNR, the probability of error admits an upper bound of the form (12.113), where the constant C_U is obtained by averaging the number of neighbors for each path $\mathbf{x}_u$ with the path probability p_u. In this bound the normalized distance $d_{\min}$ is $\sqrt{2}$ larger than the distance $1/\sigma$ controlling the probability of error

$$P_E^{\text{AMI}} = \frac{3}{2} Q(1/(2\sigma))$$

for the usual AMI detector formed by the ternary slicer of Fig. 12.7 followed by a mapper. This $\sqrt{2}$ increase in signal space distance explains the 3dB improvement of the Viterbi decoder over the standard AMI decoder observed in [23]. □

12.3.2 Pointwise MAP Detection

Let

$$\mathbf{X} = \big[\, X(1) \cdots X(t) \cdots X(N+1) \,\big]^T \tag{12.115}$$

be the random vector representing the trajectory of the MC over interval $1 \leq t \leq N+1$ and let $\{\mathbf{x}_u,\, 1 \leq u \leq U\}$ denote the vectors representing the different trellis trajectories. The MAP sequence detection problem considered in the last subsection finds the trellis sequence

$$\hat{\mathbf{x}}^{\mathrm{MAP}} = \arg \max_{\mathbf{x}_u \in \mathcal{U}} P[\mathbf{X} = \mathbf{x}_u | \mathbf{y}] \tag{12.116}$$

that maximizes the a-posteriori probability of the entire MC trajectory given observation vector $\mathbf{y}$. However, instead of maximizing the a-posteriori probability for an entire trajectory, it is also possible to perform the maximization at each time instant. This leads to a K-ary detection problem, where at time t, we seek to find the state

$$\hat{x}_t^{\mathrm{P}} = \arg \max_{1 \leq k \leq K} P[X(t) = k | \mathbf{y}] \tag{12.117}$$

maximizing the a-posteriori distribution of state $X(t)$ given $\mathbf{y}$.

At this point it is important to recognize that the two problems (12.116) and (12.117) are different. Specifically, the trajectory

$$\hat{\mathbf{x}}^{\mathrm{P}} \stackrel{\triangle}{=} (\hat{x}_t^{\mathrm{P}},\, 1 \leq t \leq N+1) \tag{12.118}$$

traced by performing the maximization (12.117) for each t need not coincide with solution $\hat{\mathbf{x}}^{MAP}$ of the pathwise maximization. In fact $\hat{\mathbf{x}}^{\mathrm{P}}$ need not form a trellis trajectory of $\mathcal{U}$, since it may contain some transitions that are not allowed, i.e., which are assigned a zero probability by the one-step transition probability distribution $\mathbf{P}$. This is, of course, rare, but this phenomenon may occur at low SNR.

Forward–backward algorithm: The pointwise MAP estimate can be evaluated by using the forward-backward or BCJR algorithm which was introduced independently by Baum et al. [25] and by Bahl et al. [26]. The first step in the derivation of this algorithm consists in recognizing that the maximization of the discrete a-posteriori probability distribution

$$p_{X(t)|\mathbf{Y}}(k, t|\mathbf{y}) \stackrel{\triangle}{=} P[X(t) = k | \mathbf{Y} = \mathbf{y}]$$

is equivalent to the maximization of the joint hybrid probability distribution/density

$$p_{X(t),\mathbf{Y}}(k, t; \mathbf{y}) \stackrel{\triangle}{=} p_{X(t)|\mathbf{Y}}(k, t|\mathbf{y}) f_{\mathbf{Y}}(\mathbf{y}) \tag{12.119}$$

where $f_{\mathbf{Y}}(\mathbf{y})$ denotes the probability density of the observation vector $\mathbf{Y}$ which, as shown in (12.72), regroups all observations over the interval $1 \leq t \leq N$. This vector can be split into the vectors

$$\begin{aligned} \mathbf{Y}_t^- &= \big[\, Y(1)\; Y(2) \cdots\; Y(t-1) \,\big]^T \\ \mathbf{Y}_t^+ &= \big[\, Y(t)\; Y(t+1) \cdots\; Y(N) \,\big]^T \end{aligned} \tag{12.120}$$

representing the observations in the past and future of the current time t. Note that because $Y(t)$ depends on the transition between states $X(t)$ and $X(t+1)$, it must be allocated to the future, not the past, of time t. Then the hybrid distribution/density (12.119) can be denoted as $p_{X(t),\mathbf{Y}_t^-,\mathbf{Y}_t^+}(k,t;\mathbf{y}_t^-,\mathbf{y}_t^+)$, and by conditioning jointly with respect to $X(t)$ and $\mathbf{Y}_t^-$, we obtain

$$p_{X(t),\mathbf{Y}_t^-,\mathbf{Y}_t^+}(k,t;\mathbf{y}_t^-,\mathbf{y}_t^+) = f_{\mathbf{Y}_t^+|X(t),\mathbf{Y}_t^-}(\mathbf{y}_t^+|k,t;\mathbf{y}_t^-) \cdot p_{X(t),\mathbf{Y}_t^-}(k,t;\mathbf{y}_t^-)\,. \tag{12.121}$$

But since $X(t)$ is a Markov process, its future and past are conditionally independent given its present value. Consequently, given $X(t)$, $\mathbf{Y}_t^+$ and $\mathbf{Y}_t^-$ are conditionally independent, so we have

$$f_{\mathbf{Y}_t^+|X(t),\mathbf{Y}_t^-}(\mathbf{y}_t^+|k,t;\mathbf{y}_t^-) = f_{\mathbf{Y}_t^+|X(t)}(\mathbf{y}_t^+|k,t)\,. \tag{12.122}$$

Let us denote

$$\alpha(k,t) \triangleq p_{X(t),\mathbf{Y}_t^-}(k,t;\mathbf{y}_t^-) \tag{12.123}$$

$$\beta(k,t) \triangleq f_{\mathbf{Y}_t^+|X(t)}(\mathbf{y}_t^+|k,t) \tag{12.124}$$

with $1 \le k \le K$, where for simplicity, the observation vectors $\mathbf{y}_t^-$ and $\mathbf{y}_t^+$ have been dropped from the arguments of α and β, since they are known. The factorization (12.121)–(12.122) can be rewritten as

$$\begin{aligned} P(k,t) &\triangleq P[X(t)=k|\mathbf{Y}=\mathbf{y}]f_{\mathbf{Y}}(\mathbf{y}) \\ &= \alpha(k,t)\beta(k,t)\,, \end{aligned} \tag{12.125}$$

so that the pointwise MAP estimate can be expressed as

$$\hat{x}_t^{\mathrm{P}} = \arg \max_{1\le k\le K} (\alpha(k,t)\beta(k,t))\,. \tag{12.126}$$

It turns out that the discrete valued functions $\alpha(k,t)$ and $\beta(k,t)$ admit forward and backward recursions, respectively, which justify the name of the algorithm. To derive the forward recursion for $\alpha(k,t)$, observe that the joint hybrid probability distribution/density of $X(t)$, $X(t+1)$, and

$$\mathbf{Y}_{t+1}^- = \begin{bmatrix} \mathbf{Y}_t^- \\ Y(t) \end{bmatrix}$$

admits the factorization

$$p_{X(t),X(t+1),\mathbf{Y}_{t+1}^-}(k,\ell,t;\mathbf{y}_{t+1}^-) = f_{Y(t)|X(t),X(t+1)}(y_t|k,\ell) \cdot p_{X(t+1)|X(t)}(\ell|k)p_{X(t),\mathbf{Y}_t^-}(k,t;\mathbf{y}_t^-)\,, \tag{12.127}$$

where we have used again the fact that the Markov property of $X(t)$ implies that $X(t+1)$ and $\mathbf{Y}_t^-$ are conditionally independent given $X(t)$. In this factorization, we recognize $p_{k\ell} = p_{X(t+1)|X(t)}(\ell|k)$ and the observation model (12.65) implies

$$f_{Y(t)|X(t),X(t+1)}(y_t|k,\ell) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\big(-\frac{1}{2\sigma^2}(y_t - h(k,\ell))^2\big) . \quad (12.128)$$

Taking into account the definition (12.123) of $\alpha(k,t)$, by marginalizing the distribution $p_{X(t),X(t+1),\mathbf{Y}_{t+1}^-}$ with respect to $X(t)$, we obtain

$$\alpha(\ell,t+1) = \frac{1}{(2\pi\sigma^2)^{1/2}} \sum_{k=1}^{K} \alpha(k,t) p_{k\ell} \exp\big(-\frac{1}{2\sigma^2}(y_t - h(k,\ell))^2\big) . \quad (12.129)$$

The recursion (12.129) can be used to propagate the probability distribution $\alpha(k,t)$ forward in time, starting with the initial condition

$$\alpha(k,1) = \pi_k(1) , \quad (12.130)$$

where $\pi_k(1) = P[X(1) = k]$ is the initial probability distribution of the MC.

Similarly, by observing that the vector $\mathbf{Y}_t^+$ admits the partition

$$\mathbf{Y}_t^+ = \begin{bmatrix} Y(t) \\ \mathbf{Y}_{t+1}^+ \end{bmatrix}$$

we find that the hybrid conditional probability distribution/density of $X(t+1)$ and $\mathbf{Y}_t^+$ given $X(t)$ can be factored as

$$\begin{aligned} p_{X(t+1),\mathbf{Y}_t^+|X(t)}(\ell;\mathbf{y}_t^+|k,t) &= f_{\mathbf{Y}_{t+1}^+|X(t),X(t+1),Y(t)}(\mathbf{y}_{t+1}^+|k,\ell,t;y_t) \\ &\cdot f_{Y(t)|X(t),X(t+1)}(y_t|k,\ell) p_{X(t+1)|X(t)}(\ell|k) . \end{aligned} \quad (12.131)$$

Again, the Markov property of $X(t)$ implies that given $X(t+1)$, the future observations $\mathbf{Y}_{t+1}^+$ are conditionally independent of the previous state $X(t)$ and observation $Y(t)$, so in (12.131) we have

$$f_{\mathbf{Y}_{t+1}^+|X(t),X(t+1),Y(t)}(\mathbf{y}_{t+1}^+|k,\ell,t;y_t) = f_{\mathbf{Y}_{t+1}^+|X(t+1)}(\mathbf{y}_{t+1}^+|\ell,t) . \quad (12.132)$$

Recognizing that this last function is $\beta(\ell,t+1)$, by marginalizing (12.131) with respect to $X(t+1)$, we obtain

$$\beta(k,t) = \frac{1}{(2\pi\sigma^2)^{1/2}} \sum_{\ell=1}^{K} \beta(\ell,t+1) p_{k\ell} \exp\big(-\frac{1}{2\sigma^2}(y_t - h(k,\ell))^2\big) , \quad (12.133)$$

which can be used to propagate $\beta(k,t)$ backwards in time. The initial condition for this recursion can be selected as

$$\beta(\ell, N+1) = 1 \tag{12.134}$$

for all ℓ. To justify this choice, observe that

$$\begin{aligned}\beta(k,N) &= \sum_{\ell=1}^{K} p_{X(N+1),Y(N)|X(N)}(\ell, y_N|k,N) \\ &= \sum_{\ell=1}^{K} f_{Y(N)|X(N),X(N+1)}(y_N|k,\ell,N) p_{X(N+1)|X(N)}(\ell|k) \\ &= \frac{1}{(2\pi\sigma^2)^{1/2}} \sum_{\ell=1}^{K} p_{k\ell} \exp\big(-\frac{1}{2\sigma^2}(y_N - h(k,\ell))^2\big) \end{aligned} \tag{12.135}$$

coincides with recursion (12.133) for $t = N$, provided $\beta(\ell, N+1)$ satisfies (12.134).

It is interesting to note that the forward and backward recursions (12.129) and (12.133) admit a compact matrix representation. Let $\mathbf{M}(t)$ be the $K \times K$ matrix whose (k,ℓ)-th entry is given by

$$m_{k\ell}(t) = \begin{cases} \dfrac{1}{(2\pi\sigma^2)^{1/2}} \exp\big(-\dfrac{c(k,\ell,t)}{2\sigma^2}\big) & \text{for } (k,\ell) \in \mathcal{T} \\ 0 & \text{otherwise}\,, \end{cases} \tag{12.136}$$

where $c(k,\ell,t)$ is the path metric defined in (12.80). Note that all entries of $\mathbf{M}(t)$ are nonnegative, and $\mathbf{M}(t)$ has the same graph as $\mathbf{P}$. Let also

$$\boldsymbol{\alpha}(t) = \big[\, \alpha(1,t) \cdots \alpha(k,t) \cdots \alpha(K,t) \,\big] \tag{12.137}$$

denote the $1 \times K$ row vector representing the discrete function $\alpha(\cdot, t)$. Then the forward recursion (12.129)–(12.130) can be rewritten in matrix form as

$$\boldsymbol{\alpha}(t+1) = \boldsymbol{\alpha}(t)\mathbf{M}(t) \tag{12.138}$$

with

$$\boldsymbol{\alpha}(1) = \boldsymbol{\pi}(1)\,. \tag{12.139}$$

Similarly, let

$$\boldsymbol{\beta}(t) = \big[\, \beta(1,t) \cdots \beta(k,t) \cdots \beta(K,t) \,\big]^T \tag{12.140}$$

denote the $K \times 1$ column vector representing the function $\beta(\cdot, t)$. Then the backward recursion (12.133) can be expressed as

$$\boldsymbol{\beta}(t) = \mathbf{M}(t)\boldsymbol{\beta}(t+1) \tag{12.141}$$

with initial condition

$$\boldsymbol{\beta}(N+1) = \mathbf{e}\,, \tag{12.142}$$

where $\mathbf{e}$ is the vector whose entries are all one, as shown in (12.4). In this respect, it is worth observing that the forward and backward recursions can be expressed in terms of the same matrix $\mathbf{M}(t)$, but the forward recursion involves row vectors, whereas the backward expression involves column vectors.

By observing that the a-posteriori probability distribution $P[X(t) = k|\mathbf{y}]$ must be normalized, we find from (12.125) that it can be expressed as

$$P[X(t) = k|\mathbf{y}] = \alpha_k(t)\beta_k(t)/\gamma(t) \tag{12.143}$$

where the normalizing constant $\gamma(t)$ is the row times column inner product

$$\gamma(t) = \boldsymbol{\alpha}(t)\boldsymbol{\beta}(t) = \sum_{k=1}^{K} \alpha_k(t)\beta_k(t) \, . \tag{12.144}$$

Substituting recursions (12.138) and (12.141) inside (12.144) yields

$$\gamma(t) = \boldsymbol{\alpha}(t)\mathbf{M}(t)\boldsymbol{\beta}(t+1) = \boldsymbol{\alpha}(t+1)\boldsymbol{\beta}(t+1) = \gamma(t+1) \, , \tag{12.145}$$

so γ is actually constant.

Computational complexity: Assume that the MC trellis is regular, so that the same number T of transitions is allowed out of each state. For example, for the MC of Fig. 12.4, $T = 2$. Then the matrix notation (12.138), (12.141) for the forward and backward recursions indicates that KT scalar multiplications are required to advance vectors $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\beta}(t)$ by one time unit. In contrast, each stage of the Viterbi algorithm requires implementing the ACS operation (12.85) for each state, so that $O(K)$ operations are required by the Viterbi algorithm. Thus the Viterbi algorithm has a factor of T advantage in computational complexity over the forward-backward algorithm. When the transition matrix $\mathbf{P}$ is dense, i.e., when $T = K$, so that all state transitions are allowed, this difference can be significant. This explains why, even though the Viterbi and forward-backward algorithms were discovered during the same time frame, for a long time the Viterbi algorithm was preferred. In this respect it is useful to observe that, because it maximizes the a-posteriori probability $P[X(t) = k|\mathbf{Y}]$, the probability of error of the pointwise MAP estimate $\hat{X}_t^{\mathrm{P}}$ is slighly lower than that of the t-th component $\hat{X}_t^{\mathrm{MAP}}$ of the sequence estimate $\hat{\mathbf{X}}_{\mathrm{MAP}}$, i.e.,

$$P[\hat{X}_t^{\mathrm{P}} \neq X(t)] \leq P[\hat{X}_t^{\mathrm{MAP}} \neq X(t)] \, .$$

However this advantage is very small, typically a small fraction of a dB. So in spite of its slight performance advantage, until recently, the forward-backward algorithm was rarely implemented in practical systems, due to its higher computational cost. This situation changed overnight when it was discovered [2,27] that iterative equalization and/or decoding algorithms can offer an advantage of several dBs over systems that treat equalization and decoding, or decoding stages of concatenated codes, as separate noninteracting blocks. The key feature of iterative decoding algorithms is that they require the exchange of soft

information, representing a relative likelihood measure of different state values, based on the given observations. Since the forward–backward algorithm computes explicitly the a-posteriori distribution of the different state values given the observation vector $\mathbf{Y}$, it is ideally suited for the implementation of iterative decoding algorithms. By comparison, the classical Viterbi algorithm does not provide an explicit reliability estimate for the values of the states at different times based on $\mathbf{Y}$. However, Hagenauer and Hoeher [28] introduced a variant of the Viterbi algorithm called the soft Viterbi algorithm (SOVA), which computes state reliability values by performing a backward recursion over a time window selected to ensure that survivors have merged. But these reliability values are only rough approximations of the a-posteriori probability distribution $P[X(t) = k|\mathbf{Y}]$, so it is often preferable to implement the forward-backward algorithm in spite of its slightly larger computational load.

Max-log-MAP algorithm: Since the a-posteriori state probability distribution is obtained by performing the normalization (12.143), the common scale factor $(2\pi\sigma^2)^{-1/2}$ can be removed from the forward and backward recursions (12.129) and (12.133). Then if we denote

$$V_f(k,t) \triangleq -2\sigma^2 \ln(\alpha(k,t)) \tag{12.146}$$

$$V_b(k,t) \triangleq -2\sigma^2 \ln(\beta(k,t)) \,, \tag{12.147}$$

the recursions (12.129) and (12.133) can be rewritten in logarithmic form as

$$V_f(\ell, t+1) = \\ -2\sigma^2 \ln\Big(\sum_{k:\,(k,\ell)\in\mathcal{T}} \exp\big(-\frac{1}{2\sigma^2}(V_f(k,t) + c(k,\ell,t))\big)\Big) \tag{12.148}$$

$$V_b(k,t) = \\ -2\sigma^2 \ln\Big(\sum_{\ell:\,(k,\ell)\in\mathcal{T}} \exp\big(-\frac{1}{2\sigma^2}(V_b(\ell,t+1) + c(k,\ell,t))\big)\Big) \,. \tag{12.149}$$

These recursions specify the log-MAP algorithm, which is strictly equivalent to the forward-backward algorithm. The a-posteriori probability distribution (12.143) can be expressed in terms of the forward and backward value functions V_f and V_b as

$$P[X(t) = k|\mathbf{y}] = \frac{\exp(-(V_f(k,t) + V_b(k,t))/(2\sigma^2))}{\sum_{\ell=1}^{K} \exp(-(V_f(\ell,t) + V_b(\ell,t))/(2\sigma^2)} \,. \tag{12.150}$$

Note that, strictly speaking, the functions $V_f(k,t)$ and $V_b(k,t)$ defined by (12.146) and (12.137) are the *negative* of the scaled logarithms of forward and backward components $\alpha(k,t)$ and $\beta(b,t)$ of the a-posteriori probability $P[X(t) = k|\mathbf{y}]$ of state k at time t. So, applying minimization operations to

these functions is performing a maximization on the logarithmic components of the a-posteriori probability. At this point, it is convenient to introduce the $\min^*$ function which for n positive real numbers x_i is defined as

$$\min^*(x_1, \cdots, x_n) \stackrel{\triangle}{=} -\ln\big(\sum_{i=1}^{n} \exp(-x_i)\big) . \tag{12.151}$$

This function satisfies the properties

$$\min^*(x, y) = \min(x, y) - \ln(1 + \exp(-|x - y|)) \tag{12.152}$$

$$\min^*(x, y, z) = \min^*(\min^*(x, y), z) . \tag{12.153}$$

With this notation, the log-MAP algorithm (12.148)–(12.149) can be written as

$$V_f(\ell, t+1) = 2\sigma^2 \min^*_{k:\,(k,\ell)\in\mathcal{T}}\big(\frac{1}{2\sigma^2}(V_f(k, t) + c(k, \ell, t))\big) \tag{12.154}$$

$$V_b(k, t) = 2\sigma^2 \min^*_{\ell:\,(k,\ell)\in\mathcal{T}}\big(\frac{1}{2\sigma^2}(V_b(\ell, t+1) + c(k, \ell, t))\big) , \tag{12.155}$$

where the forward and backward recursions are formally identical to Viterbi algorithms, except that the min operator in the usual Viterbi recursion is replaced by the operator $\min^*$. But the decomposition (12.152) indicates that whenever the arguments of the $\min^*$ operator are sufficiently far apart, the $\min^*$ operator can be approximated by the min operator. By applying this approximation to recursions (12.154)–(12.155) and noting that for an arbitrary positive scale factor s

$$s \min(\frac{x}{s}, \frac{y}{s}) = \min(x, y)$$

we obtain the max-log-MAP algorithm

$$V_f(\ell, t+1) = \min_{k:\,(k,\ell)\in\mathcal{T}}(V_f(k, t) + c(k, \ell, t)) \tag{12.156}$$

$$V_b(k, t) = \min_{\ell:\,(k,\ell)\in\mathcal{T}}(V_b(\ell, t+1) + c(k, \ell, t)) , \tag{12.157}$$

which is now expressed in the form of forward and backward Viterbi algorithms for the MC trellis with path metric (12.80). So although the forward-backward/log-MAP algorithm has a higher complexity than the Viterbi algorithm, by approximating the $\min^*$ operator by a min operator, it is possible to approximate the log-MAP algorithm by the max-log-MAP algorithm, whose complexity is only twice that of the Viterbi algorithm. It was shown in [29] that the max-log-MAP algorithm is identical to a form of the soft Viterbi algorithm proposed by Battail [30].

Remark: Note that, strictly speaking, the functions $V_f(k, t)$ and $V_b(k, t)$ defined by (12.146) and (12.137) are the *negative* of the scaled logarithms of forward and backward components $\alpha(k, t)$ and $\beta(b, t)$ of the a-posteriori probability $P[X(t) = k|\mathbf{y}]$ of state k at time t. So, even though recursions (12.156

and (12.157) are expressed in terms of minimization operations, they really represent maximimizations of logarithmic components of the a-posteriori state probability, which explains why the algorithm is called the max-log-MAP algorithm.

12.4 Example: Channel Equalization

To illustrate the detection of partially observed Markov chains, we consider in this section the MAP sequence equalization of a binary pulse amplitude modulated (PAM) signal transmitted though a linear ISI channel. The DT observation signal at the output of the receiver front-end can be expressed as

$$Y(t) = s(t) + V(t) \tag{12.158}$$

with

$$s(t) = \sum_{k=0}^{L} h_k I(t-k)\,, \tag{12.159}$$

where the transmitted symbols $\{I(t),\, t \geq 1\}$ are independent identically distributed and take the values ± 1 with probability $1/2$. The noise $V(t)$ is a zero mean WGN sequence with variance σ^2. Note that the design of an analog receiver front-end leading to a DT model of the form (12.158)–(2.159) is not a trivial matter. It was shown by Forney [14] that an optimal structure consists of applying a CT filter matched to the received pulse (the convolution of the channel impulse response with the transmit pulse), followed by a baud rate sampler and a DT noise whitening filter. For transmission systems such that the received pulse has less than 100% excess bandwidth, i.e., with a bandwidth less than the baud frequency, another optimal front-end structure [31] consists of passing the signal through a lowpass filter with bandwidth equal to the baud frequency, followed by a Nyquist sampler with sampling period $T/2$, where T denotes the baud period. The advantage of this structure is that it remains optimal when the channel is not known exactly [32]. However, in this case the observation $Y(t)$ and noise $V(t)$, as well as the impulse response h_k are 2-dimensional vectors. Also, the variance of the additive WGN is twice as large as for the whitened matched filter front end receiver, due to the 100% excess bandwidth of the analog front end filter. For simplicity, we restrict our attention to the case when $Y(t)$ is scalar, leaving the extension to the case of vector observations as an excercise for the reader.

12.4.1 Markov Chain Model

Note that the transmitted symbol sequence $\{I(t)\}$ is in one-to-one correspondence with a bit stream $\{B(t)\}$ where

$$B(t) = \frac{1}{2}(I(t)+1) \tag{12.160}$$

is either zero or one, so with respect to the bit sequence $\{B(t)\}$, the ISI signal $s(t)$ can be expressed as

$$s(t) = 2\sum_{k=0}^{L} h_k B(t-k) - \sum_{k=0}^{L} h_k \,. \tag{12.161}$$

In the model (12.159), it is assumed that L is finite, so that the channel transfer function

$$H(z) = \sum_{k=0}^{L} h_k z^{-k} \tag{12.162}$$

is FIR. This property plays an essential role in the constructing of an MC model of equation (12.159), and thus in formulating the equalization problem as an MAP sequence detection problem. In practice, the sampled channel h_k does not have finite length, but it decays sufficiently rapidly to allow the truncation of its tail.

The Markov chain state $X(t)$ corresponding to the observation model (12.159) represents an encoding of the symbols $I(t-k)$ (or equivalently the bits $B(t-k)$) with $1 \le k \le L$ stored in the channel. For example, we can use for $X(t)$ the binary expansion

$$\begin{aligned} X(t) &= 1 + \sum_{k=1}^{L} 2^{k-1} B(t-k) \\ &= \frac{1}{2}[1 + 2^L + \sum_{k=1}^{L} 2^{k-1} I(t-k)] \,. \end{aligned} \tag{12.163}$$

Under this mapping, $X(t)$ takes values in the set $\mathcal{S} = \{1, \cdots, 2^L\}$, so the total number of states $K = 2^L$ grows exponentially with the order L of the FIR filter $H(z)$. Furthermore, it can be verified from (12.163) that, if symbols $(i_{t-1}, i_{t-2}, \cdots, i_{t-L})$ are mapped into state x_t, then the opposite symbols $(-i_{t-1}, -i_{t-2}, \cdots, -i_{t-L})$ are mapped into state $1 + 2^L - x_t$. In other words, reversing the sign of the symbols stored in the channel maps the state x_t into its mirror image with respect to the center $(1 + 2^L)/2$ of the state labels. Observe also that the latest bit $B(t)$ does not appear in the encoding of $X(t)$, but in the encoding of $X(t+1)$, and the state dynamics can be expressed as

$$X(t+1) = 1 + [2(X(t) - 1) + B(t)] \bmod 2^L \,, \tag{12.164}$$

which clearly indicates that $X(t)$ is a Markov process. The recursion (12.164) just implements a shift register, so as a new bit $B(t)$ enters the register, the oldest bit $B(t-L)$ is pushed out. Note that, since $B(t)$ is binary, there are only $T = 2$ transitions out of each state, each with probability $1/2$, thus all trellis paths are equally likely.

Table 12.2. State mapping for $L = 2$.

State	Bits $B(t-1)$, $B(t-2)$	Symbols $I(t-1)$, $I(t-2)$
1	0 , 0	-1, -1
2	1 , 0	1, -1
3	0, 1	-1, 1
4	1, 1	1 , 1

Example 12.4

For $L = 2$ the mapping (12.163) is illustrated in Table 12.2 where the first column lists all four possible states, the second column lists the corresponding bits ($B(t-1)$, $B(t-2)$), and the third column the matching symbols ($I(t-1)$, $I(t-2)$).
In this case, the state transition diagram corresponding to dynamics (12.164) is shown in Fig. 12.16. Each transition is labeled by the corresponding value of $B(t)$. The one-step transition matrix for the corresponding MC is given by

$$\mathbf{P} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix},$$

and it is easy to verify that the MC is irreducible and aperiodic. □

Since the state $X(t)$ encodes the transmitted symbols $I(t-k)$ with $1 \leq k \leq L$, the ISI signal

$$s(t) = \sum_{k=0}^{L} h_k I(t-k) = h(X(t), X(t+1)) \tag{12.165}$$

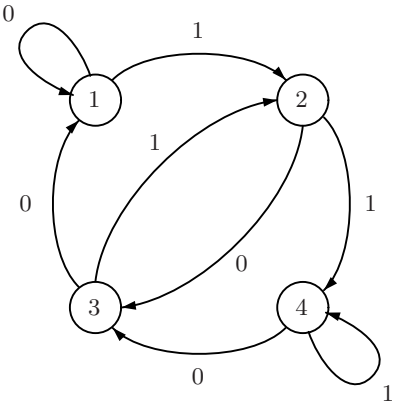

Fig. 12.16. State transition diagram of the MC modelling the channel memory.

is clearly a function of state transitions, so the equalization problem can be viewed as a partially observed MC detection problem.

Example 12.4, continued

For the case when

$$H(z) = 0.8 - 0.6z^{-1} + 0.1z^{-2}\,, \tag{12.166}$$

the matrix $\mathbf{H}$ with entries $h(k,\ell)$ for $1 \le k,\, \ell \le 4$ is given by

$$\mathbf{H} = \begin{bmatrix} -0.3 & 1.3 & x & x \\ x & x & -1.5 & 0.1 \\ -0.1 & 1.5 & x & x \\ x & x & -1.3 & 0.3 \end{bmatrix},$$

where the entries labeled with x's correspond to forbidden transitions. Note that the matrix $\mathbf{H}$ is skew-centrosymmetric, which means that entries of $\mathbf{H}$ on opposite sides of its center have an opposite value. This property is due to the mirror image symmetry of the state labeling (12.163) under symbol sign reversal, and to the fact that, if the symbols $\{i_t,\, i_{t-1},\, i_{t-2}\}$ produce an output

$$s_t = h_0 i_t + h_1 i_{t-1} + h_2 i_{t-2},$$

the opposite symbols $\{-i_t,\, -i_{t-1},\, -i_{t-2}\}$ will produce output $-s_t$. □

12.4.2 Performance Analysis

Consider now the error analysis of MAP sequence equalization. Unlike the decoding problem of convolutional codes, given an MC trellis path $\mathbf{x}_u$ and its set $\mathcal{S}(t, \mathbf{x}_u)$ of simple error events for an arbitrary time t, there does not exist a mapping which transforms $\mathbf{x}_u$ and $\mathcal{S}(t, \mathbf{x}_u)$ into any other trellis path $\mathbf{x}_w$ and its set of simple error events $\mathcal{S}(t, \mathbf{x}_w)$ in such a way that distances d_{uv} between trellis paths are preserved. Since each trellis path is equally likely with probability

$$p_u = 1/U\,,$$

by combining (12.102) and (12.104), the upper bound for the probability of error can be expressed as

$$P[E] \le \frac{1}{U} \sum_{u=1}^{U} \sum_{\mathbf{x}_v \in \mathcal{S}(t,\mathbf{x}_u)} Q\left(\frac{d_{uv}}{2}\right). \tag{12.167}$$

Thus in theory, to characterize the performance of MAP sequence detection, for each reference path $\mathbf{x}_u$, we would need to evaluate the distance between $\mathbf{s}_u$ and each signal $\mathbf{s}_v$ corresponding to a path $\mathbf{x}_v$ in the simple error event set $\mathcal{S}(t, \mathbf{x}_u)$. This represents an unrealistically large amount of computation. Fortunately, for high SNR, the bound becomes

$$P[E] \leq C_U Q(\frac{d_{\min}}{2}) \,, \tag{12.168}$$

where $d_{\min}$ is the minimum normalized distance betwen signals $\mathbf{s}_u$ and $\mathbf{s}_v$ corresponding to paths $\mathbf{x}_u$ and $\mathbf{x}_v$ which differ from each other in a connected pattern.

Due to the linearity of expressions (12.159) and (12.161) relating the ISI signal $s(t)$ to the corresponding symbols $\{I(t)\}$ or bits $\{B(t)\}$, it turns out that the evaluation of $d_{\min}$ can be performed without examining each reference path $\mathbf{x}_u$. Specifically, let $\{B_u(t)\}$ and $\{B_v(t)\}$ denote respectively the bit sequences corresponding to paths $\mathbf{x}_u$ and $\mathbf{x}_v$. According to (12.161) the signal difference

$$s_u(t) - s_v(t) = 2 \sum_{k=0}^{L} h_k D(t-k) \,, \tag{12.169}$$

where

$$D(t) = B_u(t) - B_v(t) \tag{12.170}$$

denotes the difference between the two bit sequences. Since $B_u(t)$ and $B_v(t)$ take values 0 or 1, $D(t)$ takes values in the ternary set $\{1, 0, -1\}$.

Therefore, the squared minimum normalized distance $d^2_{\min}$ can be expressed as

$$d^2_{\min} = \frac{4}{\sigma^2} \min_{D(\cdot)\in\mathcal{D}} \sum_{t=1}^{\infty} \Big(\sum_{k=0}^{L} h_k D(t-k) \Big)^2 \,, \tag{12.171}$$

where $\mathcal{D}$ denotes the set of nonzero sequences $\{D(t),\, t \geq 1\}$. Without loss of generality, since the ISI channel is time-invariant, we can assume $D(t) \equiv 0$ for $t \leq 0$ and $D(1) \neq 0$ in (12.171). The minimization (12.171) can be accomplished by associating to it an enlarged state trellis, and then recognizing that the resulting trellis minimization is a shortest path problem.

Since $D(t)$ takes values over $\{-1, 0, 1\}$, the number of states needed to encode the values $D(t-k)$ for $1 \leq t \leq L$ is 3^L instead of 2^L for the state mapping (12.163). For instance, we can use the mapping

$$\begin{aligned} \xi(t) &= 1 + \sum_{k=1}^{L} 3^{k-1} (D(t-k) + 1) \\ &= \frac{1}{2}(1 + 3^L) + \sum_{k=1}^{L} 3^{k-1} D(t-k) \,. \end{aligned} \tag{12.172}$$

Note again that the mapping (12.172) is such that if state ξ_t encodes $\{d_{t-1}, d_{t-2}, \cdots, d_{t-L}\}$, the sign reversed sequence $\{-d_{t-1}, -d_{t-2}, \cdots, -d_{t-L}\}$ is represented by state $1 + 3^L - \xi_t$, so the mapping is symmetric with respect to the middle label under sign reversal of the difference values $D(t-k)$ for $1 \leq k \leq L$.

The difference $D(t)$ does not appear in the encoding of $\xi(t)$, but in the encoding of $\xi(t+1)$, and the state transitions are specified by the recursion

$$\xi(t+1) = 1 + [3\xi(t) + D(t) - 2] \mod 3^L \,, \tag{12.173}$$

so the dynamics of $\xi(t)$ can be described by a finite state machine where three different transitions can take place out of a fixed state ξ_t depending on whether $D(t)$ takes values -1, 0 or 1.

Example 12.5

For $L = 2$, the mapping (12.172) is depicted in Table 12.3. Each column shows the state $\xi(t)$, and the differences $D(t-1)$ and $D(t-2)$ it encodes. The trellis diagram for the state transition dynamics (12.173) is shown in Fig. 12.17 for $L = 2$. □

Table 12.3. State mapping (12.172) for $L = 2$.

State	1	2	3	4	5	6	7	8	9
$D(t-1)$	–1	0	1	–1	0	1	–1	0	1
$D(t-2)$	–1	–1	–1	0	0	0	1	1	1

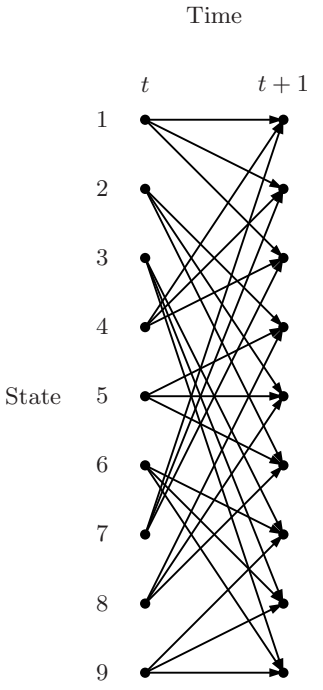

Fig. 12.17. Trellis diagram corresponding to the state dynamics (12.173) for $L = 2$.

Given the state mapping (12.172), we can express the difference signal

$$\sum_{k=1}^{L} h_k D(t-k) = g(\xi(t), \xi(t+1)) \tag{12.174}$$

as a function of the transition betwen state $\xi(t)$ and state $\xi(t+1)$. This function can be represented by a matrix $\mathbf{G}$ with entries $g(k,\ell)$, where $1 \le k, \ell \le 3^L$.

Example 12.5, continued

For the FIR channel $H(z)$ of order $L = 2$ given by (12.166), the matrix $\mathbf{G}$ with entries $g(k,\ell)$ for $1 \le k,\ \ell \le 9$ is given by

$$\mathbf{G} = \begin{bmatrix} -0.3 & 0.5 & 1.3 & x & x & x & x & x & x \\ x & x & x & -0.9 & -0.1 & 0.7 & x & x & x \\ x & x & x & x & x & x & -1.5 & -0.7 & 0.1 \\ -0.2 & 0.6 & 1.4 & x & x & x & x & x & x \\ x & x & x & -0.8 & 0 & 0.8 & x & x & x \\ x & x & x & x & x & x & -1.4 & -0.6 & 0.2 \\ -0.1 & 0.7 & 1.5 & x & x & x & x & x & x \\ x & x & x & -0.7 & 0.1 & 0.9 & x & x & x \\ x & x & x & x & x & x & -1.3 & -0.5 & 0.3 \end{bmatrix}, \tag{12.175}$$

where the entries corresponding to forbidden state transitions are marked by an x. Note again that the matrix $\mathbf{G}$ is skew-centrosymmetric. □

Let $\mathcal{V}$ denote the set of enlarged state trellis paths of the form

$$\boldsymbol{\xi} = (\xi_0, \xi_1, \cdots, \xi_t, \cdots) .$$

Note that since we assume $D(t) \equiv 0$ for $t \le 0$, under the mapping (12.172), the initial state for all trellis paths is

$$\xi_0 = (1 + 3^L)/2 . \tag{12.176}$$

Furthermore, since we assume $D(1) \neq 0$, i.e. $D(1) = 1$ or -1, the next state is

$$\xi_1 = (1 + 3^L)/2 - 1 \quad \text{or} \quad \xi_1 = (1 + 3^L)/2 + 1 . \tag{12.177}$$

Then, if $\mathcal{V}$ denotes the set of enlarged state trellis paths satisfying (12.176) and (12.177), expression (12.171) for the squared minimum distance $d^2_{\min}$ between paths of the original trellis can be rewritten as the solution of the minimization problem

$$d^2_{\min} = \frac{4}{\sigma^2} \min_{\boldsymbol{\xi} \in \mathcal{V}} \sum_{t=0}^{\infty} (g(\xi_t, \xi_{t+1}))^2 . \tag{12.178}$$

for the enlarged state trellis. In this expression the summation is infinite, but it is useful to recognize that if a path returns to state $(1+3^L)/2$ at some time $t_0 > 1$, which corresponds to having

$$D(t_0 - k) = 0$$

for $1 \leq k \leq L$, then the optimum trajectory after t_0 will stay in this state forever, since transitions from state $(1 + 3^L)/2$ to itself have zero cost. This corresponds to selecting

$$D(t) = 0 \text{ for } t \geq t_0 \, .$$

Thus, the minimization problem (12.178) consists of finding the shortest trellis path (the path with least metric) from state $(1+3^L)/2-1$ to state $(1+3^L)/2$. Note that by symmetry of the mapping (12.172) with respect to the center label $(1+3^L)/2-1$ under sign reversal of the sequence $\{D(t),\, t \geq 0\}$, because of the centrosymmetry of the matrix $\mathbf{G}$, the mirror image with respect to the middle label $(1+3^L)/2$ of the shortest path $\boldsymbol{\xi}$ connecting state $(1+3^L)/2-1$ to state $(1 + 3^L)/2$ is the shortest path connecting state $(1 + 3^L)/2 + 1$ to state $(1 + 3^L)/2$, and it has the same path metric as $\boldsymbol{\xi}$.

To summarize, expression (12.178) indicates that $d^2_{\min}$ can be evaluated by solving a shortest path problem over an oriented graph. The vertices of the graph are the enlarged states $1 \leq k \leq 3^L$. An edge exists between vertices k and ℓ if the dynamics (12.173) allow a transition from $\xi_t = k$ to $\xi_{t+1} = \ell$. The cost of a transition from k to ℓ is $c(k, \ell) = g^2(k, \ell)$ if the transition is allowed, and

$$c(k, \ell) = +\infty$$

if the transition is forbidden. Note that all costs are nonnegative. Several algorithms due to Bellman and Ford or Dijkstra [33], among others, can be employed to solve the shortest path problem from state $(1+3^L)/2-1$ to state $(1 + 3^L)/2$. Since the Bellman-Ford algorithm relies on Bellman's optimality principle described earlier, we focus on this algorithm.

Bellman-Ford algorithm: Given a fixed end vertex q, the algorithm computes sequentially for increasing values of n the minimum distance $d_k(n)$ from every vertex k to q for paths with n edges or less connecting k to q, starting with $n = 0$. Since the distance of node q to itself is zero, we set $d_q(n) = 0$ for all $n \geq 0$. For nodes k for which there does not exist a path with n edges or less connecting k to q, we set

$$d_k(n) = +\infty \, .$$

Since $1 \leq k \leq V$ where V denotes the total number of vertices of the graph (here $V = 3^L$), for a fixed t, the distances $d_k(n)$ for all vertices k can be represented by a vector

$$\mathbf{d}(n) = \big[\, d_1(n) \cdots \, d_k(n) \cdots \, d_V(n) \,\big]^T \, .$$

For $n = 0$, only node q can reach q without any transition, so all entries of $\mathbf{d}(0)$ are equal to $+\infty$, except

$$d_q(0) = 0\,.$$

Then suppose that the shortest paths with n edges or less have been identified for connecting all nodes $k \in V$ to q, so the vector $\mathbf{d}(n)$ has been evaluated. Then, by Bellman's optimality principle, the shortest path with $n+1$ segments or less connecting k to q must either be the shortest path with n segments connecting k to q, or it must be a path connecting k to another vertex ℓ, followed by the shortest path with n segments connecting ℓ to q. This implies that $d_k(n+1)$ can be expressed as

$$d_k(n+1) = \min(d_k(n), \min_{\ell \in N(k)} (c(k,\ell) + d_\ell(n)))\,. \tag{12.179}$$

In (12.179) the set of neighboring vertices $N(k)$ of k are the vertices ℓ such that there exists an edge connecting k to ℓ (the transition from k to ℓ is allowed). The inner minimization in (12.179) has for purpose to identify the best path connecting k to q with $n+1$ segments. This best path is obtained by comparing the costs $c(k,\ell) + d_\ell(n)$ for all paths obtained by concatenating an edge connecting k to one of its neighbors ℓ followed by the shortest n-segment path connecting ℓ to q. The algorithm (12.179) terminates as soon as the vector $\mathbf{d}(n)$ remains unchanged for one iteration, i.e.,

$$\mathbf{d}(n+1) = \mathbf{d}(n)\,,$$

since the form of the recursion (12.179) implies that in this case $\mathbf{d}(n+m) = \mathbf{d}(n)$ for all positive integers m. The vector $\mathbf{d}$ obtained when the algorithm terminates specifies the minimum distances between each vertex k and terminal vertex q.

We have yet to explain how the shortest path between each vertex k and terminal vertex q is traced. In parallel with recursion (12.179) for each n, we keep track of a vector $\mathbf{s}(n)$ where entry $s_k(n)$ represents the successor vertex of k on the shortest path with n segments connecting k to q. Whenever $d_k(n+1) = d_k(n)$, we set

$$s_k(n+1) = s_k(n)\,, \tag{12.180}$$

which indicates that the successor node is unchanged. But when $d_k(n+1) < d_k(n)$, we select

$$s_k(n+1) = \arg \min_{\ell \in N(k)} (c(k,\ell) + d_\ell(n)) \tag{12.181}$$

as the new successor of k. When the Bellman-Ford algorithm terminates, the final successor vector $\mathbf{s}$ can be used to trace the shortest path from k to q in a

manner similar to the trace-back stage of the Viterbi algorithm. Specifically, starting from vertex k, its successor s_k is the k-th entry of $\mathbf{s}$. The next vertex is obtained by reading off the s_k-th entry of $\mathbf{s}$. Proceeding sequentially, the shortest path is obtained when vertex q is reached.

Example 12.5, continued

Consider the evaluation of the squared minimum distance $d^2_{\min}$ for the enlarged state trellis corresponding to the channel $H(z)$ of order $L = 2$ given by (12.166). For this case, the enlarged state trellis depicting the allowed transitions is shown in Fig. 12.17, and the matrix $\mathbf{G}$ is given by (12.175). Recall also that the cost structure is specified by

$$c(k, \ell) = (g(k, \ell))^2$$

if the transition from k to ℓ is allowed and $c(k, \ell) = +\infty$ otherwise. Since $L = 2$, the shortest path problem we need to solve is from state 4 to state 5. We initialize the Bellman-Ford recursion with

$$\mathbf{d}(0) = \begin{bmatrix} \infty & \infty & \infty & \infty & 0 & \infty & \infty & \infty & \infty \end{bmatrix}^T .$$

Then for $n = 1$ we obtainv

$$\mathbf{d}(1) = \begin{bmatrix} \infty \\ (0.1)^2 \\ \infty \\ \infty \\ 0 \\ \infty \\ \infty \\ (0.1)^2 \\ \infty \end{bmatrix} , \quad \mathbf{s}(1) = \begin{bmatrix} x \\ 5 \\ x \\ x \\ 5 \\ x \\ x \\ 5 \\ x \end{bmatrix} ,$$

where x indicates that no successor has been determined yet. Similarly for $n = 2$ and $n = 3$,

$$\mathbf{d}(2) = \begin{bmatrix} (0.5)^2 + (0.1)^2 \\ (0.1)^2 \\ (0.7)^2 + (0.1)^2 \\ (0.6)^2 + (0.1)^2 \\ 0 \\ (0.6)^2 + (0.1)^2 \\ (0.7)^2 + (0.1)^2 \\ (0.1)^2 \\ (0.5)^2 + (0.1)^2 \end{bmatrix} , \quad \mathbf{s}(2) = \begin{bmatrix} 2 \\ 5 \\ 8 \\ 2 \\ 5 \\ 8 \\ 2 \\ 5 \\ 8 \end{bmatrix} ,$$

$$\mathbf{d}(3) = \begin{bmatrix} (0.5)^2 + (0.1)^2 \\ (0.1)^2 \\ (0.1)^2 + (0.5)^2 + (0.1)^2 \\ (0.2)^2 + (0.5)^2 + (0.1)^2 \\ 0 \\ (0.2)^2 + (0.5)^2 + (0.1)^2 \\ (0.1)^2 + (0.5)^2 + (0.1)^2 \\ (0.1)^2 \\ (0.5)^2 + (0.1)^2 \end{bmatrix} \quad , \quad \mathbf{s}(2) = \begin{bmatrix} 2 \\ 5 \\ 9 \\ 1 \\ 5 \\ 9 \\ 1 \\ 5 \\ 8 \end{bmatrix} ,$$

and finally

$$\mathbf{d}(n) = \mathbf{d}(3) \quad , \quad \mathbf{s}(n) = \mathbf{s}(3)$$

for $n \geq 4$. The shortest path $\boldsymbol{\xi}$ leaving state 5 at $t = 0$, going to state 4 at $t = 1$, and then returning to state 5 at $t = 4$ is shown below in Fig. 12.18. It corresponds to the sequence

$$D(1) = D(2) = -1$$

and $D(t) = 0$ for $t \geq 3$. The mirror image $\boldsymbol{\xi}_M$ of path $\boldsymbol{\xi}$ with respect to the center state 5 is also shown in Fig. 12.18. Leaving from state 5 at $t = 0$, it

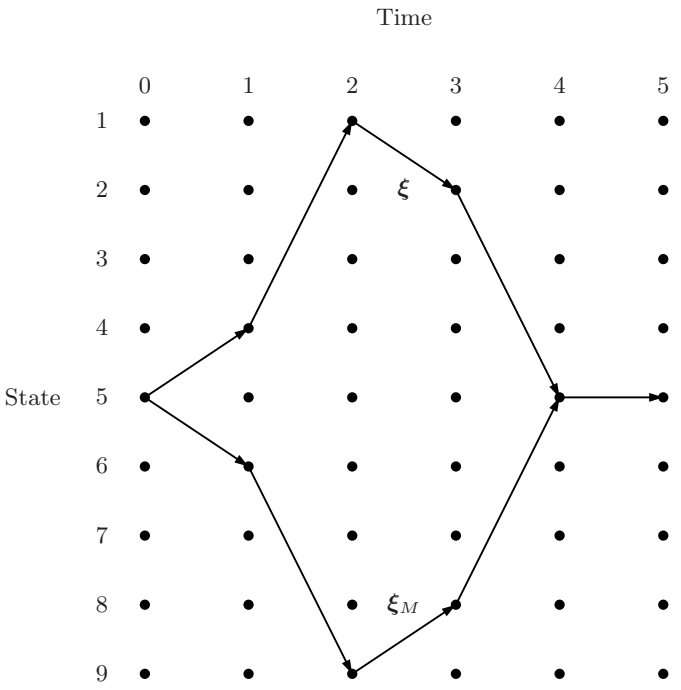

Fig. 12.18. Shortest path $\boldsymbol{\xi}$ from state 4 to state 5. Its mirror image $\boldsymbol{\xi}_M$ is the shortest path from state 6 to state 5.

goes to state 6 at $t = 1$ and then returns to state 5 at $t = 4$. It corresponds to the sign reversed sequence

$$D(1) = D(2) = 1$$

and $D(t) = 0$ for $t \geq 3$.

The squared distance $d^2_{\min}$ is the same for both $\boldsymbol{\xi}$ and its mirror image $\boldsymbol{\xi}_M$. To evaluate $d^2_{\min}$ we observe from (12.178) that $d^2_{\min}$ is obtained by adding to the minimum distance d_4 the cost of the first transition from state 5 to state 4. This gives

$$\begin{aligned} d^2_{\min} &= \frac{4}{\sigma^2}(c(5,4) + d_4) \\ &= \frac{4}{\sigma^2}[(0.8)^2 + (0.2)^2 + (0.5)^2 + (0.1)^2] = 3.76/\sigma^2 \,. \end{aligned}$$

□

Remark: If we consider recursion (12.179) and observe that for the problem we consider a vertex k has only 3 neighbors (corresponding to the 3 different values of $D(t)$), each step of the recursion requires 3×3^L operations. Noting also that a shortest path can visit each of the vertices only once, we deduce that the total number of Bellman-Ford iterations is at the most 3^L, so the total complexity of the algorithm is 3^{2L+1}. It may appear surprising to the reader that the MAP sequence detection problem and the evaluation of $d^2_{\min}$ rely on different algorithms, namely the Viterbi and Bellman-Ford algorithms. However, it is important to note that there exists an important difference between the trellis searches associated with MAP sequence detection and finding $d^2_{\min}$. For the former problem, the path metric $c(x_t, x_{t+1}, t)$ specified by (12.80) depends on observation y_t and is thus *time-dependent*. On the other hand the trellis minimization (12.178) involves the time-independent cost function $c(\xi_t, \xi_{t+1}) = (g(\xi_t, \xi_{t+1}))^2$, which makes possible the use of a faster algorithm.

Probability of error lower bound: To characterize more precisely the performance of MAP sequence equalization, it is also possible to construct a lower bound for the probability of error $P[E]$ specified by (12.102). Let $\{D_0(t), t \geq 1\}$ denote the difference sequence minimizing (12.171). Among all path indices u such that $1 \leq u \leq U$, let $\mathcal{U}_0$ denote the subset corresponding to bit sequences $\{B_u(t),\, t \geq 1\}$ such that

$$B_v(t) = B_u(t) - D_0(t) \tag{12.182}$$

is also a bit sequence. Clearly whenever $D_0(t) = 1$, we must have

$$B_u(t) = 1 \quad , \quad B_v(t) = 0 \,,$$

and when $D_0(t) = -1$, we have

$$B_u(t) = 0 \quad \text{and} \quad B_v(t) = 1 \, .$$

On the other hand if $D_0(t) = 0$, $B_u(t)$ can be either 1 or 0, and

$$B_v(t) = B_u(t) \, .$$

Intuitively, $\mathcal{U}_0$ is the set of indices u such that the set $\mathcal{S}(t, \mathbf{x}_u)$ of simple error events for trellis path $\mathbf{x}_u$ contains a path $\mathbf{x}_v$ located at distance d_{dim} from $\mathbf{x}_u$. Then if the correct path $\mathbf{x}_u$ is such that $u \in \mathcal{U}_0$, an error is commited at time t whenever $\hat{\mathbf{X}}_{\text{MAP}} = \mathbf{x}_v$, so that

$$Q(\frac{d_{\min}}{2}) = P[\hat{\mathbf{X}}_{\text{MAP}} = \mathbf{x}_v | \mathbf{X} = \mathbf{x}_u] \leq P[\hat{X}_t^{\text{MAP}} \neq x_t^u | \mathbf{X} = \mathbf{x}_u] \, . \quad (12.183)$$

Then, by weighting both sides of inequality (12.183) by the probability p_u of paths in $\mathcal{U}_0$, we find

$$C_L Q(\frac{d_{\min}}{2}) \leq P[E] \quad (12.184)$$

with

$$C_L = \sum_{u \in \mathcal{U}_0} p_u \, . \quad (12.185)$$

Comparing the lower bound (12.184) with high SNR upper bound (12.168), we find that at high SNR $P[E]$ is bracketed by two terms varying like $Q(d_{\min}/2)$, so it will itself vary like $Q(d_{\min}/2)$.

Comparison of ISI and AWGN channels: The lower and upper bounds (12.184) and (12.168) provide a characterization of the performance of MAP sequence equalization. It is now possible to answer the rather obvious question of whether there exists any advantage in communicating over an ISI channel versus communicating over an additive white Gaussian noise (AWGN) channel of the form

$$Y(t) = AI(t) + V(t) \, , \quad (12.186)$$

where the symbols $\{I(t), \, t \geq 1\}$ are independent identically distributed and take values ± 1 with probability $1/2$ and $V(t)$ is a zero-mean WGN with variance σ^2. To ensure that the comparison is fair, the same amount of energy needs to be employed to transmit one bit through both channels. Thus in (12.185), the amplitude A satisfies

$$A^2 = E = \sum_{k=0}^{L} h_k^2 \quad (12.187)$$

where E is the energy of the received waveform h_k, $0 \leq k \leq L$ if a single bit $I(1)$ is transmitted through the ISI channel (12.158)–(12.159). Accordingly $A = ||\mathbf{h}||_2$, where $\mathbf{h}$ denotes the vector formed by the channel impulse

response. After making this identification, we deduce that the probability of error for transmission over an AWGN channel is

$$P[E] = Q(\frac{d}{2}) \tag{12.188}$$

with

$$d = \frac{2A}{\sigma} = \frac{2||\mathbf{h}||_2}{\sigma} \,. \tag{12.189}$$

For MAP sequence equalization over an ISI channel, the performance is governed by $Q(d_{\min}/2)$. But the minimum of the cost function

$$J(D(\cdot)) = \sum_{t=1}^{\infty} \Big(\sum_{k=0}^{L} h_k D(t-k) \Big)^2 \tag{12.190}$$

is always less than the value of this function for the sequence

$$D_s(1) = 1 \quad , \quad D_s(t) = 0 \text{ otherwise.} \,,$$

which is

$$J(D_s(\cdot)) = \sum_{k=0}^{L} h_k^2 = ||\mathbf{h}||_2^2 \,. \tag{12.191}$$

This implies that $d_{\min} < d$ where d is the normalized distance (12.189). Hence, as we might have suspected, it is always preferable to communicate over an AWGN channel. To explain this advantage, observe that the sequence $D_s(\cdot)$ includes only one isolated bit error. An AWGN channel has the feature that all errors are isolated in the sense that there is no propagation of one error to the next. In contrast, for an ISI channel, multiple errors in close proximity may occur in the worst-case sequence $D_0(\cdot)$, so the distance $d_{\min}$ between the two closest paths is lower than the distance d between a path $\mathbf{x}_u$ and neighboring paths containing a single isolated error.

Example 12.5, continued

For the channel $H(z)$ given by (12.166) and transmitted symbols $I(t)$ taking values ± 1, we found that $d_{\min}^2 = 3.76/\sigma^2$. On the other hand,

$$||\mathbf{h}||_2^2 = (0.8)^2 + (0.6)^2 + (0.1)^2 = 1.01 \,,$$

so that

$$d^2 = 4\frac{||\mathbf{h}||_2^2}{\sigma^2} = \frac{4.04}{\sigma^2} > d_{\min}^2 \,.$$

□

12.5 Bibliographical Notes

The asymptotic decay rates of Section 12.2 for the probability of false alarm and the probability of a miss of a test for completely observed Markov chains were derived by Natarajan [34] (see also [35]), who relied on the method of types for Markov chains [36]. In contrast, the Gartner-Ellis approach that we use here to derive the same results is an adaptation of material appearing in [37]. When the Viterbi algorithm was first introduced, it was expected that it would be applicable primarily to convolutional codes with a short constraint length (a small number of states), but progress in VLSI technology has allowed its implementation in a wide variety of situations. Therefore, most of the research activity centered on the Viterbi algorithm over the last 25 years has focused on its implementation. The upper bound for the probability of error of the Viterbi algorithm was first obtained by Forney [14] for the case of ISI channels. The analysis of [14] contained a minor error which was noted in [38] and later fixed in [24]. What is usually called the Baum-Welch algorithm for HMMs includes really two contributions: the forward and backward algorithm for evaluating the a-posteriori state distribution, and an iterative parameter estimation technique which, for the case of Markov chains, coincides with the EM algorithm. Given that the work of Baum, Welch and their collaborators precedes that of Dempster, Laird and Rubin by several years, they should probably receive some credit for the EM algorithm, even though their version is less general than the one presented in [39]. It is also interesting to note that even though Welch was a co-inventor with Baum of the forward and backward algorithm, his name does not appear on [25], which focuses primarily on the hill climbing property of the iterative parameter estimation method. A detailed account of the development of the Baum-Welch algorithm can be found in Welch's 2003 Shannon Lecture [40].

12.6 Problems

12.1. Consider a binary continuous-phase frequency-shift keyed (CPFSK) signal of the form

$$s(t) = (2E/T)^{1/2} \cos(\omega_c t + \phi(t, \mathbf{I})) \,, \tag{12.192}$$

where E denotes the bit energy, T the baud period, and ω_c the carrier frequency. The information carrying phase over the n-th bit interval $nT \le t \le (n+1)T$ can be expressed as

$$\phi(t, \mathbf{I}) = 2\pi h \sum_{k=-\infty}^{n} I_k q(t - kT) \,,$$

where the symbols I_k are independent and take the binary values ± 1 with probability $1/2$ each. The modulation index $h = p/q$ where p and q denote

two relatively prime integers, and the phase function $q(t)$ can be represented as

$$q(t) = \int_{-\infty}^{t} g(u)du$$

where $g(t)$ denotes the phase pulse. For CPFSK $g(t)$ is the rectangular 1REC pulse

$$g(t) = \begin{cases} \dfrac{1}{2T} & 0 \le t \le T \\ 0 & \text{otherwise} \, . \end{cases}$$

Hence

$$q(t) = \begin{cases} 0 & t < 0 \\ \dfrac{t}{2T} & 0 \le t \le T \\ 1/2 & t > T \, , \end{cases} \tag{12.193}$$

so the phase $\phi(t, \mathbf{I})$ changes linearly by $h\pi I_n$ during the n-th bit interval.

(a) Let

$$X_n \stackrel{\triangle}{=} \frac{q}{\pi}\phi(nT, \mathbf{I}) \, . \mod (2q)$$

Verify that X_n obeys the recursion

$$X_{n+1} = (X_n + pI_n) \mod (2q) \, .$$

Explain why X_n forms a Markov chain. Find the number K of states, as well as the number T of transitions from each state.

(b) Let $h = 1/2$, which corresponds to the special case of minimum shift keying (MSK) modulation. Specify the one-step transition probability matrix $\mathbf{P}$ for the MC obtained in part (a) and draw one section of the state trellis.

(c) The signal $s(t)$ is observed in the presence of a zero-mean WGN signal $V(t)$ with variance v, so that the CT observation signal can be expressed as

$$Y(t) = s(t) + V(t) \, .$$

Assume that the carrier frequency ω_c is much larger than the baud frequency $\omega_b = 2\pi/T$. Then, by extracting the in-phase and quadrature components of $Y(t)$ as indicated in Section 9.2, and sampling them at $t = (n+1)T$, we obtain observations

$$\begin{aligned} Y_{cn} &= E^{1/2} \cos(\pi X_{n+1}/q) + V_{cn} \\ Y_{sn} &= -E^{1/2} \sin(\pi X_{n+1}/q) + V_{sn} \, , \end{aligned}$$

where V_{cn} and V_{sn} are two independent $N(0, v)$ distributed WGN sequences. Equivalently, if we introduce the complex observations and noise

$$Y_n = Y_{cn} + jY_{sn} \quad , \quad V_n = V_{cn} + jV_{sn} \, ,$$

the above observations can be expressed in complex form as

$$Y_n = E^{1/2} \exp(-j\pi X_{n+1}/q) + V_n \ ,$$

where V_n is a complex white $CN(0, 2v)$ distributed sequence. Specify the path metric corresponding to this observation model.

(d) By removing terms that are identical for all paths, show that the MAP sequence detection problem for the MC model of parts (a)–(c) reduces to finding the trellis path $\mathbf{x} = (x_n,\ 0 \le n \le N)$ such that

$$J(\mathbf{x}) = \Re\{\sum_{n=1}^{N} Y_n \exp(j\pi x_{n+1}/q)\} \tag{12.194}$$

is *maximized*.

12.2. Consider the MC model of a CPFSK signal introduced in Problem 12.1.

(a) Show that if $h = p/q$ with p even and if the initial state is known, only half of the states of the MC are visited, so a reduced state MC model can be employed.

(b) For $h = 2/3$, and assuming that the initial state is $X_1 = 0$, obtain the one-step transition probability matrix for the reduced state MC model of part a) and draw a section of the corresponding state trellis.

12.3. Suppose a CPFSK signal is observed in WGN. The variance v of the noise is known. It is not known whether the modulation index of the signal is $H_0 : h = 1/2$ or $H_1 : h = 2/3$. We seek to simultaneously test H_0 against H_1 and find the corresponding MAP state sequence. This is accomplished by joint maximization of the a-posteriori probability $P[\mathbf{x}, H_i|\mathbf{y}]$ over $i = 0,\ 1$ and $\mathbf{x} \in \mathcal{U}_i$, where $\mathcal{U}_i$ denotes the set of trellis paths under H_i.

(a) Verify that maximizing the a-posteriori probability $P[\mathbf{x}, H_i|\mathbf{y}]$ is equivalent to maximizing

$$\ln f(\mathbf{y}|\mathbf{x}, H_i) + \ln P[\mathbf{x}|H_i] + \ln \pi_i \ ,$$

where

$$\pi_0 = P[H_0] \quad \text{and} \quad \pi_1 = P[H_1]$$

denote the a-priori probabilities of the two hypotheses. Next, check that all trajectories $\mathbf{x}$ of trellises $\mathcal{U}_0$ and $\mathcal{U}_1$ have the same probability $P[\mathbf{x}|H_i]$ for $i = 0,\ 1$.

(b) Use the results of part (a) to derive a joint test between H_0 and H_1 and MAP sequence detector. Show that an implementation of the optimum test and sequence detector consists of running two Viterbi algorithms separately for trellises $\mathcal{U}_0$ and $\mathcal{U}_1$ and the path metric $J(\mathbf{x})$ given by (12.194).

Then if $\hat{\mathbf{x}}_i$ is the MAP path under hypothesis H_i with $i = 0,\ 1$, the optimum MAP test and sequence detector is obtained by maximizing

$$J(\hat{\mathbf{x}}_i) + v \ln \pi_i$$

for $i = 0,\ 1$. Of course, when the two hypotheses are equally likely a-priori, i.e.,

$$\pi_0 = \pi_1 = 1/2\ ,$$

the second term can be dropped. Verify that in this case, the optimum test just requires running the Viterbi algorithm on $\mathcal{U}_0$ and $\mathcal{U}_1$ separately and selecting the path with largest overall metric.

12.4. Consider the MC model of a CPFSK signal with modulation index $h = p/q$ derived in Problem 12.1.

(a) Let $\mathbf{x}_u$ and $\mathbf{x}_r$ denote two paths of the MC trellis $\mathcal{U}$. Introduce the path addition modulo $2q$ as

$$\begin{aligned}\mathbf{x}_w &= \mathbf{x}_u \oplus \mathbf{x}_r \\ &= (x_n^u + x_n^r \mod (2q),\ 1 \le n \le N+1)\ .\end{aligned}$$

Then if $\mathbf{x}_v$ is in the set $\mathcal{S}(t, \mathbf{x}_u)$ of simple error events for path $\mathbf{x}_u$, verify that for any path $\mathbf{x}_r$, the path $\mathbf{x}_v \oplus \mathbf{x}_r$ is in the set of simple error events for path $\mathbf{x}_u \oplus \mathbf{x}_r$. Show that the path distance d_{uv} is invariant under this transformation.

(b) Given an arbitrary path $\mathbf{x}_u$ corresponding to symbol sequence $\{I_n,\ 1 \le n \le N\}$, let $-\mathbf{x}_u$ denote the trellis path corresponding to the sign reversed sequence $\{-I_n,\ 1 \le n \le N\}$. Verify that $\mathbf{x}_u \oplus -\mathbf{x}_u$ is the all zero state path. Then conclude from part a) that to evaluate the minimum path distance $d_{\min}$, we only need to find the path $\mathbf{x}_u$ of $\mathcal{S}(t, \mathbf{0})$ which minimizes the distance between signal $\mathbf{s}_u$ and the signal $\mathbf{s}_0$ corresponding to the all zero state path $\mathbf{0}$.

(c) For $h = 1/2$, evaluate the minimum distance $d_{\min}$.

12.5. In duobinary MSK modulation, a signal of the form (12.192) is transmitted, where over interval $nT \le t \le (n+1)T$, the phase $\phi(t, \mathbf{I})$ takes the form

$$\phi(t, \mathbf{I}) = \pi \sum_{k=-\infty}^{n} D_k q(t - kT)$$

with

$$D_k = (I_k + I_{k-1})/2\ .$$

In this expression $q(t)$ is the integrated 1REC pulse given by (12.193), and the symbols $\{I_n,\ n \ge 1\}$ are independent and take the binary values ± 1 with probability $1/2$ each.

Observe that in this modulation format, the pseudo-symbol

$$D_n = \begin{cases} 1 & \text{for } I_n = I_{n-1} = 1 \\ -1 & \text{for } I_n = I_{n-1} = -1 \\ 0 & \text{for } I_n \neq I_{n-1}\,, \end{cases}$$

so the phase changes only over half of the transmission intervals, and accordingly the bandwidth of the transmission signal is smaller than for standard MSK.

(a) Consider the phase

$$\phi(nT, \mathbf{I}) = \frac{\pi}{2} \sum_{k=-\infty}^{n-1} D_k$$

at the end of the $n-1$-th signaling interval. What are its possible values? Show that the pair

$$X_n \stackrel{\triangle}{=} (\phi(nT, \mathbf{I}), I_{n-1})$$

forms a state of the MC. Find the number K of states and the number T of possible transitions out of each state. Draw the state transition diagram, and specify the one-step transition probability matrix $\mathbf{P}$

(b) By following an approach similar to the one outlined in Problem 12.1 for CPFSK, obtain the branch metric for duobinary MSK sequence decoding.

12.6. Consider the MAP sequence equalization problem for observations (12.158)–(12.159), where the transmitted symbols $I(t)$ are independent and take the binary values ± 1 with probability $1/2$, $V(t)$ is a zero-mean WGN independent of $I(t)$ with variance σ^2, and the channel

$$H(z) = 1 + az^{-1} + a^2 z^{-2}\,.$$

(a) Find the normalized minimum distance $d_{\min}(a)$ as a function of a for $0 \leq |a| < 1$. For each a, find the minimizing difference sequence $D(t)$ for $t \geq 1$.

(b) Let

$$||\mathbf{h}||^2 = 1 + a^2 + a^4\,.$$

For what values of a is

$$d^2(a) = \frac{4||\mathbf{h}||^2}{\sigma^2} > d^2_{\min}(a)\,?$$

References

1. G. Ungerboeck, "Trellis-coded modulation with redundant signal sets, Parts I and II," *IEEE Communications Magazine*, vol. 25, pp. 5–21, 1987.
2. C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo equalization," *European Trans. Telecomm.*, vol. 2, pp. 259–263, June 1998.

3. R. Koetter, A. C. Singer, and M. Tüchler, "Turbo equalization," *IEEE Signal Processing Mag.*, vol. 21, pp. 67–80, Jan. 2004.
4. G. Ferrari, G. Colavolpe, and R. Raheli, *Detection Algorithms for Wireless Communications With Applications to Wired and Storage Systems*. Chichester, England: J. Wiley & Sons, 2004.
5. K. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Applications*. Boston: Kluwer Acad. Publ., 2001.
6. L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–286, Feb. 1989.
7. L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice Hall, 1993.
8. J. G. Proakis, *Digital Communications, Fourth Edition*. New York: McGraw-Hill, 2000.
9. R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, UK: Cambridge Univ. Press, 1985.
10. R. G. Gallager, *Discrete Stochastic Processes*. Boston: Kluwer Acad. Publ., 1996.
11. R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, United Kingdom: Cambridge Univ. Press, 1994.
12. S.-I. Amari and H. Nagaoka, *Methods of Information Geometry*. Providence, RI: American Mathematical Soc., 2000.
13. A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Informat. Theory*, vol. 13, pp. 260–269, 1967.
14. J. G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Informat. Theory*, vol. 18, pp. 363–378, May 1972.
15. J. G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 31, pp. 268–278, Mar. 1973.
16. R. Bellman, "The theory of dynamic programming," *Proc. Nat. Acad. Sci.*, vol. 38, pp. 716–719, 1952.
17. R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957. Reprinted by Dover Publ., Mineola, NY, 2003.
18. C. Rader, "Memory management in a Viterbi decoder," *IEEE Trans. Commun.*, vol. 29, pp. 1399–1401, Sept. 1981.
19. R. Cypher and C. B. Shung, "Generalized trace-back technique for survivor memory management in the Viterbi algorithm," *J. VLSI Signal Proc.*, vol. 5, pp. 85–94, 1993.
20. G. Feygin and P. G. Gulak, "Architectural tradeoffs for survivor sequence memory management in Viterbi decoders," *IEEE Trans. Commun.*, vol. 41, pp. 425–429, Mar. 1993.
21. H.-L. Lou, "Implementing the Viterbi algorithm," *IEEE Signal Processing Magazine*, vol. 12, pp. 42–52, Sept. 1995.
22. J. R. Barry, E. A. Lee, and D. G. Messerschmitt, *Digital Communications, Third Edition*. New York: Springer Verlag, 2003.
23. K. Ouahada and H. C. Ferreira, "Viterbi decoding of ternary line codes," in *Proc. 2004 IEEE Internat. Conf. on Communications*, vol. 2, (Paris, France), pp. 667–671, June 2004.

24. S. Verdú, "Maximum likelihood sequence detection for intersymbol interference channels: A new upper bound on error probability," *IEEE Trans. Informat. Theory*, vol. 33, pp. 62–68, Jan. 1987.
25. L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Annals Mathematical Statistics*, vol. 41, pp. 164–171, Feb. 1970.
26. L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rates," *IEEE Trans. Informat. Theory*, vol. 20, pp. 284–287, Mar. 1974.
27. C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–127, Oct. 1996.
28. J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Globecom Conf.*, (Houston,TX), pp. 793–797, Nov. 1989.
29. M. P. C. Fossorier, F. Burkert, S. Lin, and J. Hagenauer, "On the equivalence between SOVA and max-log-MAP decoding," *IEEE Communications Letters*, vol. 5, pp. 137–139, May 1998.
30. G. Battail, "Pondération des symboles décodés par l'algorithme de Viterbi," *Annales des Telecommunications*, pp. 31–38, Jan. 1987.
31. G. M. Vastula and F. S. Hill, "On optimal detection of band-limited PAM signals with excess bandwidth," *IEEE Trans. Commun.*, vol. 29, pp. 886–890, June 1981.
32. K. M. Chugg and A. Polydoros, "MLSE for an unknown channel – Part I: Optimality considerations," *IEEE Trans. Commun.*, vol. 44, pp. 836–846, July 1996.
33. D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*. Belmont, MA: Athena Scientific, 1997.
34. S. Natarajan, "Large deviations, hypothesis testing, and source coding for finite Markov chains," *IEEE Trans. Informat. Theory*, vol. 31, pp. 360–365, May 1985.
35. V. Anantharam, "A large deviations approach to error exponents in source coding and hypothesis testing," *IEEE Trans. Informat. Theory*, vol. 36, July 1990.
36. L. D. Davisson, G. Longo, and A. Sgarro, "The error exponent for the noiseless encoding of finite ergodic Markov sources," *IEEE Trans. Informat. Theory*, vol. 27, pp. 431–438, July 1981.
37. F. den Hollander, *Large Deviations*. Providence, RI: American Mathematical Soc., 2000.
38. G. J. Foschini, "Performance bound for maximum-likelihood reception of digital data," *IEEE Trans. Informat. Theory*, vol. 21, pp. 47–50, Jan. 1975.
39. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.
40. L. R. Welch, "The Shannon lecture: Hidden Markov models and the Baum-Welch algorithm," *IEEE Information Theory Soc. Newsletter*, vol. 53, Dec. 2003.

http://www.springer.com/978-0-387-76542-6

Principles of Signal Detection and Parameter Estimation

Levy, B.C.
2008, 664 p. 101 illus., Hardcover
ISBN: 978-0-387-76542-6