The RESEARCH LABORATORY *of* ELECTRONICS *at the* MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE, MASSACHUSETTS 02139

A Framework for Non-Gaussian Signal Modeling and Estimation

Shawn M. Verbout

RLE Technical Report No. 626

June 1999

A Framework for Non-Gaussian Signal Modeling and Estimation

Shawn M. Verbout

RLE Technical Report No. 626

June 1999

This work was supported in part by the U.S. Air Force under Grant No. F49620-96-1-0072 and in part by the U.S. Army Research Laboratory under Cooperative Agreement DAAL01-96-2-001.

The Research Laboratory of Electronics
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139-4307

A Framework for Non-Gaussian Signal Modeling and Estimation
by
Shawn Matthew Verbout

Submitted to the Department of Electrical Engineering and Computer Science
on February 8, 1999, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This thesis develops a new statistical framework for analyzing and processing stationary non-Gaussian signals. The proposed framework consists of a collection of mathematical techniques for modeling such signals as well as an associated collection of model-based algorithms for solving certain basic signal processing problems. Two inference problems commonly encountered in practice are given special consideration: (i) identification of the parameter values of a non-Gaussian signal source based on a clean observation of the source output; and (ii) recovery of the source output itself based on a noisy observation and complete knowledge of the measurement model. These problems are referred to, respectively, as source identification and signal estimation.

Two probabilistic signal models are considered. The first, which is termed the ARGMIX signal model, is a direct generalization of the classical autoregressive (AR) linear-Gaussian model. Under the ARGMIX model, a signal is characterized as the output of an AR linear time-invariant (LTI) system driven by a noise process whose samples are independent and identically distributed according to a Gaussian-mixture (GMIX) density, rather than a purely Gaussian density. For this model, the source identification problem can be solved efficiently with an iterative technique designed to estimate the AR parameters of the LTI system as well as the means, variances, and weighting coefficients of the GMIX density. However, the problem of optimally estimating an ARGMIX signal in independent additive noise is shown to require a number of computations growing exponentially with the number of samples contained in the observation. For the signal estimation problem, therefore, only approximate suboptimal algorithms are proposed.

A second signal model is introduced as a way of overcoming the computational complexity of the ARGMIX structure. This new model is used to approximate an arbitrarily complicated stationary signal by representing it as the output of a finite-state hidden Markov model (HMM). Such a representation is generated by quantizing the underlying signal dynamics, i.e., by partitioning the state space of the original signal, assigning each region within this partition to a unique state of the Markov chain in the HMM, and specifying appropriate state transition probabilities for this Markov chain; output densities are then assigned to the HMM states to complete the approximation. An analytical method is given for determining the best HMM-based representation of a signal when the signal density is precisely known. Computationally efficient algorithms are derived for performing both source identification and signal estimation based on this new finite-state model. For the signal estimation problem in particular, a potentially powerful technique is proposed for dealing with independent additive noise whose samples may in general be both non-Gaussian and temporally dependent.

Thesis Supervisor: Alan V. Oppenheim
Title: Ford Professor of Engineering

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Subject Matter and Purpose

Many signals produced by real-world systems, whether natural or man-made, carry information in a form that is conveniently modeled as a random but structured pattern of fluctuations over time. A primary objective in designing a signal processor is to extract this information accurately and efficiently, so that meaningful inferences can be made about the signal source at a reasonable computational cost. In this thesis, we concentrate on solving such inference problems in cases where the signals involved do not necessarily obey the classical Gaussian probability law. The central goal of the thesis is to extend the traditional linear-Gaussian signal processing framework by developing a new set of modeling concepts and estimation techniques that can be used to solve certain basic non-Gaussian inference problems.

1.2 Preliminary Assumptions and Problem Formulation

1.2.1 Assumptions on the Measurement Model

A block diagram emphasizing the main elements of a typical inference problem that we will consider is shown in Figure 1-1. This diagram depicts a signal of interest, $\{Y_t\}$, that is initially generated by some physical source, is then subjected to an uncertain transformation (e.g., transmission over a noisy medium or passage through an imperfect measurement device), and is finally converted, via an appropriately designed signal processor, into information about the source to be used by an observer. The source signal $\{Y_t\}$ might be, for example, a telecommunications waveform, a geophysical signal, or a financial time series. We will assume throughout our work that $\{Y_t\}$ is a discrete-time, scalar-valued, stationary random process, and that it is completely characterized by a fixed, finite-dimensional parameter vector, which we denote by $\mathbf{\Psi}$. Furthermore, we will assume that the probabilistic mapping shown in Figure 1-1 takes the form of a stationary additive noise process which is statistically independent of $\{Y_t\}$; we denote this corrupting noise by $\{V_t\}$. The above assumptions on the signal and noise imply that the observation, $\{Z_t\}$, is also stationary and

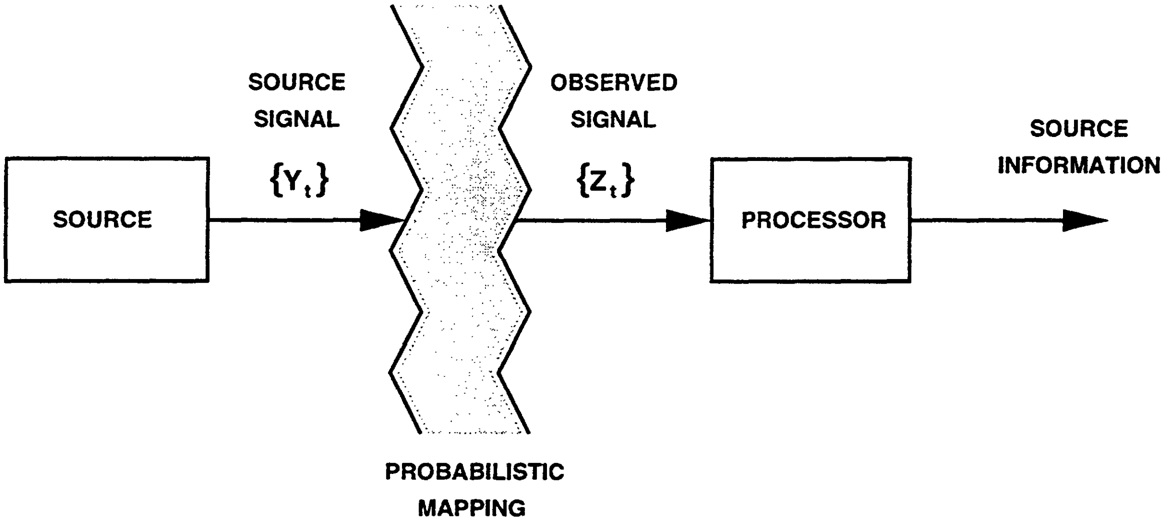


Figure 1-1: Block diagram depicting the key elements of a typical inference problem.

is defined by $Z_t = Y_t + V_t$.

1.2.2 Inference Problems to be Considered

In this thesis, we restrict our attention to two specific inference problems of the type depicted in Figure 1-1. We refer to these problems as *source identification* and *signal estimation*. In the source identification problem, it is assumed that the noise included in the observation has negligible power; hence, we take the sequence $\{V_t\}$ to be identically zero. In this problem, we are given a mathematical model for the source signal $\{Y_t\}$ as well as a finite-length sequence $y_0, y_1, \cdots, y_{N-1}$ of uncorrupted realizations of the signal. Our goal is to estimate the value of the signal parameter vector $\mathbf{\Psi}$. In the signal estimation problem, we are given mathematical models for the source signal $\{Y_t\}$ and for the additive noise process $\{V_t\}$ (including the values of all model parameters), as well as a finite-length sequence of noisy observations $z_0, z_1, \cdots, z_{N-1}$. Our goal is to estimate the underlying signal values $y_0, y_1, \cdots, y_{N-1}$. We will give more specific estimation criteria for each of these problems as we impose additional, more concrete structure on our measurement model. In general, however, we will attempt to find a maximum likelihood (ML) estimate when solving the source identification problem and a minimum mean squared error (MMSE) estimate when solving the signal estimation problem.

1.2.3 Remarks on the Problem Formulation

The problems of source identification and signal estimation as defined above are clearly idealizations of their more complicated counterparts arising in practice. For example, there exist many practical situations in which we would like to identify the parameters of a signal, but we have only corrupted observations of the signal available. On the other hand, there exist situations in which we would like to estimate a signal in noise, but we have only partial knowledge of the parametric measurement model. Both types of situations call for the solution of a joint problem involving aspects of both source identification and signal estimation; however, a joint problem of this kind may be too complex to serve as a starting point for the development of a new inference framework. Consideration of the two simplified problems described above will allow us to explore a number of important issues in non-Gaussian signal processing and still provide a suitable foundation for future investigations.

We remark further that the stationarity assumption on both the signal and noise has been introduced mainly to simplify later discussions and analysis. Although this assumption may seem somewhat restrictive, in practice it usually does not pose a serious difficulty, for many real-world signals can be considered stationary as a good working approximation. In cases where we cannot legitimately regard the signal as being stationary over the entire observation interval, we can often decompose the observation interval into a series of subintervals, and then treat the portion of the signal in each subinterval as being stationary. This strategy is commonly used, for example, in the analysis and processing of speech signals, whose statistical characteristics change dramatically over time but remain reasonably stable over brief intervals [153, 47]. In other cases where stationarity does not hold for the entire observation interval, it may be possible to transform the signal in some way so as to induce approximately stationary behavior. This technique is often used for certain financial or economic time series, which may be non-stationary only because they contain simple deterministic growth or seasonal trends over time [70, 128, 196].

1.3 Traditional Approach to the Inference Problems

1.3.1 The Linear-Gaussian Measurement Model

In order to develop feasible, working solutions to the inference problems described above, it is traditionally assumed that the stochastic structure of the source signal, as well as that of any corrupting noise present when the signal is observed, is adequately described by a Gaussian probability density function (pdf). This assumption is often synthesized from a systems point of view, namely such that the source in Figure 1-1 is defined to be a stable, linear, time-invariant (LTI) system driven by zero-mean, unit-variance white Gaussian noise. The output of the source, $\{Y_t\}$, then possesses a Gaussian pdf whose specific form is determined entirely by the impulse response of the LTI system. The corrupting noise, $\{V_t\}$, is often assumed to be zero-mean white Gaussian noise having some fixed power level. These assumptions placed on the signal and noise are typically referred to as the linear-Gaussian model.

1.3.2 Advantages of the Classical Model

There are several reasons why the classical model described above has enjoyed immense popularity in the past. Clearly, the linear-Gaussian assumption is often invoked for the sake of mathematical convenience, since it leads to the tractable derivation and analysis of theoretically optimal signal processing algorithms. But the model has also been successfully applied in a wide range of practical problems — including applications in signal analysis, filtering, prediction, and control — over a span of many decades. Its continued use and satisfactory performance in such diverse applications clearly validate the classical model as a good first-order approximation of many real-world signals and systems. Indeed, in certain situations, compelling physical arguments can be made that justify the use of the linear-Gaussian assumption via the Central Limit Theorem [125].

The body of literature that has evolved around the linear-Gaussian model has become quite rich and extensive; as a result, the mathematical theory associated with the model is now fully developed and well understood. In addition, a number of elegant and powerful algorithms have been developed in conjunction with the model. These include, for example, the methods of Levinson [108], Durbin [51], and Burg [35], which are essentially solutions to the source identification problem under the assumption that the LTI system in the model is purely autoregressive, as well as the widely used techniques of Wiener [228] and Kalman [88, 89], which are optimal solutions to the signal estimation problem under somewhat more general model assumptions. These and other algorithms developed for the linear-Gaussian model are, in general, computationally efficient, easy to implement, and fairly straightforward to analyze.

1.3.3 Limitations of the Classical Model

In spite of its many desirable properties, the linear-Gaussian model also has a number of limitations which cast doubt on its appropriateness in certain signal processing problems. Invoking the traditional Gaussian assumption actually imposes rather stringent structural constraints on the waveforms being modeled. For example, the Gaussian pdf is inherently a symmetric function; hence, any non-Gaussian signal whose pdf exhibits pronounced asymmetry is not likely to be adequately described by the classical model. In addition, we note that the pdf of any zero-mean, stationary Gaussian signal is completely characterized by the set of second-order signal moments (or, equivalently, by the autocorrelation function); in contrast, in order to specify the pdf of a non-Gaussian signal, higher-order moments (possibly an infinite number of them) are required. Furthermore, if we combine this property of sufficiency of second-order moments with the fact that the autocorrelation function is symmetric, we find that the pdf of any stationary Gaussian signal is invariant with respect to a time-reversal of the signal; on the other hand, the pdf of a stationary non-Gaussian signal is typically quite sensitive to the orientation of the time axis [201].

Yet another major limitation of the linear-Gaussian model is that it is not suitable for representing signals that exhibit sudden high-amplitude bursts or sporadic outliers. Such signals constitute a broad and important class of non-Gaussian phenomena that arise in practical settings; they are encountered in applications such as underwater acoustical analysis and signal detection [29, 52, 113, 230], low-frequency and other modes of communication [26, 107, 124, 126, 174], and exploration seismology [219], to name just a few. Signals and noise that exhibit impulsive behavior cannot be accurately represented by a linear-Gaussian model because the tails of a Gaussian pdf decay extremely rapidly, and they therefore cannot accommodate high-amplitude events. For this reason, as well as those cited earlier, the classical model may lack the flexibility needed to provide an accurate fit to certain waveforms encountered in real-world problems.

1.4 The Need for Non-Gaussian Signal Models

In practice, we rarely have perfect knowledge of the stochastic structure of either the signal or noise; hence, in most cases where the linear-Gaussian model is used, it is intended only as a nominal approximation. We have already mentioned several applications, however, in which either the signal pdf or the noise pdf deviates considerably from this nominal Gaussian assumption. A number of additional applications that are known to involve non-Gaussian phenomena can be found in [91, 225, 226]. In such applications, a critical question that must be addressed is whether a moderate amount of mismatch between the actual signal and the nominal signal model will lead to only a moderate amount of degradation in overall signal processing performance.

A commonly cited example demonstrating the potential loss in performance due to model mismatch involves the coherent reception of a deterministic waveform in additive white Gaussian noise. It is well known that the best possible detector for this problem (in the sense that it minimizes the probability of decision error) is the matched filter, which consists of a cross-correlation of the observation with the known waveform and then a comparison of the result to a fixed threshold [76]. Though it is optimal when the noise is truly Gaussian, the matched filter may suffer a dramatic decline in performance if the noise pdf deviates even slightly from the nominal Gaussian form [80, 81, 169, 173]. On the other hand, incorporating a modest amount of nonlinear signal processing based on a more realistic noise model can yield a detector that is far superior to the matched filter [127, 107, 145, 120].

Other examples have been presented in the literature that demonstrate a similar lack of robustness with the linear-Gaussian model in the problems of source identification and signal estimation [90, 117, 118, 119, 122, 146]. Such examples underscore the need for more accurate (and, unavoidably, more complex) signal models in situations where a severe loss in performance cannot be tolerated.

1.5 Proposed Approaches to the Inference Problems

1.5.1 Developing Extensions to the Classical Model

In order to overcome the limitations with the linear-Gaussian model, we seek to develop more realistic models that will allow us to solve the problems of source identification and signal estimation when they involve non-Gaussian signals. We observe, however, that the class of non-Gaussian signals is immense and extremely diverse, even when it is restricted to include only those signals that are stationary. Indeed, to define a signal to be non-Gaussian is to characterize it by default, i.e., by its failure to possess a specific, well defined statistical property. For this reason, it is virtually impossible to develop a general, unifying framework that applies equally well to all signals in the class.

Therefore, our initial approach to creating a non-Gaussian inference framework will be to focus on a narrow, well defined class of signals that is often considered under the Gaussian assumption, namely the class of linear autoregressive (AR) signals. We make a slight mathematical modification to this signal class so that it includes non-Gaussian as well as Gaussian processes, and we then attempt to develop solutions to our two inference problems with this modified model. We describe approaches based on this new model in more detail in the following subsection; we then describe a second, decidedly different signal model that has been designed to compensate for certain computational disadvantages of the initial model.

1.5.2 The ARGMIX Signal Model

The first model we will consider is intended to be a direct generalization of the classical AR linear-Gaussian model. Under the ARGMIX model, the source signal $\{Y_t\}$ is characterized as the output of an AR linear time-invariant (LTI) system driven by a white non-Gaussian noise process of a special type. More specifically, $\{Y_t\}$ is assumed to obey the Kth order difference equation

$$Y_t = \sum_{k=1}^{K} a_k Y_{t-k} + W_t, \tag{1.1}$$

where $\{a_k\}_{k=1}^{K}$ are the real-valued AR coefficients of the process and $\{W_t\}$ is a sequence whose elements are independent and identically distributed (i.i.d.) according to a Gaussian-mixture (GMIX) pdf, i.e., a pdf that is a weighted average of a finite number of Gaussian densities having arbitrary means and variances. We refer to this representation for the source signal as the ARGMIX model.

To solve the source identification problem for the ARGMIX model, recall that we must generate an estimate for the signal parameter vector $\mathbf{\Psi}$, which in this case consists of not only the AR parameters, but also the mixture parameters (i.e., the means, variances, and weighting coefficients that define the Gaussian-mixture pdf). Maximum likelihood (ML) estimates for this problem have not been directly pursued in the past because the likelihood function is unbounded in the vicinity of certain known, degenerate parameter values. In general, these degenerate values are not useful as estimates, even though, strictly speaking, they do maximize the likelihood function.

As we will see in Chapter 2, however, strategies based on finding non-degenerate *local* maxima of the likelihood function yield solutions that are useful. Indeed, Titterington et al [200] showed that the approach of locally maximizing the likelihood function is useful for the related problem of estimating the mixture parameters only, i.e., the problem in which the LTI system is known to be an identity system. These researchers, as well as others [54, 121], have empirically studied the performance of several numerical hill-climbing algorithms for computing ML estimates of the mixture parameters and have found that these algorithms often produce reasonable results. To solve the more complex identification problem in which all of the ARGMIX parameters must be estimated jointly, we show that an efficient iterative algorithm can be constructed based on the expectation-maximization

(EM) principle [48]. We refer to this iterative technique as the EMAX algorithm.

Although we are able to make considerable progress in ARGMIX source identification, unfortunately the problem of optimally estimating an ARGMIX signal in independent additive noise appears to be computationally infeasible. In the latter part of Chapter 2, we demonstrate, using the simple example in which an ARGMIX signal is corrupted by white Gaussian noise, that generating an MMSE estimate requires a number of computations growing exponentially with the number of samples contained in the observation. For the signal estimation problem, therefore, we propose only approximate, suboptimal techniques.

1.5.3 The HMM-Based Signal Model

In Chapters 3, 4, and 5, we introduce and develop a second signal model as a way of overcoming the computational difficulties encountered with the ARGMIX structure. This new model is fundamentally different from the one considered above; it is intended to represent a given stationary AR signal only approximately as the output of a finite-state hidden Markov model (HMM). A representation of this type can be constructed by quantizing the underlying dynamics of the actual signal. To carry out the construction, we first partition the state space associated with the original signal into several disjoint regions and assign each region to a unique state of the Markov chain in the approximating HMM. We then specify a set of appropriate initial state probabilities and state transition probabilities for this Markov chain. After specifying the finite-state representation of the signal dynamics in this way, we then complete the overall approximation by assigning an appropriate output pdf to each state of the HMM.

This new signal model allows us to develop computationally efficient algorithms for inference problems in which the source signal $\{Y_t\}$ is described by the more general nonlinear difference equation

$$Y_t = h(Y_{t-1}, \cdots, Y_{t-K}, W_t), \tag{1.2}$$

rather than the traditional linear form given in (1.1). However, before the HMM-based model can be used to perform either source identification or signal estimation, we must first address the basic issue of random process approximation, i.e., we must determine how to best represent the true signal by an HMM when the pdf of the signal is precisely known. This problem, which we discuss in Chapter 3, can be solved by minimizing a properly chosen distance measure between the approximate and actual densities. The solution provides us with a number of theoretical criteria that must be satisfied by the components of the optimal HMM-based approximation.

In Chapter 4, we use the theoretical criteria derived in the signal approximation problem as guidelines for developing a practical source identification algorithm. This algorithm is designed to iteratively adjust the region boundaries of the state-space partition to find the best HMM-based approximation, using only a finite-length realization of the true signal. The algorithm therefore allows us to obtain working HMM-based models of arbitrarily complicated AR signals, which can then be applied to the problem of signal estimation. Techniques for performing signal estimation based on the HMM paradigm are described in Chapter 5. The basic computational engine for these techniques is based on related existing methods that have been developed for automatic speech recognition, where HMMs are now widely used [78, 85]. Building on this previous research, we create a powerful new technique for dealing with independent additive noise whose samples may in general be both non-Gaussian and temporally dependent.

1.6 Prior Work on Non-Gaussian Inference Problems

1.6.1 Non-Gaussian Source Identification

The most popular methods for estimating parameters of non-Gaussian processes have been based on higher-order statistics (HOS) (see, for example, [123, 133, 134, 135], and associated references). Most techniques of this kind have been based on a signal model that is similar to the ARGMIX model, in that the observed process is assumed to be the output of an LTI system driven by white non-Gaussian noise. Early versions of the methods currently used were first proposed by Giannakis [64], and were further analyzed and extended by Giannakis and Mendel [65], Porat and Friedlander [148], and Tugnait [209, 210, 211]. These methods are generally robust in the presence of observation noise, are fairly easy to implement, and make few assumptions about the pdf of the AR process. However, according to Mendel [123], because they extract much of their information about the observed process by computing sample moments or cumulants above second order, HOS-based methods tend to produce high-variance parameter estimates, particularly when the length of the data record is small.

The approach based on the ARGMIX model is fundamentally different from the HOS-based approach in that it assumes a specific form for the pdf of the observed data, and is therefore entirely parametric. The Gaussian-mixture model is capable of closely approximating many densities, and has been considered by a number of researchers for this purpose (see, for example, [50, 200, 156, 54, 121]). Yet apparently only a few researchers, most notably Sengupta and Kay [171] and Zhao [232], have previously considered Gaussian-mixture models in conjunction with AR systems. Sengupta and Kay [171] have addressed the problem of ML estimation of AR parameters for ARGMIX processes in which only two Gaussian pdfs constitute the mixture, each with zero mean and known variance, but with unknown relative weighting. They used a conventional Newton-Raphson optimization algorithm that is initialized by the least-squares solution to find ML estimates for the AR parameters and for the single weighting coefficient, and showed that the performance of the ML estimate is superior to that of the standard forward-backward least-squares method. However, a potentially serious limitation of their algorithm is that it does not always converge. Moreover, because they have examined such a highly constrained version of the ARGMIX model, it is unclear whether the algorithm can be easily generalized.

In a separate investigation, Zhao, et al. [232] also considered ML estimation of the AR parameters of ARGMIX processes and derived a set of linear equations whose solution gives the ML estimate for the AR parameters when all the mixture parameters are *known*. When the mixture parameters are *unknown*, they combine these linear equations with a clever ad hoc clustering technique to produce an iterative algorithm for obtaining a joint estimate of both the AR parameters and the mixture parameters. They do not guarantee convergence of this algorithm or optimality of the estimate in any sense, but they have demonstrated empirically that the performance of their algorithm is superior to that of cumulant-based methods in certain cases. The primary limitation of their algorithm is that it cannot produce unbiased estimates of the means of the Gaussian densities in the mixture whenever two or more of the true means coincide. This unavoidable bias in turn degrades the AR parameter estimates.

1.6.2 Non-Gaussian Signal Estimation

Generating an MMSE estimate of a non-Gaussian signal in additive noise requires, in general, a processing scheme that is nonlinear; hence, methods that have been developed for the non-Gaussian signal estimation problem are commonly referred to as nonlinear filtering techniques. Most of these techniques are based on a state-space measurement model having the form

$$\mathbf{X}_t = \mathbf{H}(\mathbf{X}_{t-1}, W_t) \tag{1.3}$$

$$Y_t = G(\mathbf{X}_t) \tag{1.4}$$

$$Z_t = Y_t + V_t, \tag{1.5}$$

where, in the Kth order AR case, the state vector $\mathbf{X}_t$ is defined (without loss of generality) as $\mathbf{X}_t = (Y_t, Y_{t_1}, \cdots, Y_{t-K})$ and $G(\cdot)$ is a function that merely returns the first element of its vector argument. When this model is used, the recursion that characterizes the evolution of the density of the state vector $\mathbf{X}_t$ based on the vector of observations $\mathbf{Z}_{0:t} = (Z_0, Z_1, \cdots, Z_t)$ is given by [189]

$$f(\mathbf{x}_t \,|\, \mathbf{z}_{0:t}) = c \cdot f(\mathbf{z}_t \,|\, \mathbf{x}_t) f(\mathbf{x}_t \,|\, \mathbf{z}_{0:t-1}) \tag{1.6}$$

$$f(\mathbf{x}_{t+1} \,|\, \mathbf{z}_{0:t}) = \int f(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t) f(\mathbf{x}_t \,|\, \mathbf{z}_{0:t}) \, d\mathbf{x}_t, \tag{1.7}$$

where c is a normalizing constant. These two equations are often termed the measurement update and time update formulas, respectively. Unfortunately the recursion defined by these equations cannot usually be solved in closed form. Thus, most nonlinear filtering techniques described in the literature are practical, ad hoc methods for computing approximate solutions to (1.6) and (1.7).

By far, the most popular approach to the nonlinear filtering problem has been the extended Kalman filter, or EKF [14, 181]. With this technique, a linear Taylor series expansion of the system dynamics is calculated in the vicinity of the current state vector estimate, and then the usual linear Kalman filtering formulas are applied. Variations on the basic EKF technique have been proposed by Wishner et al [229] and by Gelb [63]. Further enhancements of the technique can be obtained by incorporating the second-order terms from the original Taylor series expansion [16, 83, 182]. It has been demonstrated empirically that the EKF and its variants often yield satisfactory estimation performance; however, for many cases in which the signal-to-noise ratio is only moderate or low, or in which the densities of certain model variables are not adequately characterized by their low-order moments, the EKF is known to diverge [188].

A wide range of alternative methods for solving (1.6) and (1.7), which are quite different from the EKF, have also appeared in the literature. Many of these techniques are based not on a Taylor series approximation of the system dynamics, but rather on a direct approximation of the posterior state density using a discrete grid of points in state space. Among these alternative techniques are the point-mass approach [32, 34, 39], in which the posterior state density is approximated by a probability mass function defined on the grid; the Gaussian-sum approach [184, 11], in which the state density is represented by a weighted combination of purely Gaussian densities (each centered at a different point on the grid); and the spline-based approach [46, 99, 102, 221, 222], in which the density is approximated using polynomial segments, and the grid points themselves serve as knots for the spline.

All of these grid-based approaches are implemented using the same basic sequence of processing stages. First, an initial set of grid points is defined in such a way that the region encompassed by the grid accounts for nearly all of the true posterior probability mass. Then, at each new time index, the values associated with these grid points are updated using the Bayesian formulas (1.6) and (1.7) and, simultaneously, the locations of the grid points are adjusted so that the grid once again encompasses most of the true probability mass. For most of the grid-based approaches, the approximate Bayesian updates are performed using numerical integration; however, other methods of updating have been suggested which use random sampling of the densities involved [38, 68, 192, 194]. In any case, the values as well as the locations of the grid points are continually modified over time so that an adequate representation of the actual state pdf is maintained.

Although reasonably good performance can be obtained using grid-based approaches in many practical problems, there are several undesirable properties associated with the methods that have been developed to date. A major drawback is that these methods are, in general, very computationally expensive. Much of the computation is spent on numerically evaluating the multidimensional integral in (1.7); specifically, if the grid contains J points, then this numerical integration requires $\mathcal{O}(J^2)$ evaluations of the measurement density. Another weakness of grid-based approaches is the lack of an optimality principle to guide the assignment of the grid parameters. Although many researchers have pointed out the flexibility of grid-based methods, none have apparently formulated the grid selection procedure as an optimization problem; instead, they provide only coarse rules of thumb indicating, for example, how the grid points should be arranged geometrically in state space at each time. In certain cases, no rules are provided at all; rather, only the possibility for redefining the grid in some useful manner (e.g., adding or removing particular grid points, or changing the spacing between existing grid points) is suggested. Because the grid itself is never optimized in any way, there exists the potential for wasted computation or for the accrual of unnecessarily large errors in the density approximation as the grid evolves over time.

1.7 Thesis Overview and Outline

The thesis consists of a total of six chapters, including this introductory chapter. The chapters that make up the core of the technical material, namely Chapters 2 through 5, fall naturally into two main parts. The first part, which consists solely of Chapter 2, examines inference problems involving the ARGMIX signal model; the second part, which consists of Chapters 3 through 5, develops the theory and algorithms for the HMM-based signal model. Below we give a brief description of the material contained in each of the remaining chapters.

In Chapter 2, we focus exclusively on source identification and signal estimation when the source signal is described by the ARGMIX model. The emphasis is placed heavily on source identification, since this is more tractable of the two problems. We develop a general iterative algorithm, which we term the EMAX algorithm, for estimating the AR parameters as well as the means, variances, and weighting coefficients of the Gaussian-mixture pdf. In the latter part of the chapter, we briefly examine the problem of estimating an ARGMIX signal that has been corrupted by independent additive white Gaussian noise. We show that an optimal solution to this problem can be readily derived, but that this solution is impractical to implement because it requires too much computation.

In Chapter 3, we begin our development of the concept that an arbitrary stationary AR process can be usefully represented by a finite-state HMM. The HMM-based signal model is introduced as an alternative to the ARGMIX model to reduce the computational burden incurred under the ARGMIX assumption. We first define optimization criteria that allow us to determine how to best approximate the true random signal by an HMM of fixed order. We then derive analytical formulas for the optimal HMM parameters. While much of our initial analysis assumes the true signal is AR having order one, we show that the results also apply to higher-order AR signals.

In Chapter 4, we develop a practical algorithm for estimating the parameter values of the best HMM-based representation of a stationary signal using only a finite-length observation. This algorithm can therefore be viewed as a way of solving the source identification problem, at least in an approximate sense. The algorithm is designed to iteratively adjust the boundaries of the regions making up the state-space partition until the optimal partition is reached. The basic ideas used to guide the iterative search are drawn from the theoretical results derived in Chapter 3.

In Chapter 5, we describe how an HMM-based representation of a non-Gaussian process can be used to solve the signal estimation problem. We begin by constructing a smoothing algorithm for the simplest case in which the signal and noise are jointly characterized by a Gaussian pdf. For this case, we show that only a few states are required in the finite-state model for the measurement to achieve near-optimal estimation performance. We then extend the basic smoothing algorithm so that it applies to the cases in which the additive noise may be non-Gaussian and even temporally correlated.

In Chapter 6, we briefly summarize the highlights of our work, discuss the main thesis contributions, and provide suggestions for future related research in non-Gaussian signal processing.

1.8 Remarks on Notation

We adopt the usual convention of writing random variables in upper case and particular realizations of random variables in lower case. If X is a random variable, then we denote its pdf by $f_X(\cdot)$. If X takes values from a set containing finitely many elements, its pdf will contain impulses (i.e., Dirac delta functions), but in such cases this pdf will be used only under appropriate integrals. If Y is also a random variable, then the conditional pdf of X given Y is written $f_{X|Y}(\cdot|\cdot)$. If these densities depend on a parameter θ then they are written as $f_X(\cdot;\theta)$ and $f_{X|Y}(\cdot|\cdot;\theta)$, respectively. Expectations and conditional expectations associated with densities that depend on a parameter θ are analogously denoted by $E\{\cdot;\theta\}$ and $E\{\cdot|\cdot;\theta\}$, respectively. Vector-valued variables, both random and deterministic, are written in boldface. If $\mathbf{x}$ is an n-dimensional vector, then the ith element of $\mathbf{x}$ is denoted by x_i for $i = 1, \cdots, n$. Finally, we use the function definition

$$\mathcal{N}(w;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(w-\mu)^2}{2\sigma^2}\right\}, \qquad -\infty < w < \infty, \tag{1.8}$$

as a compact notation for the Gaussian pdf, since this density is used frequently in the remaining chapters. A summary of much of the additional notation used in the thesis can be found in Appendix A.

Chapter 2

Using the ARGMIX Signal Model for Non-Gaussian Inference

2.1 Introduction

In this chapter, we begin the technical core of the thesis by developing an inference framework for the ARGMIX signal model, which was briefly described in the introduction. In the following two subsections, we outline the basic assumptions and notation that will be used in conjunction with the ARGMIX model, and we give concise formulations of the source identification problem and the signal estimation problem based on this model. In the third subsection, we describe how the remaining material in the chapter is organized.

2.1.1 Preliminary Assumptions and Notation

We consider a discrete-time scalar-valued random process $\{Y_t\}$ that satisfies the Kth-order autoregressive difference equation

$$Y_t = \sum_{k=1}^{K} a_k Y_{t-k} + W_t, \tag{2.1}$$

where $\{a_k\}_{k=1}^{K}$ are the real-valued AR coefficients of the process, and $\{W_t\}$ is a sequence (termed the driving process or driving noise) that consists of i.i.d. random variables having a Gaussian-mixture pdf defined by

$$f_W(w) = \sum_{i=1}^{M} \rho_i \, \mathcal{N}(w; \mu_i, \sigma_i), \tag{2.2}$$

where the weighting coefficients $\{\rho_i\}_{i=1}^{M}$ satisfy $\rho_i \geq 0$ for $i = 1, 2, \cdots, M$ and $\sum_{i=1}^{M} \rho_i = 1$. Alternatively, we can express the tth sample of the driving process as

$$W_t = \sigma(\Phi_t) U_t + \mu(\Phi_t), \tag{2.3}$$

where $\{U_t\}$ is a sequence of i.i.d., zero-mean, unit-variance Gaussian random variables, σ and μ are mappings defined by $\sigma(i) = \sigma_i$ and $\mu(i) = \mu_i$ for $i = 1, 2, \cdots, M$, $\{\Phi_t\}$ is a sequence of i.i.d. discrete-valued random variables distributed according to the probability law $\Pr(\Phi_t = i) = \rho_i$ for $i = 1, 2, \cdots, M$, and the processes $\{U_t\}$ and $\{\Phi_t\}$ are assumed statistically independent. The representation of the driving process given in (2.3) will be very useful in the derivation of our parameter estimation algorithm in Section 2.2.

2.1.2 Problem Statement and Approach to Solution

2.1.2.1 ARGMIX Source Identification

For the source identification problem, we assume that the order of the autoregression, K, and the number of constituent densities in the Gaussian-mixture pdf, M, are given, and that the parameters

$$\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_M) \tag{2.4}$$

$$\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \cdots, \sigma_M) \tag{2.5}$$

$$\boldsymbol{\rho} = (\rho_1, \rho_2, \cdots, \rho_M) \tag{2.6}$$

$$\mathbf{a} = (a_1, a_2, \cdots, a_K) \tag{2.7}$$

are unknown. In addition, we assume that the random variables $Y_{-K}, Y_{-K+1}, \cdots, Y_{N-1}$ take the values $y_{-K}, y_{-K+1}, \cdots, y_{N-1}$, respectively, and we wish to estimate the parameter vector

$$\boldsymbol{\Psi} = (\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\rho}, \mathbf{a}) \tag{2.8}$$

based on our observation. For notational convenience, we define the random vectors $\mathbf{Y} = (Y_0, Y_1, \cdots, Y_{N-1})$ and $\mathbf{Y}_t = (Y_{t-1}, Y_{t-2}, \cdots Y_{t-K})$ for $t = 0, 1, \cdots, N$, and denote the realizations of these vectors by $\mathbf{y}$ and $\mathbf{y}_t$, respectively.

As mentioned in Chapter 1, we are not strictly seeking an ML estimate because, in most cases, degenerate estimates exist that have infinite likelihood. To see how such degenerate estimates can arise, one can easily verify that if we put, say, $\hat{a}_i = 0$ for $i = 1, 2, \cdots, K$, $(\hat{\mu}_i, \hat{\sigma}_i, \hat{\rho}_i) = (0, 1, 1/M)$ for $i = 2, \cdots, M$, and $\hat{\mu}_1 = y_0$, and then let $\hat{\sigma}_1 \to 0$, then the likelihood function $f_{Y_0, \mathbf{Y}}(y_0, \mathbf{y}; \boldsymbol{\Psi}')$ will increase without bound. This assignment of parameter values corresponds to choosing the unknown AR system to be an identity system and one of the Gaussian densities in the mixture to be an impulse centered directly on one of the observations.

It is apparent from (2.2) that degenerate estimates can be obtained only if one or more of the standard deviation estimates is chosen to be zero. We may be tempted to avoid this problem by restricting all of the standard deviation estimates to be greater than some prespecified positive threshold. However, if this minimum threshold is set too low, then meaningless estimates can arise when the largest likelihood value occurs on the boundary of the restricted parameter space near a singularity at which $\hat{\sigma}_i = 0$ for some i. Yet if the threshold is set too high, we risk excluding the best available estimate, since a component of the true Gaussian-mixture pdf may have a standard deviation smaller than the artificially set threshold.

One alternative to maximizing the likelihood function is to find the parameters that achieve the largest of the finite local maxima [50]. In general, no closed-form solution exists for this estimate, and a numerical method must typically be used. Because the likelihood surface may have numerous local maxima, there is no guarantee that classical optimization techniques will find the largest local maximum. Yet Titterington [200] has found that methods based on finding local maxima (not necessarily the largest finite local maximum) yield useful estimates. Accordingly, we take the approach of searching for local maxima of the likelihood function using the generalized expectation-maximization (EM) algorithm.

More formally, if we let $\mathcal{P}$ denote the set of all possible values for the parameter vector $\boldsymbol{\Psi}$, then the estimate we seek for $\boldsymbol{\Psi}$ is any $\widehat{\boldsymbol{\Psi}}$ satisfying

$$\begin{aligned}\widehat{\boldsymbol{\Psi}} &\in \underset{\boldsymbol{\Psi}'\in\mathcal{P}}{\overline{\arg\max}}\left\{\log f_{\mathbf{Y}_0,\mathbf{Y}}(\mathbf{y}_0,\mathbf{y};\boldsymbol{\Psi}')\right\} && (2.9)\\ &= \underset{\boldsymbol{\Psi}'\in\mathcal{P}}{\overline{\arg\max}}\left\{\log f_{\mathbf{Y}_0}(\mathbf{y}_0;\boldsymbol{\Psi}') + \log f_{\mathbf{Y}|\mathbf{Y}_0}(\mathbf{y}|\mathbf{y}_0;\boldsymbol{\Psi}')\right\}, && (2.10)\end{aligned}$$

where the notation $\overline{\arg\max}_{x\in\mathcal{P}}\{g(x)\}$ stands for the set of all parameter values in $\mathcal{P}$ achieving finite local maxima of g.

Since the estimate $\widehat{\boldsymbol{\Psi}}$ is defined in terms of the likelihood function, but is not obtained through a standard global maximization, we refer to this estimate as a quasi-maximum likelihood (QML) estimate. In the sequel, we shall assume that $N \gg K$, i.e., that the number of samples in the observed sequence is much greater than the number of AR parameters to be estimated. Under this assumption, we may, as is standard in the derivation of ML estimates for Gaussian AR processes, ignore the first term of the log-likelihood function appearing on the right-hand side of (2.10) and assume that a QML estimate is any $\widehat{\boldsymbol{\Psi}}$ satisfying

$$\widehat{\boldsymbol{\Psi}} \in \underset{\boldsymbol{\Psi}'\in\mathcal{P}}{\overline{\arg\max}}\left\{\log f_{\mathbf{Y}|\mathbf{Y}_0}(\mathbf{y}|\mathbf{y}_0;\boldsymbol{\Psi}')\right\}. \tag{2.11}$$

2.1.2.2 ARGMIX Signal Estimation

In the signal estimation problem, we are not given a clean observation of $\{Y_t\}$, as we were in the previous case. Instead, we observe the signal only after it has been corrupted by additive white Gaussian noise; hence, each element of the observed sequence $\{Z_t\}$ has the form

$$Z_t = Y_t + V_t, \tag{2.12}$$

where $\{V_t\}$ is a sequence of i.i.d. random variables, each having a pdf $f_V(\cdot)$ defined by

$$f_V(v) = \mathcal{N}(v; 0, \sigma_V). \tag{2.13}$$

All random variables in the observation-noise sequence $\{V_t\}$ and in the driving-noise sequence $\{W_t\}$ are understood to be mutually independent.

For this problem, we assume that we know the true value of the signal parameter vector $\mathbf{\Psi}$ as well as that of the noise standard deviation σ_V. In addition, we assume that we are given realizations of the first N samples of the sequence $\{Z_t\}$. Given that we have observed the event $\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}$, our objective is to produce an MMSE estimate of the underlying signal realization $\mathbf{y}_{0:N-1}$. It is straightforward to show that the desired estimate $\hat{\mathbf{y}}_{0:N-1}$ is the conditional mean vector given by

$$\hat{\mathbf{y}}_{0:N-1} = E\{Y_{0:N-1}|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}; \mathbf{\Psi}\}. \tag{2.14}$$

As we will discover later in the chapter, the specific mathematical form of this estimate is fairly straightforward to derive, but the estimate itself is often impractical to compute. Thus, we suggest several possible suboptimal estimators in this case.

2.1.3 Chapter Organization

The chapter is organized in the following way. We begin by providing a brief overview of the EM and generalized EM principles. We then use this EM theory to derive an iterative method, referred to as the EMAX algorithm, which jointly estimates the AR parameters and mixture parameters of an ARGMIX process via the QML approach described above. Next, we present and discuss four separate applications of the EMAX algorithm and compare, through computer simulations, the performance of our algorithm to that of the standard least-squares technique as well as to that of previously developed algorithms based on a similar signal model. In the latter part of the chapter, we also derive a useful variant of the EMAX algorithm; this alternative technique is designed to estimate the AR parameters and the overall gain associated with the ARGMIX process based on the assumption that the functional form of the driving-noise pdf is precisely known. We then derive and analyze a theoretical solution to the ARGMIX signal estimation problem. Finally, we discuss the advantages, limitations, and possible extensions of the various estimation techniques developed in the chapter.

2.2 ARGMIX Source Identification Using the EM Principle

2.2.1 Theory of the EM and GEM Algorithms

The EM and GEM algorithms, which were first proposed by Dempster *et al.* [48], are iterative techniques for finding local maxima of likelihood functions. Although their convergence rates are slow, these algorithms converge reliably to local maxima of the likelihood function under appropriate conditions, require no derivatives of the likelihood function, and often yield equations that have an intuitively pleasing interpretation.

The EM and GEM algorithms are best suited to problems in which there is a "complete" data specification $\mathbf{Z}$, from which the original observations $\mathbf{Y}$ can be derived, and such that the expectation $E\{\log f_{\mathbf{Z}}(\mathbf{Z}; \mathbf{\Psi}') \mid \mathbf{Y} = \mathbf{y}; \mathbf{\Psi}''\}$ can be easily computed for any two parameter vectors $\mathbf{\Psi}', \mathbf{\Psi}'' \in \mathcal{P}$. For our problem, we use the complete data specification $\mathbf{Z} = (\mathbf{Y}, \mathbf{\Phi})$, where $\mathbf{\Phi}$ is the vector of pdf-selection variables defined by $\mathbf{\Phi} = (\Phi_0, \Phi_1, \cdots, \Phi_{N-1})$. With this choice of complete data, the EM algorithm as applied to our problem generates a sequence of estimates $\{\Psi^{(s)}\}_{s=1}^{\infty}$ according to the recursive formula

$$\Psi^{(s+1)} = \arg\max_{\Psi' \in \mathcal{P}} E\left\{\log f_{\mathbf{Y},\Phi|\mathbf{Y}_0}(\mathbf{Y}, \Phi|\mathbf{y}_0; \Psi') \mid \mathbf{Y} = \mathbf{y}, \mathbf{Y}_0 = \mathbf{y}_0; \Psi^{(s)}\right\}, \tag{2.15}$$

where some starting estimate $\Psi^{(0)}$ must be chosen to initialize the recursion. We now show that the sequence of estimates $\{\Psi^{(s)}\}_{s=0}^{\infty}$ defined above satisfies the inequality

$$\log f_{\mathbf{Y}|\mathbf{Y}_0}(\mathbf{y}|\mathbf{y}_0; \Psi^{(s+1)}) \geq \log f_{\mathbf{Y}|\mathbf{Y}_0}(\mathbf{y}|\mathbf{y}_0; \Psi^{(s)}) \tag{2.16}$$

for $s = 0, 1, 2, \cdots$; that is, we show that the log-likelihood value associated with our updated parameter estimate is increased at each iteration. We begin by writing the log-likelihood function for the observed data with parameters $\Psi' \in \mathcal{P}$ as

$$\log f_{\mathbf{Y}|\mathbf{Y}_0}(\mathbf{y}|\mathbf{y}_0; \Psi') = \log f_{\mathbf{Y},\Phi|\mathbf{Y}_0}(\mathbf{y}, \phi|\mathbf{y}_0; \Psi') - \log f_{\Phi|\mathbf{Y},\mathbf{Y}_0}(\phi|\mathbf{y}, \mathbf{y}_0; \Psi'). \tag{2.17}$$

Integrating both sides of (2.17) with respect to ϕ against the density $f_{\Phi|\mathbf{Y},\mathbf{Y}_0}(\phi|\mathbf{y}, \mathbf{y}_0; \Psi^{(s)})$ gives

$$\begin{aligned}\log f_{\mathbf{Y}|\mathbf{Y}_0}(\mathbf{y}|\mathbf{y}_0; \Psi') &= E\left\{\log f_{\mathbf{Y},\Phi|\mathbf{Y}_0}(\mathbf{y}, \Phi|\mathbf{y}_0; \Psi') \mid \mathbf{Y} = \mathbf{y}, \mathbf{Y}_0 = \mathbf{y}_0; \Psi^{(s)}\right\} \\ &\qquad - E\left\{\log f_{\Phi|\mathbf{Y},\mathbf{Y}_0}(\Phi|\mathbf{y}, \mathbf{y}_0; \Psi') \mid \mathbf{Y} = \mathbf{y}, \mathbf{Y}_0 = \mathbf{y}_0; \Psi^{(s)}\right\} && (2.18) \\ &\stackrel{\triangle}{=} U(\Psi', \Psi^{(s)}) - V(\Psi', \Psi^{(s)}), && (2.19)\end{aligned}$$

where the functions U and V are defined in the obvious way. Then (2.15) can be written as

$$\Psi^{(s+1)} = \arg\max_{\Psi' \in \mathcal{P}} U(\Psi', \Psi^{(s)}). \tag{2.20}$$

The definition of V together with Jensen's inequality allows us to conclude that $V(\Psi', \Psi^{(s)}) \leq V(\Psi^{(s)}, \Psi^{(s)})$ for any $\Psi' \in \mathcal{P}$. Hence, we have

$$\begin{aligned}\log f_{\mathbf{Y}|\mathbf{Y}_0}(\mathbf{y}|\mathbf{y}_0; \Psi')|_{\Psi'=\Psi^{(s+1)}} &= U(\Psi^{(s+1)}, \Psi^{(s)}) - V(\Psi^{(s+1)}, \Psi^{(s)}) && (2.21) \\ &\geq U(\Psi^{(s+1)}, \Psi^{(s)}) - V(\Psi^{(s)}, \Psi^{(s)}) && (2.22) \\ &\geq U(\Psi^{(s)}, \Psi^{(s)}) - V(\Psi^{(s)}, \Psi^{(s)}) && (2.23) \\ &= \log f_{\mathbf{Y}|\mathbf{Y}_0}(\mathbf{y}|\mathbf{y}_0; \Psi')|_{\Psi'=\Psi^{(s)}}, && (2.24)\end{aligned}$$

which implies that the EM algorithm gives a sequence of parameter estimates with increasing likelihoods. If the function U is continuous in both of its arguments, the sequence of estimates converges to a stationary point of the log-likelihood function [231].

The GEM algorithm is an alternative form of the EM algorithm that is often easier to implement. Such an algorithm chooses $\Psi^{(s+1)}$ such that

$$U(\Psi^{(s+1)}, \Psi^{(s)}) \geq U(\Psi^{(s)}, \Psi^{(s)}) \tag{2.25}$$

at each iteration s. It does not necessarily select $\Psi^{(s+1)}$ such that (2.20) is satisfied. Using the same reasoning we used to go from (2.21) to (2.24), we see that a GEM algorithm also produces a sequence of parameter estimates with increasing likelihoods. Whether the limit of this sequence of estimates is a stationary point of the likelihood function depends on the particular rule for selecting $\Psi^{(s+1)}$ from $\Psi^{(s)}$. If $\Psi^{(s+1)}$ is selected so that it is a *local* maximum of $U(\Psi', \Psi^{(s)})$ over $\Psi' \in \mathcal{P}$, then the sequence converges to a stationary point of the likelihood function [112, 231]. We will use this local-maximum rule for selecting updated parameters in our GEM algorithm. As is the case with all "hill-climbing" algorithms, the limit of the sequence of estimates generated by an EM or GEM algorithm may not be a global maximum of the likelihood function. Therefore, choosing $\Psi^{(0)}$ judiciously is the key to obtaining a good parameter estimate. A simple method for choosing $\Psi^{(0)}$ is given and empirically shown to be adequate in Section 2.3.

2.2.2 Derivation of the EMAX Algorithm

We now derive the EMAX algorithm by using a GEM method that chooses $\Psi^{(s+1)}$ to be a local maximum of $U(\Psi', \Psi^{(s)})$ over $\Psi' \in \mathcal{P}$. We let $\Psi' = (\boldsymbol{\mu}', \boldsymbol{\sigma}', \boldsymbol{\rho}', \mathbf{a}')$ and write (2.20) as

$$\begin{aligned}\Psi^{(s+1)} = \arg\max_{\mathbf{a}', \boldsymbol{\mu}', \boldsymbol{\sigma}', \boldsymbol{\rho}'} E\Big\{ &\log f_{\Phi|\mathbf{Y}_0}(\Phi|\mathbf{y}_0; \boldsymbol{\rho}') \\ &+ \log f_{\mathbf{Y}|\Phi,\mathbf{Y}_0}(\mathbf{y}|\Phi, \mathbf{y}_0; \mathbf{a}', \boldsymbol{\mu}', \boldsymbol{\sigma}') \mid \mathbf{Y} = \mathbf{y}, \mathbf{Y}_0 = \mathbf{y}_0; \Psi^{(s)}\Big\}. \end{aligned} \tag{2.26}$$

This is equivalent to solving the following two maximization problems:

$$\boldsymbol{\rho}^{(s+1)} = \arg\max_{\boldsymbol{\rho}'} E\left\{\log f_{\Phi|\mathbf{Y}_0}(\Phi|\mathbf{y}_0; \boldsymbol{\rho}') \mid \mathbf{Y} = \mathbf{y}, \mathbf{Y}_0 = \mathbf{y}_0; \Psi^{(s)}\right\} \tag{2.27}$$

$$\begin{aligned}(\mathbf{a}^{(s+1)}, \boldsymbol{\mu}^{(s+1)}, \boldsymbol{\sigma}^{(s+1)}) = \arg\max_{\mathbf{a}', \boldsymbol{\mu}', \boldsymbol{\sigma}'} E\Big\{&\log f_{\mathbf{Y}|\Phi,\mathbf{Y}_0}(\mathbf{y}|\Phi, \mathbf{y}_0; \mathbf{a}', \boldsymbol{\mu}', \boldsymbol{\sigma}') \mid \\ &\mathbf{Y} = \mathbf{y}, \mathbf{Y}_0 = \mathbf{y}_0; \Psi^{(s)}\Big\}\end{aligned} \tag{2.28}$$

To find $\boldsymbol{\rho}^{(s+1)}$ so that (2.27) is satisfied, we first define the functions $\{d_j\}_{j=1}^M$ and $\{C_j\}_{j=1}^M$ by

$$d_j(\phi) = \begin{cases} 1 & \text{if } \phi = j, \\ 0 & \text{otherwise;} \end{cases} \tag{2.29}$$

$$C_j(\phi_0, \phi_1, \cdots, \phi_{N-1}) = \sum_{t=0}^{N-1} d_j(\phi_t); \tag{2.30}$$

that is, $C_j(\Phi)$ is the number of times the symbol j appears in the vector Φ. In addition, for notational convenience, we define the function $P_{t,j}$ by

$$P_{t,j}(\Psi') = \Pr\{\Phi_t = j \mid \mathbf{Y} = \mathbf{y}, \mathbf{Y}_0 = \mathbf{y}_0; \Psi'\} \tag{2.31}$$

for all $\Psi' \in \mathcal{P}$, for $t = 0, 1, \cdots, N-1$ and $j = 1, 2, \cdots, M$. Using these definitions, the maximization in (2.27), which is over all $\boldsymbol{\rho}'$ such that $\rho'_j \geq 0$ and $\sum_{j=1}^{M} \rho'_j = 1$, can be written

$$\rho_j^{(s+1)} = \arg\max_{\rho'} E\left\{ \log \prod_{j=1}^{M} \rho_j'^{C_j(\Phi)} \;\middle|\; \mathbf{Y} = \mathbf{y}, \mathbf{Y}_0 = \mathbf{y}_0; \Psi^{(s)} \right\} \tag{2.32}$$

$$= \arg\max_{\rho'} E\left\{ \sum_{j=1}^{M} C_j(\Phi) \log \rho'_j \;\middle|\; \mathbf{Y} = \mathbf{y}, \mathbf{Y}_0 = \mathbf{y}_0; \Psi^{(s)} \right\} \tag{2.33}$$

$$= \arg\max_{\rho'} \sum_{j=1}^{M} \sum_{t=0}^{N-1} P_{t,j}(\Psi^{(s)}) \log \rho'_j \tag{2.34}$$

$$= \frac{1}{N} \sum_{t=0}^{N-1} P_{t,j}(\Psi^{(s)}), \tag{2.35}$$

where the last equality follows from Jensen's inequality.

To attempt the maximization in (2.28), we use the knowledge that the driving process is a sequence of i.i.d. Gaussian-mixture random variables to write the pdf for $\mathbf{Y}$ conditioned on Φ and $\mathbf{Y}_0$ as

$$f_{\mathbf{Y}|\Phi,\mathbf{Y}_0}(\mathbf{y}|\Phi, \mathbf{y}_0; \mathbf{a}', \boldsymbol{\mu}', \boldsymbol{\sigma}') = \prod_{t=0}^{N-1} \mathcal{N}(y_t - \mathbf{y}_t^T \mathbf{a}', \mu'_{\Phi_t}, \sigma'_{\Phi_t}) \tag{2.36}$$

$$= \prod_{t=0}^{N-1} \frac{1}{\sqrt{2\pi}\sigma'_{\Phi_t}} \exp\left\{ -\frac{1}{2\sigma'^2_{\Phi_t}} (y_t - \mathbf{y}_t^T \mathbf{a}' - \mu'_{\Phi_t})^2 \right\}. \tag{2.37}$$

Notice that the term $y_t - \mathbf{y}_t^T \mathbf{a}'$ represents the residual or prediction error obtained by using $\mathbf{a}'$ as the AR parameter vector. The function being maximized in (2.28) can then be written as

$$E\left\{ \log f_{\mathbf{Y}|\Phi,\mathbf{Y}_0}(\mathbf{y}|\Phi, \mathbf{y}_0; \mathbf{a}', \boldsymbol{\mu}', \boldsymbol{\sigma}') \mid \mathbf{Y} = \mathbf{y}, \mathbf{Y}_0 = \mathbf{y}_0; \Psi^{(s)} \right\} =$$

$$-\frac{N}{2} \log 2\pi - \sum_{t=0}^{N-1} \sum_{j=1}^{M} P_{t,j}(\Psi^{(s)}) \log \sigma'_j - \sum_{t=0}^{N-1} \sum_{j=1}^{M} P_{t,j}(\Psi^{(s)}) \frac{(y_t - \mathbf{y}_t^T \mathbf{a}' - \mu'_j)^2}{2\sigma_j'^2}. \tag{2.38}$$

Taking derivatives of this expression with respect to the quantities $\boldsymbol{\mu}'$, $\boldsymbol{\sigma}'$, and $\mathbf{a}'$ and setting the resulting expressions equal to zero yields three coupled nonlinear equations that define a stationary point of the right-hand side of (2.38). Because we are unable to solve these nonlinear equations analytically, it is difficult to find a global maximum. We instead use the method of coordinate ascent to numerically find a *local* maximum, resulting in a GEM algorithm rather than an EM algorithm. Coordinate ascent increases a multivariate function at each iteration by changing one variable at a time. If, at each iteration, the variable that is allowed to change is chosen to achieve the maximum of the function while the other variables are kept fixed, then coordinate ascent converges to a local maximum of the function [112]. Coordinate ascent is attractive because it is simple to maximize (2.38) separately over each variable as follows:

$$\begin{aligned}
&\arg\max_{\mu'_j} E\left\{\log f_{\mathbf{Y}|\mathbf{\Phi},\mathbf{Y}_0}(\mathbf{y}|\mathbf{\Phi},\mathbf{y}_0;(\mathbf{a}',\mu'_1,\cdots,\mu'_M,\sigma')) \mid \mathbf{Y}=\mathbf{y},\mathbf{Y}_0=\mathbf{y}_0;\mathbf{\Psi}^{(s)}\right\} \\
&\quad = \frac{\sum_{t=0}^{N-1} P_{t,j}(\mathbf{\Psi}^{(s)})(y_t-\mathbf{y}_t^T\mathbf{a}')}{\sum_{t=0}^{N-1} P_{t,j}(\mathbf{\Psi}^{(s)})} \qquad (2.39)\\
&\arg\max_{\sigma'_j} E\left\{\log f_{\mathbf{Y}|\mathbf{\Phi},\mathbf{Y}_0}(\mathbf{y}|\mathbf{\Phi},\mathbf{y}_0;(\mathbf{a}',\mu',\sigma'_1,\cdots,\sigma'_M)) \mid \mathbf{Y}=\mathbf{y},\mathbf{Y}_0=\mathbf{y}_0;\mathbf{\Psi}^{(s)}\right\} \\
&\quad = \sqrt{\frac{\sum_{t=0}^{N-1} P_{t,j}(\mathbf{\Psi}^{(s)})(y_t-\mathbf{y}_t^T\mathbf{a}'-\mu'_j)^2}{\sum_{t=0}^{N-1} P_{t,j}(\mathbf{\Psi}^{(s)})}} \qquad (2.40)\\
&\arg\max_{\mathbf{a}'} E\left\{\log f_{\mathbf{Y}|\mathbf{\Phi},\mathbf{Y}_0}(\mathbf{y}|\mathbf{\Phi},\mathbf{y}_0;(\mathbf{a}',\mu',\sigma')) \mid \mathbf{Y}=\mathbf{y},\mathbf{Y}_0=\mathbf{y}_0;\mathbf{\Psi}^{(s)}\right\} \\
&\quad = \left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\mathbf{\Psi}^{(s)})}{(\sigma'_j)^2}\mathbf{y}_t\mathbf{y}_t^T\right]^{-1}\left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\mathbf{\Psi}^{(s)})}{(\sigma'_j)^2}(y_t-\mu'_j)\mathbf{y}_t\right] \qquad (2.41)
\end{aligned}$$

Using the equations above, the coordinate-ascent algorithm is described as follows:

INITIALIZATION:

$$\tilde{\mu}_j^{(0)} = \mu_j^{(s)}, \qquad j=1,\cdots,M \qquad (2.42)$$

$$\tilde{\sigma}_j^{(0)} = \sigma_j^{(s)}, \qquad j=1,\cdots,M \qquad (2.43)$$

$$\tilde{\mathbf{a}}^{(0)} = \mathbf{a}^{(s)} \qquad (2.44)$$

ITERATION:

$$\tilde{\mu}_j^{(i+1)} = \frac{\sum_{t=0}^{N-1} P_{t,j}(\mathbf{\Psi}^{(s)})(y_t-\mathbf{y}_t^T\tilde{\mathbf{a}}^{(i)})}{\sum_{t=0}^{N-1} P_{t,j}(\mathbf{\Psi}^{(s)})}, \qquad j=1,\cdots,M \qquad (2.45)$$

$$\tilde{\sigma}_j^{(i+1)} = \sqrt{\frac{\sum_{t=0}^{N-1} P_{t,j}(\mathbf{\Psi}^{(s)})(y_t-\mathbf{y}_t^T\tilde{\mathbf{a}}^{(i)}-\tilde{\mu}_j^{(i+1)})^2}{\sum_{t=0}^{N-1} P_{t,j}(\mathbf{\Psi}^{(s)})}}, \qquad j=1,\cdots,M \qquad (2.46)$$

$$\tilde{\mathbf{a}}^{(i+1)} = \left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\mathbf{\Psi}^{(s)})}{(\tilde{\sigma}_j^{(i+1)})^2}\mathbf{y}_t\mathbf{y}_t^T\right]^{-1}\left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\mathbf{\Psi}^{(s)})}{(\tilde{\sigma}_j^{(i+1)})^2}(y_t-\tilde{\mu}_j^{(i+1)})\mathbf{y}_t\right]. \qquad (2.47)$$

If this recursion is iterated for $i = 0, \cdots, J-1$, then we define our parameter updates by $\mathbf{a}^{(s+1)} = \tilde{\mathbf{a}}^J$, $\boldsymbol{\mu}^{(s+1)} = \tilde{\boldsymbol{\mu}}^J$, $\boldsymbol{\sigma}^{(s+1)} = \tilde{\boldsymbol{\sigma}}^J$. For sufficiently large values of J, the updated parameters are, for practical purposes, local maxima of (2.38). Since the EMAX algorithm is a GEM algorithm that chooses the updated parameter estimates to be local maxima of (2.38), it converges to a stationary point. In summary, then, a single iteration of the EMAX algorithm consists of computing $\{P_{t,j}(\Psi^{(s)})\}$, applying (2.35), and iterating (2.45)–(2.47) until convergence.

As shown in Figure 2-1, the EMAX algorithm can be conceptually decomposed into three main steps, which are iterated to produce the final parameter estimates. Observe that the filter $1 - \sum_{i=1}^{K} a_i^{(s)} z^{-i}$ can be interpreted as the current estimate of the inverse of the AR filter. In the first block of Figure 2-1, this inverse filter is applied to the observations to produce the residual sequence $w_t^{(s)} = y_t - \mathbf{y}_t^T \mathbf{a}^{(s)}$, which can be interpreted as an estimate of the driving noise. This residual sequence is used to compute the posterior probabilities $\{P_{t,j}(\Psi^{(s)})\}$. Under the hypothesis that $\mathbf{a}^{(s)}$ is the true AR parameter vector, these residuals are statistically independent. Using the representation for the driving process given in (2.3), we may take the view that each sample of the residual sequence is a particular realization arising from one of M randomly chosen classes, where the pdf characterizing the jth of these classes is $\mathcal{N}(\cdot; \mu_j^{(s)}, \sigma_j^{(s)})$. For the tth sample of the driving noise sequence, the value of the class label j is determined by the pdf-selection variable Φ_t. Assuming that the mixture parameters are $\boldsymbol{\mu}^{(s)}$, $\boldsymbol{\sigma}^{(s)}$, and $\boldsymbol{\rho}^{(s)}$, we can easily compute the posterior probability $P_{t,j}(\Psi^{(s)})$ that the tth sample is a realization from class j using Bayes' rule; this is the operation being performed in the second block of Figure 2-1. With these posterior probabilities, we first compute the updated estimate of the weighting coefficient vector $\boldsymbol{\rho}^{(s+1)}$ according to (2.35). We then compute $\boldsymbol{\mu}^{(s+1)}$, $\boldsymbol{\sigma}^{(s+1)}$, and $\mathbf{a}^{(s+1)}$ by iterating (2.45)–(2.47) until convergence to some prespecified numerical tolerance is obtained; this operation is represented by the third block. As shown in Figure 2-1, the process is repeated, starting again from the first block, until convergence.

A single iteration of (2.45)–(2.47) has the following intuitive interpretation. The new estimate for the mean of the jth class is a weighted time average of the residuals, where the weight on the tth residual sample is proportional to the posterior probability that the sample belongs to class j. The new estimate for the variance of the jth class is a weighted time average of the square of residuals with the previously computed estimate of the mean of the jth class removed; once again, the weight on the tth residual sample is proportional to the posterior probability that the sample belongs to class j. The new estimate for the AR coefficient vector is updated via a generalized version of the Yule-Walker equations [232] using the most recent estimates of the means and variances.

We make the final observation that if the values of the parameters in any subset of the $3M$ mixture parameters are known, then the update equations for the parameter estimates can easily be modified, and the properties of the EMAX algorithm will be preserved. Specifically, we simply replace the parameter updates in (2.35), (2.45)–(2.47) with the corresponding known parameter values. Clearly, updates for the known parameters would not be performed in this case.

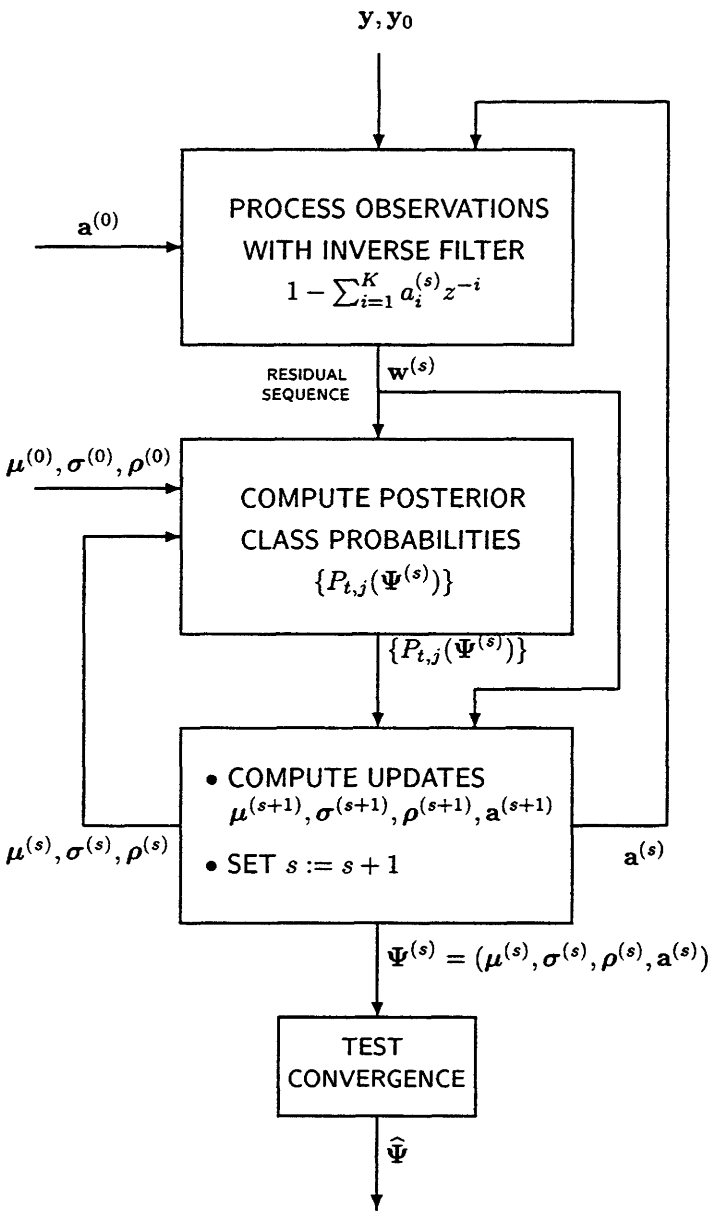

$\mathbf{y}, \mathbf{y}_0$
$\mathbf{a}^{(0)}$
PROCESS OBSERVATIONS WITH INVERSE FILTER $1 - \sum_{i=1}^{K} a_i^{(s)} z^{-i}$
RESIDUAL SEQUENCE
$\mathbf{w}^{(s)}$
$\boldsymbol{\mu}^{(0)}, \boldsymbol{\sigma}^{(0)}, \boldsymbol{\rho}^{(0)}$
COMPUTE POSTERIOR CLASS PROBABILITIES $\{P_{t,j}(\Psi^{(s)})\}$
$\{P_{t,j}(\Psi^{(s)})\}$
• COMPUTE UPDATES $\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\sigma}^{(s+1)}, \boldsymbol{\rho}^{(s+1)}, \mathbf{a}^{(s+1)}$
• SET $s := s + 1$
$\boldsymbol{\mu}^{(s)}, \boldsymbol{\sigma}^{(s)}, \boldsymbol{\rho}^{(s)}$
$\mathbf{a}^{(s)}$
$\Psi^{(s)} = (\boldsymbol{\mu}^{(s)}, \boldsymbol{\sigma}^{(s)}, \boldsymbol{\rho}^{(s)}, \mathbf{a}^{(s)})$
TEST CONVERGENCE
$\widehat{\Psi}$

Figure 2-1: Block diagram representation of the EMAX algorithm

2.3 Numerical Examples

In this section, we present several examples to illustrate the behavior and performance of the EMAX algorithm. An implementation of the EMAX algorithm using the MATLAB programming language is given in the appendix, and was used in each of the examples below. The examples were selected with several objectives in mind: (i) to verify that the EMAX algorithm behaves as expected and produces results consistent with those obtained by others on relevant ML estimation problems; (ii) to illustrate that the EMAX algorithm performs significantly better in certain estimation problems than either conventional least-squares techniques or previously proposed algorithms based on a similar data model; (iii) to demonstrate that the EMAX algorithm can be used to obtain good approximations to ML estimates in cases where the functional form for the pdf of the driving process is unknown; and (iv) to show that the EMAX algorithm can be very useful in common signal processing problems where the primary objective is to recover a signal from corrupted measurements.

For each of the examples presented here, we found that the following simple method for generating an initial parameter estimate $\Psi^{(0)} = (\boldsymbol{\mu}^{(0)}, \boldsymbol{\sigma}^{(0)}, \boldsymbol{\rho}^{(0)}, \mathbf{a}^{(0)})$ for the EMAX algorithm yielded good average performance. The vector $\mathbf{a}^{(0)}$ was computed using the forward-backward least-squares technique from traditional AR signal analysis [92, 115]. Each of the M elements of the mean vector $\boldsymbol{\mu}$ was randomly generated according to a uniform pdf having region of support $[\min_t\{w_t^{(0)}\}, \max_t\{w_t^{(0)}\}]$, where $w_t^{(0)}$ is the tth element of the residual sequence $\mathbf{w}^{(0)}$ produced by applying the filter $1 - \sum_{i=1}^{K} a_i^{(0)} z^{-i}$ to the sequence of observations. Each element of $\boldsymbol{\sigma}$ was randomly chosen according to a uniform pdf with region of support $[0, \max_t\{w_t^{(0)}\} - \min_t\{w_t^{(0)}\}]$. Finally, the elements of the weighting coefficient vector $\boldsymbol{\rho}^{(0)}$ were all set equal to $1/M$. For special cases in which certain elements of Ψ were assumed known, no initial estimate needed to be chosen.

2.3.1 Example 1: Comparison with Previous Work (Part I)

We begin with a simple example for which numerical results have already been reported by Sengupta and Kay [171]. For direct comparison of the performance of our EMAX algorithm to that of the Sengupta-Kay (S-K) algorithm, we have replicated the computer simulations carried out in their previous work. The problem considered by those authors was the ML estimation of the parameters of a fourth-order AR process whose AR coefficients are given by

$$(a_1, a_2, a_3, a_4) = (1.352, -1.338, 0.662, -0.240). \tag{2.48}$$

The driving noise for this process was assumed to consist of i.i.d. samples distributed according to the two-component Gaussian-mixture pdf

$$f_W(w) = \rho_1 \mathcal{N}(w; \mu_1, \sigma_1) + \rho_2 \mathcal{N}(w; \mu_2, \sigma_2), \qquad -\infty < v < \infty, \tag{2.49}$$

where the mixture parameters ρ_1, μ_1, σ_1, ρ_2, μ_2, and σ_2 are defined by

$$(\rho_1, \mu_1, \sigma_1) = (0.9, 0.0, 1.0); \tag{2.50}$$

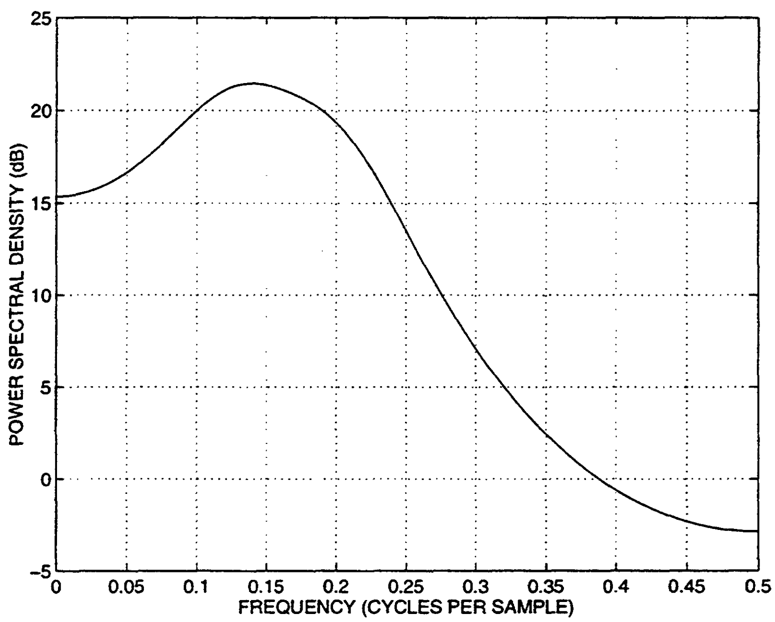


Figure 2-2: Power spectral density of fourth-order AR process discussed in Example 1.

$$(\rho_2, \mu_2, \sigma_2) = (0.1, 0.0, 10.0). \tag{2.51}$$

A plot of the power spectral density of this process is shown in Figure 2-2.

Sengupta and Kay assumed that the values of μ_1, μ_2, σ_1, and σ_2 were known, and that the values of the remaining model parameters a_1, a_2, a_3, a_4, and ρ_1 (and, of course, ρ_2, since $\rho_2 = 1-\rho_1$) were unknown. They developed a Newton-Raphson algorithm for obtaining ML estimates of the AR parameters and of the overall variance σ^2 associated with the driving process, which is given by

$$\sigma^2 = \rho_1\sigma_1^2 + (1-\rho_1)\sigma_2^2. \tag{2.52}$$

Obtaining an ML estimate of σ^2 is, in this case, equivalent to obtaining an unconstrained ML estimate of ρ_1. This is true because the parameters σ^2 and ρ_1 stand in one-to-one correspondence, and the ML estimation procedure is invariant with respect to such invertible transformations on the parameters of the log-likelihood function [149].

As was done in [171], we performed a total of 5000 trials. On each trial, a sequence of 1000 data points was generated and processed using the EMAX algorithm. The sample means and variances of the parameter estimates produced by the EMAX algorithm are presented in Table 2.1 in the column labeled EMAX-KSD (where KSD stands for *known standard deviations*). The results of a separate simulation in which the standard deviations were assumed to be unknown are also listed in Table 2.1 in the column labeled EMAX USD (where USD stands for *unknown standard deviations*). Remarkably, the sample variance of the estimates of the AR coefficients increased negligibly for the case in which the standard

	True Value	*Sample Mean* (S-K)	*Sample Mean* (EMAX-KSD)	*Sample Mean* (EMAX-USD)	*Sample Variance* (S-K)	*Sample Variance* (EMAX-KSD)	*Sample Variance* (EMAX-USD)	*Cramèr-Rao Bound* (USD)
a_1	1.352	1.3527	1.3518	1.3518	1.0219×10^{-4}	1.0727×10^{-4}	1.0782×10^{-4}	1.0491×10^{-4}
a_2	-1.338	-1.3391	−1.3378	−1.3377	2.4619×10^{-4}	2.5955×10^{-4}	2.6073×10^{-4}	2.5961×10^{-4}
a_3	0.662	0.6629	0.6619	0.6619	2.4253×10^{-4}	2.6125×10^{-4}	2.6225×10^{-4}	2.5961×10^{-4}
a_4	-0.240	-0.2404	−0.2402	−0.2402	1.0352×10^{-4}	1.0742×10^{-4}	1.0753×10^{-4}	1.0491×10^{-4}
σ^2	10.900	10.8544	10.8963	10.8946	1.2061	1.1655	2.8941	0.3149

Table 2.1: Sample means and variances for parameter estimates from Example 1. Entries were computed using results of 5000 trials for (i) the algorithm of Sengupta and Kay (S-K), (ii) the EMAX algorithm with known standard deviations (EMAX-KSD), and (iii) the EMAX algorithm with unknown standard deviations (EMAX-USD). Cramér-Rao bounds on the estimation variances, as reported by Sengupta and Kay, are also listed for the case of known standard deviations (KSD).

deviations were unknown. However, the sample variance of the estimate of the remaining parameter σ^2 increased dramatically over that for the case in which the standard deviations were known.

We observe from Table 2.1 that the estimate of σ^2 produced by the EMAX-KSD algorithm has less bias and a smaller sample variance than the corresponding estimate produced by the S-K algorithm. A possible explanation for this discrepancy is that Sengupta and Kay did not constrain their estimate of ρ_1 (which is a function of σ^2), whereas the EMAX algorithm appropriately constrains its estimate of ρ_1 to be between 0 and 1. We make two further observations from Table 2.1: (i) all of the sample means associated with the AR parameter estimates generated by the S-K algorithm exhibit slightly more bias than the sample means generated by the EMAX algorithm; and (ii) all of the sample variances of these same estimates generated by the S-K algorithm are below the Cramér-Rao bound, whereas only one of the sample variances generated by the EMAX algorithm has this property. These discrepancies may stem from the methodology used by Sengupta and Kay. They report that in approximately one percent of the trials performed for this experiment (i.e., in approximately 50 out of 5000 trials), their Newton-Raphson optimization algorithm did not converge. Whenever convergence was not obtained, the results of the corresponding trial were discarded; hence, these trials are not reflected in the statistics presented in Table 2.1. In contrast, the EMAX algorithm converged in all of the 5000 trials; hence, the results of all trials are represented in the table. The reduction in variance realized by the S-K algorithm over the EMAX algorithm may be due to the discarded trials. This conjecture is plausible if, on those occasions when the Newton-Raphson algorithm did not converge, the ML parameter estimates were relatively far from the true parameter values. If such a correlation exists between events, then it is precisely the estimates that are never obtained because of lack of convergence that distort the sample variances reported by Sengupta and Kay.

2.3.2 Example 2: Comparison with Previous Work (Part II)

Our next example illustrates that the EMAX algorithm performs significantly better in certain kinds of estimation problems than the algorithm previously proposed by Zhao et al. [232], which is based on precisely the same statistical model for the observed data as that presented in Section 2.1.1. The algorithm of Zhao, which is apparently not motivated in any respect by the EM principle, is similar in structure to the EMAX algorithm. In particular, both of these iterative algorithms use the same set of generalized normal equations to solve for the estimates of the AR parameters when given the values of the mixture parameters. In addition, at the beginning of each iteration, both algorithms use the resulting AR parameter estimates to inverse filter the observation sequence. The main difference lies in the stage of each algorithm that estimates the pdf mixture parameters from the sequence of residuals. As discussed in Section 2.2, the EMAX algorithm uses the information available in the residual sequence to climb the likelihood surface. In contrast, Zhao abandons a likelihood-based approach (citing a desire to avoid the degenerate solutions mentioned earlier) in favor of a heuristic clustering algorithm.

In the two-component mixture case, the clustering algorithm first sorts the residual samples in ascending order and then seeks out the best point at which to divide these sorted samples into two disjoint sets. The optimum point is defined as that which minimizes the average value of the sample variances associated with these two sets. Once this optimum point is found, Zhao's estimates of the means and variances of the constituent Gaussian densities are the sample means and sample variances associated with the two sets, and the estimate of the unknown weighting coefficient is simply the fraction of samples contained in each set with respect to the total number of residual samples.

We have observed that the algorithm of Zhao does not perform well when the constituent Gaussian densities in the driving-noise pdf have equal means. In this example we demonstrate that in such a case the performance of the EMAX algorithm is markedly superior to that of the Zhao algorithm. In particular, we considered the problem of estimating the parameters of an ARGMIX process whose AR coefficients are given by

$$(a_1, a_2, a_3, a_4) = (-0.1000, -0.2238, -0.0844, -0.0294). \tag{2.53}$$

The pdf for the driving noise in this case was assumed to be a two-component Gaussian-mixture pdf as in (2.49), but now with mixture parameters defined by

$$(\rho_1, \mu_1, \sigma_1) = (0.6, 0.0, 1.0); \tag{2.54}$$

$$(\rho_2, \mu_2, \sigma_2) = (0.4, 0.0, 10.0). \tag{2.55}$$

To compare the performance of the two algorithms, we performed a total of 500 trials. On each trial, a sequence of 1000 data points was generated and processed with the EMAX algorithm and Zhao algorithm. The sample means, variances, and mean square errors of the parameter estimates produced by the two algorithms are presented in Table 2.2. We note that the Zhao algorithm produces strongly biased estimates in this example. In addition, we

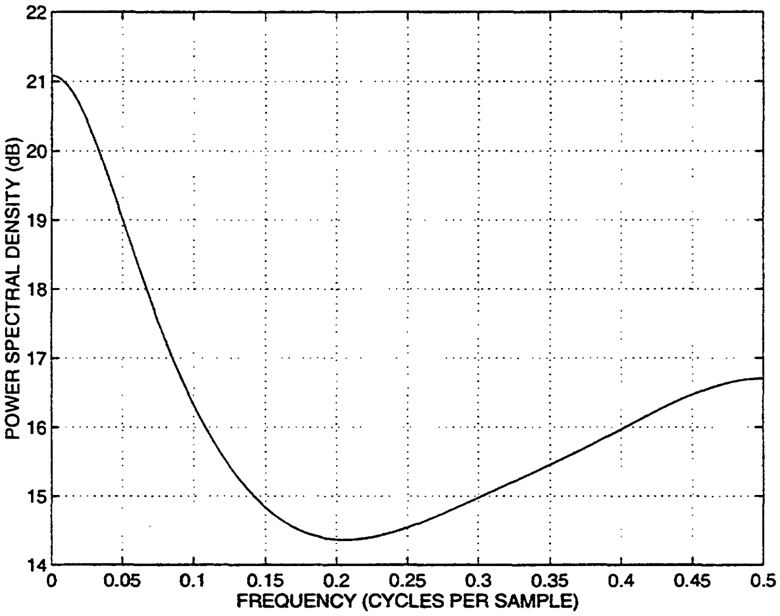


Figure 2-3: Power spectral density of fourth-order AR process discussed in Example 2.

	True Value	*Sample Mean* (EMAX)	*Sample Mean* (ZHAO)	*Sample Variance* (EMAX)	*Sample Variance* (ZHAO)	*Sample MSE* (EMAX)	*Sample MSE* (ZHAO)	*Ratio of MSE's*
a_1	-0.1000	-0.1000	-0.1150	4.975×10^{-5}	1.170×10^{-3}	4.965×10^{-5}	1.392×10^{-3}	28.03
a_2	-0.2238	-0.2238	-0.2390	5.564×10^{-5}	1.198×10^{-3}	5.553×10^{-5}	1.427×10^{-3}	25.71
a_3	-0.0844	-0.0843	-0.0983	5.539×10^{-5}	1.165×10^{-3}	5.528×10^{-5}	1.355×10^{-3}	24.52
a_4	-0.0294	-0.0289	-0.0405	5.010×10^{-5}	1.148×10^{-3}	5.029×10^{-5}	1.269×10^{-3}	25.24

Table 2.2: Sample means, variances, and mean square error (MSE) values for parameter estimates of Example 2. Entries were computed using results of 500 trials for (i) the Zhao algorithm and (ii) the EMAX algorithm. Ratios of sample MSE values (MSE of Zhao to MSE of EMAX) are also given.

note that the mean square errors associated with the EMAX algorithm are approximately 25 times smaller than those associated with the Zhao algorithm. Clearly, contributions to the mean square error for Zhao's estimates come not only from the bias term, but also from the high variance associated with her estimator.

The difficulties with the Zhao algorithm in this case may be explained by its inability to obtain good mixture parameter estimates. The quality of the mixture parameter estimates is inherently limited because the clustering algorithm essentially assigns the individual densities in the Gaussian mixture to be representatives of disjoint portions of the histogram of the residual sequence. Thus, one of the most readily observable problems with the approach, as illustrated in Figure 2-4(a), is that all of the estimated means of the constituent densities are necessarily distinct, even when the means of the true densities are identical. Figure 2-4(a) shows the true marginal pdf for the driving noise as well as typical estimates of this pdf produced by the Zhao algorithm on separate trials. Observe from the figure that, for about half of the trials, the pdf estimate produced by the Zhao algorithm is off-center to the positive side of zero, and for the other half it is off-center to the negative side. On each trial, the estimated Gaussian-mixture pdf is dominated by a single component, which attempts to model most of the histogram of the residual samples. However, the resulting overall estimate is always off-center because the smaller of the two components in the mixture attempts to model the remaining outliers, which are either much greater or much less than zero. In contrast, as shown in Figure 2-4(b), the EMAX algorithm produces pdf estimates that better approximate the true driving-noise pdf.

2.3.3 Example 3: Autoregressive Process with Laplacian Drive

In many applications, we would like to obtain ML estimates for the parameters of an AR system, but the ML problem is ill-posed because the marginal pdf characterizing the driving noise is unknown. In certain cases, however, it may be reasonable to assume that the true marginal pdf is accurately modeled by a Gaussian-mixture pdf, provided that the means, standard deviations, and weighting coefficients defining the mixture are chosen appropriately. In these cases, if we process our observations with the EMAX algorithm, then we might expect the EMAX algorithm to find the mixture parameters that yield a good approximation to the true driving-noise pdf and simultaneously to produce good approximations to the ML estimates for the AR parameters. With the present example we demonstrate the validity of this approach to the ML estimation problem.

In particular, we consider the parameter estimation problem for a fifth-order AR process whose AR coefficients are given by

$$(a_1, a_2, a_3, a_4, a_5) = (1.934, -2.048, 1.072, -0.340, 0.027). \tag{2.56}$$

The driving noise for this process consists of i.i.d. samples distributed according to a Laplacian pdf defined by

$$f_W(w) = \frac{1}{2\beta} \exp\left\{-\frac{|w|}{\beta}\right\}, \qquad -\infty < w < \infty, \tag{2.57}$$

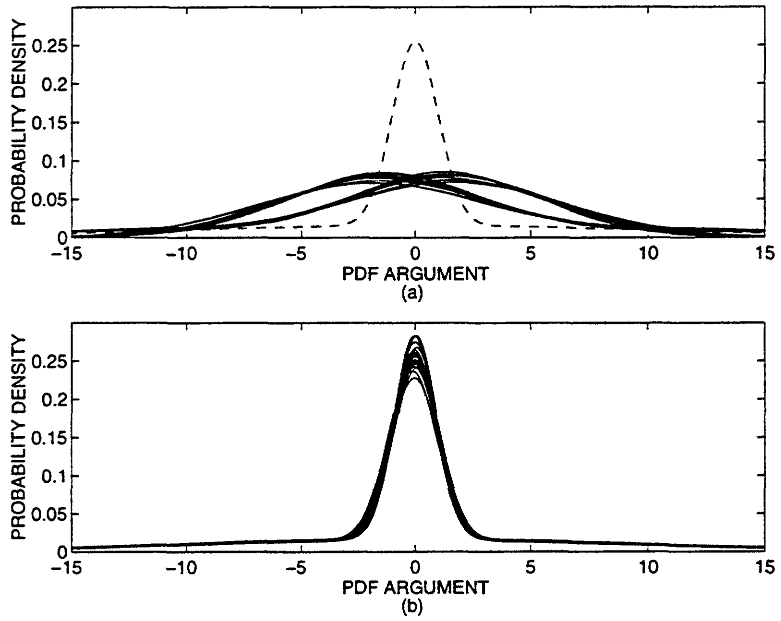


Figure 2-4: True marginal pdf (dashed curve) for driving process of Example 2 and typical estimates of the pdf (solid curves) produced by (a) the algorithm of Zhao et al. (20 estimates overlaid), and (b) the EMAX algorithm (20 estimates overlaid).

where the scale parameter β (which is related to the standard deviation σ for this density by $\sigma = \sqrt{2}\beta$) was put at $\beta = 5$. A plot of the power spectral density of this process is shown in Figure 2-5.

It is interesting to compare the performance of the EMAX algorithm to that of the exact ML estimates, which can be computed in this case. It can be shown [49] that if the samples of the driving noise for an AR process are i.i.d. and Laplacian, then the ML estimate for the AR parameter vector $\mathbf{a}$ is given by the value of $\mathbf{a}'$ that minimizes the sum of absolute residuals $\sum_{t=0}^{N-1} |y_t - \mathbf{y}_t^T\mathbf{a}'|$. An algorithm for finding such a value for $\mathbf{a}'$ was proposed by Schlossmacher [167]; this algorithm is based on the method of iteratively reweighted least squares and is therefore easy to implement on a computer.

To find parameter estimates for this problem with the EMAX algorithm, we fixed the number of Gaussian densities in the mixture at $N = 3$ and constrained the means of these constituent densities to be zero. We performed a total of 500 trials. On each trial, a sequence of 1000 data points was generated and processed with the EMAX algorithm. The sample means and sample mean square errors of the parameter estimates produced by the EMAX algorithm are presented in Table 2.3.

Also shown in Table 2.3 is a summary of the sample means and sample mean square errors of the AR parameter estimates given by two other algorithms: (i) the forward-backward least-squares method, and (ii) the ML algorithm of Schlossmacher. Experimental results shown in Table 2.3 confirm our expectation that the ML-based estimator would

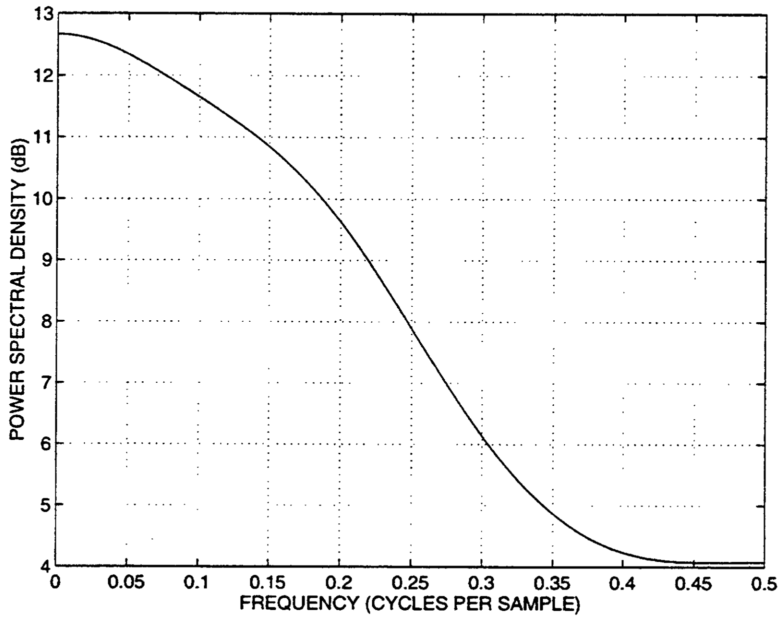


Figure 2-5: Power spectral density of fifth-order AR process discussed in Example 3.

	True Value	*Sample Mean* (LS)	*Sample Mean* (EMAX)	*Sample Mean* (ML)	*Sample MSE* (LS)	*Sample MSE* (EMAX)	*Sample MSE* (ML)
a_1	1.934	1.9311	1.9323	1.9328	1.0751×10^{-3}	6.3040×10^{-4}	5.7711×10^{-4}
a_2	-2.048	-2.0413	-2.0447	-2.0449	5.1570×10^{-3}	2.8784×10^{-3}	2.7250×10^{-3}
a_3	1.072	1.0647	1.0685	1.0697	8.4001×10^{-3}	4.7769×10^{-3}	4.2887×10^{-3}
a_4	-0.340	-0.3358	-0.3383	-0.3390	4.9714×10^{-3}	2.9190×10^{-3}	2.4228×10^{-3}
a_5	0.027	0.0256	0.0264	0.0269	1.0509×10^{-3}	6.2875×10^{-4}	5.3948×10^{-4}

Table 2.3: Sample means and sample mean square error (MSE) values for parameter estimates of Example 3. Entries were computed using results of 500 trials for (i) the standard forward-backward least squares (LS) method, (ii) the EMAX algorithm, and (iii) the ML estimation algorithm developed by Schlossmacher.

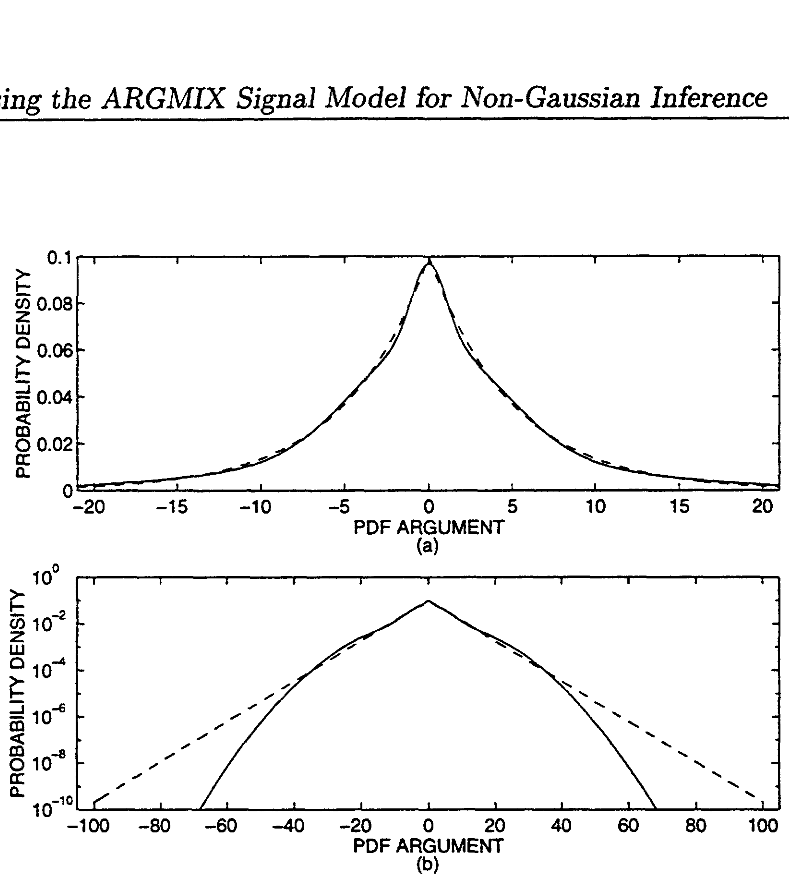


Figure 2-6: True Laplacian marginal pdf (dashed curve) for driving process of Example 3 and a typical estimate of the pdf (solid curve) produced by the EMAX algorithm, plotted using (a) linear-magnitude scale (with horizontal axis spanning ±3 standard deviations), and (b) log-magnitude scale (with horizontal axis spanning ±15 standard deviations).

perform better than the EMAX and least-squares methods since it directly exploits the fact that the driving noise is i.i.d. with a Laplacian distribution. Observe from the table that the ratio of the mean square error of the least-squares estimate to that of the ML estimate ranges approximately from 1.9 to 2.1. The ratio of the mean square error of the EMAX estimate to that of the ML estimate ranges approximately from 1.1 to 1.2. Thus, in this case the EMAX algorithm produces estimates that are much closer to the exact ML estimates than the least-squares estimates.

The superior performance of the EMAX algorithm may be attributed to the ability of its assumed Gaussian-mixture pdf to closely approximate the Laplacian pdf, as is shown for a typical case in Figure 2-6(a). It is clear from this figure that the approximation is very good over the region in which most of the samples of the driving noise reside. However, since the number of Gaussian densities in the mixture is finite, an accurate model for the Laplacian density may be obtained only over a finite region of support. Eventually, the tails of the Gaussian-mixture pdf become bounded by a function of the form $k_1 \exp\{-k_2 w^2\}$ for appropriately chosen constants k_1 and k_2. Indeed, Figure 2-6(b) reveals this phenomenon with the aid of a log-magnitude scale.

2.3.4 Example 4: Blind Equalization in Digital Communications

Our final example is an application in digital communications that has been adapted from [149]. In this example, we demonstrate that the EMAX algorithm can be used successfully in problems where the primary goal is signal reconstruction, rather than parameter estimation. In particular, we consider a communication system that uses amplitude-shift keying (ASK). In this scheme, the transmitter communicates with the receiver using an L-symbol alphabet $\mathcal{A} = \{A_i\}_{i=1}^{L}$, whose elements we take to be real numbers. To send the kth symbol of a particular message sequence $\{u_t\}$ to the receiver, the transmitter generates a pulse (having fixed shape) and modulates this pulse with the amplitude u_k. The pulse then propagates through the communication medium, which we assume is well modeled by an LTI system. Finally, the receiver processes the waveform with a linear filter to facilitate estimation of u_k.

If this filtered waveform is sampled at a rate of one sample per symbol, then the overall communication system—i.e., the transmitter, the medium, and the receiver—can be represented with an equivalent discrete-time LTI system, which we refer to as the discrete-time channel. In this case, the sampled output is the convolution of the transmitted symbol sequence $\{u_t\}$ and the impulse response $\{h_t\}$ that characterizes the discrete-time channel. If the impulse response $\{h_t\}$ is anything but a shifted and scaled unit impulse, then each sample of the output sequence will contain contributions from more than one input symbol, i.e., there will be intersymbol interference (ISI). If the characteristics of the medium are known, then the discrete-time channel is also known and the receiver can compensate for the ISI via linear equalization. Often, however, the characteristics of the medium are unknown, and the impulse response of the discrete-time channel must first be estimated in order to compensate for the ISI. One approach for accomplishing this is for the transmitter to send through the medium a training sequence that is known to the receiver. The receiver can then identify the impulse response of the discrete-time channel from the output sequence and apply the corresponding inverse filter. However, if the medium is rapidly changing, then this procedure must be performed frequently, and the effective data rate will be substantially reduced. An alternative approach is to perform blind equalization—i.e., to estimate the impulse response of the discrete-time channel from the output *without* knowing the input, and then apply the appropriate inverse filter.

We consider a scenario in which blind equalization must be performed by the receiver. We assume an ASK modulation scheme that uses the four-symbol alphabet $\mathcal{A} = \{-3, -1, 1, 3\}$. A typical 200-point input sequence to the discrete-time channel, which was generated randomly using the alphabet $\mathcal{A}$, is shown in Figure 2-7(a). We assume that the discrete-time channel has a finite impulse response $\{h_t\}$ with z-transform

$$H(z) = 1.0 - 0.65z^{-1} + 0.06z^{-2} + 0.41z^{-3}. \tag{2.58}$$

Figure 2-7(b) shows the received sequence, which is the convolution of the input sequence shown in Figure 2-7(a) and the impulse response $\{h_t\}$. It is evident from this figure that detection of the input symbols from the received sequence would be difficult without further processing.

Our blind equalization approach consists of channel estimation followed by filtering with the inverse of the estimated channel. We compare three methods for estimating the impulse response of the channel from the output sequence shown in Figure 2-7(b): (i) the forward-backward least-squares method, (ii) the fourth-order cumulant-based technique of Giannakis and Mendel [65], and (iii) the EMAX algorithm. We configured all three algorithms to estimate 18 AR coefficients. Such a configuration assumes that the discrete-time channel inverse may be accurately modeled with a system having 18 zeroes and no poles. We further configured the EMAX algorithm to estimate the means and variances of four constituent Gaussian densities. Figures 2-7(c)–(e) show the restored input sequences generated, respectively, by (i) the least-squares method, (ii) the Giannakis-Mendel algorithm, and (iii) the EMAX algorithm. It is clear from Figures 2-7(c)–(e) that the recovered sequence values produced by the EMAX algorithm are much more tightly distributed around the four true symbol values than either the recovered sequence values produced by the least squares method or those produced by the cumulant-based method. Hence, in this case we would expect superior detection performance using the EMAX algorithm.

2.4 An Alternative Version of the EMAX Algorithm

In its original form, the EMAX algorithm is capable of solving an extremely broad class of source identification problems because the ARGMIX model on which it is based offers many degrees of freedom in signal representation. As we have seen, the ARGMIX model consists of two main components for describing an unknown signal: (i) a pdf that characterizes the statistical behavior of each sample of the driving noise; and (ii) an autoregressive linear time-invariant system that induces temporal dependency among the samples of the driving noise. The power of the EMAX algorithm over conventional least-squares techniques clearly derives from the first of these components, i.e., from the flexibility of allowing the driving-noise pdf to be unknown and to have an arbitrarily complicated shape.

There are many situations arising in practice, however, in which we do not require such flexibility in a parameter estimation algorithm; in fact, in certain situations we would gladly sacrifice the flexibility of the original algorithm in exchange for a reduction in its computational complexity. In this section, we consider a useful restriction of the original estimation problem that affords such a trade-off. In particular, we focus on the case in which the driving noise is characterized by a fixed pdf whose functional form is precisely *known* except for a scale factor (i.e., a positive real number that indicates the degree of dispersion in the data distribution). In such a case, even though much prior information about the pdf is available, implementing an exact ML procedure directly may still be extremely difficult because of the complicated mathematical form of the pdf. On the other hand, using too simple an approximation to the exact ML procedure (for example, using the classical approximation based on the Gaussian-noise assumption) may lead to unacceptably poor results. For a situation of this kind, the EMAX algorithm can easily be reconfigured so that it provides a sufficiently sophisticated, yet also quite efficient and convenient, method of obtaining a good approximation to the ML solution.

To explore this idea further, let us suppose that the true density $f_W(\cdot)$ for the driving

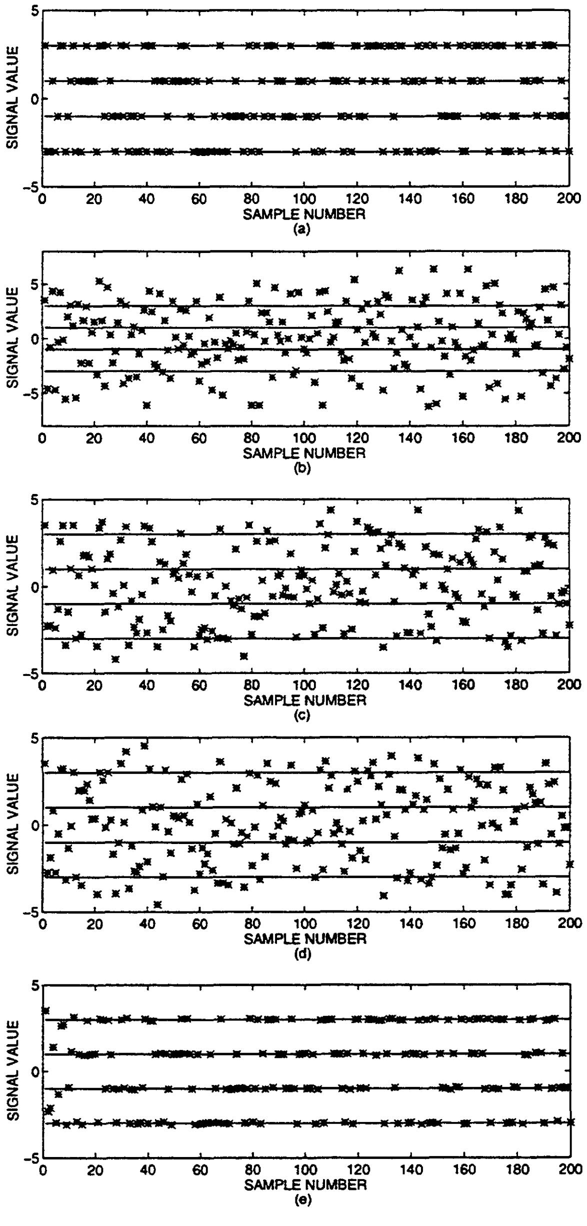


Figure 2-7: Illustration of channel equalization considered in Example 4: (a) original symbol sequence; (b) received sequence; (c) restored sequence using standard forward-backward least-squares method; (d) restored sequence using fourth-order cumulant-based Giannakis-Mendel algorithm; (e) restored sequence using EMAX algorithm assuming four-component Gaussian-mixture pdf.

noise belongs to a parameterized family of densities that is invariant with respect to scale (i.e., if the pdf for the random variable W is in the family, then the pdf for W/β is also in the family for any positive real number β). For example, the zero-mean Laplacian family used in an earlier example is scale-invariant, as is the Gaussian family, and hence also the Gaussian-mixture family for any fixed number M of mixture components. To indicate explicitly that the driving-noise pdf belongs to a scale-invariant family of densities, we shall write it as $f_W(\cdot;\beta)$ in the remainder of this section, with the understanding that

$$f_W(w;\beta) = f_{W/\beta}(w), \qquad -\infty < w < \infty. \tag{2.59}$$

For convenience, we shall also assume that $f_W(\cdot;\beta)$ is continuous at all but a finite number of points on the real line and contains no impulses.

By allowing the scale factor on an arbitrary but known driving-noise pdf to be a free parameter, we are essentially creating a generalization of the classical AR parameter estimation problem in which the zero-mean driving noise is assumed to be Gaussian, but has an unknown standard deviation that must also be estimated. The generalization follows immediately from the fact that the quantity $1/\beta$ serves as a measure of the dispersion of the distribution, since it is related to the standard deviation of the distribution through an affine transformation. To make this interpretation as direct as possible, let us assume in the sequel that the driving noise is indeed zero-mean and that the parameter value $\beta = 1$ corresponds to a standard deviation of unity, so that $1/\beta$ is exactly equal to the standard deviation for all $\beta > 0$.

Because the functional form of the pdf $f_W(\cdot;\beta)$ is assumed known for any value of β, a good approximation to this pdf for a particular value of β (say, $\beta = 1$) can be designed off-line, using a Gaussian-mixture model, before any data are observed. Such a procedure yields an approximation of the form

$$f_W(w;\beta)\Big|_{\beta=1} \approx \sum_{i=1}^{M} \rho_i \mathcal{N}(w;\mu_i,\sigma_i), \tag{2.60}$$

where the number of mixture components M is also chosen as part of the design process. Once a suitable Gaussian-mixture approximation has been obtained for the case where $\beta = 1$, it can then be adjusted to approximate any other element in original family of densities by performing a simple transformation on the mixture parameters. In particular, by using the scale-invariance property of the original family of densities, we can write

$$\begin{aligned}
f_W(w;\beta) &= f_{W/\beta}(w) && (2.61)\\
&= \beta f_W(\beta w;1) && (2.62)\\
&\approx \sum_{i=1}^{M} \beta\rho_i \mathcal{N}(\beta w;\mu_i,\sigma_i) && (2.63)\\
&= \sum_{i=1}^{M} \rho_i \mathcal{N}(w;\mu_i/\beta,\sigma_i/\beta), && (2.64)
\end{aligned}$$

where the last step follows from the fact that, for each Gaussian component $\mathcal{N}(\cdot;\mu_i,\sigma_i)$ included in the mixture, we may write

$$\beta\mathcal{N}(\beta w;\mu_i,\sigma_i) = \frac{\beta}{\sqrt{2\pi}\sigma_i}\exp\left\{-\frac{(\beta w-\mu_i)^2}{2\sigma_i^2}\right\} \tag{2.65}$$

$$= \frac{1}{\sqrt{2\pi}(\sigma_i/\beta)}\exp\left\{-\frac{[w-(\mu_i/\beta)]^2}{2(\sigma_i/\beta)^2}\right\} \tag{2.66}$$

$$= \mathcal{N}(w;\mu_i/\beta,\sigma_i/\beta). \tag{2.67}$$

We conclude from the series of equalities (2.61)–(2.64) that an approximation for any pdf in the original parameterized family can easily be generated from the initial approximation by appropriately scaling the means $\{\mu_i\}_{i=1}^M$ and standard deviations $\{\sigma_i\}_{i=1}^M$ of the Gaussian-mixture components. The weighting coefficients $\{\rho_i\}_{i=1}^M$ would remain unchanged from their initial values.

Having made these observations about Gaussian-mixture approximations, let us now derive a new version of the EMAX algorithm based on the assumption that the true driving-noise pdf is itself a Gaussian mixture which is given by

$$f_W(w;\beta) = \sum_{i=1}^{M}\rho_i\mathcal{N}(w;\mu_i/\beta,\sigma_i/\beta), \tag{2.68}$$

where the values of the parameters M, $\{\mu_i\}_{i=1}^M$, $\{\sigma_i\}_{i=1}^M$, and $\{\rho_i\}_{i=1}^M$ are again precisely known. The goal of this alternative version of the EMAX algorithm will be to generate joint QML estimates for the AR parameter vector $\mathbf{a}$ and for the scale factor β associated with the driving-noise pdf. To reflect this restriction in the new estimation problem, we re-define the parameter vector $\mathbf{\Psi}$ as

$$\mathbf{\Psi} = (\beta, \mathbf{a}). \tag{2.69}$$

The new algorithm for estimating the value of $\mathbf{\Psi}$ is now specified by the iterative formula

$$(\beta^{(s+1)},\mathbf{a}^{(s+1)}) = \arg\max_{\beta',\mathbf{a}'} E\left\{\log f_{\mathbf{Y}|\mathbf{\Phi},\mathbf{Y}_0}(\mathbf{y}|\mathbf{\Phi},\mathbf{y}_0;\beta',\mathbf{a}') \mid \mathbf{Y}=\mathbf{y},\mathbf{Y}_0=\mathbf{y}_0;\mathbf{\Psi}^{(s)}\right\}, \tag{2.70}$$

which is analogous to the original EM formula given in (2.28). (Observe that the complement of the original formula given in (2.27) is no longer needed, since the weighting coefficients are now considered fixed and known.) To obtain a more explicit expression of the function being maximized in (2.70), we can borrow the previously derived formula in (2.38) and replace the original variables μ_j' and σ_j' with their new counterparts μ_j/β' and σ_j/β', respectively; this substitution yields the modified formula

$$E\left\{\log f_{\mathbf{Y}|\mathbf{\Phi},\mathbf{Y}_0}(\mathbf{y}|\mathbf{\Phi},\mathbf{y}_0;\mathbf{a}',\beta') \mid \mathbf{Y}=\mathbf{y},\mathbf{Y}_0=\mathbf{y}_0;\mathbf{\Psi}^{(s)}\right\} =$$

$$-\frac{N}{2}\log 2\pi - \sum_{t=0}^{N-1}\sum_{j=1}^{M} P_{t,j}(\Psi^{(s)}) \log \frac{\sigma_j}{\beta'} - \sum_{t=0}^{N-1}\sum_{j=1}^{M} P_{t,j}(\Psi^{(s)}) \frac{\left[y_t - \mathbf{y}_t^T\mathbf{a}' - (\mu_j/\beta')\right]^2}{2(\sigma_j/\beta')^2}. \quad (2.71)$$

As a reminder that the search for a maximum is to be performed over only the two variables β' and $\mathbf{a}'$, we introduce a new objective function $H(\cdot)$ defined by

$$H(\beta', \mathbf{a}') = -\sum_{t=0}^{N-1}\sum_{j=1}^{M} P_{t,j}(\Psi^{(s)}) \log \frac{\sigma_j}{\beta'} - \sum_{t=0}^{N-1}\sum_{j=1}^{M} P_{t,j}(\Psi^{(s)}) \frac{\left[\beta'(y_t - \mathbf{y}_t^T\mathbf{a}') - \mu_j\right]^2}{2\sigma_j^2}, \quad (2.72)$$

which is derived from the expression on the right-hand side of (2.71) by dropping the initial constant term and substituting a somewhat more convenient (yet algebraically equivalent) form for the final term. By taking a partial derivative with respect to the variable $\mathbf{a}'$ and setting the result equal to zero, we have that the unique maximum occurs at the vector location

$$\arg\max_{\mathbf{a}'} H(\beta', \mathbf{a}')$$
$$= \left[\sum_{t=0}^{N-1}\sum_{j=1}^{M} \frac{P_{t,j}(\Psi^{(s)})}{\sigma_j^2} \mathbf{y}_t \mathbf{y}_t^T\right]^{-1} \left[\sum_{t=0}^{N-1}\sum_{j=1}^{M} \frac{P_{t,j}(\Psi^{(s)})}{\sigma_j^2} (y_t - \mu_j/\beta')\mathbf{y}_t\right]. \quad (2.73)$$

If we now take a partial derivative of $H(\cdot)$ with respect to the scale variable β', we obtain

$$\frac{\partial H}{\partial \beta'} = \frac{1}{\beta'}\sum_{t=0}^{N-1}\sum_{j=1}^{M} P_{t,j}(\Psi^{(s)}) - \sum_{t=0}^{N-1}\sum_{j=1}^{M} \frac{P_{t,j}(\Psi^{(s)})\left[\beta'(y_t - \mathbf{y}_t^T\mathbf{a}') - \mu_j\right](y_t - \mathbf{y}_t^T\mathbf{a}')}{\sigma_j^2}. \quad (2.74)$$

Observe that, for a fixed value of t, the M terms $\{P_{t,j}(\Psi^{(s)})\}_{j=1}^{M}$ necessarily sum to unity because they form a probability mass function; it follows that the value of the double summation in the first term above must be equal to N. If we first make this substitution and then set the entire expression equal to zero, we obtain (after some algebraic manipulation) the quadratic equation

$$\left[\sum_{t=0}^{N-1}\sum_{j=1}^{M} \frac{P_{t,j}(\Psi^{(s)})}{\sigma_j^2}(y_t - \mathbf{y}_t^T\mathbf{a}')^2\right]\beta'^2 - \left[\sum_{t=0}^{N-1}\sum_{j=1}^{M} \frac{P_{t,j}(\Psi^{(s)})\mu_j}{\sigma_j^2}(y_t - \mathbf{y}_t^T\mathbf{a}')\right]\beta' - N = 0. \quad (2.75)$$

For notational convenience, we rewrite this equation in the more concise form

$$\mathcal{A}\beta'^2 - \mathcal{B}\beta' - N = 0, \quad (2.76)$$

where the definitions of $\mathcal{A}$ and $\mathcal{B}$ are readily inferred from (2.75). Next, by applying the quadratic formula, we obtain the two possible solutions

$$\beta' = \begin{cases} \frac{\mathcal{B}+\sqrt{\mathcal{B}^2+4N\mathcal{A}}}{2\mathcal{A}} \\ \frac{\mathcal{B}-\sqrt{\mathcal{B}^2+4N\mathcal{A}}}{2\mathcal{A}}. \end{cases} \tag{2.77}$$

Note that since $\mathcal{A}$ is always positive, the expression under the square root sign must also be positive. Moreover, this same expression has a value whose magnitude always exceeds the magnitude of $\mathcal{B}$. Thus, while we cannot know in advance whether $\mathcal{B}$ itself will be positive or negative, we can say with certainty that the first solution given in (2.77) will always be positive and that the second will always be negative. Moreover, it can be shown (by taking a second derivative of $H(\cdot)$ with respect to β') that either of these two solutions is a local maximum of $H(\cdot)$ for a fixed value of $\mathbf{a}$. Because we seek a positive value of β' that maximizes $H(\cdot)$, we know that the unique solution must be

$$\arg\max_{\beta'>0} H(\beta', \mathbf{a}') = \frac{\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\Psi^{(s)})\mu_j}{\sigma_j^2}(y_t - \mathbf{y}_t^T\mathbf{a}')}{2\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\Psi^{(s)})}{\sigma_j^2}(y_t - \mathbf{y}_t^T\mathbf{a}')^2} + \\ \frac{\sqrt{\left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\Psi^{(s)})\mu_j}{\sigma_j^2}(y_t - \mathbf{y}_t^T\mathbf{a}')\right]^2 + 4N\left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\Psi^{(s)})}{\sigma_j^2}(y_t - \mathbf{y}_t^T\mathbf{a}')^2\right]}}{2\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\Psi^{(s)})}{\sigma_j^2}(y_t - \mathbf{y}_t^T\mathbf{a}')^2}. \tag{2.78}$$

Because (2.78) and (2.73) are highly coupled nonlinear equations, we once again resort to the technique of coordinate ascent to evaluate the optimal solution. Using the equations of optimality, we can express the coordinate-ascent algorithm as follows:

<u>INITIALIZATION:</u>

$$\tilde{\beta}^{(0)} = \beta^{(s)} \tag{2.79}$$

$$\tilde{\mathbf{a}}^{(0)} = \mathbf{a}^{(s)} \tag{2.80}$$

<u>ITERATION:</u>

$$\tilde{\beta}^{(i+1)} = \frac{\sum_t\sum_j\frac{P_{t,j}(\Psi^{(s)})\mu_j}{\sigma_j^2}(y_t - \mathbf{y}_t^T\tilde{\mathbf{a}}^{(i)})}{2\sum_t\sum_j\frac{P_{t,j}(\Psi^{(s)})}{\sigma_j^2}(y_t - \mathbf{y}_t^T\tilde{\mathbf{a}}^{(i)})^2} + \\ \frac{\sqrt{\left[\sum_t\sum_j\frac{P_{t,j}(\Psi^{(s)})\mu_j}{\sigma_j^2}(y_t - \mathbf{y}_t^T\tilde{\mathbf{a}}^{(i)})\right]^2 + 4N\left[\sum_t\sum_j\frac{P_{t,j}(\Psi^{(s)})}{\sigma_j^2}(y_t - \mathbf{y}_t^T\tilde{\mathbf{a}}^{(i)})^2\right]}}{2\sum_t\sum_j\frac{P_{t,j}(\Psi^{(s)})}{\sigma_j^2}(y_t - \mathbf{y}_t^T\tilde{\mathbf{a}}^{(i)})^2} \tag{2.81}$$

$$\tilde{\mathbf{a}}^{(i+1)} = \left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\Psi^{(s)})}{\sigma_j^2}\mathbf{y}_t\mathbf{y}_t^T\right]^{-1}\left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\Psi^{(s)})}{\sigma_j^2}(y_t - \mu/\tilde{\beta}^{(i+1)})\mathbf{y}_t\right]. \tag{2.82}$$

After this recursion has been performed for $i = 0, 1, \cdots, J-1$ (where J is chosen to be a sufficiently large integer), our parameter updates are then defined by $\mathbf{a}^{(s+1)} = \tilde{\mathbf{a}}^{(J)}$ and $\beta^{(s+1)} = \tilde{\beta}^{(J)}$, and the entire process is subsequently repeated. In summary, then, a single iteration of this modified EMAX algorithm consists of first computing the posterior probabilities $\{P_{t,j}(\Psi^{(s)})\}$ and then iterating (2.81) and (2.82) until convergence.

Recall that we must again choose values for $\mathbf{a}^{(0)}$ and $\beta^{(0)}$ to initialize our new algorithm. A logical choice for $\mathbf{a}^{(0)}$ is, as before, the parameter vector estimate obtained by applying the Yule-Walker equations to the original observed sequence. Moreover, since the parameter β is inversely proportional to the standard deviation of the driving-noise pdf, a logical choice for $\beta^{(0)}$ is the reciprocal of the sample standard deviation associated with the residual sequence $\mathbf{w}^{(0)}$, which can readily be obtained after applying the approximate inverse filter $1 - \sum_{i=1}^{K} a_i^{(0)} z_{-i}$ to the sequence of observations.

2.5 ARGMIX Signal Estimation in Additive Noise

Up to this point, we have considered only the problem of source identification as it applies to an unknown ARGMIX process. By developing a simple iterative techniques for solving this problem, we have demonstrated that good parameter estimates for ARGMIX processes can be generated by searching for local maxima of the likelihood surface. In fact, the techniques we developed in earlier sections constitute an important extension to the existing collection of classical methods, which were designed to solve the less complicated source identification problem in which the unknown process is assumed to be autoregressive but purely Gaussian.

The degree of success we were able to achieve in the source identification problem now leads us to question whether similar progress might be made in the equally important signal estimation problem. In this section, we therefore turn our attention to the problem of optimally filtering an ARGMIX process that has been corrupted by independent additive noise, under the assumption that we are given the true parameter values for both signal and noise. We soon discover, however, that the ARGMIX signal model is not well suited to the development of filtering or smoothing techniques that are both computationally efficient and globally optimal. Indeed, it appears that any algorithm designed to produce an optimal estimate of an ARGMIX signal in Gaussian noise necessarily incurs a computational cost that grows exponentially as a function of the length of the observed sequence.

After demonstrating the inherent algorithm complexity associated with the ARGMIX signal model, we discuss several alternative solutions to the signal estimation problem; these solutions are reasonable approximations which are much less computationally expensive, but which naturally lack the property of global optimality. Our discussion of these alternative methods will actually lead us to the creation of a new signal model which is altogether different from the original ARGMIX signal model, but which appears to be more practical and more general than the ARGMIX model. It is the analysis and development of this new model that will occupy us for the remaining portion of the thesis.

To demonstrate the difficulty we encounter when attempting to estimate an ARGMIX process that has been corrupted by additive noise, let us now consider a very simple yet illustrative problem of this kind. In particular, suppose we have a random signal $\{Y_t\}$ that has been generated according to the first-order AR difference equation

$$Y_t = aY_{t-1} + W_t, \tag{2.83}$$

where a is the single real-valued AR coefficient of the process and $\{W_t\}$ is a sequence of i.i.d. random variables, each distributed according to a fixed Gaussian-mixture pdf, which we denote by $f_W(\cdot)$. For simplicity, we shall assume that this driving-noise pdf $f_W(\cdot)$ consists of only two zero-mean Gaussian components, so that we may write it as

$$f_W(w) = \rho\mathcal{N}(w; 0, \sigma_1) + (1-\rho)\mathcal{N}(w; 0, \sigma_2). \tag{2.84}$$

To insure that the driving noise $\{W_t\}$ is indeed non-Gaussian (i.e., to preclude an assignment of ARGMIX parameter values that would yield a purely Gaussian signal), we impose the further conditions $0 < \rho < 1$ and $\sigma_2 \neq \sigma_1$ on the model parameters. We will assume for convenience, however, that the autoregression in (2.83) is initialized randomly at time $t = 0$ according to a Gaussian probability law, so that the signal variable Y_0 is characterized by the pdf $\mathcal{N}(\cdot; 0, \sigma_Y)$.

In contrast to the setup for the source identification problem, a clean observation of the signal $\{Y_t\}$ is not available in this case. Instead, we may observe the signal only after it has been corrupted by additive white Gaussian noise; hence, each element of the observed sequence $\{Z_t\}$ has the form

$$Z_t = Y_t + V_t, \tag{2.85}$$

where $\{V_t\}$ is a sequence of i.i.d. random variables, each having a pdf $f_V(\cdot)$ defined by

$$f_V(v) = \mathcal{N}(v; 0, \sigma_V). \tag{2.86}$$

All random variables contained in the sequences $\{V_t\}$ and $\{W_t\}$ are understood to be mutually independent.

It should be clear from the description given above that our observation model is completely characterized by the parameter vector

$$\boldsymbol{\Psi} = (a, \rho, \sigma_1, \sigma_2, \sigma_Y, \sigma_V). \tag{2.87}$$

Suppose, then, that we know the true value of each element of $\boldsymbol{\Psi}$, and furthermore that we have been furnished with realizations of the first N samples of the sequence $\{Z_t\}$. Then, given that we have observed the event $\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}$, our objective is to produce an MMSE estimate of the underlying signal realization $\mathbf{y}_{0:N-1}$. It is well known that the desired estimate $\hat{\mathbf{y}}_{0:N-1}$ is simply the conditional mean vector given by

$$\hat{\mathbf{y}}_{0:N-1} = E\{Y_{0:N-1}|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}; \Psi\}. \tag{2.88}$$

To understand why the computation of this optimal estimate becomes so complex as the observation length N gets large, let us now explore the structure of the estimate for various values of N. Our approach will be to build up the solution in a series of steps, starting with the simplest case in which $N = 1$ and progressively working toward the case in which N is allowed to be arbitrarily large.

First suppose that $N = 1$. By using our assumption about the initialization of the autoregressive formula in (2.83), together with the above description of the additive noise, we know that the observed variable Z_0 is constructed as a superposition of the independent zero-mean Gaussian random variables Y_0 and V_0, and is therefore itself a zero-mean Gaussian random variable. Therefore, the value of the optimal estimate in this case is given by the classical linear formula

$$\hat{y}_0 = \frac{\sigma_Y^2}{\sigma_Y^2 + \sigma_V^2} z_0. \tag{2.89}$$

Note that the signal estimate has an extremely simple form when $N = 1$, and that this estimate can be evaluated by performing only a single multiplication, provided that the leading scale factor is computed off-line before the realization z_0 is received.

Now consider the case in which $N = 2$. In this case, the observed vector $\mathbf{Z}_{0:1}$ no longer possesses a Gaussian pdf, since one of its two independent components — namely the signal vector $\mathbf{Y}_{0:1}$ — is not Gaussian. To verify this latter assertion, we need only observe that if the vector $\mathbf{Y}_{0:1}$ were Gaussian, then the variable Y_1, when conditioned on the event $Y_0 = y_0$, would also be Gaussian. From our model assumptions, however, we know that this cannot be true, because under such conditioning, Y_1 consists of the constant ay_0 plus a random innovation that is distributed according to a two-component Gaussian-mixture probability law. This does not necessarily imply that we can no longer use the classical techniques from linear-Gaussian estimation theory to construct an optimal estimate. On the contrary, we can still apply such techniques once we are able to decompose the new non-Gaussian estimation problem into a collection of separate Gaussian estimation problems. This point of view will allow us to apply a conventional linear processor in each of the individual Gaussian problems and then combine the resulting estimates using a special type of weighted average.

The key to decomposing the original non-Gaussian problem into purely Gaussian components is to condition the original problem on the outcome of the unobservable event $\Phi_1 = \phi_1$, where the random variable Φ_1 is (as defined in our earlier development) the pdf-selection variable corresponding to time $t = 1$. Such conditioning allows us to assume that we know which of the two Gaussian densities in the mixture gave rise to the driving-noise sample at time $t = 1$. For example, if we know with certainty that $\Phi_1 = 1$, then we may conclude that the conditional driving-noise pdf must be $\mathcal{N}(\cdot; 0, \sigma_1)$. Under this assumption, it now follows that the variable Y_1 is Gaussian when conditioned on the event $Y_0 = y_0$; hence, the entire signal vector $\mathbf{Y}_{0:1}$ is also Gaussian, as is the observed vector $\mathbf{Z}_{0:1}$.

Because we must now account for the two possibilities $\Phi_1 = 1$ and $\Phi_1 = 2$, we will need to further manipulate the expression in (2.88) in order to reduce it to simplest terms. Specifically, we now express the optimal estimator as

$$\begin{aligned}
\hat{\mathbf{y}}_{0:1} &= E\left\{\mathbf{Y}_{0:1}|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}; \boldsymbol{\Psi}\right\} && (2.90)\\
&= \int \mathbf{y} f_{\mathbf{Y}_{0:1}|\mathbf{Z}_{0:1}}(\mathbf{y}|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}; \boldsymbol{\Psi})d\mathbf{y} && (2.91)\\
&= \int \mathbf{y} \sum_{\phi=1}^{2} f_{\mathbf{Y}_{0:1},\Phi_1|\mathbf{Z}_{0:1}}(\mathbf{y}, \Phi_1 = \phi|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}; \boldsymbol{\Psi})d\mathbf{y} && (2.92)\\
&= \int \mathbf{y} \sum_{\phi=1}^{2} \Pr\{\Phi_1 = \phi|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}; \boldsymbol{\Psi}\} f_{\mathbf{Y}_{0:1}|\mathbf{Z}_{0:1},\Phi_1}(\mathbf{y}|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}, \Phi_1 = \phi; \boldsymbol{\Psi})d\mathbf{y} && (2.93)\\
&= \sum_{\phi=1}^{2} \Pr\{\Phi_1 = \phi|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}; \boldsymbol{\Psi}\} \int \mathbf{y} f_{\mathbf{Y}_{0:1}|\mathbf{Z}_{0:1},\Phi_1}(\mathbf{y}|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}, \Phi_1 = \phi; \boldsymbol{\Psi})d\mathbf{y} && (2.94)\\
&= \sum_{\phi=1}^{2} \Pr\{\Phi_1 = \phi|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}\} E\left\{\mathbf{Y}_{0:1}|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}, \Phi_1 = \phi; \boldsymbol{\Psi}\right\} && (2.95)\\
&= \sum_{\phi=1}^{2} \Pr\{\Phi_1 = \phi|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}; \boldsymbol{\Psi}\} \mathbf{C}_Y(\phi)(\mathbf{C}_Y(\phi) + \sigma_V^2 \mathbf{I})^{-1}\mathbf{z}_{0:1} && (2.96)\\
&= \sum_{\phi=1}^{2} \Pr\{\Phi_1 = \phi|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}; \boldsymbol{\Psi}\} \mathbf{C}_Y(\phi)\mathbf{C}_Z^{-1}(\phi)\mathbf{z}_{0:1} && (2.97)
\end{aligned}$$

where $\mathbf{I}$ is the identity matrix, $\sigma_V^2\mathbf{I}$ is the covariance matrix of the Gaussian noise vector $\mathbf{V}_{0:1}$, and $\mathbf{C}_Y(\phi)$ and $\mathbf{C}_Z(\phi)$ are, respectively, the conditional covariance matrices of the Gaussian signal vector $\mathbf{Y}_{0:1}$ and the Gaussian observation vector $\mathbf{Z}_{0:1}$ given that $\Phi_1 = \phi$.

One of the most immediately obvious properties of the estimate given in (2.97) is that it is actually a weighted average of two elemental linear estimates, each accounting for a unique choice of the true Gaussian pdf of the driving-noise sample at time $t = 1$. The overall estimate does not inherit the property of linearity, however, because the weighting coefficient for each term in the average is a nonlinear function of the observed data $\mathbf{z}_{0:1}$. Nonetheless, this weighting coefficient $\Pr\{\Phi_1 = \phi|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}; \boldsymbol{\Psi}\}$ can still be easily evaluated using Bayes' rule, since the parameter vector $\boldsymbol{\Psi}$ is precisely known. For example, in the case where $\Phi_1 = 1$, we can write

$$\begin{aligned}
\Pr\left\{\Phi_1 \right.&= \left. 1|\mathbf{Z}_{0:1} = \mathbf{z}_{0:1}; \boldsymbol{\Psi}\right\}\\
&= \frac{\Pr\{\Phi_1 = 1; \boldsymbol{\Psi}\} f_{\mathbf{Z}_{0:1}|\Phi_1}(\mathbf{z}_{0:1}|\Phi_1 = 1; \boldsymbol{\Psi})}{\sum_{\phi=1}^{2} \Pr\{\Phi_1 = \phi; \boldsymbol{\Psi}\} f_{\mathbf{Z}_{0:1}|\Phi_1}(\mathbf{z}_{0:1}|\Phi_1 = \phi; \boldsymbol{\Psi})} && (2.98)\\
&= \frac{\frac{\rho}{|\mathbf{C}_Z(1)|^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{z}_{0:1}^T \mathbf{C}_Z^{-1}(1)\mathbf{z}_{0:1}\right\}}{\frac{\rho}{|\mathbf{C}_Z(1)|^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{z}_{0:1}^T \mathbf{C}_Z^{-1}(1)\mathbf{z}_{0:1}\right\} + \frac{1-\rho}{|\mathbf{C}_Z(2)|^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{z}_{0:1}^T \mathbf{C}_Z^{-1}(2)\mathbf{z}_{0:1}\right\}}, && (2.99)
\end{aligned}$$

where we have used the explicit form for a bivariate zero-mean Gaussian density.

By comparing (2.89) and (2.97), we see that the optimal estimate derived for the case $N = 2$ requires considerably more computation than does the estimate for the case $N = 1$. Of course, part of this increase in computational cost — for example, the evaluation of a matrix-vector product rather than a product of two scalars — is a direct consequence of the increase in the length of the observation; this portion of the additional cost would be incurred even in a purely Gaussian estimation problem. The remaining amount of computation is our primary concern, for this is the amount we incur solely because of the non-Gaussian nature of problem. Clearly, the same kinds of matrix-vector operations that would be required to form an estimate in the purely Gaussian case must now be performed two times (i.e., for two distinct contingencies) to obtain the final estimate in (2.97); moreover, for each of the two elemental estimates computed, an associated weight factor must also be computed. Hence, when $N = 2$, the amount of computation that is needed to generate an optimal estimate in the non-Gaussian problem is at least twice the amount required in the analogous Gaussian problem.

As we might expect, the computational expense doubles yet again when $N = 3$, because in this case we must account for each of the four possible events $\mathbf{\Phi}_{1:2} = (1,1)$, $\mathbf{\Phi}_{1:2} = (1,2)$, $\mathbf{\Phi}_{1:2} = (2,1)$, and $\mathbf{\Phi}_{1:2} = (2,2)$ in order to reduce the problem to a collection of familiar Gaussian sub-problems. The exponential growth in the number of possible pdf-selection sequences (and hence the amount of computation) continues as the observation length N increases; this phenomenon is depicted in Figure 2-8. Indeed, when N is allowed to be arbitrarily large, we must account for every possible event of the form $\mathbf{\Phi}_{1:N-1} = (\phi_1, \phi_2, \cdots, \phi_{N-1})$, of which there are 2^{N-1} in all. For this general case, the expression for the estimate $\hat{\mathbf{y}}_{0:N-1}$ is given by

$$\hat{\mathbf{y}}_{0:N-1} = \sum_{\phi_1=1}^{2}\sum_{\phi_2=1}^{2}\cdots\sum_{\phi_{N-1}=1}^{2} \Pr\{\mathbf{\Phi} = \boldsymbol{\phi} | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}; \mathbf{\Psi}\} \mathbf{C}_Y(\boldsymbol{\phi}) \left(\mathbf{C}_Y(\boldsymbol{\phi}) + \sigma_V^2 \mathbf{I}\right)^{-1} \mathbf{z}_{0:N-1}, \tag{2.100}$$

which is seen to be a direct extension of the expression given in (2.97). The above estimate is clearly a weighted average of 2^{N-1} elemental estimates, each tailored to a unique realization of the state sequence $\mathbf{\Phi}_{1:N-1}$.

It should now be evident that the amount of computation involved in evaluating the optimal estimate in (2.100) will be prohibitive even when N is only moderately large. Specifically, evaluating $\hat{\mathbf{y}}_{0:N-1}$ would take more than 2^{N-1} times as many arithmetic operations than would evaluating an estimate of the same length in the case where the signal and noise vectors are independent and purely Gaussian. We further note that, although there exist alternative methods for computing the exact value of the optimal estimate in (2.100), it appears that each of these methods incurs a computational cost that grows exponentially as a function of the observation length N.

For example, one such alternative technique would be to optimally combine the output signals produced by a bank of Kalman smoothers operating in parallel on the observation

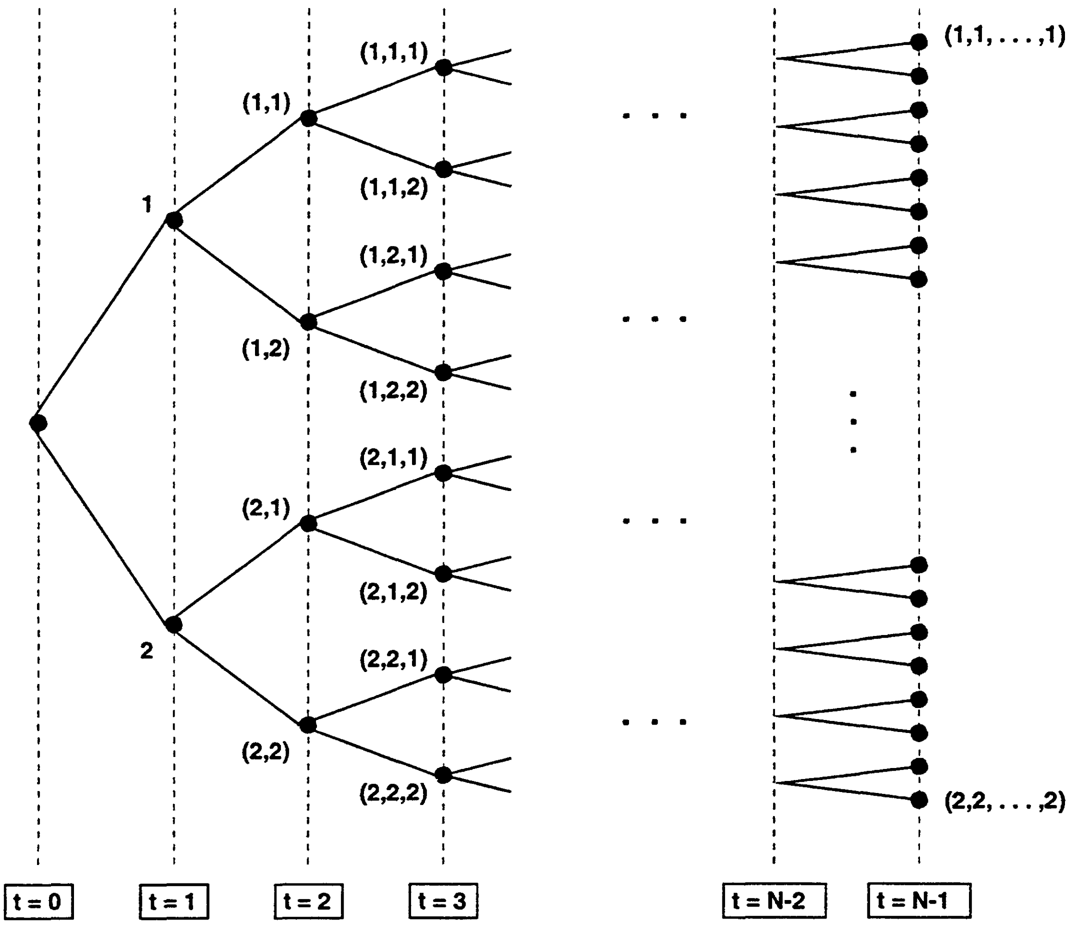


Figure 2-8: Binary tree diagram depicting exponential growth in the number of possible pdf-selection sequences that could be realized from time 0 up to time t. Vertical dashed lines correspond to fixed time indices. Each potential value of the underlying state sequence $\phi_{1:t}$ is shown in parentheses next to its associated node on the tree.

$\mathbf{z}_{0:N-1}$. But it is well known that the classical Kalman smoother is able to produce a globally optimal estimate of a signal in additive noise only if the signal and noise are jointly Gaussian and all distributional parameters are known. With this approach, therefore, each Kalman smoother would have to be configured to operate under the assumption that a particular value of the underlying state sequence $\Phi_{1:N-1}$ is in fact the true value; this is the only way that the Gaussian assumption would hold. Thus, even though each of the recursively implemented Kalman smoothers might be considered to be computationally efficient when operating in isolation, a total of 2^{N-1} such smoothers (one for each possible state sequence) would still be required to generate the overall optimal estimate.

A number of suboptimal techniques have been proposed in the literature to overcome the computational complexity of the ARGMIX signal estimation problem. One of the earliest of these techniques was put forth by Ackerson and Fu [1], who suggested that the posterior pdf of the signal variable be modeled as purely Gaussian at each time. Under this assumption, the signal estimate at a given time index would be the mean of the posterior density; at the following time index, the pdf of the predicted signal value would consist of M Gaussian components (owing to the M-fold branching process depicted in Figure 2-8), but these would be subsequently reformed into a single Gaussian pdf through a moment-matching procedure. Several methods similar to that of Ackerson and Fu were proposed a short time later [31, 165, 227].

A very different approach based on the notion of random sampling was taken by Akashi and Kumamoto [8]. They viewed the collection of all possible underlying driving-noise state sequences as a population, and they generated a suboptimal signal estimate using a relatively small number of these state sequences chosen at random from the population. This technique had the theoretical advantage that it could produce an estimate arbitrarily close to the optimal estimate if a sufficiently large number of sequences were selected. Other proposed approaches to ARGMIX signal estimation have been based on the concept of pruning the tree in Figure 2-8 to allow only the most likely branches as candidate hypotheses [179, 204]. Many such techniques bear strong similarities to methods used for tracking moving targets in a dense multi-target environment [18]. In the remainder of the thesis, we shall pursue an entirely different approach to the non-Gaussian signal estimation problem; our method is based on approximating the underlying signal with a finite-state Markov dynamical model. We begin exploring this approach in detail in Chapter 3.

2.6 Discussion

2.6.1 Remarks on the EMAX Algorithm

The computations that constitute the EMAX algorithm have an intuitively pleasing form, are easy to implement in computer code, and consume little computer memory. The empirical results presented in Section 2.3 suggest that the EMAX algorithm has at least three distinct advantages over other techniques proposed for similar estimation problems: (i) it produces high-quality estimates, since it uses the likelihood function as a guide for finding solutions, (ii) it converges reliably to a stationary point of the likelihood function, by virtue of being a generalized EM algorithm, and (iii) it is extremely versatile because the Gaussian-mixture pdf is able to model a wide range of densities very well.

Although the EMAX algorithm is a powerful method for estimating non-Gaussian signal parameters, a number of issues must still be addressed before it can be transformed into a robust signal analysis tool. For example, we have given only cursory consideration to the initialization of the EMAX algorithm in the analysis presented here. For the examples given in Section 2.3, we adopted an initialization method based on its conceptual and computational simplicity. This method worked reasonably well for the limited set of examples addressed in that section. However, since initial estimates are the key to good performance, we need an initialization procedure that will consistently lead to points of high likelihood after the algorithm has been iterated to convergence. This would be a useful direction for future work in ARGMIX parameter identification.

In addition, for certain problems (particularly those for which the Gaussian-mixture pdf contains many components), it would be useful to speed up the convergence of the EMAX algorithm. This might be accomplished by iterating the algorithm until reaching the vicinity of a local maximum, and then applying a more efficient method (e.g., the Newton-Raphson technique) to move to the peak. Also, it would be useful to detect, during the operation of the algorithm, whether a degenerate parameter estimate is being approached, so that the algorithm could be restarted elsewhere in the parameter space. Furthermore, we note that in any practical setting, our observations of the signal of interest will be corrupted by additive noise. For example, the digital communications application presented in Section 2.3.4 is a typical case in which additive noise is unavoidable. Hence, a modification of the EMAX algorithm should be devised for estimating the parameters of an ARGMIX process when noise is present.

Another issue that must be addressed is how to estimate the parameters K (the order of the autoregression) and M (the number of constituent densities in the Gaussian mixture) when these parameters are not given in advance. Moreover, we need to be able to assess the effect that incorrectly chosen values for K and M would have on the variances of the remaining parameter estimates. Because the Gaussian mixture model is quite flexible even when M is very small (say, 2 or 3), the selection of a suitable value for M could probably be easily accomplished by trial and error in most cases. A number of criteria have already been proposed for selecting an appropriate value for K. The most widely used among these include the information-theoretic criterion [6] and final prediction error metric [4], both of which were developed by Akaike, as well as the minimum description length, which was developed independently by Rissanen [158] and Schwarz [168]. The approach used in each of these criteria is essentially to augment the log-likelihood function with a penalty term which increases monotonically with the parameter K. Adding such a penalty term has the effect of counteracting the monotonic decrease of the prediction error variance that typically results when the model order is increased. It is conceivable that a new EM algorithm could be derived to identify ARGMIX signal parameters (including the parameter K) upon incorporating one of the model order estimation criteria mentioned above.

2.6.2 Suggested Future Direction for ARGMIX Signal Estimation

There is a relatively simple approach to the problem of suboptimal ARGMIX signal estimation which appears to have been overlooked in the literature, and which we mention here as a possible direction for future research. The motivation for using this approach is similar to that for using an FIR Wiener smoother as an alternative to the true Wiener smoother in the purely Gaussian case. The basic idea is to generate an estimate of the signal at each time t using only a finite-length portion of the observation in the vicinity of time t. The underlying assumption for this finite-memory estimation scheme is that any given portion of the observation is accurately characterized by a multivariate Gaussian-mixture pdf having a fixed number of components. Under this assumption, the processor itself would be a nonlinear combination of the outputs of a fixed number of FIR Wiener smoothers (one for each component in the mixture), as was the case for the optimal smoother discussed in Section 2.5. A potential difficulty in implementing the approach, however, is that the parameters characterizing the best Gaussian-mixture approximation of a portion of the observation cannot be expressed easily in terms of the parameters of the ARGMIX measurement model. Moreover, the approach may perform poorly if the dependence length induced by the AR filter in the model is large relative to the length of the portion of the observation that has been chosen for processing.

However, there exists an alternative method of implementing this approach which may overcome these difficulties. In particular, we can first pass the noisy measurement through an invertible LTI system, $\mathcal{S}$, then apply a nonlinear estimator to the filtered result, and finally pass this nonlinearly processed waveform though the inverse of the original LTI system, $\mathcal{S}^{-1}$. From the principle of reversibility [215], we know that if the nonlinear estimator applied in the second stage of this system were truly optimal for the result produced by $\mathcal{S}$, then the overall system would be optimal for the original noisy measurement. A convenient choice for the LTI system used in the first stage is the inverse of the original AR filter. Since this inverse filter is linear, we can examine its effect on the signal and noise separately. In particular, the signal becomes whitened, i.e., transformed back into the original i.i.d. Gaussian-mixture driving sequence; on the other hand, the observation noise, which was originally white and Gaussian, becomes colored by the inverse filter, but remains Gaussian. Hence, the roles of signal and noise are now essentially reversed.

Once this initial transformation has been carried out, the non-Gaussian samples in the measurement exhibit are no longer mutually dependent. Since any finite-length portion of the processed observation is now truly characterized by a multivariate Gaussian-mixture pdf, there exists a clear link between the parameters of the original ARGMIX model and the finite-memory processor that should be used for smoothing. Let us denote the transformed signal and observations by $\{\tilde{y}_t\}$ and $\{\tilde{z}_t\}$, respectively. Then to generate an optimal estimate of $\tilde{y}_t$ using, say, the subsequence of observations $\tilde{\mathbf{z}}_{t-J:t+J}$, we would need to combine estimates produced by M^{2J+1} distinct finite-length Wiener smoothers, each operating on a unique assumption about the true value of the subsequence of driving-noise states $\boldsymbol{\phi}_{t-J:t+J}$. (Recall that M is the number of components in the original Gaussian-mixture driving-noise pdf.) The appropriate weighting coefficients used to combine these estimates would be the posterior probabilities of the individual state subsequences, based on the value of $\tilde{\mathbf{z}}_{t-J:t+J}$. Once this estimation procedure has been performed for all samples of the transformed signal, the resulting sequence of estimates can then be passed through the original AR filter once again to obtain the final signal estimate.

Chapter 3

Approximating Stationary Signals with Finite-State Markov Models

3.1 Introduction

The analysis in the latter half of Chapter 2 identified some of the practical difficulties involved in developing an optimal technique for estimating a non-Gaussian signal in additive noise. Throughout that analysis, our attention was focused on an estimation problem with a particularly simple structure; specifically, the signal was a stationary ARGMIX process, the noise was a stationary white Gaussian process, and the signal and noise were assumed to be independent. It was demonstrated through a simple example that the optimal estimation scheme for the ARGMIX problem was not particularly difficult to derive or to understand; in fact, because we were able to decompose the original non-Gaussian problem into more familiar, purely Gaussian subproblems, we could express the final solution in closed form as a (nonlinear) weighted average of many different Wiener filters. However, this optimal estimation scheme was, practically speaking, impossible to implement because it consumed a prohibitive amount of computation, even in cases where the observed sequence contained only a modest number of samples.

Our immediate temptation in this situation is to search for a tractable, yet sufficiently sophisticated approximation to the best processor, with the hope that the resulting approximate scheme will perform satisfactorily in place of the optimal scheme. As we mentioned at the end of Chapter 2, this basic approach has been taken by many researchers for solving non-Gaussian problems similar to the ARGMIX signal estimation problem. In this chapter, however, we shall take a much different approach toward developing approximate signal processing algorithms — an approach which reflects a fundamental shift in paradigm for the remainder of the thesis. In particular, we will attempt to approximate the true random process with another random process whose structure is much simpler and which gives rise to algorithms that are much more computationally efficient.

The collection of random processes that we will consider as approximations is the class of finite-state hidden Markov models, or HMMs. An HMM consists of two basic components: (i) a Markov chain, which characterizes the underlying temporal structure of the HMM in terms of transitions on a discrete set of states; and (ii) a collection of probability density functions (one for each state of the Markov chain), which characterize the output of the HMM. In general, the state values assumed by the underlying Markov chain in the HMM are not directly observable. Instead, at each time index, the HMM output is actually a function of the current state; this function is random, rather than deterministic, and is completely characterized by the pdf assigned to that state.

This class of finite-state random processes was introduced as a statistical modeling tool in a series of papers by Baum and his colleagues [19, 20, 21, 22, 23]; since that time, various properties and algorithms associated with HMMs have been developed extensively by other researchers. The most widespread practical use of these models has been in the area of speech processing; specifically, they have been applied to such problems as automatic speech recognition, speaker identification, and language identification, to name just a few [78, 84, 85, 139, 144, 150, 155, 233]. More recently, HMMs have been used to approximate certain types of low-dimensional dynamical systems (e.g., systems that are chaotic), for the purpose of either predicting the output of such systems or enhancing the output after it has been contaminated with additive noise [97, 130, 157].

We develop the HMM-based approximation concept further in the next two subsections; we first outline the basic assumptions and notation that will be used in connection with the signal approximation problem, and we then give a concise formulation of the problem itself. In the third subsection, we describe how the remaining material in the chapter is organized.

3.1.1 Preliminary Assumptions and Notation

3.1.1.1 Assumptions on the True Source Signal

Our use of HMMs as approximating processes will allow us to solve problems involving a very broad class of AR signals, which includes the ARGMIX subclass as a special case. In the sequel, we will assume that the source signal being approximated, $\{Y_t\}$, is a stationary AR process described by the Kth-order nonlinear difference equation

$$Y_t = h(Y_{t-1}, Y_{t-2}, \cdots, Y_{t-K}, W_t), \tag{3.1}$$

where $h(\cdot)$ is a deterministic function and $\{W_t\}$ is a sequence of i.i.d. random variables, each distributed according to the pdf $f_W(\cdot)$.

When seeking an approximation to $\{Y_t\}$, we will find it convenient to represent this signal as the output of a nonlinear dynamical system driven by the white noise process $\{W_t\}$. In view of the process description given in (3.1), we see that a suitable definition for the state vector $\mathbf{X}_t$ of such a dynamical system is given by

$$\mathbf{X}_t = (Y_t, Y_{t-1}, \cdots, Y_{t-K+1}). \tag{3.2}$$

This definition allows us to decompose (3.1) into a dynamical equation and an output

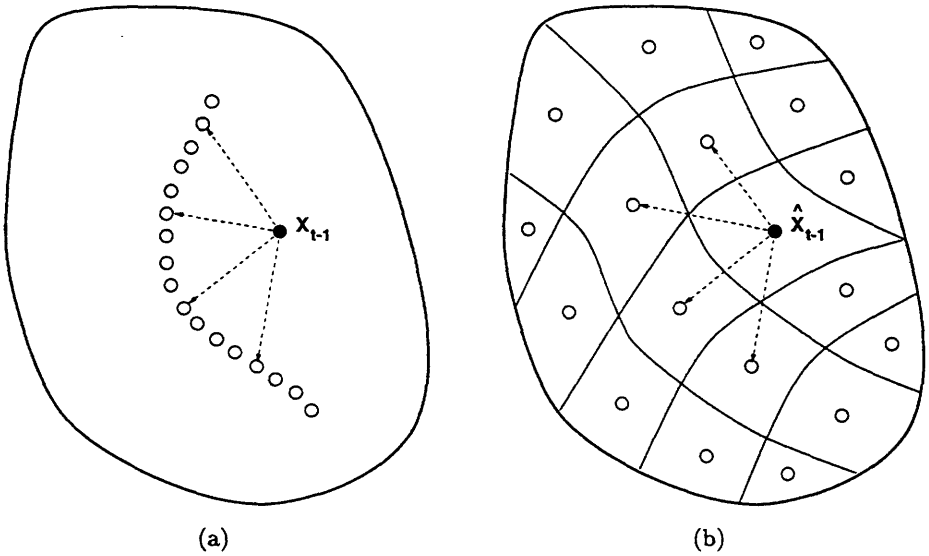


Figure 3-1: Depictions of possible transitions of the state vector from time $t-1$ to time t under (a) the true dynamical structure; and (b) the quantized dynamical structure within the partitioned state space.

equation, as shown by

$$\mathbf{X}_t = \mathbf{H}(\mathbf{X}_{t-1}, W_t) \tag{3.3}$$

$$Y_t = G(\mathbf{X}_t), \tag{3.4}$$

where the state transition function $\mathbf{H}(\cdot)$ transforms $\mathbf{X}_{t-1}$ and W_t into $\mathbf{X}_t$ (via the original regression function $h(\cdot)$) according to

$$\mathbf{H}(\mathbf{X}_{t-1}, W_t) = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \mathbf{X}_{t-1} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} h(\mathbf{X}_{t-1}, W_t) \tag{3.5}$$

and the output function $G(\cdot)$ merely extracts the first element of its vector argument. This representation of the source signal is useful because it allows us to think of the underlying dynamics of the signal in a linear-algebraic sense, i.e., in terms of a transformation of the state from one time to the next within a K-dimensional vector space, as depicted in Figure 3-1(a).

3.1.1.2 Assumptions on the HMM-Based Signal Approximation

Our approximation to the true dynamics is represented by an L-state HMM, which is required to satisfy certain constraints. In particular, we will insist that set of HMM states stand in one-to-one correspondence with a collection of regions in the original K-dimensional state space.[1] We denote these regions by $\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_L$, and we require that they satisfy the conditions

$$\mathcal{R}_1 \cup \mathcal{R}_2 \cup \cdots \cup \mathcal{R}_L = \mathbb{R}^K \tag{3.6}$$

and

$$\mathcal{R}_i \cap \mathcal{R}_j = \varnothing, \qquad i \neq j, \tag{3.7}$$

so that any point in $\mathbb{R}^K$ is contained in exactly one of the $\mathcal{R}_i$. With the regions defined in this way, we can explicitly specify a mapping between the original continuous-valued state space and the discrete-valued state set of the HMM (which we take to be, without loss of generality, the set $\{1, 2, \cdots, L\}$). This mapping, which we denote by $\theta(\cdot)$, is given by

$$\theta(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \mathcal{R}_1; \\ 2 & \text{if } \mathbf{x} \in \mathcal{R}_2; \\ \vdots & \vdots \\ L & \text{if } \mathbf{x} \in \mathcal{R}_L. \end{cases} \tag{3.8}$$

We will refer to the constraint imposed on the HMM by this mapping as the state-space partitioning constraint. A notional depiction of the resulting approximate dynamics in the partitioned state space is shown in Figure 3-1(b).

We will denote the underlying L-state Markov chain in the HMM by $\{\Theta_t\}$, and we will assume that this chain is homogeneous, i.e., that any conditional probability of the form $\Pr\{\Theta_{t+1} = j | \Theta_t = i\}$ depends only on the values of i and j and is entirely independent of the value of the time variable t. The stochastic structure of a homogeneous Markov chain is completely characterized by two distinct sets of parameters: (i) a collection of state transition probabilities, which we denote by $\{Q(i,j)\}_{i,j=1}^{L}$; and (ii) a collection of initial state probabilities, which we denote by $\{P(j)\}_{j=1}^{L}$. These parameters are defined, respectively, by

$$Q(i,j) = \Pr\{\Theta_{t+1} = j | \Theta_t = i\}, \qquad i,j = 1, 2, \cdots, L \tag{3.9}$$

[1] The term region, as it is used here and in the sequel, should be understood in an intuitive sense as a set that consists of a single piece. To be more technically precise, we may consider a region to be a nonempty connected set [2, 163], i.e., a set having the property that any two of its points can be joined by a polygonal line which also lies in the set. This technical definition will be used only on rare occasions in the remainder of the thesis, however.

and

$$P(j) = \Pr\{\Theta_0 = j\}, \qquad j = 1, 2, \cdots, L. \tag{3.10}$$

Because we will be exclusively considering the approximation of stationary processes, we impose the additional constraint that the HMM-based representation must itself be stationary. Under this constraint, the pmf of the initial state variable is identical to the marginal pmf for every other state variable in the chain. As a consequence, the initial state probabilities and state transition probabilities of the chain are related through the equation [27]

$$P(j) = \sum_{i=1}^{L} P(i)Q(i,j), \qquad j = 1, 2, \cdots, L. \tag{3.11}$$

In addition, the joint pmf characterizing a pair of successive random variables (Θ_t, Θ_{t+1}) is constant for all time. We will have occasion to refer to this joint pmf later in the chapter; its elements will be denoted by $\{R(i,j)\}_{i,j=1}^{L}$.

To complete the definition of the HMM, we now require only a collection of L densities, one for each of the L states of the Markov chain. We shall assume that these densities, once specified, remain fixed for all time, as do the parameters characterizing the Markov chain. However, we will adopt two separate notations for these HMM output densities, depending on whether we are describing an approximation for the sequence of state vectors $\{\mathbf{X}_t\}$ or the sequence of signal variables $\{Y_t\}$. In the former case, we will denote the output densities by $\{f_i(\cdot)\}_{i=1}^{L}$; in the latter case, we use the notation $\{g_i(\cdot)\}_{i=1}^{L}$. It is understood that the densities $f_j(\cdot)$ and $g_j(\cdot)$ correspond to state j of the underlying Markov chain, so that we may write

$$f_j(\mathbf{x}) = f_{\tilde{\mathbf{X}}_t|\Theta_t}(\mathbf{x}|\Theta_t = j), \qquad j = 1, 2, \cdots, L \tag{3.12}$$

and

$$g_j(y) = f_{\tilde{Y}_t|\Theta_t}(y|\Theta_t = j), \qquad j = 1, 2, \cdots, L \tag{3.13}$$

for all values of the time index t. We will assume that any output pdf included in an HMM is continuous throughout its domain (except possibly on a subset having zero measure) and contains no impulses.

3.1.2 Problem Statement and Approach to Solution

We will assume that all of the quantities that define the true source signal — i.e., the order of the autoregression, K, the regression function itself, $h(\cdot)$, and the driving-noise pdf, $f_W(\cdot)$ — are precisely known. Under these assumptions, together with the stationarity constraint, we can (at least in principle) derive the exact form of the pdf that characterizes the true source signal $\{Y_t\}$, or equivalently, the pdf that characterizes the true state-vector sequence $\{\mathbf{X}_t\}$. Hence, we will freely assume that these pdfs are also given. The problem we consider in this chapter will concern a finite-length portion of the true state-vector sequence given by

$$\mathbf{X}_{0:N-1} = (\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_{N-1}). \tag{3.14}$$

We can think of this subsequence as the collection of underlying state vectors associated with a measurement of the source signal that we will make at some point in the future. Our objective is to approximate the random subsequence $\mathbf{X}_{0:N-1}$ with another subsequence $\tilde{\mathbf{X}}_{0:N-1}$ given by

$$\tilde{\mathbf{X}}_{0:N-1} = (\tilde{\mathbf{X}}_0, \tilde{\mathbf{X}}_1, \cdots, \tilde{\mathbf{X}}_{N-1}), \tag{3.15}$$

where the elements of $\tilde{\mathbf{X}}_{0:N-1}$ are the outputs of an L-state HMM. Our HMM-based approximation must satisfy both the stationarity constraint and the state-space partitioning constraint described earlier.

Our approach to finding a suitable approximation will be to attempt to fit the pdf of the approximate subsequence $\tilde{\mathbf{X}}_{0:N-1}$ to the pdf of the true subsequence $\mathbf{X}_{0:N-1}$. The attributes of an optimal fit must be defined in an appropriate statistical sense to be made more precise later in the chapter. Observe that the approximating pdf is entirely characterized by the parameters of the HMM, i.e., by the initial state probabilities $\{P(i)\}_{i=1}^{L}$, the state transition probabilities $\{Q(i,j)\}_{i,j=1}^{L}$, and the state output densities $\{f_i(\cdot)\}_{i=1}^{L}$. Thus, we seek optimal values for these parameters expressed in terms of the true pdf. Because all of the HMM parameters depend upon the mapping $\theta(\cdot)$ defined in (3.8), our selection of the best possible state-space partition will be a critical part of the overall optimization procedure.

In most situations, our actual objective is not to approximate the state-vector subsequence $\mathbf{X}_{0:N-1}$, but rather to approximate the signal subsequence $\mathbf{Y}_{0:N-1}$ defined by

$$\mathbf{Y}_{0:N-1} = (Y_0, Y_1, \cdots, Y_{N-1}). \tag{3.16}$$

It turns out, however, that the optimal state-vector approximation $\tilde{\mathbf{X}}_{0:N-1}$ is not only much easier to derive, but can also be used to generate an optimal approximation

$$\tilde{\mathbf{Y}}_{0:N-1} = (\tilde{Y}_0, \tilde{Y}_1, \cdots, \tilde{Y}_{N-1}) \tag{3.17}$$

of the true signal subsequence. In particular, the elements of $\tilde{\mathbf{Y}}_{0:N-1}$ are individually defined by

$$\tilde{Y}_t = G(\tilde{\mathbf{X}}_t), \qquad t = 0, 1, \cdots, N-1, \tag{3.18}$$

where the function $G(\cdot)$ returns the first element of its vector argument. It is straightforward to show that the signal approximation $\tilde{\mathbf{Y}}_{0:N-1}$ is once again the output of an HMM; in fact, this HMM has exactly the same Markov chain parameters as those derived for $\tilde{\mathbf{X}}_{0:N-1}$. However, an important change to the model is that a univariate pdf $g_i(\cdot)$ must now be assigned to state i of the HMM to take the place of the K-variate pdf $f_i(\cdot)$ that was assigned earlier. In light of the relationship in (3.18), we see that this new pdf can readily be derived from the old pdf by integrating out the unnecessary elements, as shown by

$$g_i(y) = \int_{(y,\mathbf{y}_{t-1:t-K})\in\mathcal{R}_i} f_i(y,\mathbf{y}_{t-1:t-K})\,d\mathbf{y}_{t-1:t-K}. \tag{3.19}$$

3.1.3 Chapter Organization

The remainder of the chapter is organized in the following way. We begin by motivating and defining a figure of merit by which we can assess the quality of a given HMM-based approximation; this allows us to formulate the approximation procedure as a well defined optimization problem over a subclass of HMMs. We then give a detailed formulation and solution of the approximation problem in the case where the true signal is taken to be a stationary, first-order random process which is in general non-Gaussian. The results obtained through this theoretical analysis are then implemented using a numerical, gradient-based algorithm; with this algorithm, we develop several different HMM-based approximations for a specific first-order AR linear-Gaussian random process, and we demonstrate that the accuracy of the approximation improves as the number of states included in the model is increased. We then show that our theoretical results can be readily generalized so that they apply equally well to higher-order stationary AR processes. Finally, we provide a discussion of the key concepts developed in the chapter, including an analysis of the advantages and limitations of certain assumptions we have placed on the true and approximate processes.

3.2 Establishing Criteria for a Good Approximation

Thus far we have stated that the mathematical structure of our signal approximation will take the form of an HMM, but we have not yet established criteria that would allow us to determine whether a particular approximation is the best within a specified class. We begin this section by describing a universal metric by which the best possible signal approximation can be identified. The optimization procedure based on this metric would, roughly speaking, seek to maximize the confusion of an observer who is trying to distinguish between the true and approximate processes using complete statistical information about each process. Because this metric is difficult to analyze, however, we ultimately settle on an alternative, information-theoretic figure of merit known as Kullback-Leibler distance, which provides a measure of similarity between the pdf of the true signal and the pdf of the approximate signal.

3.2.1 Minimax Probability-of-Error Approach to Approximation

Let us first consider the following experiment, which, for the sake of simplicity, involves the approximation of only a single random variable, rather than an entire random process: Let $f_Y(\cdot)$ be the known pdf of the random variable Y, and let $\mathcal{F} = \{f_{\tilde{Y}}(\cdot)\}$ be a collection of feasible approximations of $f_Y(\cdot)$. We wish to find the best approximation contained in the set $\mathcal{F}$. To determine the quality of a particular approximation, we seek the assistance of an independent observer and carry out our quality test in the form of a game. Specifically, suppose the observer is informed that he will be given, in each of a series of experimental trials, a set of realizations $\{z_t\}_{t=0}^{N-1}$ from the N random variables $\{Z_t\}_{t=0}^{N-1}$, which are independent and identically distributed according to the pdf $f_Z(\cdot)$. His task during each experimental trial is to render a decision indicating which of the following two hypotheses is true:

$$\begin{aligned} H_0 &: f_Z(\cdot) = f_{\tilde{Y}}(\cdot) \\ H_1 &: f_Z(\cdot) = f_Y(\cdot) \end{aligned}$$

The observer is furnished with complete descriptions of both $f_Y(\cdot)$ and $f_{\tilde{Y}}(\cdot)$, and he is told that the hypotheses H_0 and H_1 are equally likely to be true. Our game is structured such that, when the observer guesses correctly, we must pay him a fixed amount of money; when the observer guesses incorrectly, he must pay us a fixed amount. The goal of either player in the game is to maximize his total winnings over the series of experimental trials.

Clearly, with his complete knowledge of the experimental setup, the observer can implement the best possible statistical test for deciding between H_0 and H_1, namely the likelihood ratio test (LRT) or Neyman-Pearson test [15, 94, 132, 147, 215]. Thus, to render his decision, he uses the optimal rule

$$\text{Declare} \left\{ \begin{array}{c} H_0 \text{ true} \\ H_1 \text{ true} \end{array} \right\} \text{when } \ell(\mathbf{z}_{0:N-1}) \left\{ \begin{array}{c} < \\ \geq \end{array} \right\} 0, \tag{3.20}$$

where $\ell(\cdot)$ is the log-likelihood ratio defined by

$$\ell(\mathbf{z}_{0:N-1}) = \log \frac{f_{\mathbf{Y}_{0:N-1}}(\mathbf{z}_{0:N-1})}{f_{\tilde{\mathbf{Y}}_{0:N-1}}(\mathbf{z}_{0:N-1})}. \tag{3.21}$$

Of course, we are well aware that the observer will use the best available decision rule on each trial; to do otherwise would be to unnecessarily increase the chance of a monetary loss. It can easily be shown that, when the observer uses this optimal rule, his expected winnings will increase directly with the probability that he makes a correct decision. If we now use our knowledge of the two densities involved, as well as our knowledge of the observer's gaming strategy, we can quantify his winnings simply by calculating this probability. This calculation yields

$$\begin{aligned} &\Pr\{\text{Correct decision}\} \\ &\qquad = \Pr\{H_0 \text{ true, Declare } H_0\} + \Pr\{H_1 \text{ true, Declare } H_1\} &(3.22) \\ &\qquad = \Pr\{H_0 \text{ true}\}\Pr\{\text{Declare } H_0 \mid H_0 \text{ true}\} \\ &\qquad\qquad + \Pr\{H_1 \text{ true}\}\Pr\{\text{Declare } H_1 \mid H_1 \text{ true}\} &(3.23) \\ &\qquad = \tfrac{1}{2}\Pr\{\ell(\mathbf{Z}_{0:N-1}) < 0 \mid H_0 \text{ true}\} + \tfrac{1}{2}\Pr\{\ell(\mathbf{Z}_{0:N-1}) \geq 0 \mid H_1 \text{ true}\} &(3.24) \\ &\qquad = \tfrac{1}{2}\int_{-\infty}^{0} f_{\ell|H_0}(\xi|f_Z = f_{\tilde{Y}})\, d\xi + \tfrac{1}{2}\int_{0}^{\infty} f_{\ell|H_1}(\xi|f_Z = f_Y)\, d\xi, &(3.25) \end{aligned}$$

where $f_{\ell|H_0}(\cdot)$ and $f_{\ell|H_1}(\cdot)$ are the conditional densities of the log-likelihood ratio $\ell(\cdot)$ given that H_0 is true and that H_1 is true, respectively. Note in the final step above that we have indicated explicitly the dependence of the observer's winnings on the approximate pdf $f_{\tilde{Y}}(\cdot)$.

The probability expressed in (3.25) will be fixed once we have specified the pdf $f_{\tilde{Y}}(\cdot)$. We are allowed to select *any* pdf from the set $\mathcal{F}$. Clearly, if we are to maximize our winnings, we should choose the pdf that makes the above probability as small as possible. That is, the optimal pdf $f^*_{\tilde{Y}}(\cdot)$ will be the one that satisfies

$$f^*_{\tilde{Y}}(\cdot) \quad = \quad \arg\min_{f_{\tilde{Y}}(\cdot)\in\mathcal{F}} \left\{ \tfrac{1}{2}\int_{-\infty}^{0} f_{\ell|H_0}(\xi|f_Z = f_{\tilde{Y}})\,d\xi + \tfrac{1}{2}\int_{0}^{\infty} f_{\ell|H_1}(\xi|f_Z = f_Y)\,d\xi \right\}. \quad (3.26)$$

Although we have expressed the optimal density here in terms of the observer's probability of correct decision, we could equivalently represent it in terms of his probability of error. From this alternative perspective, the optimal density in $\mathcal{F}$ is the one causes the most confusion for an observer who is trying to discriminate between $f_Y(\cdot)$ and $f_{\tilde{Y}}(\cdot)$ using a statistically optimal test.

3.2.2 Kullback-Leibler Distance as a Figure of Merit

While it is very useful to reason through an experiment such as the one described above to find the best, most general measure of approximation quality, unfortunately the conclusions we have drawn cannot be readily applied because the optimization problem in (3.26) is extremely difficult to solve. We must therefore search for an alternative metric which also indicates the degree of similarity between two distributions, but which is much more mathematically tractable. Although several such metrics are available, we now turn our attention to a popular information-theoretic metric that is particularly well suited to our approximation problem, namely Kullback-Leibler distance.[2]

If $f_Y(\cdot)$ and $f_{\tilde{Y}}(\cdot)$ are univariate probability density functions, then the Kullback-Leibler distance between them, which we denote by $\mathcal{D}(f_Y, f_{\tilde{Y}})$, is defined by

$$\mathcal{D}(f_Y, f_{\tilde{Y}}) = \int_{\mathcal{Y}} f_Y(y) \log \frac{f_Y(y)}{f_{\tilde{Y}}(y)}\,dy, \quad (3.27)$$

where $\mathcal{Y}$ represents the region of support for $f_Y(\cdot)$. If, instead, $f_Y(\cdot)$ and $f_{\tilde{Y}}(\cdot)$ are univariate probability mass functions, then the appropriate definition of Kullback-Leibler distance is

$$\mathcal{D}(f_Y, f_{\tilde{Y}}) = \sum_{y\in\mathcal{Y}} f_Y(y) \log \frac{f_Y(y)}{f_{\tilde{Y}}(y)}, \quad (3.28)$$

[2] The Kullback-Leibler distance measure derives its name from the authors who originally introduced it into the information theory literature [105]. This metric is also discussed at length in [106], and has since been investigated extensively by a number of other researchers [13, 43, 53, 175, 176]. It is referred to by many different names in the literature, including cross entropy, relative entropy, directed divergence, information divergence, I-divergence, and discrimination information.

where $\mathcal{Y}$ now represents the set of possible outcomes under the pmf $f_Y(\cdot)$. When applying either of the above definitions in the sequel, we will use the conventions

$$a \log \frac{a}{b} = \begin{cases} 0 & \text{if } a = 0 \text{ and } b \geq 0; \\ \infty & \text{if } a > 0 \text{ and } b = 0, \end{cases} \tag{3.29}$$

which follow from limiting arguments using the continuity of the function $a \log(a/b)$ on the set $\{(a, b)|(a, b) \in (0, \infty) \times (0, \infty)\}$. Note that both (3.27) and (3.28) have obvious extensions to the case in which the functions $f_Y(\cdot)$ and $f_{\tilde{Y}}(\cdot)$ are multivariate.

The Kullback-Leibler distance serves as a convenient measure of our ability to discriminate between two classes of random observations. Suppose, for example, that $f_Y(y) = f_{\tilde{Y}}(y)$ for $y \in \mathbb{R}$ (except possibly on a set of measure zero). In this case, if we were given observations that were equally likely to be realizations of Y or of $\tilde{Y}$, we should not expect to be able to determine the true source of the observations any better than could a simple coin flip. This is a very special case in that it represents the smallest achievable degree of discriminability between two distributions and simultaneously yields the smallest possible value of $\mathcal{D}(\cdot)$, for we have from (3.27) that $\mathcal{D}(f_Y, f_{\tilde{Y}}) = 0$. Now let us consider the opposite extreme in which $f_Y(\cdot)$ and $f_{\tilde{Y}}(\cdot)$ have non-overlapping regions of support, i.e., the case in which $f_{\tilde{Y}}(y) = 0$ whenever $f_Y(y) > 0$, and vice versa. In this case, we could determine the true source of any observations given to us with absolute certainty, even if we were given only a single realization from either Y or $\tilde{Y}$. Accordingly, in this case we have that $\mathcal{D}(f_Y, f_{\tilde{Y}}) = \infty$. This demonstrates that using Kullback-Leibler distance as a measure of discriminability is at least reasonable in each of the two extreme cases.

Strictly speaking, the function $\mathcal{D}(\cdot)$ is not a true distance function by the standard mathematical definition [161]. In particular, although it is true that $\mathcal{D}(f_Y, f_{\tilde{Y}}) \geq 0$ with equality if and only if $f_Y(y) = f_{\tilde{Y}}(y)$ almost everywhere, it is not true in general that $\mathcal{D}(\cdot)$ satisfies either the symmetry property

$$\mathcal{D}(f_Y, f_{\tilde{Y}}) = \mathcal{D}(f_{\tilde{Y}}, f_Y) \tag{3.30}$$

or the triangle inequality

$$\mathcal{D}(f_Y, f_{\tilde{Y}}) \leq \mathcal{D}(f_Y, f'_Y) + \mathcal{D}(f'_Y, f_{\tilde{Y}}). \tag{3.31}$$

Nonetheless, it is useful to adopt the notion that the "distance" between the $f_Y(\cdot)$ and $f_{\tilde{Y}}(\cdot)$ increases as $\mathcal{D}(f_Y, f_{\tilde{Y}})$ increases, in the sense that the associated random variables Y and $\tilde{Y}$ become easier to distinguish. In Appendix D, we discuss in detail how the Kullback-Leibler distance relates to other, more familiar statistical measures of approximation quality.

3.3 Optimal HMM-Based Approximation of a First-Order AR Process

Having established a suitable metric for assessing approximation quality, we now seek to apply the above concepts in a very simple, illustrative case. Specifically, in this section we derive an optimal HMM-based approximation to a stationary signal $\{Y_t\}$ which is assumed to obey the first-order nonlinear difference equation

$$Y_t = h(Y_{t-1}, W_t), \tag{3.32}$$

where $h(\cdot)$ is a deterministic function and $\{W_t\}$ is a sequence of i.i.d. random variables described by the pdf $f_W(\cdot)$. We assume that complete descriptions of the functions $h(\cdot)$ and $f_W(\cdot)$ are given. In addition, since the pdf of the random process $\{Y_t\}$ can be determined exactly from this given information, we assume that it, too, is known.

3.3.1 Some Preliminary Observations

Before attempting a detailed problem formulation, let us first discuss certain basic aspects of the first-order signal approximation problem. Observe from (3.32) that the scalar-valued signal variable Y_t by itself constitutes a suitable state vector for the dynamical system at time t. Thus, since the state vector is only one-dimensional in this case, we can consider the state space to be the real line, and we can think of the disjoint "regions" in state space referred to earlier to be disjoint intervals whose union makes up the real line. In this one-dimensional example, therefore, a segmentation of the state space into L regions can be conveniently described by a collection of $L+1$ distinct points $d_0, d_1, \cdots, d_L$ on the real line, as depicted in Figure 3-2. We refer to these as breakpoints, and we assume that they satisfy the conditions

$$-\infty = d_0 < d_1 < \cdots < d_{L-1} < d_L = \infty. \tag{3.33}$$

We will sometimes use the vector notation $\mathbf{d}$ to refer to the ordered collection of breakpoints $(d_0, d_1, \cdots, d_L)$.

The above conditions imply that the mapping $\theta(\cdot)$, which enforces the state-space partitioning constraint, is now defined by

$$\theta(y) = \begin{cases} 1 & \text{if } -\infty < y < d_1; \\ 2 & \text{if } d_1 \leq y < d_2; \\ \vdots & \\ L & \text{if } d_{L-1} \leq y < \infty. \end{cases} \tag{3.34}$$

We will make extensive use of this mapping as we derive the optimal finite-state approximation to the first-order AR signal $\{Y_t\}$. Recall from our definition of an HMM-based representation that the densities $\{f_i(\cdot)\}_{i=1}^{L}$ describing the state vector were restricted in that their regions of support were not allowed to overlap. In the present case, since the state variable at time t and the signal variable at time t are identical, these same region-of-support constraints also apply to the densities $\{g_i(\cdot)\}_{i=1}^{L}$. In particular, the region of support for the function $g_i(\cdot)$ will be the interval $[d_{i-1}, d_i]$.

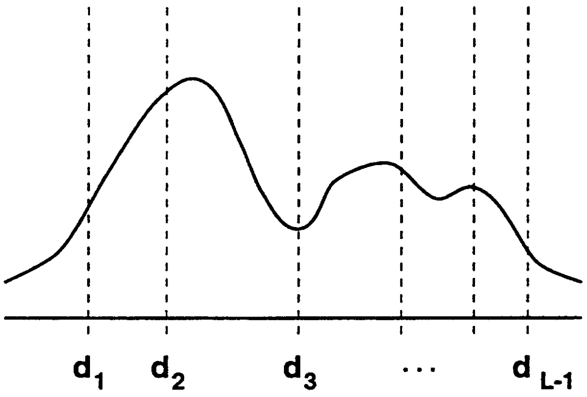


Figure 3-2: Partitioning of the state space in the first-order signal approximation problem via the breakpoints $d_0, d_1, \cdots, d_L$. The resulting segmentation of the state-variable pdf is also indicated.

3.3.2 Formulation of the Approximation Problem

The Kullback-Leibler distance between the densities for the true signal vector $\mathbf{Y}$ and the HMM-based approximate signal vector $\tilde{\mathbf{Y}}$ is given by[3]

$$\mathcal{D}(f_{\mathbf{Y}}, f_{\tilde{\mathbf{Y}}}) = \int f_{\mathbf{Y}}(\mathbf{y}) \log \frac{f_{\mathbf{Y}}(\mathbf{y})}{f_{\tilde{\mathbf{Y}}}(\mathbf{y})} \, d\mathbf{y}. \tag{3.35}$$

For our purposes, however, it will be more convenient to work with the alternative version of this expression given by

$$\mathcal{D}(f_{\mathbf{Y}}, f_{\tilde{\mathbf{Y}}}) = \int f_{\mathbf{Y}}(\mathbf{y}) \log f_{\mathbf{Y}}(\mathbf{y}) \, d\mathbf{y} - \int f_{\mathbf{Y}}(\mathbf{y}) \log f_{\tilde{\mathbf{Y}}}(\mathbf{y}) \, d\mathbf{y}. \tag{3.36}$$

Because the true pdf $f_{\mathbf{Y}}(\cdot)$ is fixed and known, the first term on the right hand side above is completely independent of the parameters we will choose for the approximating pdf $f_{\tilde{\mathbf{Y}}}(\cdot)$. Thus, minimizing the original objective function $\mathcal{D}(f_{\mathbf{Y}}, f_{\tilde{\mathbf{Y}}})$ is equivalent to maximizing the modified objective function $\mathcal{D}'(f_{\mathbf{Y}}, f_{\tilde{\mathbf{Y}}})$ defined by

$$\mathcal{D}'(f_{\mathbf{Y}}, f_{\tilde{\mathbf{Y}}}) = \int f_{\mathbf{Y}}(\mathbf{y}) \log f_{\tilde{\mathbf{Y}}}(\mathbf{y}) \, d\mathbf{y}. \tag{3.37}$$

The maximization of $\mathcal{D}'(f_{\mathbf{Y}}, f_{\tilde{\mathbf{Y}}})$ is to be carried out over a set of approximate densities having a very special structure. Specifically, each density in the set can be characterized by a tuple, which we denote by Ψ', consisting of all parameters needed to specify an L-state HMM-based representation of the true signal; these parameters include the breakpoint vector $\mathbf{d}$, the initial state probabilities $\{P(i)\}_{i=1}^{L}$, the state transition probabilities

[3] In this section and in many of the remaining sections in the chapter, we will use the symbol $\mathbf{Y}$ to refer to the signal vector being approximated, in place of the more cumbersome symbol $\mathbf{Y}_{0:N-1}$.

$\{Q(i,j)\}_{i,j=1}^{L}$, and the output densities $\{g_i(\cdot)\}_{i=1}^{L}$. Let us denote by $\mathcal{P}$ the collection of all such tuples that satisfy the constraints mentioned in the preceding subsection as well as those outlined in the introduction to the chapter.[4] Our problem is then to find the best such tuple Ψ^*, which is defined by

$$\Psi^* = \arg\max_{\Psi' \in \mathcal{P}} \int f_{\mathbf{Y}}(\mathbf{y}) \log f_{\tilde{\mathbf{Y}}}(\mathbf{y}; \Psi')\, d\mathbf{y}. \tag{3.38}$$

3.3.3 Derivation of the Approximate Signal Density

We now wish to derive an expression for the pdf of the approximate signal vector $\tilde{\mathbf{Y}} = (\tilde{Y}_0, \tilde{Y}_1, \cdots, \tilde{Y}_{N-1})$ in terms of the parameters that characterize its associated HMM. We proceed in the usual way by first accounting for all possible values of the underlying state sequence $\Theta = (\Theta_0, \Theta_1, \cdots, \Theta_{N-1})$ and then conditioning the HMM output on each of these contingencies. By following this approach, we arrive at the initial pdf expression

$$f_{\tilde{\mathbf{Y}}}(\mathbf{y}) = \sum_{\boldsymbol{\theta}} \Pr\{\Theta = \boldsymbol{\theta}\} f_{\tilde{\mathbf{Y}}|\Theta}(\mathbf{y} \mid \Theta = \boldsymbol{\theta}). \tag{3.39}$$

We can introduce a bit more detail into this expression by taking advantage of two special properties of the HMM structure, namely that (i) the state sequence Θ obeys the Markov property; and (ii) the elements of the output sequence $\tilde{\mathbf{Y}}$ are statistically independent when conditioned on a particular value of Θ. Using property (i), we can write

$$\Pr\{\Theta = \boldsymbol{\theta}\} = \Pr\{\Theta_0 = \theta_0\} \cdot \prod_{t=1}^{N-1} \Pr\{\Theta_t = \theta_t \mid \Theta_{t-1} = \theta_{t-1}\} \tag{3.40}$$

$$= P(\theta_0) \cdot \prod_{t=1}^{N-1} Q(\theta_{t-1}, \theta_t). \tag{3.41}$$

From property (ii), we have that

$$f_{\tilde{\mathbf{Y}}|\Theta}(\mathbf{y} \mid \Theta = \boldsymbol{\theta}) = \prod_{t=0}^{N-1} f_{\tilde{Y}_t|\Theta_t}(y_t \mid \Theta_t = \theta_t) \tag{3.42}$$

$$= \prod_{t=0}^{N-1} g_{\theta_t}(y_t). \tag{3.43}$$

[4] For the moment, we leave the structure of each pdf $g_i(\cdot)$ unconstrained. This will not pose any difficulty during the analysis presented in this chapter. In later chapters, however, we will find it convenient from a practical standpoint to restrict this pdf to be a Gaussian mixture with a fixed number of components.

By substituting (3.41) and (3.43) back into (3.39), we obtain the alternative expression for $f_{\tilde{\mathbf{Y}}}(\cdot)$ given by

$$f_{\tilde{\mathbf{Y}}}(\mathbf{y}) = \sum_{\theta_0=1}^{L} \sum_{\theta_1=1}^{L} \cdots \sum_{\theta_{N-1}=1}^{L} P(\theta_0) \cdot \prod_{t=1}^{N-1} Q(\theta_{t-1}, \theta_t) \prod_{t=0}^{N-1} g_{\theta_t}(y_t). \tag{3.44}$$

Based on this new expression, it appears, at least upon first inspection, that the pdf of $\tilde{\mathbf{Y}}$ is a very complex function; specifically, the pdf is represented above as a sum of L^N terms, where each term accounts for a possible realization of the underlying state sequence. In fact, almost all of the terms appearing in the summation in (3.44) are equal to zero; the sole exception is the term corresponding to the particular state sequence

$$\boldsymbol{\theta} = (\theta(y_0), \theta(y_1), \cdots, \theta(y_{N-1})). \tag{3.45}$$

Using this fact, we can now express the pdf of $\tilde{\mathbf{Y}}$ in the much simpler form

$$f_{\tilde{\mathbf{Y}}}(\mathbf{y}) = P(\theta(y_0)) \cdot \prod_{t=1}^{N-1} Q(\theta(y_{t-1}), \theta(y_t)) \cdot \prod_{t=0}^{N-1} g_{\theta(y_t)}(y_t). \tag{3.46}$$

3.3.4 Decomposition of the Objective Function

Now that we have derived an expression for the pdf of the approximate signal vector, let us once again turn our attention toward the maximization of our objective function $\mathcal{D}'(f_{\mathbf{Y}}, f_{\tilde{\mathbf{Y}}})$. It is apparent from (3.37) that we first need an expression for the natural logarithm of the pdf $f_{\tilde{\mathbf{Y}}}(\cdot)$. Using (3.46), we easily have that

$$\log f_{\tilde{\mathbf{Y}}}(\mathbf{y}) = \log P(\theta(y_0)) + \sum_{t=1}^{N-1} \log Q(\theta(y_{t-1}), \theta(y_t)) + \sum_{t=0}^{N-1} \log g_{\theta(y_t)}(y_t). \tag{3.47}$$

Upon substituting this expression into (3.37), we obtain a more explicit form of the objective function given by

$$\begin{aligned} \mathcal{D}'(f_{\mathbf{Y}}, f_{\tilde{\mathbf{Y}}}) &= \int f_{\mathbf{Y}}(\mathbf{y}) \log P(\theta(y_0))\, d\mathbf{y} \\ &\quad + \int f_{\mathbf{Y}}(\mathbf{y}) \left[\sum_{t=1}^{N-1} \log Q(\theta(y_{t-1}), \theta(y_t)) \right] d\mathbf{y} \\ &\quad + \int f_{\mathbf{Y}}(\mathbf{y}) \left[\sum_{t=1}^{N-1} \log g_{\theta(y_t)}(y_t) \right] d\mathbf{y} && (3.48) \\ &\triangleq \mathcal{D}_1 + \mathcal{D}_2 + \mathcal{D}_3, && (3.49) \end{aligned}$$

where the terms $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ are defined in the obvious way. Observe that these three terms involve different components of the HMM, and can therefore be maximized separately.[5] This is precisely the strategy we shall pursue in the next three subsections.

3.3.4.1 Maximization of $\mathcal{D}_1$

We begin by solving for the values of the initial state probabilities $\{P(i)\}_{i=1}^{L}$ that maximize the term $\mathcal{D}_1$. Note first that $\mathcal{D}_1$ can be written as

$$\mathcal{D}_1 = \int f_{\mathbf{Y}}(\mathbf{y}) \log P(\theta(y_0))\, d\mathbf{y} \tag{3.50}$$

$$= \int_{-\infty}^{\infty} f_{Y_0}(y_0) \log P(\theta(y_0))\, dy_0, \tag{3.51}$$

where in the latter step we have eliminated the superfluous variables $y_1, y_2, \cdots, y_{N-1}$ by integrating them out. To simplify the expression for $\mathcal{D}_1$ even further, it will be convenient to represent the integral in (3.51) as a sum of integrals taken over disjoint portions of the real line. In particular, we use collection of breakpoints $\{d_0, d_1, \cdots, d_L\}$ to segment the real line according to

$$\mathbb{R} = [d_0, d_1] \cup [d_1, d_2] \cup \cdots \cup [d_{L-1}, d_L], \tag{3.52}$$

and then write $\mathcal{D}_1$ as

$$\mathcal{D}_1 = \sum_{j=1}^{L} \int_{d_{j-1}}^{d_j} f_{Y_0}(y_0) \log P(\theta(y_0))\, dy_0. \tag{3.53}$$

Next, recall that the function $\theta(\cdot)$ is, by definition, constant over any interval of the form $[d_{j-1}, d_j]$, and can therefore be factored out of each of the above integrals. This leads to the expression

$$\mathcal{D}_1 = \sum_{j=1}^{L} \left(\int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\, dy_0 \right) \log P(j). \tag{3.54}$$

Under the assumption that the breakpoints are fixed and that all constraints on the initial state probabilities are satisfied, the above sum will be largest if we use the assignment

$$P(j) = \int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\, dy_0, \qquad j = 1, 2, \cdots, L. \tag{3.55}$$

A proof of this claim can be found in Appendix B.

[5] Actually, this is not entirely true, since the initial state probabilities $\{P(i)\}_{i=1}^{L}$ and the state transition probabilities $\{Q(i,j)\}_{i,j=1}^{L}$ are coupled through the stationarity constraint given in (3.11). However, as we shall soon discover, a fruitful strategy in this case is to proceed with maximizing the terms separately and then to verify that the optimal solutions do indeed satisfy the constraint.

3.3.4.2 Maximization of $\mathcal{D}_2$

We now turn to the problem of maximizing $\mathcal{D}_2$ through an appropriate choice of the state transition probabilities $\{Q(i,j)\}_{i,j=1}^{L}$. First, observe that we can write $\mathcal{D}_2$ as

$$\mathcal{D}_2 = \int f_{\mathbf{Y}}(\mathbf{y}) \sum_{t=1}^{N-1} \log Q(\theta(y_{t-1}), \theta(y_t))\, d\mathbf{y} \tag{3.56}$$

$$= \sum_{t=1}^{N-1} \int f_{\mathbf{Y}}(\mathbf{y}) \log Q(\theta(y_{t-1}), \theta(y_t))\, d\mathbf{y} \tag{3.57}$$

$$= \sum_{t=1}^{N-1} \int f_{Y_{t-1},Y_t}(y_{t-1}, y_t) \log Q(\theta(y_{t-1}), \theta(y_t))\, dy_{t-1}\, dy_t, \tag{3.58}$$

where in the second step we have interchanged the order of summation and integration, and in the last step we have once again eliminated the superfluous variables within each integral, i.e., all variables having indices other than $t-1$ or t. At this point, however, we can simplify (3.58) even further by using the fact that the process $\{Y_t\}$ is stationary, and therefore that the condition

$$f_{Y_{t-1},Y_t}(y_0, y_1) = f_{Y_0,Y_1}(y_0, y_1) \tag{3.59}$$

is satisfied for all $y_0, y_1 \in \mathbb{R}$ and for $t = 1, 2, \cdots, N-1$. This implies that (3.58) can be written as

$$\mathcal{D}_2 = \sum_{t=1}^{N-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{Y_0,Y_1}(y_0, y_1) \log Q(\theta(y_0), \theta(y_1))\, dy_0\, dy_1 \tag{3.60}$$

$$= (N-1) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{Y_0,Y_1}(y_0, y_1) \log Q(\theta(y_0), \theta(y_1))\, dy_0\, dy_1. \tag{3.61}$$

As a final step in reducing this expression to simplest terms, we once again invoke the two-part strategy in which we first decompose each integral into a sum of L integrals over segments of the form $[d_{j-1}, d_j]$, and then use the fact that the function $\theta(\cdot)$ is constant over each such segment so that we can factor it out of the integral. Applying this strategy yields the new formulas

$$\mathcal{D}_2 = (N-1) \sum_{i=1}^{L} \sum_{j=1}^{L} \int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{Y_0,Y_1}(y_0, y_1) \log Q(\theta(y_0), \theta(y_1))\, dy_0\, dy_1 \tag{3.62}$$

$$= (N-1) \sum_{i=1}^{L} \sum_{j=1}^{L} \left(\int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{Y_0,Y_1}(y_0, y_1)\, dy_0\, dy_1 \right) \log Q(i,j). \tag{3.63}$$

The decomposition of the joint bivariate pdf $f_{Y_0,Y_1}(\cdot)$ implied by (3.63) is depicted in Figure 3-3.

Now observe that, for a fixed value of i, the elements $Q(i,1), Q(i,2), \cdots, Q(i,L)$ must

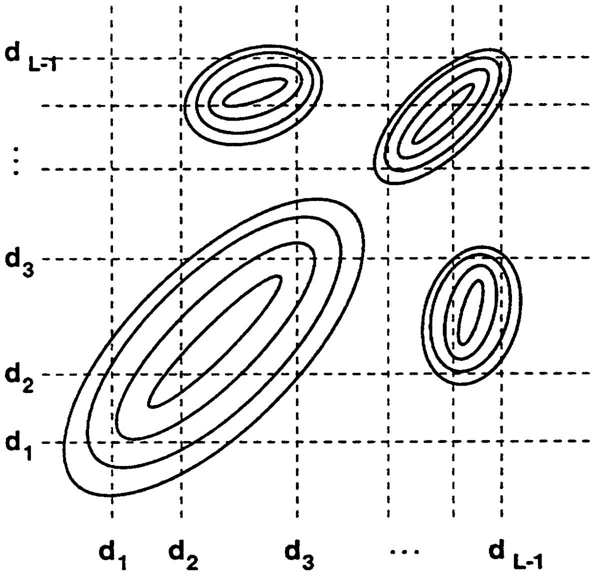


Figure 3-3: Contour plot of joint bivariate pdf of two successive state variables and its corresponding decomposition into rectangular regions by the breakpoints $d_0, d_1, \cdots, d_L$.

form a pmf. Moreover, there are L such pmfs that make up the entire collection of state transition probabilities $\{Q(i,j)\}_{i,j=1}^{L}$, and each of these L pmfs is entirely independent of the others; hence, we can solve for each pmf separately. It can be shown (as before, using the arguments given in Appendix B) that the elements $Q(i,j)$ that maximize $\mathcal{D}_2$, subject to the usual normalization constraints, are given by

$$Q(i,j) = \frac{\int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{Y_0,Y_1}(y_0,y_1)\, dy_0\, dy_1}{\int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\, dy_0}, \qquad i,j = 1,2,\cdots,L. \tag{3.64}$$

3.3.4.3 Maximization of $\mathcal{D}_3$

Finally, we consider maximizing the term $\mathcal{D}_3$. As before, we can apply the usual manipulations to integrals and summations within $\mathcal{D}_3$, taking advantage of the stationarity of the process $\{Y_t\}$ as well as the special structure of the indexing function $\theta(\cdot)$ and the breakpoint vector $\mathbf{d}$, to obtain

$$\mathcal{D}_3 = \int f_{\mathbf{Y}}(\mathbf{y}) \sum_{t=1}^{N-1} \log g_{\theta(y_t)}(y_t)\, d\mathbf{y} \tag{3.65}$$

$$= \sum_{t=1}^{N-1} \int f_{\mathbf{Y}}(\mathbf{y}) \log g_{\theta(y_t)}(y_t)\, d\mathbf{y} \tag{3.66}$$

$$= \sum_{t=1}^{N-1} \int_{-\infty}^{\infty} f_{Y_t}(y_t) \log g_{\theta(y_t)}(y_t)\, dy_t \tag{3.67}$$

$$= N \int_{-\infty}^{\infty} f_{Y_0}(y_0) \log g_{\theta(y_0)}(y_0)\, dy_0 \tag{3.68}$$

$$= N \sum_{j=1}^{L} \int_{d_{j-1}}^{d_j} f_{Y_0}(y_0) \log g_j(y_0)\, dy_0 \tag{3.69}$$

Under the assumption that $\mathbf{d}$ is fixed, we can now maximize each of the L terms in (3.69) separately, since they are entirely uncoupled. Once again, from arguments presented in Appendix B, it follows that the optimal output densities $\{g_j(\cdot)\}_{j=1}^{L}$ of the HMM are given by

$$g_j(y) = \begin{cases} \dfrac{f_{Y_0}(y)}{\int_{d_{j-1}}^{d_j} f_{Y_0}(u)du} & \text{if } d_{j-1} < y \leq d_j; \\ 0 & \text{otherwise;} \end{cases} \qquad j = 1, 2, \cdots, L. \tag{3.70}$$

3.3.4.4 Verification of the Stationarity Constraint

Thus far, we have adopted the strategy of maximizing each of the terms $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ without regard to the constraint that the derived Markov chain $\{\Theta_t\}$ must be stationary. This requirement has no effect on the output densities associated with the states, but it does place a simultaneous restriction on the initial state probabilities and the state transition probabilities. We now verify that the solutions obtained to the unconstrained maximization problems above actually meet this constraint. Recall that we must have

$$\sum_{i=1}^{L} P(i)Q(i,j) = P(j), \qquad j = 1, 2, \cdots, L. \tag{3.71}$$

If we now insert into this expression the optimal values we have already obtained for $P(i)$ and $Q(i,j)$, we find that

$$\sum_{i=1}^{L} P(i)Q(i,j) = \sum_{i=1}^{L} \left(\int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\, dy_0 \right) \left(\frac{\int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\, d\mathbf{y}_{0:1}}{\int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\, dy_0} \right) \tag{3.72}$$

$$= \sum_{i=1}^{L} \int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\, d\mathbf{y}_{0:1} \tag{3.73}$$

$$= \int_{d_{j-1}}^{d_j} \sum_{i=1}^{L} \int_{d_{i-1}}^{d_i} f_{Y_0,Y_1}(y_0, y_1)\, dy_1\, dy_0 \tag{3.74}$$

$$= \int_{d_{j-1}}^{d_j} \int_{-\infty}^{\infty} f_{Y_0,Y_1}(y_0, y_1)\, dy_1\, dy_0 \tag{3.75}$$

$$= \int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\, dy_0 \tag{3.76}$$

$$= P(j), \tag{3.77}$$

which is what we wished to show.

3.3.5 Identification of Optimal Breakpoints

When solving each of the maximization problems above, we assumed the values of the breakpoints $d_0, d_1, \cdots, d_L$ were fixed. Consequently, for any given collection of valid breakpoint values, we can now obtain optimal solutions for the HMM parameters directly from the formulas in (3.55), (3.64), and (3.70). Clearly, however, any change in the values of the breakpoints will bring about a corresponding change in the values of these conditionally optimal solutions. The only remaining part of the overall maximization problem, and the part which we presently consider, is the optimal selection of the breakpoints. To find the optimal breakpoints, let us first reconstruct the original objective function $\mathcal{D}' = \mathcal{D}_1 + \mathcal{D}_2 + \mathcal{D}_3$ using all of the optimal solutions just obtained for a fixed value of the breakpoint vector $\mathbf{d}$.

Note that if we take the optimal values of the initial state probabilities from (3.55) and substitute them back into the derived expression for $\mathcal{D}_1$ (given in (3.54)), we find that the largest possible value of $\mathcal{D}_1$, conditioned on a particular value of $\mathbf{d}$, is given by

$$\mathcal{D}_1^* = \sum_{j=1}^{L} \left(\int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\, dy_0 \right) \log \left(\int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\, dy_0 \right). \tag{3.78}$$

Next, if we take the optimal values of the state transition probabilities from (3.64) and substitute them back into the derived expression for $\mathcal{D}_2$ (given in (3.63)), we conclude that the maximum conditional value of $\mathcal{D}_2$ is given by

$$\begin{aligned}
\mathcal{D}_2^* &= (N-1) \sum_{i=1}^{L} \sum_{j=1}^{L} \left(\int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\, d\mathbf{y}_{0:1} \right) \log \left(\frac{\int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\, d\mathbf{y}_{0:1}}{\int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\, dy_0} \right) \\
&= (N-1) \sum_{i=1}^{L} \sum_{j=1}^{L} \left(\int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\, d\mathbf{y}_{0:1} \right) \log \left(\int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\, d\mathbf{y}_{0:1} \right) \\
&\quad - (N-1) \sum_{i=1}^{L} \sum_{j=1}^{L} \left(\int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\, d\mathbf{y}_{0:1} \right) \log \left(\int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\, dy_0 \right) \\
&= (N-1) \sum_{i=1}^{L} \sum_{j=1}^{L} \left(\int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\, d\mathbf{y}_{0:1} \right) \log \left(\int_{d_{i-1}}^{d_i} \int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\, d\mathbf{y}_{0:1} \right) \\
&\quad - (N-1) \sum_{i=1}^{L} \left(\int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\, dy_0 \right) \log \left(\int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\, dy_0 \right),
\end{aligned} \tag{3.79}$$

where in the last step we have simplified the double integral in the second term by first interchanging the order of the inner summation and the outer integration, and then by integrating out the superfluous variable y_1.

Finally, if we take the optimal values of the HMM output densities from (3.70) and substitute them back into the derived expression for $\mathcal{D}_3$ (given in (3.69)), we find that the maximum value of $\mathcal{D}_3$, conditioned on the value of $\mathbf{d}$, is given by

$$\begin{aligned}
\mathcal{D}_3 &= N\sum_{j=1}^{L}\int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\log\left(\frac{f_{Y_0}(y_0)}{\int_{d_{j-1}}^{d_j} f_{Y_0}(u)\,du}\right)dy_0 && (3.80)\\
&= N\sum_{j=1}^{L}\int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\log f_{Y_0}(y_0)\,dy_0\\
&\quad - N\sum_{j=1}^{L}\int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\log\left(\int_{d_{j-1}}^{d_j} f_{Y_0}(u)\,du\right)dy_0 && (3.81)\\
&= N\sum_{j=1}^{L}\int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\log f_{Y_0}(y_0)\,dy_0\\
&\quad - N\sum_{j=1}^{L}\left(\int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\,dy_0\right)\log\left(\int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\,dy_0\right). && (3.82)
\end{aligned}$$

Let us now reconstruct the maximum conditional value of the overall objective function by calculating the sum of the three components above. This yields the expression

$$\begin{aligned}
\mathcal{D}' &= \mathcal{D}_1 + \mathcal{D}_2 + \mathcal{D}_3 && (3.83)\\
&= (N-1)\sum_{i=1}^{L}\sum_{j=1}^{L}\left(\int_{d_{i-1}}^{d_i}\int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\,d\mathbf{y}_{0:1}\right)\cdot\\
&\qquad\qquad \log\left(\int_{d_{i-1}}^{d_i}\int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\,d\mathbf{y}_{0:1}\right)\\
&\quad - 2(N-1)\sum_{i=1}^{L}\left(\int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\,dy_0\right)\log\left(\int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\,dy_0\right)\\
&\quad - N\sum_{i=1}^{L}\int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\log f_{Y_0}(y_0)\,dy_0. && (3.84)
\end{aligned}$$

Observe, however, that the last term on the right-hand side above can be written as

$$N\sum_{i=1}^{L}\int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\log f_{Y_0}(y_0)\,dy_0 = N\int_{-\infty}^{\infty} f_{Y_0}(y_0)\log f_{Y_0}(y_0)\,dy_0, \qquad (3.85)$$

and is therefore invariant with respect to the breakpoint vector $\mathbf{d}$. This allows us to drop the final term from the objective function $\mathcal{D}'$, and subsequently remove the factor of $N-1$ that multiplies the two remaining terms. We can then restate our present goal as

$$\mathbf{d}^* = \underset{-\infty=d_0<d_1<\cdots<d_L=\infty}{\arg\max} \mathcal{D}''(\mathbf{d}), \tag{3.86}$$

where $\mathcal{D}''$ is the new objective function given by

$$\begin{aligned}\mathcal{D}''(\mathbf{d}) = (N-1)\sum_{i=1}^{L}\sum_{j=1}^{L}\left(\int_{d_{i-1}}^{d_i}\int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\,d\mathbf{y}_{0:1}\right)\cdot \\ \log\left(\int_{d_{i-1}}^{d_i}\int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\,d\mathbf{y}_{0:1}\right) \\ -\sum_{j=1}^{L}\left(\int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\,dy_0\right)\log\left(\int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\,dy_0\right) \\ -\sum_{i=1}^{L}\left(\int_{d_{i-1}}^{d_i} f_{Y_1}(y_1)\,dy_1\right)\log\left(\int_{d_{i-1}}^{d_i} f_{Y_1}(y_1)\,dy_1\right).\end{aligned} \tag{3.87}$$

Although the above expression for $\mathcal{D}''(\mathbf{d})$ may at first seem cumbersome (particularly in view of the fact that the last two terms are equal and could therefore be consolidated), we have written it in such a way that it can now be easily identified as a familiar information-theoretic measure. In particular, $\mathcal{D}''(\mathbf{d})$ represents the mutual information between any pair (Θ_t, Θ_{t+1}) of successive state variables in the underlying Markov chain whose parameter values are conditionally optimal given $\mathbf{d}$. Hence, maximizing $\mathcal{D}''(\cdot)$ is equivalent to maximizing $I(\Theta_t, \Theta_{t+1})$, the mutual information between the discrete random variables Θ_t and Θ_{t+1}.

To see this more directly, let us assume, without loss of generality, that the variables in question are Θ_0 and Θ_1. Then for a given value of $\mathbf{d}$, the marginal pmfs for these random variables, which we denote by P_0 and P_1, respectively, are given by[6]

$$P_0(j;\mathbf{d}) = \int_{d_{j-1}}^{d_j} f_{Y_0}(y_0)\,dy_0 \qquad j = 1,2,\cdots,L \tag{3.88}$$

$$P_1(i;\mathbf{d}) = \int_{d_{i-1}}^{d_i} f_{Y_1}(y_1)\,dy_1 \qquad i = 1,2,\cdots,L \tag{3.89}$$

and their joint pmf is given by

$$R(i,j;\mathbf{d}) = \int_{d_{i-1}}^{d_i}\int_{d_{j-1}}^{d_j} f_{\mathbf{Y}_{0:1}}(\mathbf{y}_{0:1})\,d\mathbf{y}_{0:1} \qquad i,j = 1,2,\cdots,L. \tag{3.90}$$

[6] Of course, from our earlier derivation, we know that P_0 and P_1 must be identical (owing to the stationarity constraint), but we nonetheless use distinct symbols here to emphasize the mutual information concept.

By substituting these expressions back into (3.87), we can write the objective function much more plainly as

$$\begin{aligned}\mathcal{D}''(\mathbf{d}) &= \sum_{i=1}^{L}\sum_{j=1}^{L} R(i,j;\mathbf{d}) \log R(i,j;\mathbf{d}) \\ &\quad + \sum_{j=1}^{L} P_0(j;\mathbf{d}) \log P_0(j;\mathbf{d}) + \sum_{i=1}^{L} P_1(i;\mathbf{d}) \log P_1(i;\mathbf{d}) \qquad (3.91) \\ &= I(\Theta_0,\Theta_1;\mathbf{d}). \qquad (3.92)\end{aligned}$$

This representation makes it clear that we should choose the value of the breakpoint vector $\mathbf{d}$ that yields the largest possible mutual information between successive state variables of our HMM. While there exists no general closed-form solution for such a value, we now have a very useful rule for finding an optimal set of breakpoints.

3.4 Generation of Numerical Approximations

We now demonstrate that an optimal HMM-based representation of a particular AR signal can be constructed using a special gradient-descent technique derived in Appendix E. At the core of our example lies the true signal to be approximated, which, for the purpose of analytical tractability, we have chosen to be a first-order AR Gaussian process. We compare realizations generated by several different approximations of this process and present tables of the parameters characterizing each HMM. In addition, at the end of the section, we examine the probability of error of an optimal detector, which is designed to determine whether it has been given a realization of the true process or an approximate process.

3.4.1 Statistical Characterization of the True Random Process $\{Y_t\}$

Throughout this section, we assume that the true source signal $\{Y_t\}$ obeys the first-order linear difference equation given by

$$Y_t = aY_{t-1} + W_t, \qquad (3.93)$$

where a is a real number satisfying $-1 < a < 1$ and $\{W_t\}$ is a sequence consisting of i.i.d. Gaussian random variables whose pdf $f_W(\cdot)$ is given by

$$f_W(w) = \mathcal{N}(w, 0, 1). \qquad (3.94)$$

Observe that we can equivalently describe the process $\{Y_t\}$ as the output of a linear time-invariant system whose impulse response $\{h_t\}$ is the discrete-time sequence defined by

$$h_t = \begin{cases} 0, & t < 0, \\ a^t, & t \geq 0, \end{cases} \qquad (3.95)$$

and whose input is the white Gaussian noise sequence $\{W_t\}$. The constraint imposed on the autoregressive parameter a insures that the system described by the above impulse response is stable, and hence that the output of the system, $\{Y_t\}$, is stationary,

From our discussion in earlier sections, we know that, in order to solve for the best parameter values of any HMM-based representation of $\{Y_t\}$, we will require the marginal pdf for the random variable Y_t as well as the joint pdf of the pair of random variables (Y_t, Y_{t+1}). Since $\{Y_t\}$ is zero-mean, both of these densities are completely characterized by their second-order moments. Let us first calculate the variance of the single random variable Y_t. Using the fact that the elements of $\{W_t\}$ are statistically independent and have unit variance, we can write

$$\begin{aligned}\mathrm{Var}\{Y_t\} &= \mathrm{Var}\left\{\sum_{k=0}^{\infty} a^k W_{t-k}\right\} &(3.96)\\ &= \sum_{k=0}^{\infty}(a^k)^2 \mathrm{Var}\{W_{t-k}\} &(3.97)\\ &= \frac{1}{1-a^2}. &(3.98)\end{aligned}$$

Therefore, the pdf for the random variable Y_t can be expressed as

$$f_{Y_t}(y) = \frac{\sqrt{1-a^2}}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(1-a^2)y^2\right\}. \tag{3.99}$$

Next, let us solve for the covariance of the pair of random variables (Y_t, Y_{t+1}). Using the autoregressive equation that relates these two variables, we have

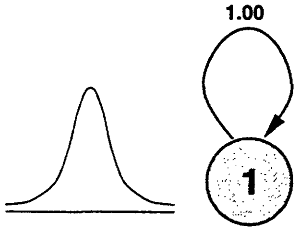
$$\begin{aligned}\mathrm{Cov}\{Y_t, Y_{t+1}\} &= E\{Y_t(aY_t + W_t)\} &(3.100)\\ &= a\mathrm{Var}\{Y_t\} &(3.101)\\ &= \frac{a}{1-a^2}. &(3.102)\end{aligned}$$

From the results in (3.98) and (3.102), we conclude that the covariance matrix $\mathbf{C}$ of the random vector $\mathbf{Y}_{t:t+1}$ must have the form

$$\mathbf{C} = \begin{bmatrix} \frac{1}{1-a^2} & \frac{a}{1-a^2} \\ \frac{a}{1-a^2} & \frac{1}{1-a^2} \end{bmatrix}. \tag{3.103}$$

It is straightforward to show that the determinant of this matrix is given by

$$|\mathbf{C}| = \frac{1}{1-a^2} \tag{3.104}$$



i	d_i	$P(i)$	$Q(i,\cdot)$
0	$-\infty$	—	—
1	∞	1.00	1.00

Table 3.1: Diagrammatic representation and parameter definitions for 1-state HMM-based approximation of the AR Gaussian process $Y_t = 0.8Y_{t-1} + W_t$.

and that its inverse is

$$\mathbf{C}^{-1} = \begin{bmatrix} 1 & -a \\ -a & 1 \end{bmatrix}. \tag{3.105}$$

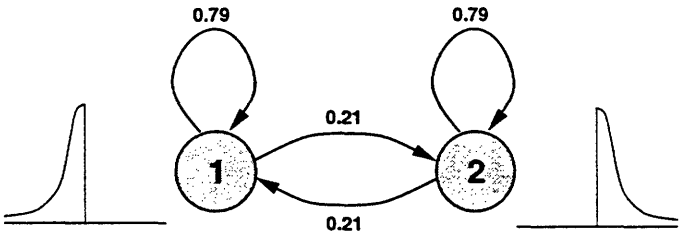
Using these expressions, we can express the pdf of $\mathbf{Y}_{t:t+1}$ as

$$f_{\mathbf{Y}_{t:t+1}}(\mathbf{y}) = \frac{\sqrt{1-a^2}}{2\pi} \exp\left\{ -\tfrac{1}{2}\mathbf{y}^T \begin{bmatrix} 1 & -a \\ -a & 1 \end{bmatrix} \mathbf{y} \right\}. \tag{3.106}$$

3.4.2 Descriptions of Various HMM-Based Approximations of $\{Y_t\}$

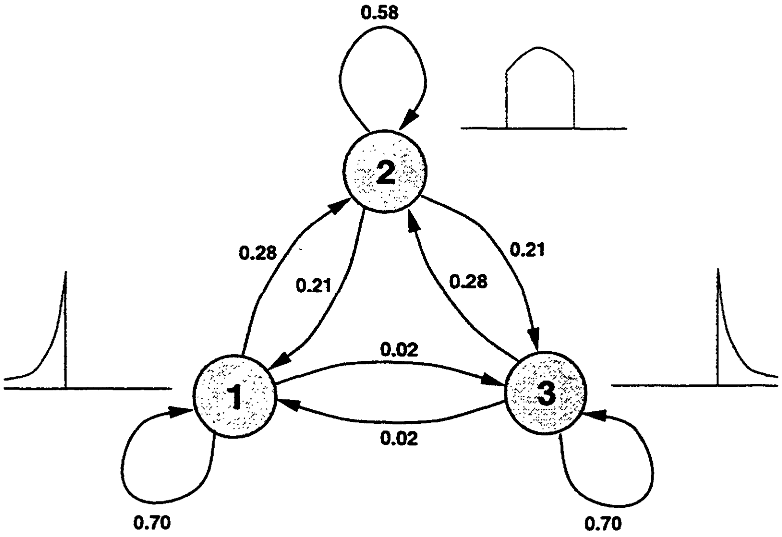
To conduct our finite-state modeling experiment, we arbitrarily selected the parameter value $a = 0.8$. Then, using the expressions for the signal densities given in (3.99) and (3.106), together with the numerical optimization technique derived in Appendix E, we created several distinct HMM-based approximations for the true signal. The single factor distinguishing these approximations was the number of states making up the underlying Markov chain within each HMM.

Our collection of approximations consisted of a 1-state HMM, a 2-state HMM, a 3-state HMM, a 5-state HMM, and a 7-state HMM. Complete parametric descriptions of these finite-state approximations are given in Tables 3.1, 3.2, 3.3, 3.4, and 3.5, respectively. Note that each of the first three tables have been augmented with a corresponding diagram of the components of the HMM so that its dynamics and its output can be more easily visualized. Such diagrams become rather unwieldy for higher-order models, however, so they have been omitted from the remaining two tables. We remark in addition that the probabilities listed in each table have been rounded to two decimal places for succinctness. Thus, although certain probabilities appear to be equal to zero, all probabilities are, in fact, strictly positive.



i	d_i	$P(i)$	$Q(i,\cdot)$	
0	$-\infty$	—	—	—
1	0.00	0.50	0.79	0.21
2	∞	0.50	0.21	0.79

Table 3.2: Diagrammatic representation and parameter definitions for 2-state HMM-based approximation of the AR Gaussian process $Y_t = 0.8Y_{t-1} + W_t$.



i	d_i	$P(i)$	$Q(i,\cdot)$		
0	$-\infty$	—	—	—	—
1	-0.86	0.30	0.70	0.28	0.02
2	0.86	0.40	0.21	0.58	0.21
3	∞	0.30	0.02	0.28	0.70

Table 3.3: Diagrammatic representation and parameter definitions for 3-state HMM-based approximation of the AR Gaussian process $Y_t = 0.8Y_{t-1} + W_t$.

i	d_i	$P(i)$	$Q(i,\cdot)$				
0	$-\infty$	—	—	—	—	—	—
1	-1.89	0.13	0.58	0.34	0.07	0.01	0.00
2	-0.59	0.23	0.18	0.45	0.30	0.07	0.00
3	0.59	0.28	0.03	0.25	0.44	0.25	0.03
4	1.89	0.23	0.00	0.07	0.30	0.45	0.18
5	∞	0.13	0.00	0.01	0.07	0.34	0.58

Table 3.4: Parameter definitions for 5-state HMM-based representation of the AR Gaussian process $Y_t = 0.8Y_{t-1} + W_t$.

i	d_i	$P(i)$	$Q(i,\cdot)$						
0	$-\infty$	—	—	—	—	—	—	—	—
1	-2.51	0.06	0.52	0.34	0.12	0.02	0.00	0.00	0.00
2	-1.40	0.13	0.16	0.38	0.31	0.12	0.03	0.00	0.00
3	-0.45	0.19	0.04	0.21	0.35	0.27	0.11	0.02	0.00
4	0.45	0.21	0.01	0.08	0.24	0.34	0.24	0.08	0.01
5	1.40	0.19	0.00	0.02	0.11	0.27	0.35	0.21	0.04
6	2.51	0.13	0.00	0.00	0.03	0.12	0.31	0.38	0.16
7	∞	0.06	0.00	0.00	0.00	0.02	0.12	0.34	0.52

Table 3.5: Parameter definitions for 7-state HMM-based representation of the AR Gaussian process $Y_t = 0.8Y_{t-1} + W_t$.

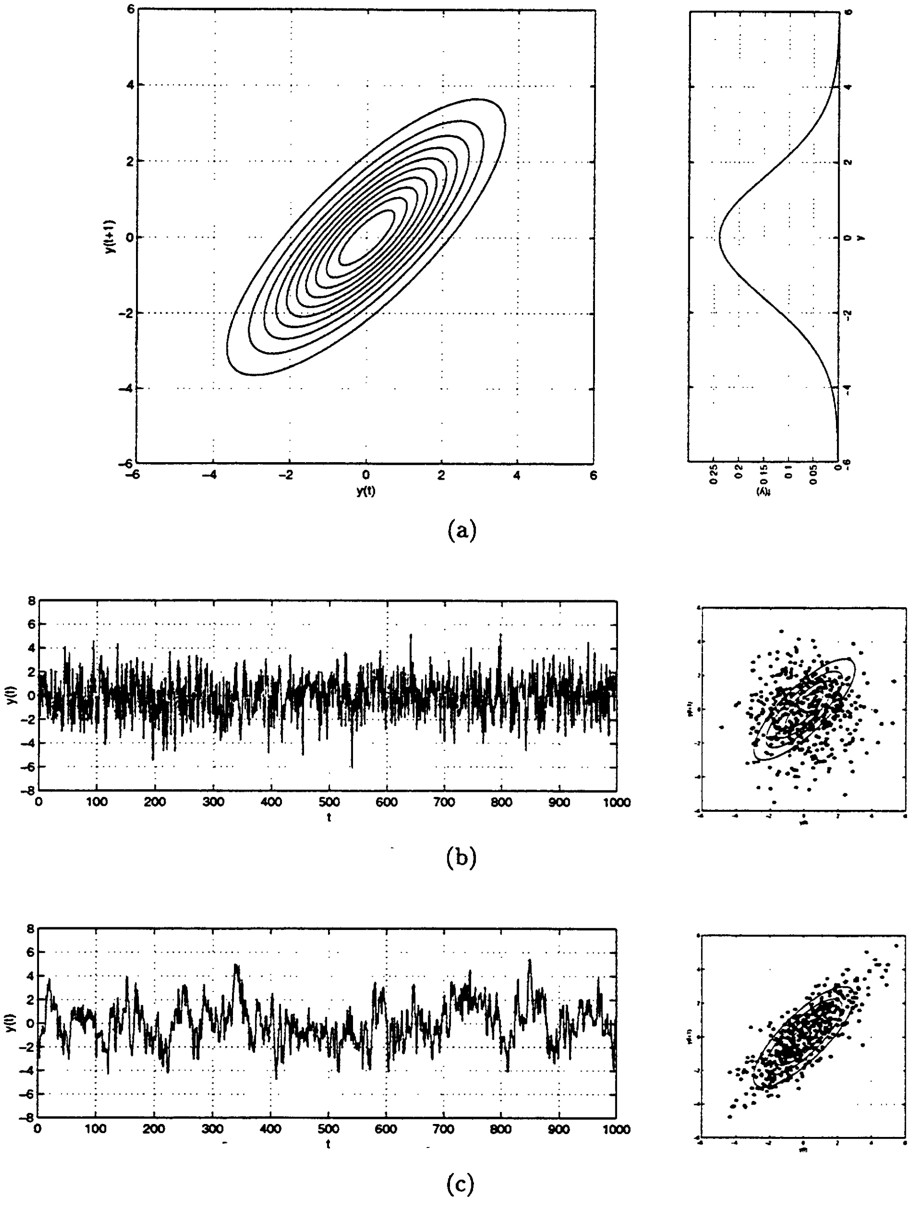
Some additional notable aspects of our HMM-based approximations are highlighted in Figures 3-4 through 3-8. Each of these figures characterizes a particular finite-state model from the set of five models that were constructed; the information presented in each figure allows us to better understand the underlying dynamics of the associated HMM and to compare the attributes of the output of the HMM to that of the true Gaussian process.

In the top portion of each figure, we show a contour plot of the critical bivariate Gaussian pdf from (3.106) as well as a plot of its univariate projection (i.e., the associated marginal pdf), whose functional form is given in (3.99). Superimposed on each of these plots is a collection of lines corresponding to the optimal breakpoints determined for the model; these lines help us to see how both the original one-dimensional state space and the two-dimensional coordinate plane were segmented into appropriate regions of integration. In the bottom portion of each figure, we show realizations from the HMM and from the true Gaussian process, as well as corresponding scatter plots that were constructed from successive sample values occurring within each realization. These two sets of plots allow us to quickly assess, by means of a direct visual comparison, the quality of a given HMM-based representation of the original signal.

From the plots shown in Figure 3-4, we can examine the key characteristics of our 1-state approximation. Recall that any 1-state HMM is inherently a memoryless process, i.e., the random variables that make up the process exhibit no temporal dependence whatsoever. This fundamental property of the 1-state model is clearly evident in both the realization and the scatter plot shown in Figure 3-4(b). Specifically, note from the scatter plot in this example that, although the marginal distribution of the samples from each realization appear closely matched, the dispersion pattern of the points about the origin is approximately the same in all directions — an indication that successive samples of the process are independent.

In Figure 3-5, we can see the temporal correlation structure of the approximate process begin to take shape, owing to the the dynamics of its underlying 2-state Markov chain. Nonetheless, the output of the 2-state HMM still appears to be a rather coarse representation of the true process, since it is capable only of switching back and forth between an approximate positive random value and an approximate negative random value.

As we can see from Figures 3-6, 3-7, and 3-8, however, the finite-state approximation gets progressively better as more states are incorporated into the model. In particular, from the plots shown in Figure 3-8, we see that the 7-state HMM is capable of producing a realization that is nearly indistinguishable — at least in its observable statistical attributes — from a realization of the true random process.

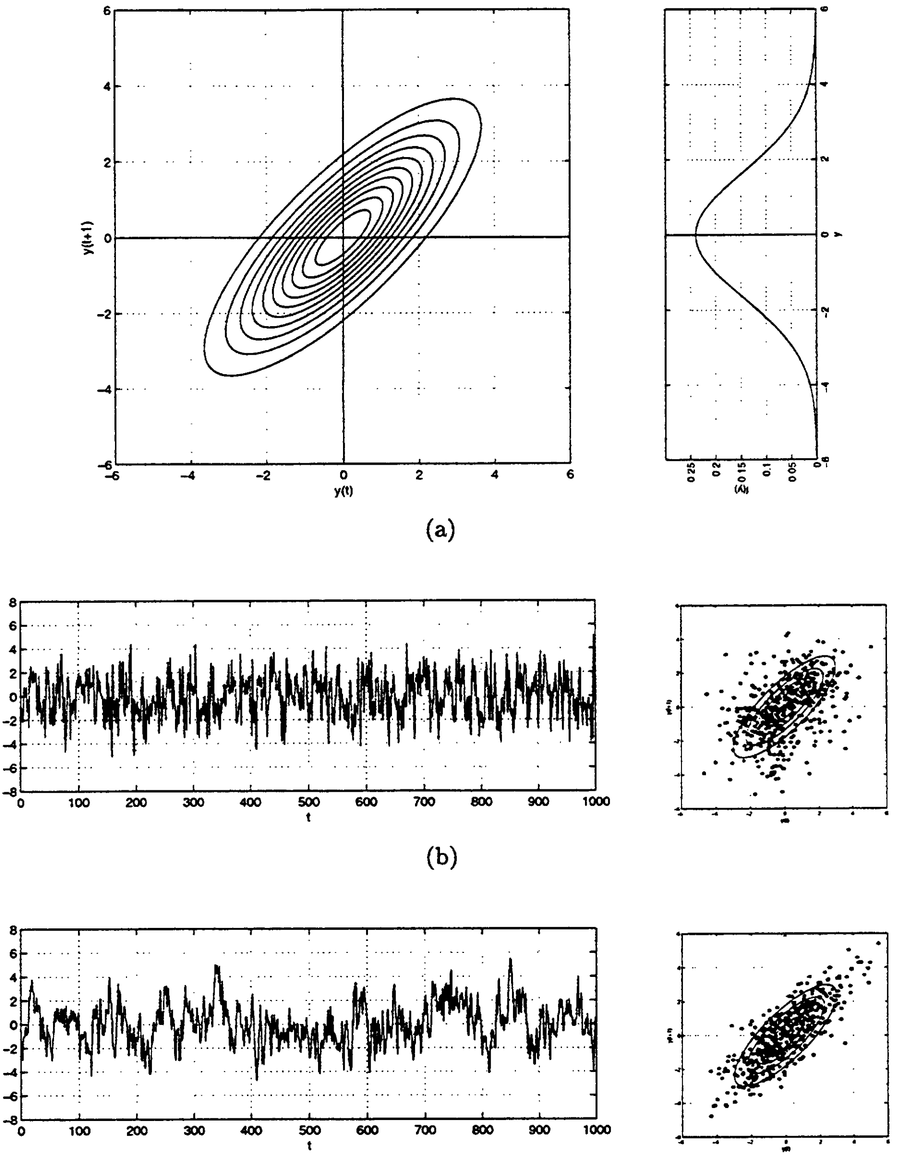


(a)

(b)

(c)

Figure 3-4: Modeling of the AR Gaussian process $Y_t = 0.8Y_{t-1} + W_t$ using a 1-state HMM: (a) (left) contour plot of bivariate Gaussian pdf for the pair of random variables (Y_t, Y_{t+1}); (right) plot of its univariate projection, the marginal pdf for the single random variable Y_t; (b) (left) realization $\tilde{\mathbf{y}}_{0:999}$ from 1-state HMM; (right) corresponding scatter plot of the pairs $(\tilde{y}_t, \tilde{y}_{t+1})$; (c) (left) realization $\mathbf{y}_{0:999}$ from true AR Gaussian process; (right) corresponding scatter plot of the pairs (y_t, y_{t+1}).



(a)

(b)

(c)

Figure 3-5: Modeling of the AR Gaussian process $Y_t = 0.8Y_{t-1} + W_t$ using a 2-state HMM: (a) (left) contour plot of bivariate Gaussian pdf for the pair of random variables (Y_t, Y_{t+1}); (right) plot of its univariate projection, the marginal pdf for the single random variable Y_t; (b) (left) realization $\tilde{\mathbf{y}}_{0:999}$ from 2-state HMM; (right) corresponding scatter plot of the pairs $(\tilde{y}_t, \tilde{y}_{t+1})$; (c) (left) realization $\mathbf{y}_{0:999}$ from true AR Gaussian process; (right) corresponding scatter plot of the pairs (y_t, y_{t+1}).

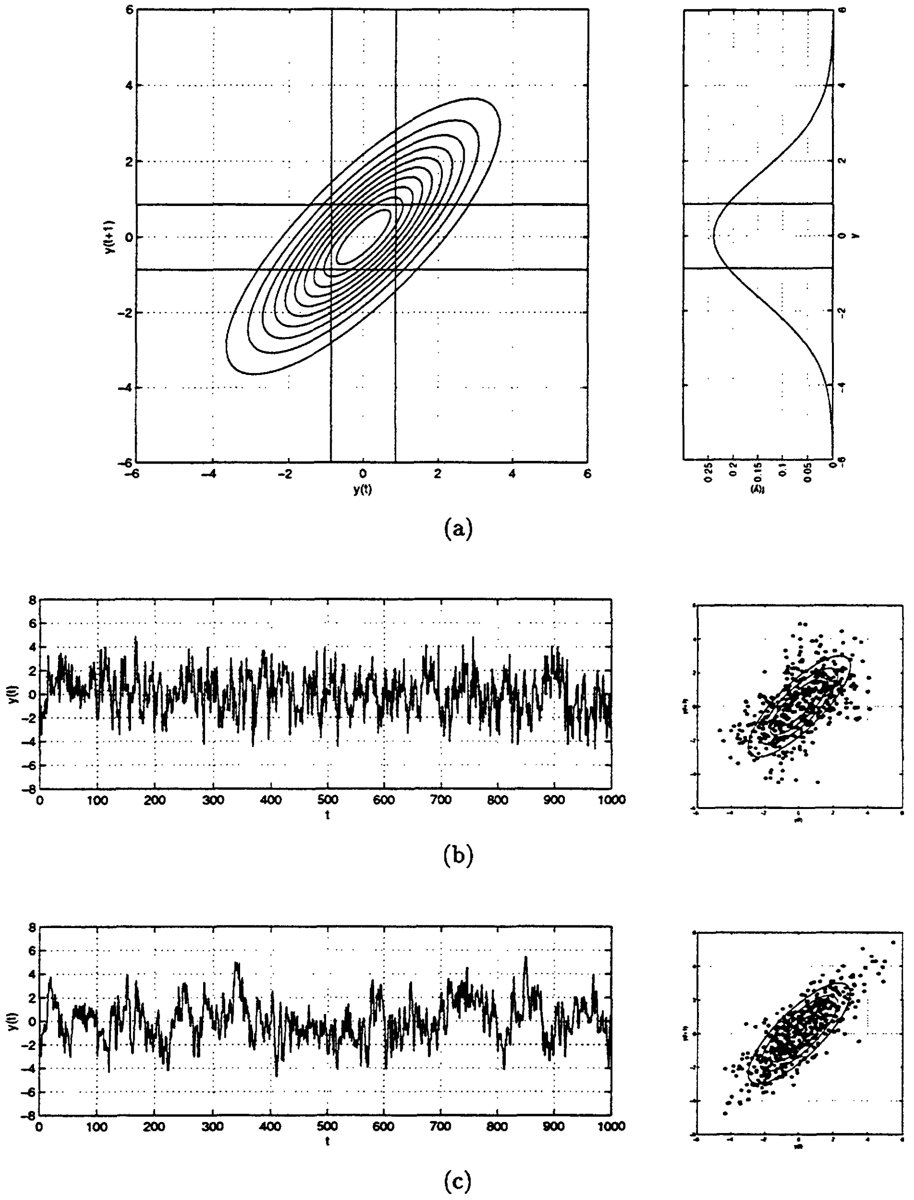
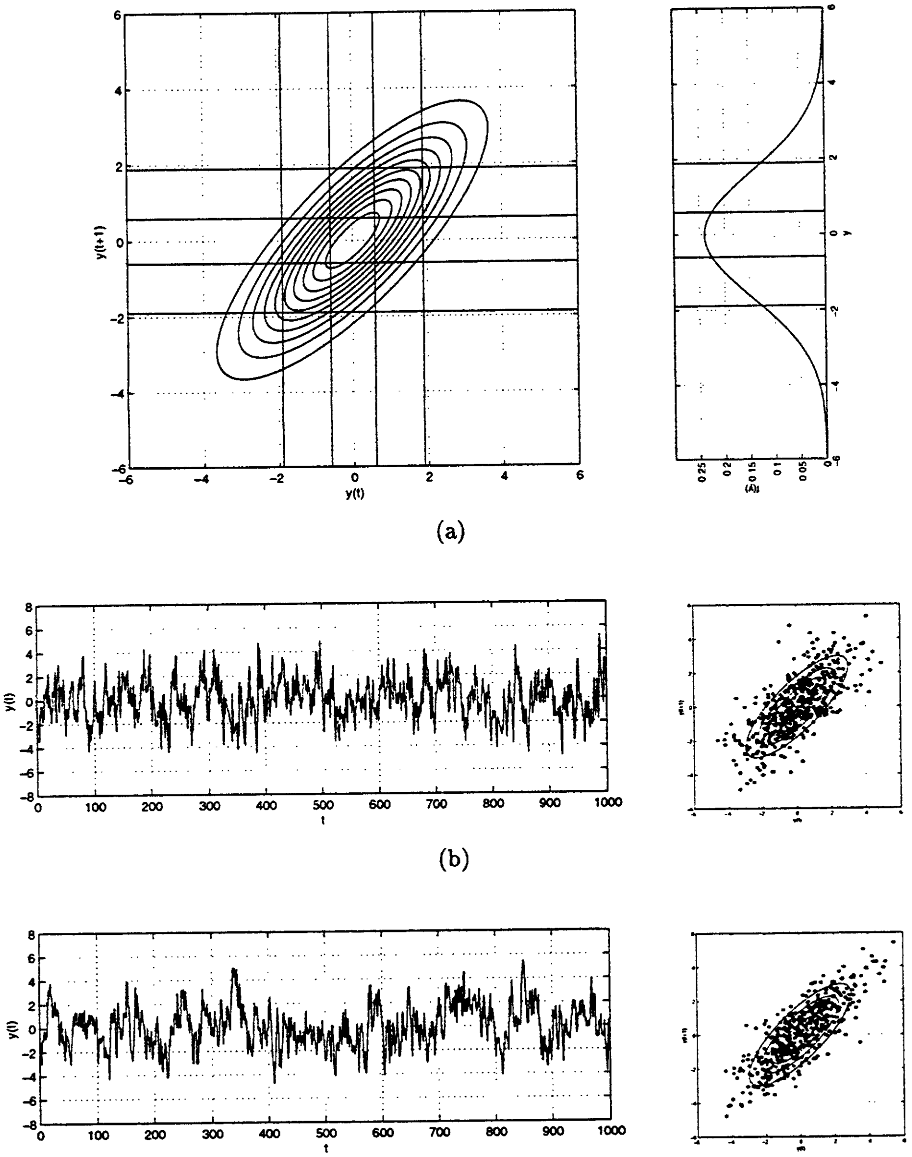


Figure 3-6: Modeling of the AR Gaussian process $Y_t = 0.8Y_{t-1} + W_t$ using a 3-state HMM: (a) (left) contour plot of bivariate Gaussian pdf for the pair of random variables (Y_t, Y_{t+1}); (right) plot of its univariate projection, the marginal pdf for the single random variable Y_t; (b) (left) realization $\tilde{\mathbf{y}}_{0:999}$ from 3-state HMM; (right) corresponding scatter plot of the pairs $(\tilde{y}_t, \tilde{y}_{t+1})$; (c) (left) realization $\mathbf{y}_{0:999}$ from true AR Gaussian process; (right) corresponding scatter plot of the pairs (y_t, y_{t+1}).



(a)

(b)

(c)

Figure 3-7: Modeling of the AR Gaussian process $Y_t = 0.8Y_{t-1} + W_t$ using a 5-state HMM: (a) (left) contour plot of bivariate Gaussian pdf for the pair of random variables (Y_t, Y_{t+1}); (right) plot of its univariate projection, the marginal pdf for the single random variable Y_t; (b) (left) realization $\tilde{\mathbf{y}}_{0:999}$ from 5-state HMM; (right) corresponding scatter plot of the pairs $(\tilde{y}_t, \tilde{y}_{t+1})$; (c) (left) realization $\mathbf{y}_{0:999}$ from true AR Gaussian process; (right) corresponding scatter plot of the pairs (y_t, y_{t+1}).

3.4.3 Verification that Approximation Improves with Model Order

While it is useful to assess the relative quality of various finite-state models — as we have just done with the aid of Figures 3-4 through 3-8 — it is also very important to verify such comparisons by using quantitative methods that are well understood. In this brief section, we demonstrate, using a classical quantitative test, that our earlier qualitative assessment was correct, i.e., that the finite-state approximations do indeed become progressively better as more states are added to the underlying Markov chain.

Specifically, let us consider the following experiment: Suppose we are given an observation of length N, and we know that it is a realization of either the true AR Gaussian process defined earlier or an HMM-based approximation of this process. Moreover, we know that these possibilities are equally likely to be true. All parameters of both processes, including the finite-state model order, L, are assumed known. We wish to determine, with minimum probability of error, which process gave rise to the observation.

It is well known that the best test to apply in this case is the likelihood ratio test (LRT). If we denote the realization by $\mathbf{z}_{0:N-1}$, and we let H_0 and H_1 represent the hypotheses that the realization was generated by the approximate and true processes, respectively, then the LRT for this situation can be expressed as

$$\text{Declare} \left\{ \begin{array}{c} H_0 \text{ true} \\ H_1 \text{ true} \end{array} \right\} \text{when } \ell_0(\mathbf{z}_{0:N-1}) \left\{ \begin{array}{c} \geq \\ < \end{array} \right\} \ell_1(\mathbf{z}_{0:N-1}), \tag{3.107}$$

where $\ell_0(\cdot)$ and $\ell_1(\cdot)$ are the log-likelihood functions associated with the approximate and true processes, respectively, and are given by

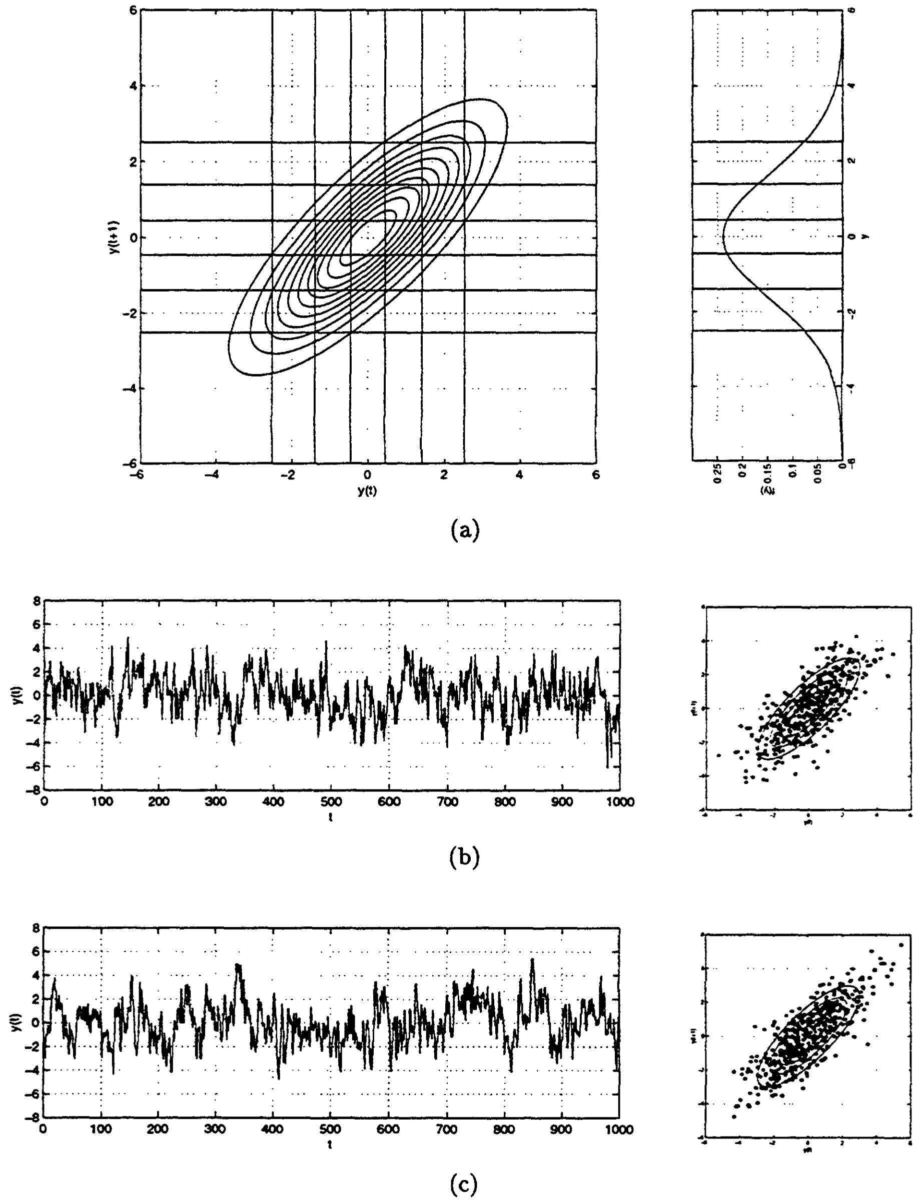
$$\begin{aligned} \ell_0(\mathbf{z}_{0:N-1}) = \log P(\theta(z_0)) &+ \sum_{t=1}^{N-1} \log Q(\theta(z_{t-1}), \theta(z_t)) \\ &- N \log(\sqrt{2\pi}\sigma_Y) - \sum_{t=0}^{N-1} \frac{y_t^2}{2\sigma_Y^2} - \sum_{j=1}^{L} N_j(\mathbf{z}_{0:N-1}) P(j) \end{aligned} \tag{3.108}$$

and

$$\begin{aligned} \ell_1(\mathbf{z}_{0:N-1}) = &- \log(\sqrt{2\pi}\sigma_Y) - \frac{z_0^2}{2\sigma_Y^2} \\ &- (N-1) \log(\sqrt{2\pi}\sigma_W) - \sum_{t=1}^{N-1} \frac{(z_t - a z_{t-1})^2}{2\sigma_W^2}. \end{aligned} \tag{3.109}$$

Here we have used the notation $N_j(\mathbf{z}_{0:N-1})$ to represent the number of occurrences in the realization $\mathbf{z}_{0:N-1}$ of a value that would have been generated in state j of the HMM.

An analytical expression for the probability of error, which we denote by P_{err}, is very difficult to obtain for this problem. Thus, we have resorted to approximating P_{err} by applying the above LRT in a series of experimental trials. Specifically, after fixing values for L and N, we generated a total of 10000 realizations (of which half were from the true



(a)

(b)

(c)

Figure 3-8: Modeling of the AR Gaussian process $Y_t = 0.8Y_{t-1} + W_t$ using a 7-state HMM: (a) (left) contour plot of bivariate Gaussian pdf for the pair of random variables (Y_t, Y_{t+1}); (right) plot of its univariate projection, the marginal pdf for the single random variable Y_t; (b) (left) realization $\tilde{\mathbf{y}}_{0:999}$ from 7-state HMM; (right) corresponding scatter plot of the pairs $(\tilde{y}_t, \tilde{y}_{t+1})$; (c) (left) realization $\mathbf{y}_{0:999}$ from true AR Gaussian process; (right) corresponding scatter plot of the pairs (y_t, y_{t+1}).

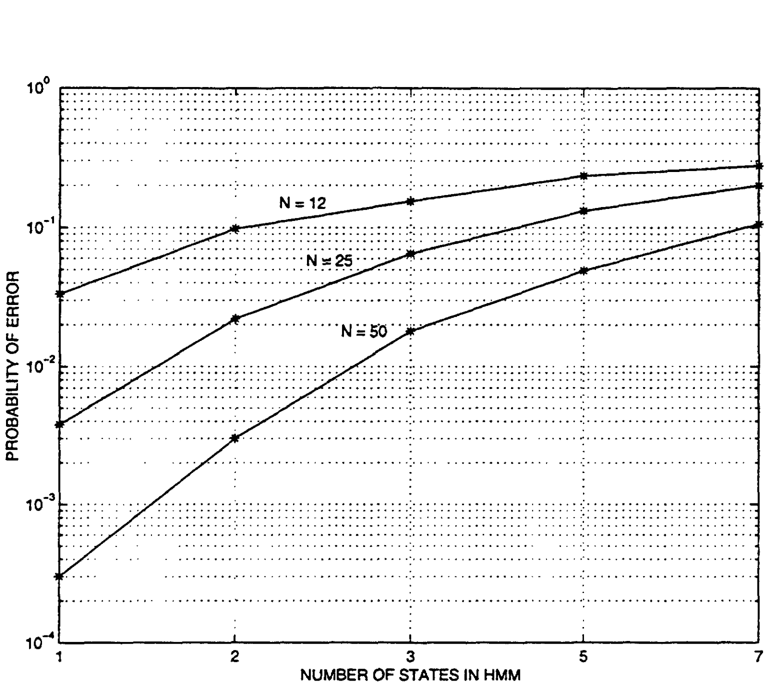


Figure 3-9: Results of applying an optimal detector to determine whether an observed data sequence is a realization of the true AR Gaussian signal $Y_t = 0.8Y_{t-1} + W_t$ or an approximation to this process by an L-state HMM. Plots of probability of error versus L are shown for observation lengths of $N = 12$, $N = 25$, and $N = 50$.

process and the other half from the approximate process) and applied the test in (3.107) to each realization. The fraction of incorrect decisions made by the optimal test served as the estimate of P_{err}.

In Figure 3-9, we present a plot of the probability of error versus the number of states included in the HMM. A total of three curves are shown on this plot, corresponding to the cases in which the length of the observed sequence was 12 samples, 25 samples, or 50 samples. As we can clearly see from these curves, the HMM-based approximations do indeed become better as L increases; that is, a high-order approximation leads to greater confusion on the part of an optimal detector than does a relatively low-order approximation. We note in addition that no pair of curves plotted in this figure ever intersect; this demonstrates simply that, for a fixed model order L, the ability of an optimal processor to discriminate between the true and approximate processes increases uniformly (or equivalently, P_{err} decreases uniformly) as the number of observed samples, N, increases.

3.5 Generalization of the Optimal Solution to Higher-Order AR Signals

Thus far in the chapter, we have focused our attention almost exclusively on the problem of approximating a first-order stationary AR process. For this first-order case, we found that the state vector of the associated dynamical system was merely a scalar, and that the state space was therefore just the real line. In fact, it was precisely because of this uncomplicated structure that we were drawn to the first-order approximation problem as a starting point; it made certain theoretical concepts easy to visualize and to understand.

As we shall soon discover, however, a number of subtleties and complexities crop up as we begin to consider problems of higher dimension. In this final section, we discuss many of the important issues that arise when we attempt to apply the concepts developed for the first-order case to the approximation of AR processes having order $K > 1$. We remark in addition that much of the material in this section is presented purely for heuristic purposes; our main objective is to identify and understand the issues involved in higher-order problems, rather than to develop techniques for generating concrete numerical solutions to these problems.

3.5.1 Specification of the State-Space Partition

Recall from our earlier discussion that, when using the finite-state approach to approximate the dynamics of a one-dimensional system, we first decomposed the real line into a collection of L disjoint intervals and then created a one-to-one correspondence between this collection of intervals and the set of L states of the approximating Markov chain. The intervals themselves could be readily specified via the $L+1$-dimensional tuple of ordered breakpoints $(d_0, d_1, \cdots, d_L)$, in which the first and last elements were constrained, respectively, by the equations $d_0 = -\infty$ and $d_L = +\infty$. Once values were specified for the remaining $L-1$ elements in the tuple, corresponding expressions could be written down immediately for the optimal values of the HMM parameters, i.e., for the initial state probabilities $\{P(i)\}_{i=1}^{L}$, the state transition probabilities $\{Q(i,j)\}_{i,j=1}^{L}$, and the output densities $\{g_i(\cdot)\}_{i=1}^{L}$. The search for the best values of the remaining $L-1$ breakpoints formed the core of the approximation problem.

In the higher-order case, however, the state space is $\mathbb{R}^K$, rather than $\mathbb{R}$, and the relevant subsets of the state space thus become full-fledged regions, rather than mere intervals. Consequently, a partitioning of this higher-dimensional space into L disjoint regions can no longer be accomplished simply by specifying the values of $L-1$ numbers on the real line as before. Instead, we must now specify a collection of contours or surfaces that would form the region boundaries within $\mathbb{R}^K$. We demonstrate this notion in Figure 3-10 for the case in which $K = 2$. Clearly, even in this two-dimensional case the definition of region boundaries is considerably more complex than it was in the one-dimensional case.

Indeed, even if a suitable description of the region boundaries could be found (say, for example, some low-order polynomial description), then for any given set of boundary values we would still be left with the problem of evaluating the best corresponding set of HMM parameters. This in turn could only be accomplished through the involved procedure of integrating the signal pdf over irregularly shaped regions in a multi-dimensional space. Moreover, even if this latter step could be achieved using a reasonable amount of computation, we would then require a method for determining the optimal set of boundaries, i.e., the partition of $\mathbb{R}^K$ that ultimately yields the best finite-state approximation of the actual signal according to the Kullback-Leibler distance metric.

It is straightforward to show, however, that the fundamental optimization principle guiding the search for the best partition remains exactly the same for the case $K > 1$ as

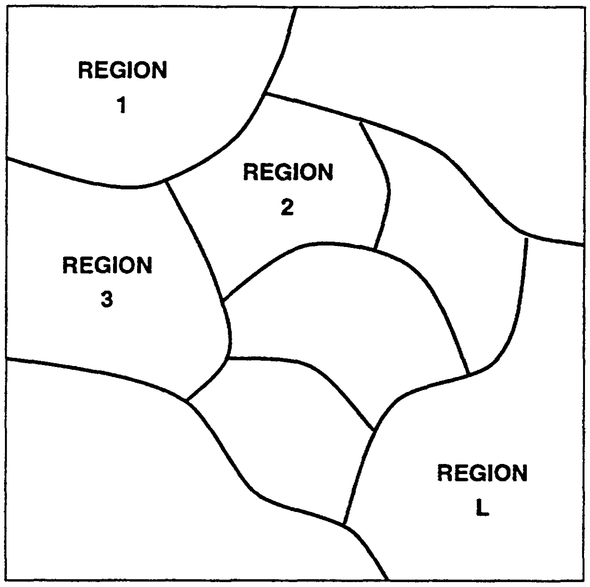


Figure 3-10: Partitioning of a two-dimensional state space into L disjoint regions.

it was for the case $K = 1$; in particular, to find the optimal partition of $\mathbb{R}^K$, we should adjust the boundaries of its L constituent regions so as to maximize the mutual information between successive state variables of the underlying Markov chain.

3.5.2 Evaluation of the HMM Parameters

Let us suppose for the moment that a tractable description of the region boundaries (or equivalently, of the regions themselves) is available, so that we may concentrate on the subsequent step of finding the best HMM parameters associated with a particular set of boundaries. Throughout this section, we will assume that a partition of the state space has already been specified, and we will denote the L regions that make up the space by $\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_L$.

By using the same basic principles of optimality that were applied in the first-order problem, we readily conclude that the best choice for the ith element of the initial state pmf for the underlying Markov chain is given by

$$P(i) = \int_{\mathbf{y} \in \mathcal{R}_i} f_{\mathbf{Y}_{t:t+K-1}}(\mathbf{y}) d\mathbf{y}, \tag{3.110}$$

i.e., it is equal to the unconditional probability that the original state vector $\mathbf{X}_t$ will lie in the region $\mathcal{R}_i$. Furthermore, we conclude that the best choice for the output pdf associated with the ith state of the Markov chain is given by

$$f_i(\mathbf{x}) = \begin{cases} \frac{1}{P(i)} f_{\mathbf{Y}_{t:t-K+1}}(\mathbf{x}) & \text{if } \mathbf{x} \in \mathcal{R}_i, \\ 0 & \text{otherwise,} \end{cases} \tag{3.111}$$

i.e., it is defined to be nonzero only on the region $\mathcal{R}_i$, but on this region it is a scaled version of the original pdf for $\mathbf{X}_t$. If we prefer to work instead with the univariate output densities $\{g_i(\cdot)\}_{i=1}^L$, we can derive the optimal choices for these from their multivariate counterparts $\{f_i(\cdot)\}_{i=1}^L$ defined above. In particular, the optimal output densities are given by

$$g_i(y) = \frac{1}{P(i)} \int_{(y,\mathbf{y}_{t-1:t-K})\in\mathcal{R}_i} f_{\mathbf{Y}_{t:t-K+1}}(y, \mathbf{y}_{t-1:t-K})\, d\mathbf{y}_{t-1:t-K}. \tag{3.112}$$

With regard to these univariate densities, we note that there is a significant difference between the case in which $K > 1$ and the previously considered case in which $K = 1$. Specifically, in contrast to the first-order case, the collection of functions $\{g_i(\cdot)\}_{i=1}^L$ may now have overlapping regions of support on the real line, despite the fact that the original regions in state space are disjoint. For example, in Figure 3-10 we can see that the projections of Region 1 and Region 2 onto the real line will certainly not be disjoint.

The above solutions for $P(i)$ and $f_i(\cdot)$ (or, equivalently, for $P(i)$ and $g_i(\cdot)$) are straightforward extensions of the results we obtained for the case $K = 1$. However, the solution for the critical state transition probability $Q(i,j)$ is somewhat less direct. Recall that $Q(i,j)$ can be expressed as

$$Q(i,j) = \frac{R(i,j)}{P(i)}, \tag{3.113}$$

where $R(i,j) = \Pr\{\Theta_t = i, \Theta_{t+1} = j\}$. It therefore suffices to find an expression for $R(i,j)$. We know from our earlier analysis that, in terms of the dynamics of the original process, the optimal value for the quantity $R(i,j)$ is simply the joint probability that the following two events will occur:

(i) $\mathbf{X}_t \in \mathcal{R}_i$

(ii) $\mathbf{X}_{t+1} \in \mathcal{R}_j$

To get a concise expression for this joint probability, let us first go back to the definitions of $\mathbf{X}_t$ and $\mathbf{X}_{t+1}$, which are given by

$$\mathbf{X}_t = (Y_t, Y_{t-1}, \cdots, Y_{t-K+1}) \tag{3.114}$$

$$\mathbf{X}_{t+1} = (Y_{t+1}, Y_t, \cdots, Y_{t-K+2}). \tag{3.115}$$

Now, if we insist on the condition $\mathbf{X}_t \in \mathcal{R}_i$, but place no constraints on the subsequent sample in the process, Y_{t+1}, then for the augmented state vector $(Y_{t+1}, \mathbf{X}_t)$ we have that

$$(Y_{t+1}, \mathbf{X}_t) \in \mathbb{R} \times \mathcal{R}_i. \tag{3.116}$$

Similarly, if we impose the restriction $\mathbf{X}_{t+1} \in \mathcal{R}_j$, but place no constraints on the preceding sample in the process, Y_{t-K+1}, then for the augmented vector $(\mathbf{X}_{t+1}, Y_{t-K+1})$ we have that

$$(\mathbf{X}_{t+1}, Y_{t-K+1}) \in \mathcal{R}_j \times \mathbb{R} \tag{3.117}$$

But observe that since

$$(Y_{t+1}, \mathbf{X}_t) = (\mathbf{X}_{t+1}, Y_{t-K+1}) = (Y_{t+1}, Y_t, \cdots, Y_{t-K+1}), \tag{3.118}$$

we can immediately combine (3.116) and (3.117) to obtain the consolidated condition

$$(Y_{t+1}, Y_t, \cdots, Y_{t-K+1}) \in \mathcal{R}_{ij}, \tag{3.119}$$

where $\mathcal{R}_{ij}$ denotes the region in $\mathbb{R}^{K+1}$ given by the set intersection

$$\mathcal{R}_{ij} = (\mathbb{R} \times \mathcal{R}_i) \cap (\mathcal{R}_j \times \mathbb{R})\,. \tag{3.120}$$

From this last result, we finally have that

$$R(i,j) = \int_{\mathbf{y} \in \mathcal{R}_{ij}} f_{\mathbf{Y}_{t:t+K}}(\mathbf{y})\, d\mathbf{y}. \tag{3.121}$$

This is the appropriate extension of our result from the first-order case, in which we had

$$R(i,j) = \int_{[d_{i-1}, d_i] \times [d_{j-1}, d_j]} f_{\mathbf{Y}_{t:t+1}}(\mathbf{y})\, d\mathbf{y}. \tag{3.122}$$

The issues that arise in the approximation of higher-order autoregressive processes will also be encountered in Chapter 4. There, we address the related modeling problem of finding optimal HMM parameter values under the assumption that we are given a finite number of observations of the true AR random process, rather than the actual pdf of the process. Since the true AR process may have order $K > 1$, obtaining a solution will require practical methods for representing irregularly shaped regions in a high-dimensional space, as well as methods for estimating densities and integrals of densities over such regions.

3.6 Discussion

3.6.1 Necessity of Constraints for Finite-State Approximation

In the formulation of the signal approximation problem stated at the beginning in the chapter, we imposed a number of rather stringent constraints on our finite-state signal model. Here we attempt to explain why these constraints were needed to make the approximation problem mathematically tractable.

First, recall that we made extensive use of the state-space partitioning constraint, the purpose of which was to enforce a mapping between the L disjoint regions making up a partition of the original state space and the L states of the signal model. To maintain logical consistency with this constraint, we also defined the support set for the output pdf associated with each state to be precisely the same as the region assigned to that state; hence, the support sets themselves were not allowed to overlap. When taken together, these constraints enabled us to infer the state of the Markov chain unambiguously from the HMM output at each time step, and they therefore greatly simplified the resulting optimization problem. It is important to note that if these constraints had not been imposed on the model, the optimization problem might have been intractable. If, for example, we had allowed the support sets for the HMM output densities to overlap, then in order to calculate the probability of a given output sequence, we would have had to account for every possible underlying state sequence of the Markov chain that could have produced this output. This may well have consumed a prohibitive amount of computation, since the number of such state sequences grows exponentially with the observation length.

We also imposed the constraint that the underlying dynamics of our finite-state approximation were to be structured in the form of a Markov chain. While this constraint was intuitively appealing since the true signal was in fact known to be Markov, it too was introduced to make the optimization problem tractable. Recall that we had already defined a quantization function, $\theta(\cdot)$, to effect the mapping between the original state space associated with the true signal and the finite state set associated with the approximation. Therefore, in the absence of the Markov-chain constraint, the most direct method of creating a finite-state approximation of the true state vector sequence $\{\mathbf{X}_t\}$ would simply have been to apply this quantization function to each vector in the sequence. This would have created an entirely new random process $\{\theta(\mathbf{X}_t)\}$ in which each element assumed values on the set $\{1, 2, \cdots, L\}$; however, this new process would not necessarily possess the Markov property. Thus, when evaluating the probability of a given finite-state sequence under this approach, we would not have realized the computational savings made possible by the Markov-chain constraint.

3.6.2 Alternative Criteria for Assessing Approximation Quality

The criteria we proposed earlier in the chapter for determining the quality of an approximation — i.e., the maximin probability-of-error criterion and the Kullback-Leibler distance metric — were based on the principle that we should match the shapes of the true and approximate signal densities in an appropriate way. Such criteria are useful from the standpoint of general-purpose signal representation; that is, the criteria themselves are actually independent of any specific signal processing task, but nonetheless allow us to develop a signal model that is likely to perform reasonably well at most tasks, provided that the order of the model is sufficiently high.

Sometimes, however, it may be desirable to use other, more task-specific ways of assessing approximation quality. In the context of a well defined signal processing task, the most logical criteria for determining what makes the best signal approximation are usually obvious once we know how the approximation is going to be used in performing the task. Consider, for example, the task of signal estimation, in which the objective is to produce an estimate of the source signal $\{Y_t\}$ that has been corrupted by an independent additive noise process $\{V_t\}$. Suppose that the noise has a simple statistical description (e.g., it is white and Gaussian) and that $\{\tilde{Y}_t\}$ is our HMM-based representation of $\{Y_t\}$. Furthermore, suppose that, under the structural constraints imposed by this HMM assumption, an optimal processor to the signal estimation problem can easily be derived. Let us denote by $\widetilde{\tilde{Y}}_t(\mathbf{Z}; \mathbf{\Psi})$ the value generated by such a processor at time t, where $\mathbf{Z}$ is our finite-length corrupted observation and $\mathbf{\Psi}$ is a multi-dimensional tuple representing all of the parameters needed to completely specify the HMM for $\{\tilde{Y}_t\}$. Then, if the signal estimate we desire is an MMSE estimate, we should select the optimal tuple of HMM parameters according to the rule

$$\hat{\Psi} = \arg\min_{\Psi \in \mathcal{P}} E\left\{(\widetilde{\hat{Y}}_t(\mathbf{Z};\Psi) - Y_t)^2\right\}, \tag{3.123}$$

where $\mathcal{P}$ is the collection of all tuples Ψ that yield a valid HMM-based signal approximation. Once we know the value of the optimal tuple, we would then have a precise characterization of the best approximate signal $\{\tilde{Y}_t\}$.

If an approximation can be obtained by solving (3.123), its performance in the signal estimation problem is guaranteed to be at least as good as the performance achieved by any other approximation in $\mathcal{P}$, including an approximation which is optimal under the Kullback-Leibler measure. However, the solution to (3.123) may not perform well when applied to a task other than signal estimation, simply because any two distinct signal processing tasks can have radically different objective functions. In view of this lack of "task robustness," we may require a number of different signal approximations, one for each task that must be performed. These might include, for example, the tasks of detection, classification, enhancement, or compression. A task-dependent approach of this kind may indeed yield better overall performance, but it will also lead to a considerable increase in complexity in the design of a signal processing system. In particular, each new signal model obtained would require additional storage, and all of the models together would have to be manipulated by a higher-level, centralized decision system, whose purpose would be to determine which model is appropriate for the particular task at hand.

3.6.3 Advantages and Limitations of Using HMMs

We have seen that a hidden Markov model can be used to approximate a real-world signal concisely by capturing the dynamical behavior of the signal's most important statistical features; consequently, it has the potential of greatly simplifying any signal analysis that must be performed. In addition, if the true signal is known to be stationary and can be described as the output of a known dynamical system, then it is clear, at least in principle, that an HMM could be designed to represent the signal with arbitrarily high accuracy (via the quantization approach described earlier) by partitioning the state space infinitely finely. Another clear benefit of using HMMs is that their mathematical properties have been investigated by researchers from widely varied disciplines over a period of more than three decades; as a result, such models are now very well understood. Moreover, a number of sophisticated, computationally efficient algorithms have already been developed for many signal processing applications involving HMMs.

On the other hand, the use of HMMs for signal approximation also has a number of limitations. These limitations must also be taken into account, since the set of signals that we might attempt to approximate with HMMs is virtually limitless. We begin by observing that an HMM is capable of modeling temporal dependencies in the true signal only to the extent allowed by its coarse, finite-state dynamical structure, i.e., its underlying Markov chain. Among all conceivable dynamical models, the Markov chain possesses one of the simplest of memory structures available, second only to that of a process that is entirely memoryless. In contrast, the signal that is being approximated might be extremely complex, and it might not even obey the Markov property. Thus, perhaps the most serious limitation of HMMs is that the probabilistic structure of an HMM may be too simplistic to capture the intricate detail in the true signal — detail that might be critical in successfully performing the signal processing task for which the HMM will be used.

There are undoubtedly many cases in which an HMM-based approximation would be suitable, provided that we allow a sufficient number of states in its Markov chain and a sufficient number of degrees of freedom to describe each of its output densities. In such cases, however, we may discover another of the potentially serious limitations of using HMMs, namely that the specification of an HMM could require solving for an enormous number of parameters. For example, it is not inconceivable that an HMM-based representation of a given signal may require as many as 50 states, and that specifying the output pdf in each state may require as many as 10 parameters. This means that we would need 50 initial probabilities, 2500 state transition probabilities, and 500 pdf parameters in order to specify the HMM completely. Even if we could find optimal values for more than 3000 parameters, the resulting model would be considered impractical in many signal processing situations.

Fortunately, as we will discover in Chapter 5, it is not always necessary or desirable to incorporate a large number of parameters into an HMM simply so that we can represent the true signal at its finest level of statistical detail. Quite the contrary, in certain signal processing applications, a rather coarse HMM-based approximation of the true signal is adequate to achieve nearly optimal performance.

Chapter 4

Building Finite-State Markov Models from Observations

4.1 Introduction

In Chapter 3, we considered the problem of how to find the best HMM-based representation of a stationary random signal given exact knowledge of the signal pdf. This was a useful starting point for our analysis of HMMs because it compelled us to consider how finite-state modeling should be performed when complete information about the true signal is available. In a real-world signal processing situation, however, we rarely have such a large amount of prior knowledge about any of the signals that make up the measurement. Thus, although our analysis from the previous chapter produced a number of useful theoretical guidelines for finite-state modeling, certain assumptions associated with the approximation problem addressed there were somewhat unrealistic.

In this chapter, we adopt a more practical viewpoint and consider the problem of how to construct an HMM-based representation of the true random process when we have only a finite-length observation of this process available, rather than a complete probabilistic description of it. Thus, the problem we consider here is essentially one of HMM source identification. In the following two subsections, we give an outline of the assumptions and notation that will be used in connection with the HMM source identification problem, and we provide a concise formulation of the problem itself. In the third subsection, we describe how the remaining material in the chapter is organized.

4.1.1 Preliminary Assumptions and Notation

In the latter part of Chapter 3, we discussed some of the complexities involved in the finite-state modeling of AR processes having order greater than one. Clearly, we must address the issues raised in that discussion before developing our source identification algorithm. We pointed out, for example, that arbitrary multi-dimensional regions in a state-space partition cannot be easily represented or manipulated with finite memory and computing resources. In addition, the optimal output densities associated with the states of the HMM (which

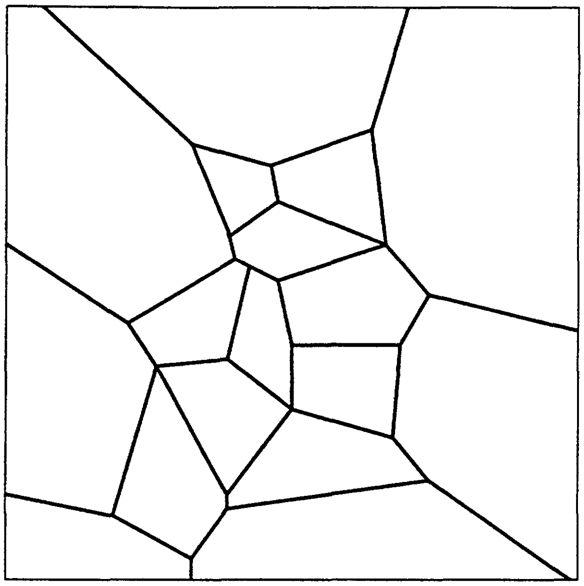


Figure 4-1: Depiction of a typical Voronoi partition in two-dimensional space.

were expressed directly in terms of the true signal pdf in Chapter 3) are free of restrictions, and hence may also require a large amount of storage to represent accurately.

A further complication is that, in addition to addressing the issue of computational complexity, we must also deal with the issue of uncertainty, since the exact signal pdf is unknown. In particular, observe that quantities such as the optimal initial state probabilities and state transition probabilities of the Markov chain, as well as the mutual information between successive state variables in the chain, are all unambiguously defined when the pdf of the true random process is given. In the present case, however, these quantities will have to be estimated using only a finite-length signal realization. In the remainder of this subsection, we describe the techniques that will be used to address these issues.

We will represent regions in $\mathbb{R}^K$ efficiently using a simple geometric construction known as a Voronoi partition [152]. A Voronoi partition having L regions can be completely characterized by L distinct points in the space. Let us denote these points by $\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_L$ and their corresponding regions by $\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_L$. The region $\mathcal{R}_j$ is defined by

$$\mathcal{R}_j = \{\mathbf{x} \in \mathbb{R}^K \mid D(\mathbf{x}, \mathbf{c}_j) \leq D(\mathbf{x}, \mathbf{c}_i), \quad i = 1, 2, \cdots, L\} \tag{4.1}$$

where the notation $D(\mathbf{x}, \mathbf{c})$ represents the Euclidean distance between the points $\mathbf{x}$ and $\mathbf{c}$. In other words, the set $\mathcal{R}_j$ contains all points in $\mathbb{R}^K$ that are closer to the point $\mathbf{c}_j$ than to any other point $\mathbf{c}_i$, $i \neq j$. We will occasionally refer to the region $\mathcal{R}_i$ as a Voronoi region and to its associated point $\mathbf{c}_i$ as the anchor point for the region. An example of a randomly generated Voronoi partition in two-dimensional space is shown in Figure 4-1. In view of the simple rule given in (4.1), we see that a major advantage of using this construction is savings in memory and computation; at the end of the chapter, we will discuss several limitations associated with using this type of partition.

At certain stages of our source identification algorithm, we will need to know which region within the current Voronoi partition contains the data point $\mathbf{x}_t$. For this purpose it is convenient to introduce a class label for the tth data point, which we denote by ω_t. This class label will take exactly one of the values in the set $\{1, 2, \cdots L\}$. Once a set of anchor points $\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_L\}$ has been fixed, the value of ω_t is defined according to the formula

$$\omega_t = \underset{j \in \{1,2,\cdots,L\}}{\arg\min}\ D(\mathbf{c}_j, \mathbf{x}_t). \tag{4.2}$$

The tuple consisting of all such class labels will be denoted by

$$\mathbf{\Omega} = (\omega_0, \omega_1, \cdots, \omega_{N-1}) \tag{4.3}$$

and will be referred to as the classification sequence.

Suppose we have fixed a set of anchor points and have performed the categorization by region described above. In order to assess the quality of the partition represented by this set of anchor points, we must then estimate the value of mutual information associated with this categorization. Recall from Chapter 3 that for this calculation we require values for both the joint and marginal pmfs for the state variables of the underlying Markov chain. We will use empirical estimates for these pmfs given by

$$P(j; \mathbf{\Omega}) = \frac{1}{N} \sum_{t=0}^{N-1} \gamma_j(\omega_t), \qquad j = 1, 2, \cdots, L \tag{4.4}$$

and

$$R(i, j; \mathbf{\Omega}) = \frac{1}{N-1} \sum_{t=1}^{N-1} \gamma_{ij}(\omega_{t-1}, \omega_t), \qquad i, j = 1, 2, \cdots, L, \tag{4.5}$$

where $\gamma_j(\cdot)$ and $\gamma_{ij}(\cdot)$ are binary-valued indicator functions defined respectively by

$$\gamma_j(\omega) = \begin{cases} 1 & \text{if } \omega = j \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

and

$$\gamma_{ij}(\omega_1, \omega_2) = \begin{cases} 1 & \text{if } (\omega_1, \omega_2) = (i, j) \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

Observe from (4.4) and (4.5) that we have expressed the pmf estimates with an explicit dependence on the classification sequence $\mathbf{\Omega}$. These pmf estimates measure, respectively, the fraction of times that the symbol j occurs in this classification sequence and the fraction of transitions of the form (i, j) that occur in the sequence. Using these pmf estimates, we can compute the associated estimate of the mutual information given by

$$I(\mathbf{\Omega}) = \sum_{i=1}^{L}\sum_{j=1}^{L} R(i,j;\mathbf{\Omega}) \log \frac{R(i,j;\mathbf{\Omega})}{P(i;\mathbf{\Omega})P(j;\mathbf{\Omega})}. \tag{4.8}$$

Because our goal is to find the classification sequence $\mathbf{\Omega}$ that maximizes this measure of mutual information, we will often refer to the function $I(\cdot)$ as the objective function in the remainder of the chapter.

Finally, we will assume for convenience that the output pdf associated with the ith state of our HMM is a Gaussian mixture having M_i constituent elements, and is defined by

$$g_i(y) = \sum_{j=1}^{M_i} \mathcal{N}(y; \mu_{ij}, \sigma_{ij}), \qquad -\infty < y < \infty. \tag{4.9}$$

4.1.2 Problem Statement and Approach to Solution

We will assume that the autoregressive order of the signal, K, the number of states in the HMM-based signal approximation, L, and the number of components in each Gaussian-mixture output pdf, M_i, are precisely known. In addition, we assume that we have a finite-length realization of the source signal given by

$$\mathbf{y}_{-K+1:N-1} = (y_{-K+1}, y_{-K+2}, \cdots, y_{N-1}). \tag{4.10}$$

Under the same state vector definition used in Chapter 3, i.e., $\mathbf{X}_t = (Y_t, Y_{t-1}, \cdots, Y_{t-K+1})$, the assumption that we have the sequence of signal values above is equivalent to the assumption that we have the sequence of state vector values given by

$$\mathbf{x}_{0:N-1} = (\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{N-1}), \tag{4.11}$$

since $\mathbf{y}_{-K+1:N-1}$ could be reconstructed perfectly from $\mathbf{x}_{0:N-1}$ and vice versa. From this sequence of N data points in K-dimensional space, we wish to estimate the parameter values of the best HMM-based representation of the source signal; these include the values of the initial state probabilities and state transition probabilities of the underlying Markov chain, as well as the means, standard deviations, and weighting coefficients of the Gaussian-mixture densities associated with the states of the chain. It is understood that the HMM to be estimated must satisfy both the stationary constraint and the state-space partitioning constraint described in Chapter 3.

Our approach to finding the best HMM-based representation of the source signal will be to construct an iterative, ad hoc algorithm which implements the theoretical guidelines established in Chapter 3. We decompose the algorithm into two basic parts: (i) estimation of the optimal Voronoi partition of the state space; and (ii) estimation of the HMM parameters based on this optimal partition. To solve the first of these two subproblems, we develop an algorithm which selects an appropriate initial state-space partition and then systematically adjusts the boundaries of this partition to increase the value of the objective function (i.e., the mutual information between successive state variables of the Markov chain) at each step. Once the best partition has been reached, the state-vector realizations contained in each region of the partition are used to generate empirical estimates of the HMM parameter values.

4.1.3 Chapter Organization

The chapter consists mainly of a description of the components of our HMM source identification algorithm. First, we describe a procedure for finding the best partition of the state space associated with the observed signal; then, we describe how this partition can be used to estimate the parameters of the Markov chain as well as the parameters of the densities associated with the states of the chain. At the end of the chapter, we discuss a number of advantages and limitations of our algorithm, and we identify several open issues as potential directions for future work.

4.2 Estimation of the Optimal State-Space Partition

The purpose of the first part of our HMM source identification algorithm is to estimate the best Voronoi partition of the state space based on the given sequence of state-vector realizations $\mathbf{x}_{0:N-1}$. This part of the algorithm is made up of three stages: (i) selection of a suitable initial Voronoi partition of the state space, (ii) iterative refinement of the Voronoi partition, and (iii) termination of the iterative procedure. We describe each of these parts of the algorithm in the following three subsections. In Figure 4-2, we show plots of the output produced by this part of the algorithm in a particular source identification problem. For the case depicted here, the true AR signal has order two, and therefore the state space is two-dimensional.

4.2.1 Selection of an Initial Partition

We specify an initial Voronoi partition of our K-dimensional state space by choosing values for the L anchor points $\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_L$. To simplify the selection process, we restrict the anchor points to lie in the given set of N state vector realizations $\{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{N-1}\}$. In particular, we use a randomized procedure whereby we choose each of the L anchor points successively from the set of N realizations, without replacement, assuming after each selection that all remaining realizations are equally likely candidates. This procedure has the advantage that it yields an approximate random sample from the true marginal pdf of the state vector, provided that N is large relative to L and to the dependence length associated with the original random signal $\{Y_t\}$.

Once the initial partition has been specified in this way, we can assess the quality of the partition by computing its associated objective function value. We evaluate the objective function by performing the following three steps: first, we categorize each state vector realization according to its region number using the minimum distance formula in (4.2); then, from the resulting classification sequence, $\mathbf{\Omega}$, we compute the empirical estimates

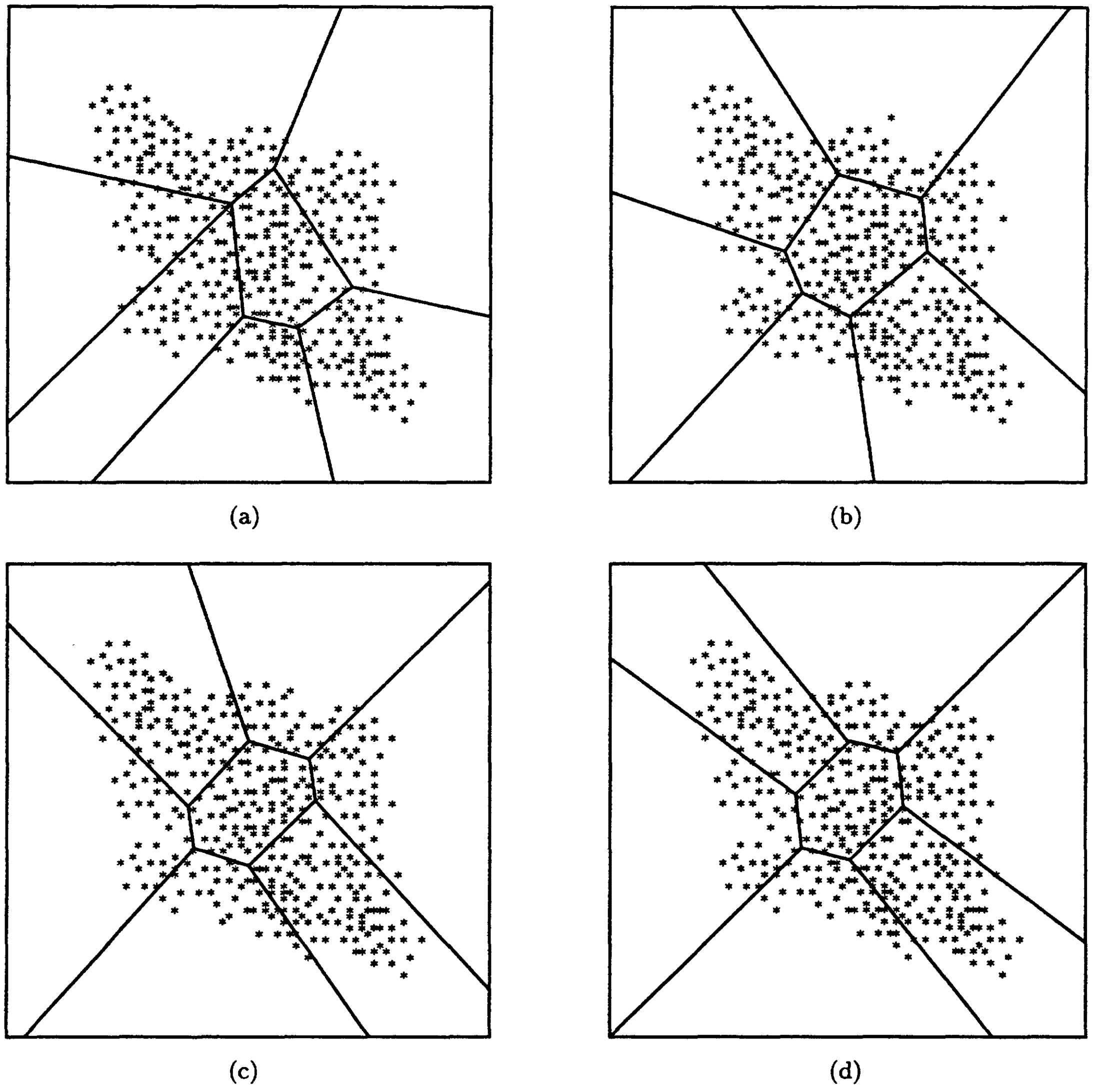


Figure 4-2: Voronoi partition at various stages of the iterative refinement algorithm: (a) initial partition; (b) partition after 5 iterations; (c) partition after 10 iterations; (d) partition after 20 iterations.

$\{P(i;\Omega)\}_{i=1}^{L}$ and $\{R(i,j;\Omega)\}_{i,j=1}^{L}$ using (4.4) and (4.5); finally, we calculate the corresponding estimate of mutual information using (4.8).

This process of selecting a set of anchor points and evaluating the corresponding partition can be repeated to find a different set with a higher objective function value. In fact, we could in principle find the best such partition by testing all distinct sets of L anchor points that could be drawn from the pool of N data points. However, even for modest values of L and N, the amount of computation required to find the best initial set using this technique would be prohibitive. An attractive alternative to this exhaustive search is to choose a fixed number of sets of anchor points at random (say J_{init}, where $J_{\text{init}} \ll \frac{N!}{L!(N-L)!}$), compute the mutual information corresponding to each set, and retain the set that yields the largest mutual information as the specification of the initial partition. A detailed step-by-step description of this initialization procedure is given in Figure 4-3.[1]

[1] In this figure and in the following two figures, we use several new notational symbols, which we define here. First, we refer to the set of time indices associated with our observations as $\mathcal{T} = \{0, 1, \cdots, N-1\}$. We also use the expression $\mathcal{S}' = \mathrm{RS}(\mathcal{S}; j)$ to indicate that $\mathcal{S}'$ is a randomly selected subset of $\mathcal{S}$ consisting of j elements. Finally, we define δ_t to be an ordered tuple of length N whose tth element is equal to 1 and whose remaining elements are equal to 0.

INITIALIZATION PROCEDURE

$\boxed{0}$ DESCRIPTION:

Initialize loop counter and objective function value.

OPERATION:

$$n \longleftarrow 0$$
$$I_{\max} \longleftarrow 0$$

$\boxed{1}$ DESCRIPTION:

Select L points at random from the given set of N and use these as anchor points for the next Voronoi partition to be tested.

OPERATION:

$$\{t_1, t_2, \cdots, t_L\} \longleftarrow \mathrm{RS}(\mathcal{T}; L)$$
$$\mathbf{c}_j \longleftarrow \mathbf{x}_{t_j}, \qquad j = 1, 2, \cdots, L$$

$\boxed{2}$ DESCRIPTION:

Compute the class label for each data point by determining the identity of the nearest anchor point.

OPERATION:

$$\omega_t \longleftarrow \underset{j \in \{1,2,\cdots,L\}}{\arg\min} D(\mathbf{c}_j, \mathbf{x}_t), \qquad t \in \mathcal{T}$$
$$\Omega \longleftarrow (\omega_0, \omega_1, \cdots, \omega_{N-1})$$

$\boxed{3}$ DESCRIPTION:

Calculate empirical estimates of the marginal and joint probability mass functions associated with state variables of the underlying Markov chain.

OPERATION:

$$P(j) \longleftarrow \frac{1}{N} \sum_{t=0}^{N-1} \gamma_j(\omega_t), \qquad j = 1, 2, \cdots, L;$$
$$R(i,j) \longleftarrow \frac{1}{N-1} \sum_{t=1}^{N-1} \gamma_{ij}(\omega_{t-1}, \omega_t), \qquad i, j = 1, 2, \cdots, L.$$

Figure 4-3: Description of algorithm for selecting an initial Voronoi partition. *(Continued on following page.)*

$\boxed{4}$ DESCRIPTION:

Compute the value of the objective function, which is the estimated mutual information between successive state variables of the Markov chain.

OPERATION:

$$I \longleftarrow \sum_{i=1}^{L}\sum_{j=1}^{L} R(i,j) \log \frac{R(i,j)}{P(i)P(j)}$$

$\boxed{5}$ DESCRIPTION:

Check if the current value of the objective function is the largest value encountered thus far. If so, store this value as well as the classification sequence that produced it; if not, proceed to the next step.

OPERATION:

```
if $I > I_{\max}$ then
        $I_{\max} \longleftarrow I$
        $\Omega_{\max} \longleftarrow \Omega$
else
        goto $\boxed{6}$
endif
```

$\boxed{6}$ DESCRIPTION:

Increment the loop counter and check if a sufficient number of subsets of the data have been tested. If so, proceed to the body of the algorithm; if not, test another subset.

OPERATION:

```
$n \longleftarrow n + 1$

if $n = J_{\text{init}}$ then
        goto $\boxed{\text{ITERATIVE REFINEMENT PROCEDURE}}$
else
        goto $\boxed{1}$
endif
```

Figure 4-3: *(Continued from previous page.)* Description of initialization algorithm.

4.2.2 Iterative Refinement of the Iteration

To improve upon the Voronoi partition generated by the initialization procedure, we now make a series of small adjustments to the anchor points of the partition in such a way that the objective function value increases at every step. This stage of the algorithm, which we refer to as the iterative refinement procedure, is described in detail in Figure 4-5.

During the iterative refinement procedure, we remove our original restriction that the anchor points be elements of the set of data points $\{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{N-1}\}$, so that the locations of the anchor points will now be unconstrained. However, the anchor points will not be adjusted directly during the refinement, since this could entail a computationally expensive search in the vicinity of each anchor point within $\mathbb{R}^K$. Instead, at each iteration we will adjust the anchor points indirectly by making small changes to the regions that they represent. These changes to the regions will in turn be implemented by changing the class labels of certain data points contained in each region.

Thus, adjustments to the partition are ultimately made by reassigning data points to different regions. Observe that several alternative class labels can be hypothesized for each data point in order to find the best label for this point under the current partition. (The definition of the best class label for a particular point will be given later; roughly, it is the label which is most likely to yield the greatest increase to the objective function value when the labels of all other points are held constant.) To eliminate unnecessary computation, we make use of two key principles at each iteration. First, we use the principle that it is more important to examine data points that are near boundaries of the current partition than to examine those that are far away; second, for each point that is examined, we use the principle that it is more important to test class labels corresponding to nearby anchor points rather than those of distant anchor points. Accordingly, each iteration performed within the iterative refinement procedure consists of the following sequence of steps: (i) finding data points near boundaries; (ii) finding anchor points in the vicinity of each boundary point; (iii) testing the profitability of reassigning each data point to a nearby anchor; (iv) switching class labels of points under test, if appropriate; and (v) defining an updated set of anchor points based on the new class labels.

The first step of each iteration is therefore to separate all of the data points into those that are interior points and those that are boundary points. For each data point, we make this determination based on the class labels of the m other points closest to it. (Here we assume m is an algorithm parameter that must be specified in advance.) In particular, if all m of the neighboring points have the same class label as the point under test, then this point is defined to be an interior point. Otherwise, the point is defined to be a boundary point. We show examples of interior and boundary points in Figure 4-4. If, after this boundary test, a given data point has been classified as an interior point, then its class label will remain fixed for the current iteration. However, if the point has been classified as a boundary point, then at least one alternative class label will be tested. The set of candidate labels for this point will be precisely those of its m neighboring points; hence, the m nearest neighbors not only indicate whether a particular point is a boundary point, they also give us a convenient way of determining which anchor points are nearby.

Suppose now that $\mathbf{x}_t$ is known to be a boundary point. We wish to determine whether

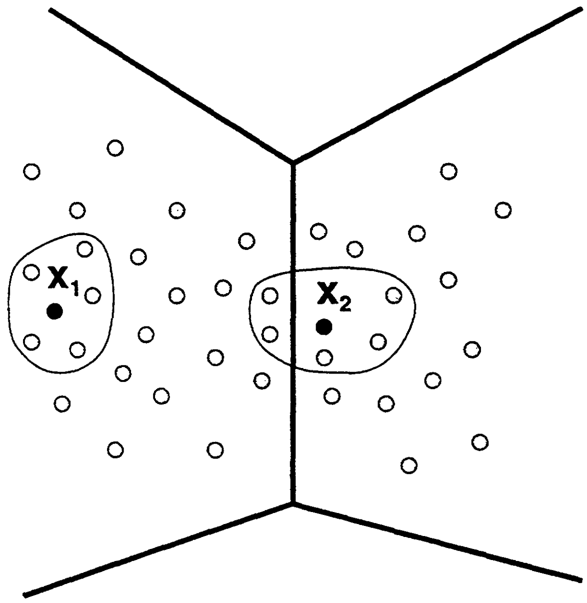


Figure 4-4: Depiction of a portion of the Voronoi partition during an iteration of the algorithm. In this case, the five nearest neighbors of a given point determine whether it is an interior point, such as $\mathbf{x}_1$, or a boundary point, such as $\mathbf{x}_2$.

the value of the objective function would increase or decrease as a result of changing its class label ω_t. We will make this determination under the assumption that the class labels of all data points other than $\mathbf{x}_t$ remain fixed. Under this assumption, the current classification sequence

$$\mathbf{\Omega} = (\omega_0, \omega_1, \cdots, \omega_{t-1}, \omega_t, \omega_{t+1}, \cdots, \omega_{N-1}) \tag{4.12}$$

would change to the new classification sequence

$$\mathbf{\Omega}' = (\omega_0, \omega_1, \cdots, \omega_{t-1}, \omega_t', \omega_{t+1}, \cdots, \omega_{N-1}), \tag{4.13}$$

where ω_t' represents a class label (different from ω_t) of one of the m nearest neighbors of $\mathbf{x}_t$. To calculate the effect of this change on the value of the objective function, we first determine its effect on the marginal pmf $P(\cdot; \mathbf{\Omega})$ and the joint pmf $R(\cdot, \cdot; \mathbf{\Omega})$. Let us express the modified pmfs $P(\cdot; \mathbf{\Omega}')$ and $R(\cdot, \cdot; \mathbf{\Omega}')$ as additively perturbed versions of the original pmfs given by

$$P(i; \mathbf{\Omega}') = P(i; \mathbf{\Omega}) + \Delta P(i; \mathbf{\Omega}, \mathbf{\Omega}') \tag{4.14}$$

and

$$R(i, j; \mathbf{\Omega}') = R(i, j; \mathbf{\Omega}) + \Delta R(i, j; \mathbf{\Omega}, \mathbf{\Omega}') \tag{4.15}$$

for $i, j = 1, 2, \cdots, L$. Then, relative to the original classification sequence, the new classification sequence has one more element with the class label ω_t' and one fewer element with class label ω_t as a result of the change; hence, the perturbation ΔP must have the form

$$\Delta P(i; \boldsymbol{\Omega}, \boldsymbol{\Omega}') = \frac{1}{N}\gamma_i(\omega_t') - \frac{1}{N}\gamma_i(\omega_t). \tag{4.16}$$

Moreover, since the new classification sequence now contains the transitions $(\omega_{t-1}, \omega_t')$ and $(\omega_t', \omega_{t+1})$ but no longer contains the transitions (ω_{t-1}, ω_t) and (ω_t, ω_{t+1}), we conclude that the perturbation ΔR must have the form

$$\begin{aligned}\Delta R(i, j; \boldsymbol{\Omega}, \boldsymbol{\Omega}') = &\frac{1}{N-1}\gamma_{ij}(\omega_{t-1}, \omega_t') + \frac{1}{N-1}\gamma_{ij}(\omega_t', \omega_{t+1}) \\ &- \frac{1}{N-1}\gamma_{ij}(\omega_{t-1}, \omega_t) - \frac{1}{N-1}\gamma_{ij}(\omega_t, \omega_{t+1}).\end{aligned} \tag{4.17}$$

Working with these simple perturbations allows us to quickly assess the effect on the objective function, since it essentially saves us from recalculating the marginal and joint pmfs using the original formulas in (4.4) and (4.5). Once the above perturbations have been applied via (4.14) and (4.15), the value of the objective function $I(\boldsymbol{\Omega}')$ resulting from the change in the classification sequence can now be computed using (4.8) with $\boldsymbol{\Omega}$ replaced by $\boldsymbol{\Omega}'$. The net change in the objective function is given by

$$\Delta I(\boldsymbol{\Omega}, \boldsymbol{\Omega}') = I(\boldsymbol{\Omega}') - I(\boldsymbol{\Omega}). \tag{4.18}$$

If, for a particular boundary point, there is only a single alternative class label to be tested via the above marginal measure, and if this new label would yield an increase in mutual information according to (4.18), then the current label is replaced by the new label; otherwise, the current label is left unchanged. If there are multiple alternative labels to be tested in this way, then the current class label is replaced by the alternative label that yields the largest change in the objective function (provided that this change is greater than zero). The remaining data points are then processed in a similar way. In the descriptions given in Figure 4-5, we refer to boundary points whose labels could be changed to increase mutual information as positive-potential boundary points.

The new class labels for the data points are only tentative, however. This is because the points themselves do not necessarily fall into disjoint Voronoi regions under their present categorization; their new labels merely indicate the general directions in which region boundaries should be modified. Class labels become permanently reassigned through a two-step procedure. First, the anchor points of a new Voronoi partition are defined based on the tentative class labels; then, each data point is once again re-labeled according to its nearest anchor point. To obtain the new value for the anchor point $\mathbf{c}_j$, we compute the centroid of all points $\mathbf{x}_t$ having the class label $\omega_t = j$. That is, $\mathbf{c}_j$ is now given by

$$\mathbf{c}_j = \frac{\sum_{t=0}^{N-1} \gamma(\omega_t)\mathbf{x}_t}{\sum_{t=0}^{N-1} \gamma(\omega_t)}. \tag{4.19}$$

Once these anchors have been computed, each new class label ω_t is calculated using (4.2).

It is hoped that the Voronoi partition resulting from this change will have a greater objective function value than did its predecessor. However, this is not always the case for at least two reasons. First, the decision to change a class label is made individually for each boundary point, rather than jointly over all boundary points; second, the two-stage construction of the partition at the end of the iteration may introduce some error through the approximation of boundary surfaces. (As we will demonstrate in the discussion at the end of the chapter, the final partition boundary is always a portion of a K-dimensional hyperplane under the Voronoi constraint, even though the most recent class label changes may collectively indicate that the boundary should be curved in some way.) Nonetheless, the iterative refinement proceeds as described above until the value of the objective function undergoes a decrease, rather than an increase. When this occurs, the termination procedure is invoked.

ITERATIVE REFINEMENT PROCEDURE

$\boxed{0}$ <u>DESCRIPTION:</u>

Assign initial values for the classification sequence and construct the set of m nearest neighbors for each data point.

<u>OPERATION:</u>

$$\begin{aligned}
\boldsymbol{\Omega} &\longleftarrow \boldsymbol{\Omega}_{\max} \\
\tau_0(t) &\longleftarrow t, \qquad t \in \mathcal{T} \\
\mathcal{T}_i(t) &\longleftarrow \mathcal{T} \setminus \{\tau_0(t), \cdots, \tau_{i-1}(t)\}, \qquad t \in \mathcal{T}; \quad i = 1, 2, \cdots, m \\
\tau_i(t) &\longleftarrow \mathop{\arg\min}_{\tau \in \mathcal{T}_i(t)} \{D(\mathbf{x}_\tau, \mathbf{x}_t)\}, \qquad t \in \mathcal{T}; \quad i = 1, 2, \cdots, m
\end{aligned}$$

$\boxed{1}$ <u>DESCRIPTION:</u>

For each data point, construct the set of class labels associated with its m nearest neighbors. Once this set has been determined, remove from it all class labels having the same value as that of the point itself. Define the set of boundary points to be those data points having at least one neighbor labeled differently from itself.

<u>OPERATION:</u>

$$\begin{aligned}
\mathcal{W}_t^m &\longleftarrow \{\omega_{\tau_1(t)}, \omega_{\tau_2(t)}, \cdots .\omega_{\tau_m(t)}\}, \qquad t \in \mathcal{T} \\
\overline{\mathcal{W}}_t^m &\longleftarrow \{\omega \in \mathcal{W}_t^m \mid \omega \neq \omega_t\} \qquad t \in \mathcal{T} \\
\mathcal{T}_{\mathbf{bdy}} &\longleftarrow \{t \in \mathcal{T} \mid \overline{\mathcal{W}}_t^m \neq \varnothing\}
\end{aligned}$$

$\boxed{2}$ <u>DESCRIPTION:</u>

For each boundary point, test alternative classifications for it and determine which of these would result in the greatest gain in the value of the objective function, assuming that the labels of all other points remain fixed. Define the set of positive-potential boundary points to be those that would increase the current value of the objective function if given the best class label.

<u>OPERATION:</u>

$$\begin{aligned}
\omega_t^* &\longleftarrow \mathop{\arg\max}_{\omega \in \overline{\mathcal{W}}_t^m} \{\Delta I(\boldsymbol{\Omega}, \boldsymbol{\Omega} + (\omega - \omega_t)\boldsymbol{\delta}_t)\}, \qquad t \in \mathcal{T}_{\mathbf{bdy}}; \\
\mathcal{T}_{\mathbf{bdy}}^{+} &\longleftarrow \{t \in \mathcal{T}_{\mathbf{bdy}} \mid \Delta I(\boldsymbol{\Omega}, \boldsymbol{\Omega} + (\omega_t^* - \omega_t)\boldsymbol{\delta}_t) > 0\}.
\end{aligned}$$

Figure 4-5: Description of algorithm for iteratively refining the Voronoi partition. *(Continued on following page.)*

$\boxed{3}$ DESCRIPTION:

Define a new classification sequence by replacing the current class labels associated with positive-potential boundary points with the best alternative class labels.

OPERATION:

$$\boldsymbol{\Omega}' \longleftarrow \boldsymbol{\Omega} + \sum_{t\in\mathcal{T}^{+}_{\mathbf{bdy}}} (\omega_t^{*} - \omega_t)\boldsymbol{\delta}_t$$

$\boxed{4}$ DESCRIPTION:

For each set of data points with a given class label, compute the centroid location and make this the new anchor point for the class.

OPERATION:

$$\mathbf{c}_j \longleftarrow \frac{\sum_{t=0}^{N-1}\gamma_j(\omega'_t)\mathbf{x}_t}{\sum_{t=0}^{N-1}\gamma_j(\omega'_t)}, \qquad j = 1, 2, \cdots, L$$

$\boxed{5}$ DESCRIPTION:

Re-compute the class label for each data point by determining the identity of the nearest anchor point.

OPERATION:

$$\omega'_t \longleftarrow \mathop{\arg\min}_{j\in\{1,2,\cdots,L\}} \{D(\mathbf{c}_j, \mathbf{x}_t)\}, \qquad t \in \mathcal{T}$$

$$\boldsymbol{\Omega}' \longleftarrow (\omega'_0, \omega'_1, \cdots, \omega'_{N-1})$$

$\boxed{6}$ DESCRIPTION:

Check if the new classification sequence yields an overall increase in the value of the objective function. If so, then update the current classification sequence accordingly and perform another iteration; if not, proceed to the termination stage of the algorithm.

OPERATION:

```
if $\Delta I(\boldsymbol{\Omega}, \boldsymbol{\Omega}') > 0$ then
    $\boldsymbol{\Omega} \longleftarrow \boldsymbol{\Omega}'$
    goto $\boxed{1}$
else
    goto $\boxed{\text{TERMINATION PROCEDURE}}$
endif
```

Figure 4-5: *(Continued from previous page.)* Description of iterative refinement algorithm.

4.2.3 Termination of the Iterative Refinement Procedure

The termination procedure is designed to test whether the most recent partition refinement has yielded a local maximum or a global maximum of the objective function. This procedure introduces random perturbations into the current partition as a way of probing the parameter space and testing whether the objective function value can be further increased. If a particular perturbation is successful, the termination procedure then sends control back to the iterative refinement procedure for further adjustments to the partition. Thus, the termination procedure may actually be invoked more than once during a single execution of the overall algorithm. We give a detailed description of this procedure in Figure 4-6.

The procedure is furnished with the identities of the positive-potential boundary points from the most recent partition refinement, as well as the corresponding alternative class labels that have been deemed individually the most profitable. Clearly, when the current labels of all of these points are switched to their best alternatives and the new partition is subsequently specified, the objective function value does not increase; otherwise, the termination procedure would not have been invoked. Therefore, rather than switching the labels of all of these points at once, the termination procedure switches the labels of only a subset of these points; this subset is chosen at random from all possible subsets of the positive-potential boundary points.

If switching the class labels of the particular subset selected does not increase the value of the objective function, then another subset is tried. This procedure continues until either the objective function value is increased or the maximum allowable number of subsets, say J_{term}, is reached. (The number J_{term} is an algorithm parameter that must be specified in advance.) If the maximum number of subsets has been tried and the objective function value has not been increased, the algorithm terminates, and the current partition (i.e., the one supplied to the termination procedure upon invocation) is declared to be the best one.

TERMINATION PROCEDURE

$\boxed{0}$ DESCRIPTION:

Initialize the loop counter.

OPERATION:

$$n \longleftarrow 0$$

$\boxed{1}$ DESCRIPTION:

Select at random a subset of the positive-potential boundary points.

OPERATION:

$$\{k_i\} \longleftarrow \mathrm{RS}(\{0,1\};1), \qquad i = 1, 2, \cdots, |\mathcal{T}^+_{\mathrm{bdy}}|$$

$$k \longleftarrow \sum_{i=1}^{|\mathcal{T}^+_{\mathrm{bdy}}|} k_i$$

$$\overline{\mathcal{T}}^+_{\mathrm{bdy}} \longleftarrow \mathrm{RS}(\mathcal{T}^+_{\mathrm{bdy}}; k)$$

$\boxed{2}$ DESCRIPTION:

Define a new classification sequence by replacing the current class labels associated with the randomly selected subset of positive-potential boundary points with the best alternative class labels.

OPERATION:

$$\boldsymbol{\Omega}' \longleftarrow \boldsymbol{\Omega} + \sum_{t \in \overline{\mathcal{T}}^+_{\mathrm{bdy}}} (\omega_t^* - \omega_t)\boldsymbol{\delta}_t$$

$\boxed{3}$ DESCRIPTION:

Compute the centroid location associated with the points in each class and make this the new anchor point for the class.

OPERATION:

$$\mathbf{c}_j \longleftarrow \frac{\sum_{t=0}^{N-1} \gamma_j(\omega_t')\mathbf{x}_t}{\sum_{t=0}^{N-1} \gamma_j(\omega_t')}, \qquad j = 1, 2, \cdots, L$$

Figure 4-6: Description of algorithm to terminate the iterative refinement procedure. *(Continued on following page.)*

$\boxed{4}$ DESCRIPTION:

Re-compute the class label for each data point by determining the identity of the nearest anchor point.

OPERATION:

$$\omega'_t \longleftarrow \mathop{\arg\min}_{j\in\{1,2,\cdots,L\}} \{D(\mathbf{c}_j,\mathbf{x}_t)\}, \qquad t\in\mathcal{T}$$

$$\boldsymbol{\Omega}' \longleftarrow (\omega'_0,\omega'_1,\cdots,\omega'_{N-1})$$

$\boxed{5}$ DESCRIPTION:

Check if the new classification sequence yields an increase in the value of the objective function. If so, then update the current classification sequence accordingly and return to the iterative refinement procedure; if not, then increment the loop counter and test another subset of positive-potential boundary points.

OPERATION:

```
if $\Delta I(\boldsymbol{\Omega},\boldsymbol{\Omega}') > 0$ then
    $\boldsymbol{\Omega} \longleftarrow \boldsymbol{\Omega}'$
    goto $\boxed{\text{ITERATIVE REFINEMENT PROCEDURE}}$
else
    $n \longleftarrow n+1$
    if $n = J_{\text{term}}$ then
        end
    else
        goto $\boxed{1}$
    endif
endif
```

Figure 4-6: *(Continued from previous page.)* Description of termination algorithm.

4.3 Estimation of Optimal HMM Parameters

Once a suitable state-space partition has been found using the three-stage algorithm described above, we can compute estimates of the HMM parameter values. In particular, we need to estimate the initial state probabilities and state transition probabilities for the Markov chain as well as the means, standard deviations, and weighting coefficients for the Gaussian-mixture densities associated with the states of the chain. Since the estimation of Markov chain parameters is decoupled from the estimation of output density parameters, the algorithms used to generate these estimates can be executed in any order.

Let us denote by Ψ^* the tuple of final class labels for the data points $\{\mathbf{x}_0, \cdots, \mathbf{x}_{N-1}\}$. To estimate the Markov chain parameters, we use the empirical formulas given in (4.4) and (4.5), as was done during the search for the best partition. However, an additional processing step is now required to obtain the final estimates, namely the conversion of the joint state probabilities $\{R(i,j;\Psi^*)\}_{i,j=1}^{L}$ to the state transition probabilities $\{Q(i,j;\Psi^*)\}_{i,j=1}^{L}$. This step can be carried out via the formula

$$Q(i,j;\Psi^*) = \frac{R(i,j;\Psi^*)}{\sum_{k=1}^{L} R(i,k;\Psi^*)}, \tag{4.20}$$

which normalizes the rows of the array R such that each row becomes a valid pmf.

Estimating the densities associated with the states of the Markov chain is somewhat more involved. For this problem, we use the procedure presented in Figure 4-7, whose computational structure is based on the EM principle. This procedure must be applied separately to the data points in each region of the optimal partition to estimate all L densities. It need not be applied directly to the data points in $\mathbb{R}^K$, however. If we wish to build an HMM having scalar-valued output (so as to approximate a sequence of signal values rather than state vector values), we can simply extract the first element of each state-vector realization in a given Voronoi region and proceed to estimate a univariate output pdf based on these scalar measurements. We will concentrate on this univariate case here.

To estimate a given output density, the algorithm must be supplied with initial estimates of all of the Gaussian-mixture parameters. To generate these initial estimates, we could, for example, try many different randomly selected sets of mixture parameters and then use the particular set that yields the highest likelihood value. After the algorithm has been initialized, a single iteration then proceeds as follows. The new estimate for the weighting coefficient associated with the jth mixture component is defined to be the average posterior probability that each observation was produced by component j. The new estimate for the mean of the jth mixture component is defined to be a weighted average of the observed samples, where the weight placed on the tth sample is proportional to the posterior probability that this sample was generated by component j. The new estimate for the variance of the jth component is a weighted average of the squares of the observations (after the previously computed estimate of the mean of the jth component has been subtracted out); the weights used in this calculation are precisely the same as those used to update the jth mean. This iterative updating procedure continues until there is a negligible change in the tuple of estimated parameter values from one iteration to the next.

PDF ESTIMATION PROCEDURE

0 DESCRIPTION:

Assign initial values to the means, standard deviations, and weighting coefficients of the M-component Gaussian mixture. In addition, assign values to the N observed samples.

OPERATION:

$$\begin{aligned}
\rho_j &\longleftarrow \rho_j^{(0)}, \qquad j = 1, 2, \cdots, M \\
\mu_j &\longleftarrow \mu_j^{(0)}, \qquad j = 1, 2, \cdots, M \\
\sigma_j &\longleftarrow \sigma_j^{(0)}, \qquad j = 1, 2, \cdots, M \\
\mathbf{\Psi}' &\longleftarrow (\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\rho}) \\
y_t &\longleftarrow y_t^{(0)}, \qquad t = 0, 1, \cdots, N-1
\end{aligned}$$

1 DESCRIPTION:

Using the current pdf parameter estimates, compute the posterior probability that the tth observation came from the jth mixture component.

OPERATION:

$$P_{t,j} \longleftarrow \frac{\frac{\rho_j}{\sqrt{2\pi}\sigma_j} \exp\left\{-\frac{(y_t-\mu_j)^2}{2\sigma_j^2}\right\}}{\sum_{k=1}^{M} \frac{\rho_k}{\sqrt{2\pi}\sigma_k} \exp\left\{-\frac{(y_t-\mu_k)^2}{2\sigma_k^2}\right\}}, \qquad \begin{array}{l} t = 0, 1, \cdots, N-1 \\ j = 1, 2, \cdots, M \end{array}$$

Figure 4-7: EM algorithm for estimating parameters of HMM output pdf. *(Continued on following page.)*

$\boxed{2}$ DESCRIPTION:

Update the pdf parameter estimates using the posterior probabilities just computed. Define the weighting coefficient for the jth mixture component to be the arithmetic average of all posterior probabilities associate with that component. Define the jth mean to be a weighted average of the observed values, where the weights are the posterior probabilities associated with the jth mixture component. Using these same weighting terms, define the jth variance to be a weighted average of the squares of the mean-adjusted observed values.

OPERATION:

$$\rho_j \longleftarrow \frac{1}{N}\sum_{t=0}^{N-1} P_{t,j}, \qquad j = 1, 2, \cdots, M$$

$$\mu_j \longleftarrow \frac{\sum_{t=0}^{N-1} P_{t,j} y_t}{\sum_{t=0}^{N-1} P_{t,j}}, \qquad j = 1, 2, \cdots, M$$

$$\sigma_j \longleftarrow \sqrt{\frac{\sum_{t=0}^{N-1} P_{t,j}(y_t - \mu_j)^2}{\sum_{t=0}^{N-1} P_{t,j}}}, \qquad j = 1, 2, \cdots, M$$

$$\Psi \longleftarrow (\mu, \sigma, \rho)$$

$\boxed{3}$ DESCRIPTION:

Check if the distance between the current and previous parameter vector estimates is below a predetermined tolerance level. If so, then terminate the algorithm. If not, then save the current estimate and return to step 1 and perform another iteration.

OPERATION:

```
if $D(\Psi, \Psi') < T$ then
        end
else
        $\Psi' \longleftarrow \Psi$
        goto $\boxed{1}$
endif
```

Figure 4-7: *(Continued from previous page.)* EM algorithm for estimating parameters of HMM output pdf.

4.4 Discussion

4.4.1 Prior Work in HMM Parameter Identification

A number of algorithms have been developed by other researchers for estimating the parameters of an HMM under various assumptions, including a key assumption that we have used here, namely that the output pdf for each state of the HMM is a Gaussian mixture. The most popular of these existing algorithms is referred to as the Baum-Welch algorithm, which was originally developed and analyzed in a series of papers by Baum et al [20, 22, 23]. This technique is now recognized as an implementation of the EM algorithm; it has been used extensively in the construction of HMM-based phonetic models for modern speech recognition systems [78, 84, 154]. Alternative algorithms have also been developed within the speech processing community; these include, most notably, the gradient-based algorithm developed by Levinson et al [109]. However, none of the existing techniques just mentioned can be used to solve the HMM-based source identification problem as we have defined it here, for these algorithms have been designed to optimize a likelihood-based criterion rather than a mutual information criterion; moreover, they are not equipped to handle the critical state-space partitioning constraint, and they therefore generally produce HMM-based approximations which lie outside of the set of valid solutions defined in the chapter introduction.

4.4.2 Quality Assessment for the Initial Partition

We note that the random selection approach that we have used in the initialization procedure described in Section 4.2.1 allows us to assess, in a probabilistic sense, the quality of our initial partition relative to all possible initial partitions. Specifically, observe that each of the L-element subsets of $\{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{N-1}\}$ specifies a unique initial partition (provided that the N original data points are distinct), and that each partition can in turn be evaluated using its associated mutual information. In fact, these subsets can be arranged in ascending order according their objective function values. It is straightforward to show that a randomly chosen subset among these ordered subsets has an objective function value in the (100α)th percentile with probability $1-\alpha$, where $0<\alpha<1$. Furthermore, if J_{init} of the subsets are chosen at random, then the subset among these with the largest objective function value lies in the (100α)th percentile with probability $1-\alpha^{J_{\text{init}}}$. Therefore, if we wish to know, for example, the smallest number of initial subsets that should be tested so that the subset with the largest objective function value is in the 95th percentile with probability 0.9 or higher, we need only solve

$$J_{\text{init}} = \min\left\{J \in \{1,2,3,\cdots\} \mid 1-0.95^J > 0.9\right\} \tag{4.21}$$

The above minimization yields a value of $J = 29$; thus, in this case we would find the best of 29 L-element subsets chosen at random from the N given data points. Alternatively, rather than starting with a required exceedance probability, we might instead place a restriction on the amount of computation we wish to perform in the initialization procedure. Such a restriction would translate directly into a bound on J_{init}. With this bound, we could then assess the quality of our partition by computing performance values represented by the pair $(100\alpha, 1 - \alpha^{J_{\text{init}}})$ for $0 < \alpha < 1$. A similar quantitative analysis to the one just described would also apply to the termination procedure, since this procedure also makes use of the random subset selection technique.

4.4.3 Implicit Constraints Imposed by the Voronoi Assumption

Our decision to use the special Voronoi construction for our state-space partition was made in order to minimize computational complexity and memory consumption. But this decision actually places a severe constraint on the form of an admissible region within a partition. In particular, any Voronoi region is inherently a convex set; hence, as we will demonstrate below, the region boundaries in such a partition must always be planar (i.e., portions of K-dimensional hyperplanes). The cost incurred (in terms of sacrificed approximation quality) as a result of using the Voronoi structure, rather than a more general partition structure which could model curved region boundaries, is unknown and may be extremely difficult to measure.

The convexity property of a Voronoi region can be derived directly from its definition. To see this, suppose the point $\mathbf{x} \in \mathbb{R}^K$ is known to be an element of the Voronoi region $\mathcal{R}_j$, so that it satisfies the distance inequalities

$$D(\mathbf{x}, \mathbf{c}_j) \leq D(\mathbf{x}, \mathbf{c}_i), \qquad i = 1, 2, \cdots, L. \tag{4.22}$$

Upon squaring both sides of each inequality, the entire set of L inequalities continues to hold and can be expressed in the form

$$(\mathbf{x} - \mathbf{c}_j)^T(\mathbf{x} - \mathbf{c}_j) \leq (\mathbf{x} - \mathbf{c}_i)^T(\mathbf{x} - \mathbf{c}_i), \qquad i = 1, 2, \cdots, L. \tag{4.23}$$

Though at first it may appear that the above inequalities are quadratic in $\mathbf{x}$, in fact the terms of second order cancel each other; thus, the expressions are actually linear in $\mathbf{x}$ and can, after considerable algebraic manipulation, ultimately be written as

$$\frac{\left(\mathbf{x} - \frac{1}{2}(\mathbf{c}_i + \mathbf{c}_j)\right)^T (\mathbf{c}_j - \mathbf{c}_i)}{\|\mathbf{c}_j - \mathbf{c}_i\|} \geq 0, \qquad i = 1, 2, \cdots, L. \tag{4.24}$$

In Figure 4-8, we give a geometric interpretation (in two-dimensional space) of a typical inequality from this latter set of L inequalities. Observe that the ith inequality above is actually imposed on the inner product of two vectors, namely $\overline{\mathbf{x}} = \mathbf{x} - \frac{1}{2}(\mathbf{c}_i + \mathbf{c}_j)$, which is simply a re-expression of the vector $\mathbf{x}$ relative to the midpoint between $\mathbf{c}_i$ and $\mathbf{c}_j$, and $\overline{\mathbf{c}} = (\mathbf{c}_j - \mathbf{c}_i)/\|\mathbf{c}_j - \mathbf{c}_i\|$, which is the unit vector that points in the direction from $\mathbf{c}_i$ to $\mathbf{c}_j$. The inequality itself implies that only those points $\mathbf{x} \in \mathbb{R}^K$ yielding an inner product that is either positive or zero can lie in the region $\mathcal{R}_j$. In other words, each of the inequalities above, with the exception of the trivial one in which $i = j$, can be thought of as representing a closed half-space in K dimensions. The hyperplane forming the boundary of this half-space is the plane that bisects the line segment connecting $\mathbf{c}_i$ and $\mathbf{c}_j$. It follows that if we take all of the (nontrivial) inequalities simultaneously, we have a new representation of the Voronoi

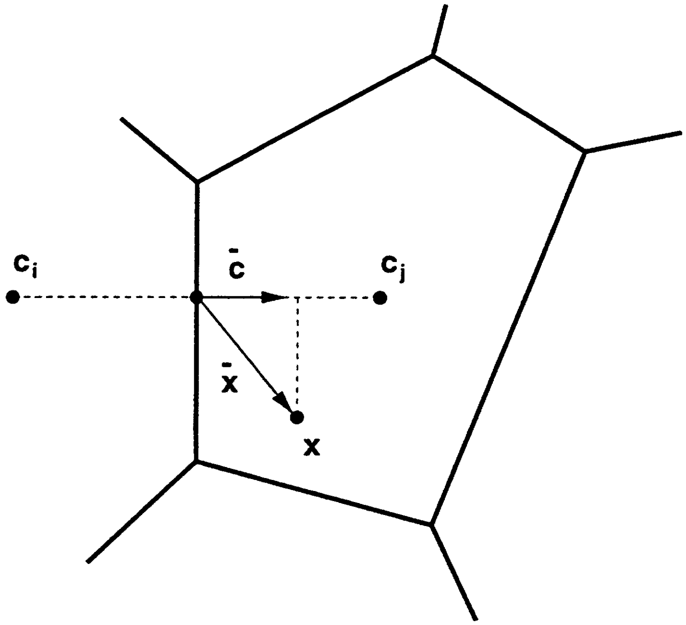


Figure 4-8: Representation of a Voronoi region as an intersection of many half-spaces. The inner product of $\bar{\mathbf{x}}$ and $\bar{\mathbf{c}}$ must be positive for any point $\mathbf{x}$ in the region $\mathcal{R}_j$.

region $\mathcal{R}_j$ as the intersection of $L-1$ half-spaces; hence, the bounding surface enclosing the Voronoi region is formed from portions of the associated $L-1$ bounding hyperplanes. Since the line segment connecting any two points in a closed half-space is itself contained in the half-space, we have that a closed half-space is a convex set. Finally, since the intersection of a finite number of convex sets is a convex set, it follows that a Voronoi region is convex.

4.4.4 Assumptions on the HMM State Output Densities

As part of the HMM source identification algorithm described in this chapter, we assumed that the output pdf associated with each state of the HMM was a Gaussian mixture. Of course, we know that this assumption cannot be true in general, since the region of support for a Gaussian-mixture pdf is all of $\mathbb{R}$, whereas the actual projection of a Voronoi region onto the real line is often a bounded interval. However, in practice this violation of the assumption does not typically cause any difficulties. In addition, as we will discover in Chapter 5, the Gaussian-mixture assumption is convenient not only because it can be specified by a small number of parameters, but also because it offers several practical advantages in the signal estimation problem and other related problems.

Furthermore, we can use the EM-based iterative algorithm presented in Section 4.3 to estimate the parameters of such a density; this algorithm is very easy to implement and is reasonably efficient. Other algorithms (many of them similar to the one described in Figure 4-7) for solving this estimation problem have been proposed independently in several different contexts and could also be used [3, 77, 136, 191]. For situations in which the Gaussian-mixture assumption does not provide an adequate representation of the true signal, we can choose from a wide range of other, more general density estimation methods (see, for example, [104, 129, 140, 159, 203, 214, 218, 223, 224] for a number of classical approaches to density estimation, or [170, 178, 198, 199] for an overview of more modern techniques.

4.4.5 Open Issues in HMM Parameter Identification

For the purpose of developing our source identification algorithm, we assumed that the parameters $K, L, M_1, M_2, \cdots, M_L$ were given. In practice, however, the values of these parameters are not typically known and therefore must be estimated. More precisely, we must estimate K, which is truly an unknown parameter of the signal, and we must select reasonable values for the remaining parameters, which are merely being used to approximate the signal. As we mentioned in Chapter 2, several methods already exist for estimating the autoregressive order, K, even in cases where the signal has been generated by a nonlinear system. However, there appear to be no clear guidelines for choosing values of the remaining model parameters. Undoubtedly, the most critical of these remaining parameters is the HMM order, L, since this parameter is the only one that affects the dynamical structure of the approximation. In general, the most appropriate choice for L will depend on the specific signal processing task for which the HMM-based approximation will be used. A potentially rich area for future research is to develop a technique for optimally selecting the HMM order, a priori, based on a description of the signal processing task, so that a tedious process of trial and error can be avoided.

Chapter 5

Using Finite-State Markov Models for Signal Estimation

5.1 Introduction

In Chapter 4, we developed a set of practical numerical techniques for performing source identification based on the finite-state signal model introduced earlier. However, these techniques constitute only a part of our overall finite-state signal processing framework as defined in the early portion of the thesis. To complete the framework, we now turn our attention to the inference problem that we have not yet addressed, namely the problem of signal estimation. In the next two subsections, we give some assumptions and notation that will be used in connection with the signal estimation problem, and we provide a concise formulation of the estimation problem itself. In the third subsection, we describe how the remaining material in the chapter is organized.

5.1.1 Preliminary Assumptions and Notation

As usual, we will assume that $\{Y_t\}$ is a stationary signal of interest, and that our corrupted observation of this signal, $\{Z_t\}$, consists of samples defined by

$$Z_t = Y_t + V_t, \tag{5.1}$$

where $\{V_t\}$ is a stationary noise process which is statistically independent of $\{Y_t\}$. Throughout the chapter, we will assume that $\{Y_t\}$ is the output of an L-state HMM whose state space is, without loss of generality, the set of integers $\{1, 2, \cdots, L\}$. The underlying Markov chain from which $\{Y_t\}$ is generated will be denoted by $\{\Theta_t\}$; as in previous chapters, the initial state probabilities and state transition probabilities of this Markov chain will be denoted, respectively, by $\{P(i)\}_{i=1}^{L}$ and $\{Q(i,j)\}_{i,j=1}^{L}$. We denote the output densities of the HMM by $\{g_i(\cdot)\}_{i=1}^{L}$. Although these output densities can in general be arbitrarily complicated functions, in the sequel we will assume that the ith density, $g_i(\cdot)$, is a Gaussian mixture made up of M_i constituent elements, and is defined by

$$g_i(y) = \sum_{k=1}^{M_i} \rho_{ik} \mathcal{N}(y; \mu_{ik}, \sigma_{ik}), \qquad -\infty < y < \infty. \tag{5.2}$$

Because we will be examining several different types of signal estimation problems, the definition of the additive noise process $\{V_t\}$ will be modified as appropriate as we progress through the chapter. In all cases, however, this noise process will also be viewed as the output of a finite-state HMM. In most of the cases considered, $\{V_t\}$ will be made up of i.i.d. random variables; hence, the HMM representing $\{V_t\}$ will be degenerate, i.e., it will consist of only a single state. When it becomes necessary to refer to the temporal dynamics of the noise, we will use the notation $\{\Theta'_t\}$ to represent the underlying Markov chain from which the noise samples are generated. We will assume that this Markov chain has L' states, given by $\{1, 2, \cdots, L'\}$, and that the initial state probabilities and state transition probabilities for this chain are given by $\{P'(i)\}_{i=1}^{L'}$ and $\{Q'(i,j)\}_{i,j=1}^{L'}$, respectively. As in the HMM-based representation of the signal, each of the output densities will be taken to be a Gaussian mixture. The ith output density, which we denote by $g'_i(\cdot)$ and which is assumed to consist of M'_i Gaussian components, is defined by

$$g'_i(v) = \sum_{k=1}^{M'_i} \rho'_{ik} \mathcal{N}(v; \mu'_{ik}, \sigma'_{ik}), \qquad -\infty < v < \infty. \tag{5.3}$$

5.1.2 Problem Statement and Approach to Solution

We assume throughout the chapter that only the finite-length portion of the observed signal $\{Z_t\}$ between $t = 0$ and $t = N - 1$ is available for estimating signal values of interest. When given a realization of the random vector $\mathbf{Z}_{0:N-1}$, our goal is to attempt to determine the value that has been taken by the underlying signal vector $\mathbf{Y}_{0:N-1}$. More precisely, our objective is to obtain, for each signal value y_t ($t = 0, 1, \cdots, N-1$), the MMSE estimate $\hat{y}_t(\mathbf{z}_{0:N-1})$ defined by

$$\hat{y}_t(\mathbf{z}_{0:N-1}) = \arg\min_{y(\cdot)\in\mathcal{Y}} E\left\{(Y_t - y(\mathbf{z}_{0:N-1}))^2 | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\right\}, \tag{5.4}$$

where $\mathcal{Y}$ represents the set of all real-valued functions of a real N-dimensional argument.[1] This type of estimation problem is commonly referred to as a smoothing problem, and its solution is typically termed an optimal smoother [14, 164, 193, 197]. For this problem, all parameter values characterizing both the signal and the noise are assumed to be precisely known.

As we have already pointed out in earlier chapters, the solution to (5.4) is given by the

[1] To simplify notation in the remainder of the chapter, we will often suppress the argument in the functional expression $\hat{y}_t(\mathbf{z}_{0:N-1})$ and use the abbreviated symbol $\hat{y}_t$ to refer to the estimate.

conditional expectation

$$\hat{y}_t = E\{Y_t | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\}. \tag{5.5}$$

However, from the standpoint of generating a specific numerical estimate of the signal based on our observations, this symbolic expression is not immediately helpful. Rather, it provides only a starting point from which we may ultimately derive a more concrete solution. Much of the material this chapter is aimed at exploiting the stochastic structure of the signal and noise outlined above in order to develop computationally efficient techniques for evaluating the right-hand side of (5.5).

5.1.3 Chapter Organization

The chapter is organized in the following way. First, we examine the form of the above conditional expectation in the case where the corrupting noise is white and Gaussian, and we describe an efficient recursive algorithm for evaluating the expectation in this simple case. We then analyze the various components of this estimation algorithm to determine the approximate number of arithmetic operations required to generate the final signal estimate. Next, we compare the estimation performance achieved using several different HMM-based representations of a stationary AR Gaussian signal, so that we can quantify the improvement in estimator quality as a function of the number of states in the HMM and as a function of the SNR. We then extend the basic estimation algorithm developed for the case in which the noise is white and Gaussian to more complex cases in which the noise is allowed to be both non-Gaussian and non-white. We also demonstrate how these algorithms can be configured to perform signal separation, i.e., the estimation of each of several statistically independent non-Gaussian signals that have been additively combined. Finally, we describe how the finite-state modeling paradigm can be effectively applied to signal processing problems other than smoothing, including the problems of filtering, prediction, and multi-class hypothesis testing. We also suggest a general method for applying the finite-state framework to the problem of signal estimation in non-stationary noise.

5.2 Estimating a Signal in Additive White Gaussian Noise

5.2.1 Characterization of Signal, Noise, and Observation

We begin by examining the simplest possible case in which the noise process $\{V_t\}$ is made up of zero-mean i.i.d. Gaussian random variables. This constraint on $\{V_t\}$ implies that it can be represented by a one-state HMM whose output pdf is a one-component Gaussian mixture. In keeping with the notation established in (5.3), we define the pdf for each noise sample V_t by

$$g'_1(v) = \mathcal{N}(v; 0, \sigma'_{11}). \tag{5.6}$$

From this definition of the additive noise process, it should be immediately evident that the observed process $\{Z_t\}$ is itself the output of an HMM whose structure is very similar to the HMM for $\{Y_t\}$. In particular, because the noise is white and therefore contributes no additional complexity to the temporal structure of the observation when it is combined with the signal, the initial state probabilities and state transition probabilities in the HMM for $\{Z_t\}$ are exactly the same as those in the HMM for $\{Y_t\}$. In fact, the only difference between these two HMMs lies in the output densities associated with the states of their respective Markov chains. Fortunately, we can easily derive the output pdf for the observed signal when its Markov chain is in state i. Since the signal and noise are statistically independent, this output pdf, which we denote by $h_i(\cdot)$, is obtained by convolving the signal and noise output densities $g_i(\cdot)$ and $g'_1(\cdot)$, and hence is given by

$$\begin{aligned} h_i(z) &= g_i(z) * g'_1(z) && (5.7)\\ &= \sum_{k=1}^{M_i} \rho_{ik}\mathcal{N}\left(z; \mu_{ik}, \sqrt{\sigma_{ik}^2 + \sigma_{11}'^2}\right), \qquad -\infty < z < \infty. && (5.8) \end{aligned}$$

5.2.2 Decomposition of the Conditional Signal Mean

Having established the stochastic structure for the signal, the noise, and the observation in this case, let us now take a closer look at the conditional expectation in (5.5), and attempt to decompose it into more manageable pieces. We begin by expressing this expectation in the form of an integral, and we then introduce further conditioning on possible values of the discrete state variable for the underlying Markov chain. This yields

$$\begin{aligned} &E\{Y_t|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\}\\ &\quad= \int_{-\infty}^{\infty} y_t f_{Y_t|\mathbf{Z}_{0:N-1}}(y_t|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1})\, dy_t && (5.9)\\ &\quad= \int_{-\infty}^{\infty} y_t \left[\sum_{i=1}^{L} f_{Y_t,\Theta_t|\mathbf{Z}_{0:N-1}}(y_t, i|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1})\right] dy_t && (5.10)\\ &\quad= \int_{-\infty}^{\infty} y_t \left[\sum_{i=1}^{L} \Pr\{\Theta_t = i|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\}\cdot\right.\\ &\qquad\qquad \left. f_{Y_t|\Theta_t,\mathbf{Z}_{0:N-1}}(y_t|\Theta_t = i, \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1})\right] dy_t. && (5.11) \end{aligned}$$

We can now simplify this last expression somewhat by observing that

$$\begin{aligned} &f_{Y_t|\Theta_t,\mathbf{Z}_{0:N-1}}(y|\Theta_t = i, \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1})\\ &\qquad\qquad = f_{Y_t|\Theta_t,Z_t}(y|\Theta_t = i, Z_t = z_t), \qquad \infty < y < \infty. && (5.12) \end{aligned}$$

This equality follows directly from the properties of our HMM-based signal model, for if we are given the true value of the underlying state variable Θ_t, then the quantity Y_t is independent of all other signal variables $\{Y_s|s \neq t\}$, and therefore (owing to the independence of the additive noise) is independent of all other observations $\{Z_s|s \neq t\}$. Indeed, when we are given the value of Θ_t, the only observed signal variable that contains any information about Y_t is Z_t itself. If we now introduce this simplification into (5.11), and reverse the order of integration and summation, we can write

$$\begin{aligned}
&E\{Y_t \,|\, \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\} \\
&\quad = \sum_{i=1}^{L} \Pr\{\Theta_t = i | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\} \int_{-\infty}^{\infty} y_t f_{Y_t|\Theta_t,Z_t}(y_t|\Theta_t = i, Z_t = z_t)\, dy_t \\
&\quad = \sum_{i=1}^{L} \Pr\{\Theta_t = i | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\} E\{Y_t|\Theta_t = i, Z_t = z_t\}. \qquad (5.13)
\end{aligned}$$

Now recall that when $\Theta_t = i$, the conditional density characterizing Y_t is a Gaussian mixture made up of M_i Gaussian components. Thus, we can think of the process of generating the output Y_t in two separate stages, where the first stage consists of selecting at random (according to the pmf constructed from the weighting coefficients) exactly one of the M_i Gaussian components from the mixture, and the second stage consists of generating a realization according to the selected Gaussian pdf. Based on this two-stage output-generation paradigm, we can introduce an additional discrete state variable to keep track of the Gaussian component that has been selected at time t. Let us denote this state variable by Φ_t. Then the expectation appearing on the right-hand side of (5.13) can be decomposed further by conditioning on all possible outcomes for Φ_t. This yields the new expression

$$\begin{aligned}
&E\{Y_t|\Theta_t = i, Z_t = z_t\} \\
&\quad = \sum_{j=1}^{M_i} \Pr\{\Phi_t = j|\Theta_t = i, Z_t = z_t\} E\{Y_t|\Theta_t = i, \Phi_t = j, Z_t = z_t\}. \qquad (5.14)
\end{aligned}$$

Upon combining (5.14) and (5.13), we find that the optimal estimate of Y_t given $\mathbf{Z}_{0:N-1}$ can be written as

$$\begin{aligned}
\hat{y}_t = &\sum_{i=1}^{L} \Pr\{\Theta_t = i | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\} \cdot \\
&\sum_{j=1}^{M_i} \Pr\{\Phi_t = j|\Theta_t = i, Z_t = z_t\} E\{Y_t|\Phi_t = j, \Theta_t = i, Z_t = z_t\}. \qquad (5.15)
\end{aligned}$$

Although this estimate now appears to have a more complex structure than it did originally in (5.5), it is nonetheless in a form that is much more amenable to the development of an optimal estimation algorithm, as we will see in the coming sections.

5.2.3 Analysis of Components of the Optimal Estimate

Note that the right-hand side of (5.15) is made up of three basic types of components: (i) posterior state probabilities associated with the underlying Markov chain, which have the form $\Pr\{\Theta_t = i|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\}$; (ii) posterior sub-state probabilities associated with the Gaussian-mixture pdf for a particular state, which have the form $\Pr\{\Phi_t = j|\Theta_t = i, Z_t = z_t\}$; and (iii) expectations conditioned on state and sub-state outcomes, which have the form $E\{Y_t|\Phi_t = j, \Theta_t = i, Z_t = z_t\}$. We shall refer to these quantities as terms of type I, type II, and type III, respectively. In the remainder of this subsection, we will examine and attempt to evaluate each of these terms, beginning with terms of type III and working in reverse order until we finally consider terms of type I.

To begin, observe that if we know the events $\Theta_t = i$ and $\Phi_t = j$ have occurred, then the conditional pdf for Y_t is purely Gaussian. Moreover, if we know that $Z_t = z_t$, then, although this additional knowledge may affect the parameters of the conditional pdf, the pdf itself remains Gaussian, since Y_t and Z_t are (conditionally) jointly Gaussian by assumption. It follows that a term of type III is linear (or, more correctly, affine) in the observed quantity z_t. Specifically, for such a term we can write

$$E\{Y_t|\Theta_t = i, \Phi_t = j, Z_t = z_t\} = \frac{\sigma_{ij}^2}{\sigma_{ij}^2 + \sigma_{11}'^2}(z_t - \mu_{ij}). \tag{5.16}$$

Let us next consider terms of type II. Note that a term of this kind is merely the posterior probability that the Gaussian-mixture state variable Φ_t has taken the value j given that the Markov chain is currently in state i and the current value of our observation is z_t. This probability can be expressed via Bayes' rule as

$$\begin{aligned}\Pr\{\Phi_t = j|\Theta_t = i, Z_t = z_t\} &= \frac{\Pr\{\Phi_t = j|\Theta_t = i\} f_{Z_t|\Theta_t,\Phi_t}(z_t|\Theta_t = i, \Phi_t = j)}{f_{Z_t|\Theta_t}(z_t|\Theta_t = i)} \\ &= \frac{\rho_{ij}\mathcal{N}\left(z_t; \mu_{ij}, \sqrt{\sigma_{ij}^2 + \sigma_{11}'^2}\right)}{\sum_{k=1}^{M_i} \rho_{ik}\mathcal{N}\left(z_t; \mu_{ik}, \sqrt{\sigma_{ik}^2 + \sigma_{11}'^2}\right)}.\end{aligned} \tag{5.17}$$

In this form, the posterior sub-state probabilities can now be easily computed in terms of known model parameters.

The only remaining quantities that must be computed in order to produce the optimal estimate are terms of type I. It turns out, however, that these terms are the most difficult of all three types to evaluate, owing to the fact that they depend on the entire observed sequence $\mathbf{z}_{0:N-1}$, rather than only on a single observed sample, as did terms of type II and type III. A term of type I can still be calculated efficiently, but the procedure for performing this calculation is rather involved. A description of this procedure can be found in Appendix F.

5.3 Analysis of Computation for HMM-Based Estimation

In this section, our goal is to determine the amount of computation required to generate an estimate with the formula in (5.15). For the analysis presented here, we assume for convenience that the Gaussian-mixture pdf assigned to each state has a fixed number of components, say M, rather than a state-dependent number. We wish to derive an expression for the total computational cost as a function of the model parameters M and L, and as a function of the number of samples N in the given realization $\mathbf{z}_{0:N-1}$. To accomplish this, we first determine the cost incurred in producing individual terms of types I, II, and III, and we then quantify the amount of computation required to combine these terms when forming the final estimate. Throughout the following analysis, we assume that the basic arithmetic operations of addition, subtraction, multiplication, and division all require the same number of primitive computer instructions; thus, any such operation will be assigned a single unit of computational cost.

It turns out that the components of the estimation formula above are listed in order of decreasing complexity. Thus, let us consider these components in reverse order, beginning with terms of type III. Recall that any term of type III has the form

$$E\{Y_t | \Phi_t = j, \Theta_t = i, Z_t = z_t\} = \mu_{ij} + \frac{\sigma_{ij}^2}{\sigma_{ij}^2 + \sigma_{11}'^2}(z_t - \mu_{ij}). \tag{5.18}$$

Although this expression contains certain quantities that could, in order to reduce computational expense, be computed and stored in advance for use during the algorithm, its most important attribute from a computational standpoint is that it requires a fixed number of arithmetic operations (let us say J operations altogether), independent of the model parameters L and M.

A term of type II has the form

$$\Pr\{\Phi_t = j | \Theta_t = i, Z_t = z_t\} = \frac{\rho_{ij}\mathcal{N}\left(z_t; \mu_{ij}, \sqrt{\sigma_{ij}^2 + \sigma_{11}'^2}\right)}{\sum_{k=1}^{M_i} \rho_{ik}\mathcal{N}\left(z_t; \mu_{ik}, \sqrt{\sigma_{ik}^2 + \sigma_{11}'^2}\right)}. \tag{5.19}$$

These posterior probabilities, which are derived using Bayes' rule, can be constructed during the estimation algorithm by normalizing a collection of M weighted Gaussian pdf values so that they sum to unity. If we assume that each evaluation of a Gaussian pdf consumes G arithmetic operations, then computing all M of the unnormalized likelihood values would take $M(G+1)$ multiplications. Now suppose that, once these values have been computed, they can be stored in memory temporarily until all of them can be appropriately scaled. Computing the normalizing denominator in the above expression then requires $M-1$ additions, and the subsequent scaling of the original values requires M divisions, bringing the total number of operations performed to $M(G+3)-1$. However, we can view these operations as being distributed over all M terms; hence, in order to evaluate a single term of type II, we need essentially $G+3$ operations.

Let us finally determine the amount of computation needed to evaluate terms of type I. Such terms are computed during the estimation algorithm using special recursive formulas (derived in Appendix F), and are therefore inherently more complicated to analyze. A term of type I is given by

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{L} \alpha_t(j)\beta_t(j)}. \tag{5.20}$$

where $\alpha_t(\cdot)$ and $\beta_t(\cdot)$ are the recursively computed forward and backward variables. The above expression represents the normalization of L likelihood values, one for each state of the underlying Markov chain. By the same reasoning we used earlier, we see that evaluating all L of the probabilities requires L initial multiplications to obtain the unnormalized likelihood values, then $L-1$ additions to compute the scaling term, and finally L divisions to perform the normalization, for a total of $3L-1$ operations. Because these operations are distributed over the L terms, each term requires essentially 3 operations after $\alpha_t(\cdot)$ and $\beta_t(\cdot)$ have been computed.

But we must now turn our attention to the evaluation of these forward and backward variables. First, we observe (see Appendix F for details) that the expression for the particular value $\alpha_t(i)$ is given by

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{L} Q(i,j)\alpha_t(j) \right] h_i(z_t). \tag{5.21}$$

Since each term in the bracketed summation requires one multiplication, and there are L terms in all, the sum itself requires L multiplications and $L-1$ additions, for a total of $2L-1$ operations. After the sum has been computed, it is multiplied by an M-component Gaussian-mixture pdf value, which consumes GM operations (plus one operation for the subsequent multiplication). Thus, to evaluate the single quantity $\alpha_t(i)$, we need $2L+GM$ operations.[2]

A somewhat different result holds when computing the quantity $\beta_t(i)$, which is given by

$$\beta_t(i) = \sum_{j=1}^{L} Q(i,j)h_j(z_{t+1})\beta_{t+1}(j). \tag{5.22}$$

Evaluating the pdf again takes GM operations, but this pdf value is multiplied by two other numbers, yielding a total of $GM+2$ operations for each term in the summation. Since there are L terms in all, the sum requires $(GM+2)L$ multiplications and $L-1$ additions, for a grand total of $(GM+2)L+(L-1)$ operations. Combining the computational requirements for $\alpha_t(i)$, $\beta_t(i)$, and $\gamma_t(i)$, we see that $(GM+5)L+GM+2$ operations are needed to evaluate a term of type I.

Now let us begin putting the above pieces together to determine the computational

[2] We have implicitly assumed here that all values of the forward variable (and, for that matter, the backward variable as well) are computed only once and then stored in computer memory for the duration of the HMM-based estimation algorithm, so that they can later be accessed on demand at no expense. If this were not the case, the amount computational cost would clearly be much greater than the amount stated here.

requirement for the original estimation formula in (5.15). To begin, we note that each term in the inner summation of the estimation formula requires the multiplication of a term of type II and a term of type III. Thus, each term takes $J + G + 4$ operations to evaluate. Since there are M such terms to be computed for a fixed value of i in the outer summation, the inner sum takes $(J + G + 5)M - 1$ operations to evaluate, including the $M - 1$ additions needed to form the sum. After the inner sum has been computed, it is multiplied by a term of type I, and thus a single term in the outer summation consumes $(GM + 5)L + (J + 2G + 5)M + 2$ operations. Finally, since the outer summation consists of L such terms, the overall number of operations required (including the $L - 1$ final additions) is $(GM + 5)L^2 + [(J + 2G + 5)M + 3]L - 1$. In later sections, we shall assume that the first term of this expression is the dominant term, so that a reasonable first-order approximation to computational cost is cML^2 operations per sample, where c is an appropriately chosen constant. Thus, assuming all other parameters are held fixed, we see that total computation is linear in the number of Gaussian components in each output pdf, M, and is quadratic in the number of states in the underlying Markov chain, L.

Of course, the expression we have just derived represents the computational requirement only for a single time index t. To obtain the number of operations needed to evaluate the entire waveform estimate, we multiply this number by N, the number of samples in the observation. Thus, the amount of computation required to generate a waveform estimate depends linearly on N when the model parameters L and M are held constant.

5.4 HMM-Based Performance Evaluation for the Gaussian Problem

Thus far in this chapter, we have discussed only the methods involved in HMM-based estimation. In later sections, we will also discuss certain extensions of these methods so that they can be applied in more complex signal processing problems. But for the moment, let us shift our focus away from the details of algorithm derivation, and instead consider how our HMM-based procedure performs in a specific signal estimation problem. We devote this section to a discussion of an experiment which uses computer-simulated signals and noise to determine how the performance of our new HMM-based method changes as a function of two major parameters: (i) the number of states in the finite-state signal model; and (ii) the signal-to-noise ratio (SNR) characterizing the observation.

The experiment is designed to address a simple, purely Gaussian signal estimation problem, so that the globally optimum processor is known exactly and can be implemented with ease. In particular, the true source signal $\{Y_t\}$ for this problem is assumed to obey the second-order difference equation

$$Y_t = 0.75Y_{t-1} + 0.2Y_{t-2} + W_t, \tag{5.23}$$

where the sequence $\{W_t\}$ consists of i.i.d. Gaussian random variables, each having a mean of zero and a standard deviation of unity. The additive noise sequence $\{V_t\}$, on the other hand, is assumed to consist of i.i.d. Gaussian random variables, each having a mean of zero

i	$P(i)$	$Q(i,\cdot)$
1	1.00	1.00

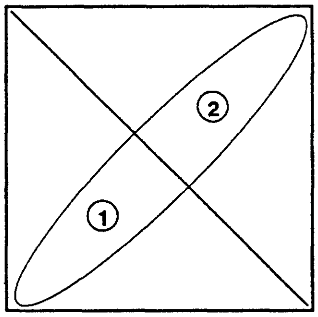
i	$\mu(i)$	$\sigma(i)$	$\rho(i)$
1	0.00	2.93	1.00

Table 5.1: Parameter definitions for 1-state HMM representation of the AR Gaussian process $Y_t = 0.75Y_{t-1} + 0.2Y_{t-2} + W_t$. From top to bottom: specification of initial state probability and state transition probability for Markov chain; specification of the mean, standard deviation, and weighting coefficient for Gaussian-mixture pdf associated with the single state.

and a standard deviation σ'_{11}, whose value is specified according to the SNR level being tested.

We will examine the signal estimation performance achieved by using each of five distinct HMM-based representations of $\{Y_t\}$, specifically representations containing one, two, three, five, and nine states. The parameter values for each signal model used in the experiment are generated directly from realizations of $\{Y_t\}$ using the model-building algorithms described in Chapter 4. Because the true signal $\{Y_t\}$ can be characterized with a state-space representation in which the state vector is two-dimensional, the states in each HMM-based representation of $\{Y_t\}$ are made to correspond to disjoint regions in the two-dimensional coordinate plane. The output pdf assigned to each state of each HMM (with the exception of the one-state HMM) is a Gaussian mixture having three components. For the one-state HMM, the output pdf is taken to be the Gaussian marginal pdf of the true signal $\{Y_t\}$. The parameter values for all five of the finite-state signal models are given in Tables 5.1 through 5.5.

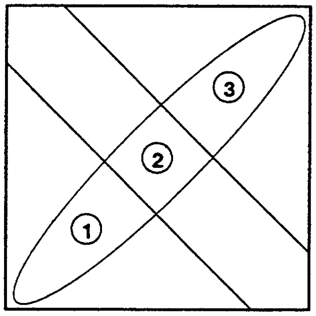
Before we begin to assess the estimation performance associated with each of the finite-state models defined above, let us first try to gain an appreciation for the differences among these models by examining the statistical structure of their output signals. A simple, qualitative way of doing this is to generate a suitably long realization of the output signal from each HMM, and then visually compare and contrast the temporal patterns that are present in the resulting collection of realizations. In Figures 5-1(a) through 5-1(e), we show plots of output waveforms generated by each of the finite-state models used in the experiment. This series of plots is ordered from top to bottom according to the number of states in the signal model. Note that each successive waveform in the series possesses the same basic shape as its predecessor, but also contains a significant amount of detail that was not present before. The similarity among these waveforms results from the fact that the underlying sequence of state variable values for each waveform was generated from



i	$P(i)$	$Q(i,\cdot)$	
1	0.50	0.93	0.07
2	0.50	0.07	0.93

i	$\mu(i)$			$\sigma(i)$			$\rho(i)$		
1	-4.60	-2.14	-0.59	1.63	1.16	0.78	0.23	0.46	0.30
2	0.59	2.14	4.60	0.78	1.16	1.63	0.30	0.46	0.23

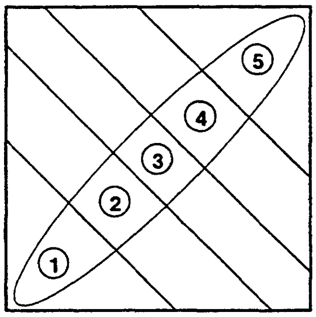
Table 5.2: Parameter definitions for 2-state HMM representation of the AR Gaussian process $Y_t = 0.75Y_{t-1} + 0.2Y_{t-2} + W_t$. From top to bottom: partitioning of underlying two-dimensional state space (superimposed on an elliptical equi-probability contour of the true state-vector pdf); specification of initial state probabilities and state transition probabilities for Markov chain; specification of means, standard deviations, and weighting coefficients for Gaussian-mixture pdf associated with each state.



i	$P(i)$	$Q(i,\cdot)$		
1	0.31	0.91	0.09	0
2	0.38	0.07	0.86	0.07
3	0.31	0	0.09	0.91

i	$\mu(i)$			$\sigma(i)$			$\rho(i)$		
1	-5.48	-3.90	-2.31	1.31	0.71	0.71	0.22	0.25	0.53
2	-0.86	0.15	1.09	0.64	0.56	0.58	0.38	0.37	0.25
3	2.31	3.90	5.48	0.71	0.71	1.31	0.53	0.25	0.22

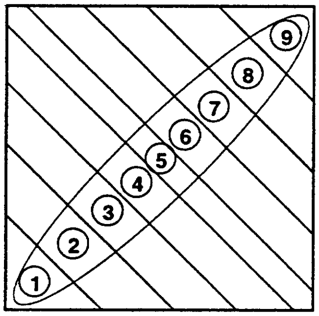
Table 5.3: Parameter definitions for 3-state HMM representation of the AR Gaussian process $Y_t = 0.75Y_{t-1} + 0.2Y_{t-2} + W_t$. From top to bottom: partitioning of underlying two-dimensional state space (superimposed on an elliptical equi-probability contour of the true state-vector pdf); specification of initial state probabilities and state transition probabilities for Markov chain; specification of means, standard deviations, and weighting coefficients for Gaussian-mixture pdf associated with each state.



i	$P(i)$	$Q(i,\cdot)$				
1	0.16	0.86	0.14	0	0	0
2	0.23	0.10	0.74	0.16	0	0
3	0.22	0	0.17	0.66	0.17	0
4	0.23	0	0	0.16	0.74	0.10
5	0.16	0	0	0	0.14	0.86

i	$\mu(i)$			$\sigma(i)$			$\rho(i)$		
1	-6.57	-4.96	-3.38	1.03	0.94	0.76	0.10	0.26	0.64
2	-2.12	-1.38	-0.67	0.63	0.50	0.50	0.43	0.39	0.18
3	-0.74	-0.11	0.45	0.43	0.36	0.52	0.25	0.28	0.47
4	0.67	1.38	2.12	0.50	0.50	0.63	0.18	0.39	0.43
5	3.38	4.96	6.57	0.76	0.94	1.03	0.64	0.26	0.10

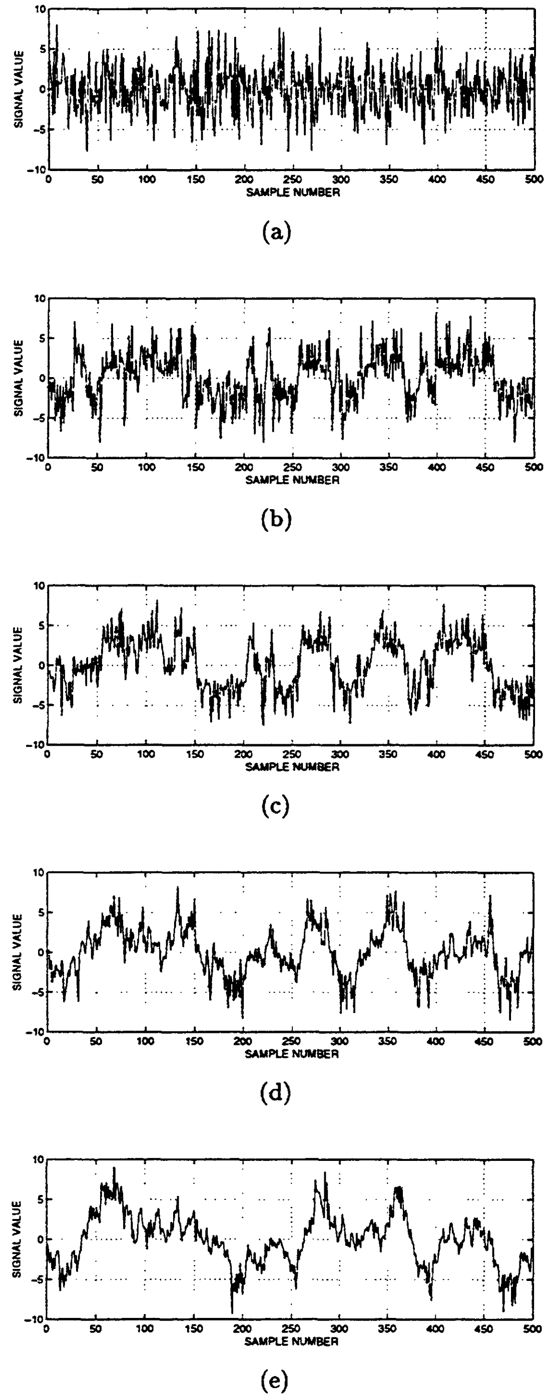
Table 5.4: Parameter definitions for 5-state HMM representation of the AR Gaussian process $Y_t = 0.75Y_{t-1} + 0.2Y_{t-2} + W_t$. From top to bottom: partitioning of underlying two-dimensional state space (superimposed on an elliptical equi-probability contour of the true state-vector pdf); specification of initial state probabilities and state transition probabilities for Markov chain; specification of means, standard deviations, and weighting coefficients for Gaussian-mixture pdf associated with each state.



i	$P(i)$	$Q(i,\cdot)$								
1	0.074	0.85	0.15	0	0	0	0	0	0	0
2	0.111	0.10	0.69	0.20	0.01	0	0	0	0	0
3	0.126	0	0.18	0.59	0.22	0.01	0	0	0	0
4	0.125	0	0	0.22	0.52	0.24	0.02	0	0	0
5	0.128	0	0	0.02	0.23	0.50	0.23	0.02	0	0
6	0.125	0	0	0	0.02	0.24	0.52	0.22	0	0
7	0.126	0	0	0	0	0.01	0.22	0.59	0.18	0
8	0.111	0	0	0	0	0	0.01	0.20	0.69	0.10
9	0.074	0	0	0	0	0	0	0	0.15	0.85

i	$\mu(i)$			$\sigma(i)$			$\rho(i)$		
1	-7.10	-6.04	-4.86	1.13	0.58	0.56	0.27	0.27	0.46
2	-4.18	-3.50	-2.86	0.49	0.49	0.53	0.25	0.41	0.34
3	-2.70	-2.40	-1.75	0.47	0.42	0.50	0.12	0.27	0.61
4	-1.57	-1.18	-0.58	0.46	0.35	0.44	0.24	0.27	0.49
5	-0.74	-0.27	0.28	0.37	0.41	0.48	0.11	0.29	0.60
6	0.58	1.18	1.57	0.44	0.35	0.46	0.49	0.27	0.24
7	1.75	2.40	2.70	0.50	0.42	0.47	0.61	0.27	0.12
8	2.86	3.50	4.18	0.53	0.49	0.49	0.34	0.41	0.25
9	4.86	6.04	7.10	0.56	0.58	1.13	0.46	0.27	0.27

Table 5.5: Parameter definitions for 9-state HMM representation of the AR Gaussian process $Y_t = 0.75Y_{t-1} + 0.2Y_{t-2} + W_t$. From top to bottom: partitioning of underlying two-dimensional state space (superimposed on an elliptical equi-probability contour of the true state-vector pdf); specification of initial state probabilities and state transition probabilities for Markov chain; specification of means, standard deviations, and weighting coefficients for Gaussian-mixture pdf associated with each state.



(a)

(b)

(c)

(d)

(e)

Figure 5-1: Waveforms generated by increasingly accurate HMM representations of the AR Gaussian process $Y_t = 0.75Y_{t-1} + 0.2Y_{t-2} + W_t$: (a) output of one-state HMM; (b) output of two-state HMM; (c) output of three-state HMM; (d) output of five-state HMM; (e) output of nine-state HMM.

a common pseudo-random noise sequence. The progressive increase in signal detail is a natural consequence of the corresponding increase in model complexity.

The waveform produced by the one-state HMM, shown in Figure 5-1(a), is the coarsest possible finite-state representation of the actual signal, because any HMM with only a single state is capable of modeling only the marginal statistics of the signal. (Such an HMM cannot model any temporal dependence exhibited by the original signal, since the output samples of a one-state HMM are necessarily statistically independent.) On the other hand, the waveform produced by the two-state HMM exhibits some temporal correlation, but the representation is coarse, abruptly switching back and forth between two gross output levels over time. As we continue to scan through this series of plots, we observe a greater and greater degree of refinement in signal structure, until finally we come to the waveform produced by the nine-state HMM. This waveform, which is shown Figure 5-1(e), is almost indistinguishable in character from a waveform that would be generated by the original autoregressive linear model in (5.23).

Although it is both interesting and useful to examine realizations of the output signals of various finite-state models, as we have done with Figure 5-1, this exercise does not necessarily help us to predict the estimation performance that will be achieved by each HMM. In the remaining portion of this section, we will quantify the performance associated with each model and compare its performance to that of the optimal Wiener smoother. For the particular estimation problem we have chosen, where the observation is a Gaussian signal combined with independent white Gaussian noise, a reasonable measure of performance is the gain in signal-to-noise ratio (SNR) achieved by the estimator. Of course, in order to calculate this gain, we must have suitable definitions for both the input SNR and the output SNR, which we denote, respectively, by $\mathrm{SNR_{in}}$ and $\mathrm{SNR_{out}}$. For the first of these quantities, $\mathrm{SNR_{in}}$, we will use the classical definition given by

$$\mathrm{SNR_{in}} = 10\log_{10}\frac{E\{Y_t^2\}}{E\{V_t^2\}}. \tag{5.24}$$

For the remaining quantity, $\mathrm{SNR_{out}}$, the definition is not as straightforward, for it requires that we view the output estimate $\hat{Y}_t$ as being composed of the original signal value Y_t together with an additive noise component U_t, which actually represents the estimation error. In other words, we need to express $\hat{Y}_t$ in the form

$$\hat{Y}_t = Y_t + U_t \tag{5.25}$$

$$= Y_t + (\hat{Y}_t - Y_t). \tag{5.26}$$

With this construction, we can define the output SNR as[3]

$$\mathrm{SNR_{out}} = 10\log_{10}\frac{E\{Y_t^2\}}{E\{(\hat{Y}_t - Y_t)^2\}}. \tag{5.27}$$

[3] Because the term $(\hat{Y}_t - Y_t)$ is more naturally viewed as an error term rather than an additive noise term, the quantity $\mathrm{SNR_{out}}$ is also commonly referred to as the signal-to-error ratio, or SER.

In all cases, the value of SNR_{in} can be expressed in closed form in terms of the signal covariance matrix $\mathbf{C}_Y$ and the noise variance $\sigma_{11}'^2$ that is needed to achieve the desired SNR level in a specific case. The input SNR is given by

$$\text{SNR}_{\text{in}} = 10 \log_{10} \frac{\operatorname{tr}(\mathbf{C}_Y)}{N\sigma_{11}'^2}, \tag{5.28}$$

where $\operatorname{tr}(\cdot)$ represents the matrix trace operator and N is the observation length. Furthermore, whenever the Wiener smoother is applied to the observation, the quantity SNR_{out} can also be expressed in closed form. This quantity is given by

$$\text{SNR}_{\text{out}} = 10 \log_{10} \frac{\operatorname{tr}(\mathbf{C}_Y)}{\operatorname{tr}\left(\mathbf{C}_Y - \mathbf{C}_Y(\mathbf{C}_Y + \sigma_{11}'^2\mathbf{I})^{-1}\mathbf{C}_Y\right)}. \tag{5.29}$$

Whenever one of the HMM-based smoothers is applied, however, we must resort to an estimate for the denominator in the expression for SNR_{out}. This estimate is obtained simply by taking the arithmetic mean of the squared values of the actual error waveform.

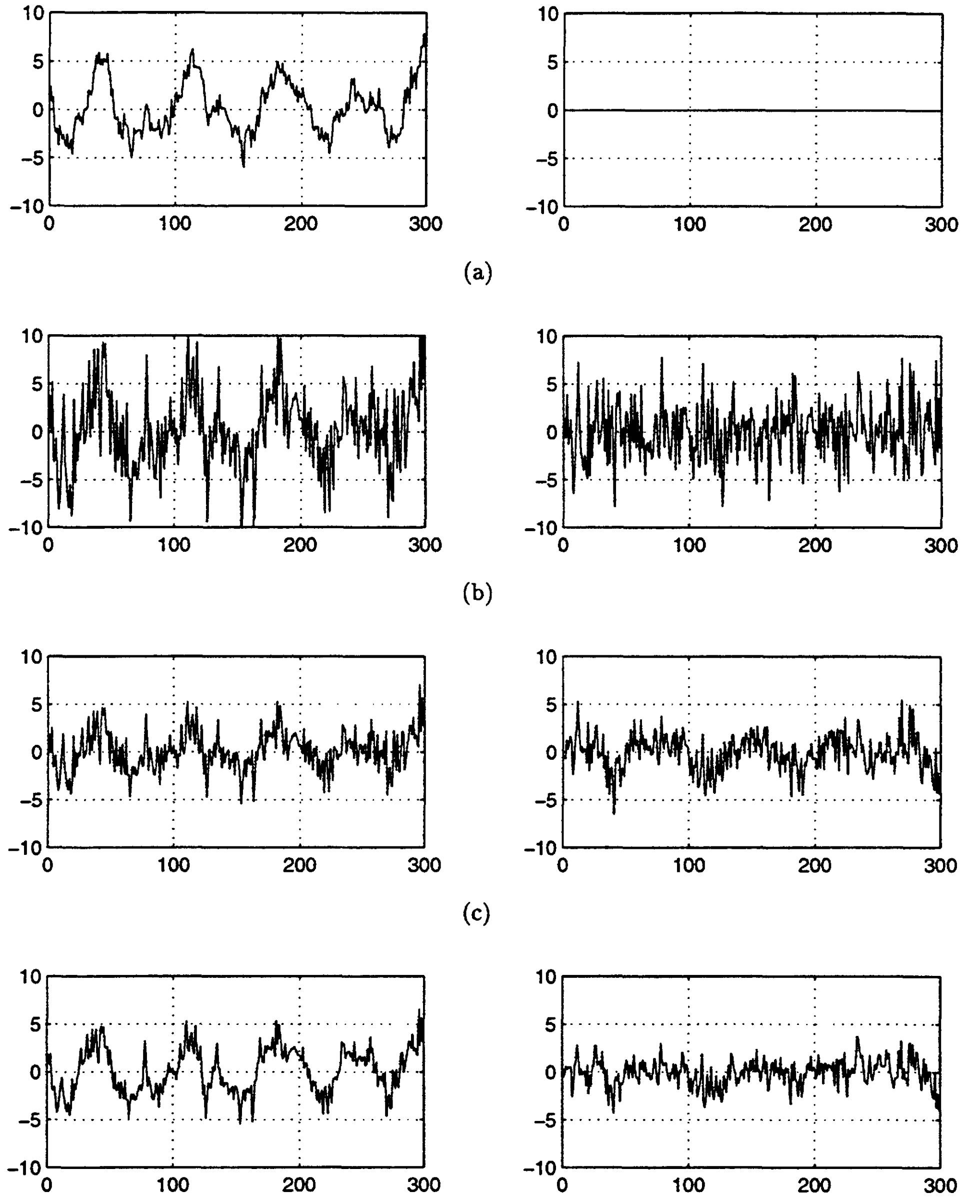
For our experiment, estimation performance was measured for each signal model at input SNR levels of –10 dB, –5 dB, 0 dB, 5 dB, and 10 dB. For each input SNR level, a total of 1000 waveforms were processed by each of the finite-state estimators described above. In addition, each waveform was processed by the Wiener smoother, whose output is given by

$$\hat{\mathbf{y}}_{0:N-1} = \mathbf{C}_Y(\mathbf{C}_Y + \sigma_{11}'^2\mathbf{I})^{-1}\mathbf{z}_{0:N-1}. \tag{5.30}$$

Each input waveform was 300 samples in length. In Figure 5-2, we show the results of a single experimental trial for which the input SNR level was 0 dB. The collection of waveforms plotted in this figure includes realizations of the original source signal, the noisy observed signal, and the estimates generated by each HMM-based smoother and by the Wiener smoother. Each of these waveforms is shown next to its associated residual, which was created by subtracting the original from the estimate.

In Figures 5-3(a) and 5-3(b), we give a graphical summary of the performance of the HMM-based smoothers, as well as that of the globally optimal Wiener smoother. Let us now consider each of these plots in turn. Observe that a single curve on the plot shown in Figure 5-3(a) indicates the output SNR that was achieved by a particular finite-state estimation algorithm as a function of the input SNR.[4] It is clear from this figure that the nine-state HMM performs nearly as well as the optimal Wiener smoother. (Note that we

[4] To interpret these curves properly, we must be aware that the plotted values for the output SNR are somewhat deceiving, for in certain cases they seem to suggest that extraordinary estimation gains have been made, especially at low input SNR levels. For example, it appears that the one-state HMM achieves an estimation gain of more than 10 dB when the input SNR is –10 dB. To understand why this is true, we must keep in mind that an optimal estimator will produce a value close to the prior mean of the underlying signal whenever the input SNR is extremely small, since very little new information can be extracted from the observation itself. In this case, since our one-state HMM-based estimator is nothing more than an optimal memoryless Wiener smoother (i.e., a one-sample FIR Wiener filter), and since the underlying signal has a prior mean of zero, the estimator tends to produce values very near zero. Of course, this produces a residual



(a)

(b)

(c)

(d)

Figure 5-2: Plots of estimated versions of the original waveform (left-hand column) and their corresponding residual waveforms (right-hand column) after subtracting out the original: (a) original waveform; (b) original waveform combined with additive noise (0 dB SNR); (c) estimate of original waveform produced by one-state HMM, and its residual; (d) estimate and residual produced by two-state HMM. *(Continued on following page.)*

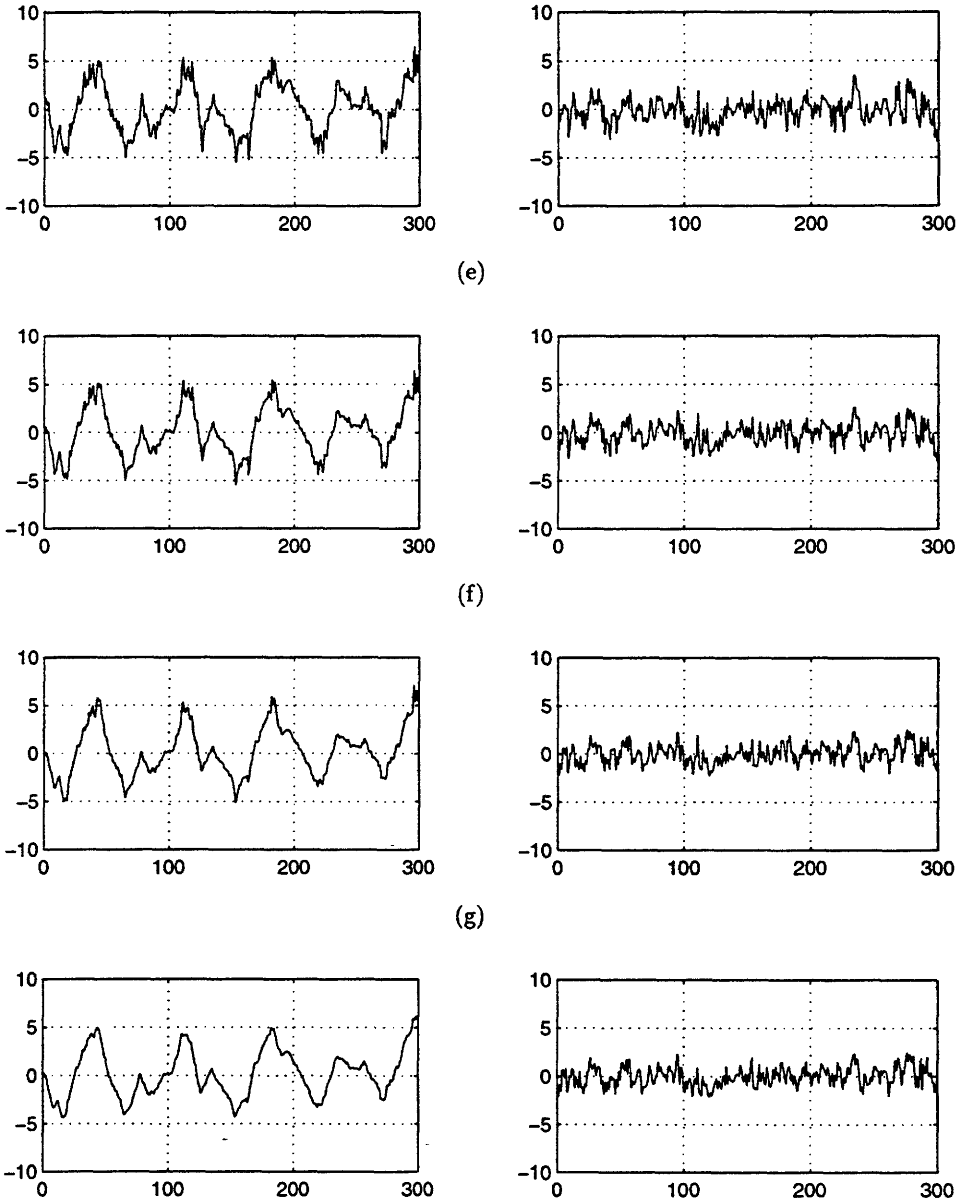


Figure 5-2: *(Continued from previous page.)* (e) estimate and residual produced by three-state HMM; (f) estimate and residual produced by five-state HMM; (g) estimate and residual produced by nine-state HMM; (h) estimate and residual produced by optimal Wiener smoother.

have labeled the performance curve associated with the Wiener smoother as "INF-STATE" to indicate that this curve could be also achieved by using an HMM with an infinite number of states.) In addition, we see that the performance of the five-state HMM is within 1 dB of the performance of the Wiener smoother over the entire range of input SNR levels tested. Thus, we have the rather surprising conclusion that, for this purely Gaussian problem, near-optimal performance can be achieved by using only a very coarse (albeit well-designed) finite-state signal model.[5] As one might predict, however, performance can degrade rapidly if the model becomes too coarse. This conclusion can be drawn from the performance curves associated with the three-state, two-state, and one-state models. We note in particular that the estimator based on the one-state HMM (which is, as we have already mentioned, the best possible memoryless operation that can be performed on the observed signal), yields only a slight improvement in SNR, even at the highest input SNR level tested; in fact, for this case, its performance lags behind that of the Wiener smoother by approximately 3 dB.

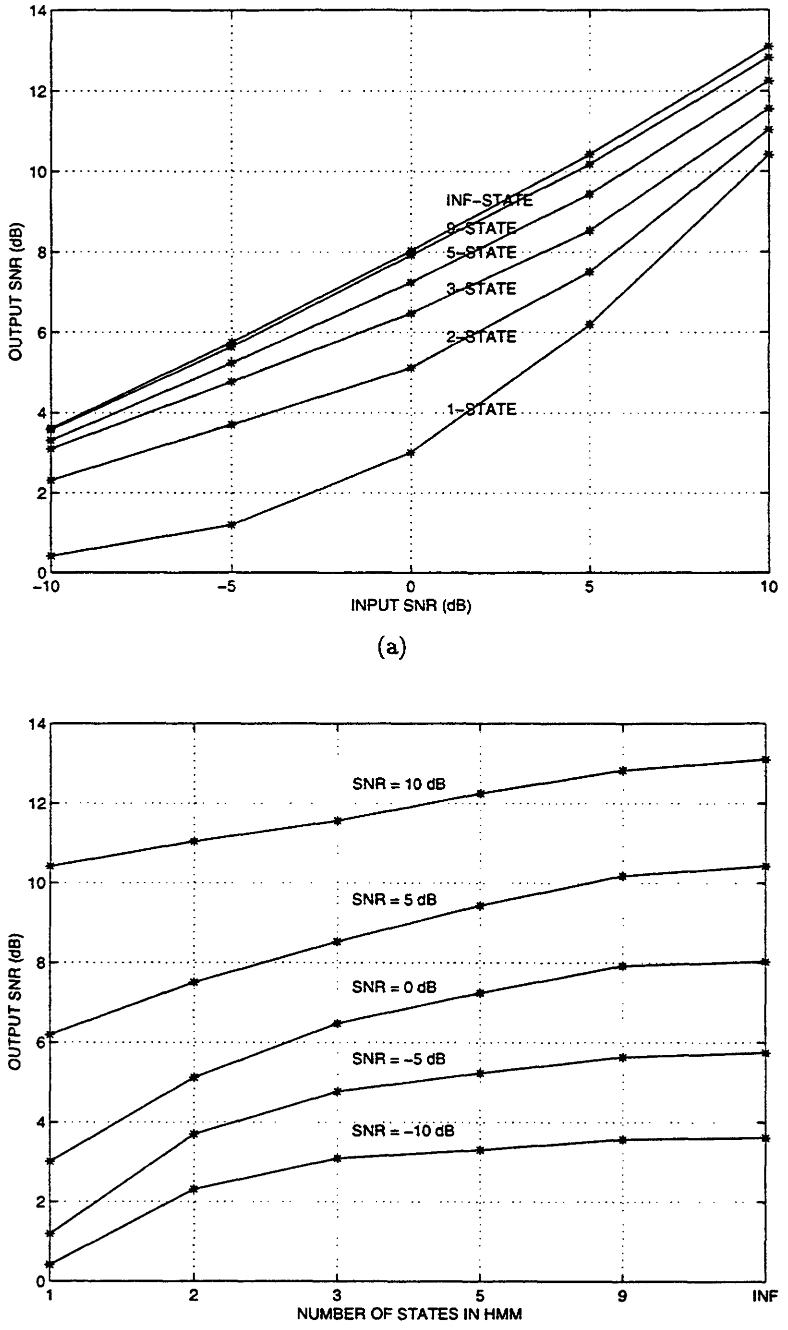
Figure 5-3(b) displays exactly the same data shown in Figure 5-3(a), but in a slightly different format. Specifically, a single curve on this plot now indicates the output SNR that was achieved at a fixed input SNR level as a function of the number of states used in the HMM-based estimation algorithm. Thus, each successive point on the curve explicitly indicates the marginal value of adding the corresponding number of states to the model.

5.5 Extensions of the Basic Signal Estimation Algorithm

In the previous section, we addressed only the problem of estimating a stationary Gaussian signal that has been corrupted by independent additive white Gaussian noise. We chose to consider this classical estimation problem first not only because of its analytical simplicity, but also because a theoretical bound on performance was available for this case. Specifically, the performance of the globally optimal solution — i.e., the Wiener smoother — could be calculated in advance and compared directly to the performance achieved by using various finite-state HMMs for the underlying signal and noise combination. Clearly, the recursive algorithms we used to solve this purely Gaussian problem could be applied just as easily to the problem of estimating a non-Gaussian signal in additive white Gaussian noise. The only modification required in such a case would be to replace the original HMM, which was designed to represent the Gaussian signal, with a new HMM designed to represent the non-Gaussian signal. The sequence of computations subsequently performed to generate a signal estimate would be exactly the same as before. Thus, we already have at our disposal a method for estimating any stationary signal (provided, of course, that the signal is adequately characterized by an HMM) — either Gaussian or non-Gaussian — that has

error signal that is approximately the same as the original signal, which in turn causes the output SNR to be approximately 0 dB, even though the input SNR was -10 dB. Therefore, because the estimation gains will usually appear to be substantial at very small input SNR levels, the output SNR must be interpreted with care. At moderate to high input SNR levels, the difference $\mathrm{SNR}_{\mathrm{out}} - \mathrm{SNR}_{\mathrm{in}}$ can be interpreted more directly as a reduction in the original noise power.

[5] We will present examples later in the chapter suggesting that this same conclusion may extend to more complicated non-Gaussian estimation problems as well.



(a)

(b)

Figure 5-3: Performance comparison among different HMM-based estimators for the AR Gaussian process $Y_t = 0.75Y_{t-1} + 0.2Y_{t-2} + W_t$ in various levels of additive Gaussian noise: (a) output SNR as a function of input SNR for each of the HMMs used; (b) output SNR as a function of the number of states in the HMM for each noise level.

been additively combined with white Gaussian noise.

A problem we have not yet discussed, however, is that of estimating a stationary signal in the presence of additive non-Gaussian noise. For this more challenging problem, there are two fundamental variations to be considered, according to whether the samples of the noise are temporally independent or temporally dependent. These two variations of the problem — which will be addressed, respectively, in the next two subsections — lead to modifications of the basic estimation algorithm that have different degrees of complexity and, accordingly, that require different amounts of computation per output sample. When addressing either of these variations, we shall restrict the scope of the estimation problem in the usual way by assuming that any non-Gaussian density characterizing either signal or noise can be adequately modeled by a Gaussian-mixture density, provided that the mixture contains a sufficient (but finite) number of elements.

After we have considered these two main variations of the estimation problem in some detail, we then show, in the third subsection, how the corresponding algorithms can be extended even further by applying them to the related problem of signal separation — i.e., the reconstruction of many individual signals of interest that have been additively combined (and perhaps also corrupted by additive noise).

5.5.1 Estimating a Signal in Additive White Non-Gaussian Noise

We begin our discussion by demonstrating how to extend the previously derived estimation algorithm to handle the case in which the samples of additive non-Gaussian noise are i.i.d. As before, we assume that the signal and noise are independent, and that the tth element in our sequence of observations $\{Z_t\}$ is given by

$$Z_t = Y_t + V_t. \tag{5.31}$$

Here, the sequence $\{Y_t\}$ once again represents the output of a stationary L-state HMM, but the sequence $\{V_t\}$ consists of i.i.d. random variables, each now having a Gaussian-mixture pdf $g'_1(\cdot)$ defined by

$$g'_1(v) = \sum_{j=1}^{M'} \rho'_{1j} \mathcal{N}(v; \mu'_{1j}, \sigma'_{1j}). \tag{5.32}$$

As we will soon see, even though this modest change in the density of the noise is the only modification to the observation model considered earlier, it leads to a substantially more complex estimator than the one we derived in the Gaussian-noise case. The added complexity stems from the need to keep track of a new discrete-valued state variable at each time t that indicates which of the M' Gaussian components in the above Gaussian-mixture pdf was the true density for the noise sample added at time t. Recall that a discrete variable of this kind was introduced earlier during the development of the original estimation algorithm; specifically, we used the sub-state variable Φ_t to indicate which component of the Gaussian-mixture pdf (associated with the current state of the Markov chain) was selected to generate the signal value at time t. We now introduce an analogous variable Φ'_t to indicate which component of the Gaussian-mixture pdf for the noise was selected at time t.

With these state variables defined, we can develop a new estimation formula similar to the one developed for the Gaussian-noise case. Specifically, through appropriate conditioning on the potential outcomes for the state variables of the model, we may write

$$\begin{aligned} E\{Y_t|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\} \\ &= \sum_{i=1}^{L} \Pr\{\Theta_t = i|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\} E\{Y_t|\Theta_t = i, Z_t = z_t\} && (5.33) \\ &= \sum_{i=1}^{L} \Pr\{\Theta_t = i|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\} \cdot \\ &\quad \sum_{j=1}^{M_i} \sum_{j'=1}^{M'} \Pr\{\Phi_t = j, \Phi'_t = j'|\Theta_t = i, Z_t = z_t\} \cdot \\ &\qquad E\{Y_t|\Theta_t = i, \Phi_t = j, \Phi'_t = j', Z_t = z_t\}. && (5.34) \end{aligned}$$

We see from this last expression that there are three basic types of quantities that must be computed in order to produce the desired estimate. In particular, we need to compute posterior probabilities of the form $\Pr\{\Theta_t = i|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\}$ (appearing in the outer summation), posterior probabilities of the form $\Pr\{\Phi_t = j, \Phi'_t = j'|\Theta_t = i, Z_t = z_t\}$ (appearing in the inner summation), and conditional expectations of the form $E\{Y_t|\Phi_t = j, \Phi'_t = j', \Theta_t = i, Z_t = z_t\}$. Once we have shown how each of these quantities is computed, our algorithm for estimating a stationary signal in white non-Gaussian noise will be completely specified.

Let us first consider the conditional expectation appearing in (5.34). Note that, under the conditions assumed for this expectation, the random variable Y_t has a purely Gaussian pdf. The parameters of this pdf will naturally depend on the values given for the state variables Θ_t, Φ_t, and Φ'_t, as well as for the observed variable z_t; the conditional expectation itself will be affine in z_t. It is straightforward to show that this expectation is given by

$$E\{Y_t|\Theta_t = i, \Phi_t = j, \Phi'_t = j', Z_t = z_t\} = \mu_{ij} + \frac{\sigma_{ij}^2}{\sigma_{ij}^2 + \sigma'^2_{1j'}}(z_t - \mu_{ij} - \mu'_{1j'}). \qquad (5.35)$$

To describe how the remaining posterior probabilities are computed, we first reiterate a key observation about the sequence $\{Z_t\}$ that we made in our earlier discussion. In particular, this sequence, like the sequence $\{Y_t\}$, is the output of a stationary L-state HMM. In fact, the parameters defining the underlying Markov chain for $\{Z_t\}$ (i.e., the initial state probabilities and the state transition probabilities) are exactly the same as those for $\{Y_t\}$. The only difference between the HMM for $\{Z_t\}$ and the HMM for $\{Y_t\}$ lies in the output densities associated with the states of their respective Markov chains. If the Markov chain for $\{Y_t\}$ is in state i at time t, the conditional pdf characterizing the output sample Y_t is given by

$$g_i(y) = \sum_{j=1}^{M_i} \rho_{ij} \mathcal{N}(y; \mu_{ij}, \sigma_{ij}). \tag{5.36}$$

Since the signal and noise are assumed statistically independent, we can derive the corresponding conditional pdf for Z_t simply by convolving the two pdfs $g_i(\cdot)$ and $g'_1(\cdot)$. For the previously considered case in which the noise was purely Gaussian, this convolution resulted only in a modification of the variance of each component in the original Gaussian-mixture pdf for the signal; consequently, the overall number of components in the resulting Gaussian mixture did not change. However, because the pdf for the noise is now also a Gaussian mixture (which in general has more than a single component), the convolution procedure could greatly increase the number of components in the conditional pdf for Z_t. In fact, it can be readily verified that the result of the convolution in this case is given by

$$h_i(z) = \sum_{j=1}^{M_i} \sum_{j'=1}^{M'} \rho_{ij} \rho'_{1j'} \mathcal{N}\left(z; \mu_{ij} + \mu'_{1j'}, \sqrt{\sigma_{ij}^2 + \sigma'^2_{1j'}}\right), \tag{5.37}$$

which is a weighted sum of $M_i M'$ Gaussian components. But note that, in spite of the M'-fold increase in the number of parameters needed to describe the conditional output pdf, the pdf itself is still merely a Gaussian mixture with a finite number of constituent elements. Therefore, the fundamental mathematical structure of the observed signal $\{Z_t\}$ in the non-Gaussian-noise case is identical to that of the observed signal in the Gaussian-noise case; specifically, either signal is the output of an HMM defined such that the conditional pdf associated with each state of its underlying Markov chain is a Gaussian-mixture pdf. Moreover, in view of the foregoing description of $\{Z_t\}$, it is clear that the values of all parameters defining this structure can be easily calculated in advance once $\{Y_t\}$ and $\{V_t\}$ are specified.

This is a significant observation because it means that precisely the same algorithms that were used in the Gaussian-noise case can be used once again to compute all of the required posterior state probabilities. In particular, we can calculate the probability $\Pr\{\Theta_t = i | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\}$ by applying the recursions already developed for the forward and backward variables $\alpha_t(i)$ and $\beta_t(i)$, and subsequently by combining the resulting values to form the equivalent quantity $\gamma_t(i)$. Moreover, we can calculate the remaining probability $\Pr\{\Phi_t = j, \Phi'_t = j' | \Theta_t = i, Z_t = z_t\}$ through a direct application of Bayes' rule, which in this case is given by

$$\begin{aligned}
&\Pr\{\Phi_t = j, \Phi'_t = j' | \Theta_t = i, Z_t = z_t\} \\
&= \frac{\Pr\{\Phi_t = j, \Phi'_t = j' | \Theta_t = i\} \Pr\{Z_t = z_t | \Phi_t = j, \Phi'_t = j', \Theta_t = i\}}{\sum_{k=1}^{M_i} \sum_{k'=1}^{M'} \Pr\{\Phi_t = k, \Phi'_t = k' | \Theta_t = i\} \Pr\{Z_t = z_t | \Phi_t = k, \Phi'_t = k', \Theta_t = i\}}
\end{aligned}$$

$$= \frac{\rho_{ij}\rho'_{1j'}\mathcal{N}\left(z_t;\mu_{ij}+\mu'_{1j'},\sqrt{\sigma_{ij}^2+\sigma'^2_{1j'}}\right)}{\sum_{k=1}^{M_i}\sum_{k'=1}^{M'}\rho_{ik}\rho'_{1k'}\mathcal{N}\left(z_t;\mu_{ik}+\mu'_{1k'},\sqrt{\sigma_{ik}^2+\sigma'^2_{1k'}}\right)}. \tag{5.38}$$

With these rather straightforward modifications to the original estimation algorithm, we are now equipped to handle the case in which the added noise is non-Gaussian and white. Note, however, that in return for this added capability we must pay a premium in the form of an increase in computation. To see this, suppose for convenience that all of the M_i are identical, e.g., that $M_i = M$ for $i = 1, 2, \cdots, L$, and recall from our earlier computational analysis that the required number of operations in the Gaussian-noise case was approximately cML^2 per sample. Now, since the number of components in each output pdf is MM', rather than M, the computational expense increases directly by a factor of M' to approximately $cMM'L^2$ operations per sample.

5.5.2 Estimating a Signal in Additive Colored Non-Gaussian Noise

We now turn our attention to the case in which the additive noise is not only non-Gaussian, but also colored (i.e., consists of samples that are temporally dependent). For this case, we assume that the corrupting sequence $\{V_t\}$ possesses a probabilistic structure similar to that of the signal $\{Y_t\}$, i.e., it is the output of an stationary finite-state HMM. More specifically, we assume that at each time t, the Markov chain $\{\Theta'_t\}$ associated with the corrupting sequence can be in any of the L' possible states $\{1, 2, \cdots, L'\}$, and we denote the initial state probabilities and state transition probabilities for this chain by $\{P'(i)\}_{i=1}^{L'}$ and $\{Q'(i,j)\}_{i,j=1}^{L'}$, respectively. Furthermore, we assume that the output pdf associated with the ith state of the Markov chain is a Gaussian mixture having M'_i constituent elements, and we express this pdf as

$$g_i(v) = \sum_{j=1}^{M'_i} \rho'_{ij}\mathcal{N}(v;\mu'_{ij},\sigma'_{ij}), \qquad i = 1, 2, \cdots, L'. \tag{5.39}$$

It is understood that the L' output densities specified above remain constant for each time t. All parameters defining both the HMM for the noise and the HMM for the signal are assumed known.

Because the noise sequence now has temporal dynamics induced by its underlying Markov chain, the optimal estimation formula for this case includes an additional layer of complexity that was not present in the previously considered white-noise case. To derive the new formula, we will use the variables Θ_t and Θ'_t to indicate the states of the respective Markov chains for the signal and noise at time t, and, just as before, we will use the variables Φ_t and Φ'_t to indicate the components of the respective Gaussian-mixture densities that were selected to generate the signal and noise values at time t. Once again, through appropriate conditioning on the potential outcomes for these state variables of the model, we may write

$$E\{Y_t|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\}$$

$$= \sum_{i=1}^{L} \sum_{i'=1}^{L'} \Pr\{\Theta_t = i, \Theta'_t = i' | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\} \cdot$$
$$E\{Y_t | \Theta_t = i, \Theta'_t = i', Z_t = z_t\} \tag{5.40}$$
$$= \sum_{i=1}^{L} \sum_{i'=1}^{L'} \Pr\{\Theta_t = i, \Theta'_t = i' | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\} \cdot$$
$$\sum_{j=1}^{M_i} \sum_{j'=1}^{M'_{i'}} \Pr\{\Phi_t = j, \Phi'_t = j' | \Theta_t = i, \Theta'_t = i', Z_t = z_t\} \cdot$$
$$E\{Y_t | \Theta_t = i, \Theta'_t = i', \Phi_t = j, \Phi'_t = j', Z_t = z_t\}. \tag{5.41}$$

Note that this last expression has the same basic form as (5.34), except that we have now incorporated the additional state variable Θ'_t into both the posterior probabilities and the conditional expectation. As in the previous case, the expectation in this expression is easy to compute, since the random variable Y_t has a Gaussian pdf when conditioned on the values of the model variables Θ_t, Θ'_t, Φ_t, Φ'_t, and Z_t. In this case, the expectation takes the slightly different form

$$E\{Y_t | \Theta_t = i, \Theta'_t = i', \Phi_t = j, \Phi'_t = j', Z_t = z_t\}$$
$$= \mu_{ij} + \frac{\sigma_{ij}^2}{\sigma_{ij}^2 + \sigma'^2_{i'j'}} (z_t - \mu_{ij} - \mu'_{i'j'}). \tag{5.42}$$

To compute the posterior probabilities appearing in (5.41), we can use the same techniques developed earlier, but we must first establish that the observed sequence $\{Z_t\}$ is again the output of a finite-state HMM (albeit one with many more possible states than the one derived in the white-noise case just considered). To see that $\{Z_t\}$ can indeed be characterized in this way, first observe that at a given time t, the values of Θ_t and Θ'_t provide current and complete descriptions of the individual processes $\{Y_t\}$ and $\{V_t\}$, respectively, in the sense that no further information could be provided about either process that would improve our predictions about its future behavior. Since $\{Z_t\}$ is merely the sum of $\{Y_t\}$ and $\{V_t\}$, it follows that the pair of values (Θ_t, Θ'_t) also summarizes all relevant information currently available about the underlying dynamics of $\{Z_t\}$, and therefore may be considered a suitable state variable for $\{Z_t\}$. Now, since the individual signal state variable Θ_t may assume any of the L values $\{1, 2, \cdots, L\}$, and the noise state variable Θ'_t may simultaneously assume any of the L' values $\{1, 2, \cdots, L'\}$, we conclude that the new composite state variable (Θ_t, Θ'_t) may assume any of the LL' values $\{(1,1), (1,2), \cdots, (L, L')\}$.

Furthermore, based on the parameter specifications for the individual Markov chains $\{\Theta_t\}$ and $\{\Theta'_t\}$, as well as on the assumption that these two chains are statistically independent, we can directly calculate the parameter values that characterize the new Markov "super-chain" $\{(\Theta_t, \Theta'_t)\}$. In particular, the initial state probabilities for this super-chain, which we denote by $\{P''((i,i'))\}$, are given by

$$\begin{aligned} P''((i,i')) &= \Pr\{\Theta_0 = i, \Theta'_0 = i'\} && (5.43)\\ &= \Pr\{\Theta_0 = i\}\Pr\{\Theta'_0 = i' | \Theta_0 = i\} && (5.44)\\ &= \Pr\{\Theta_0 = i\}\Pr\{\Theta'_0 = i'\} && (5.45)\\ &= P(i)P'(i'). && (5.46) \end{aligned}$$

The state transition probabilities, which we denote by $\{Q''((i,i'),(j,j'))\}$, are given by

$$\begin{aligned} Q''((i,i'),(j,j')) &= \Pr\{\Theta_{t+1} = j, \Theta'_{t+1} = j' | \Theta_t = i, \Theta'_t = i'\} && (5.47)\\ &= \Pr\{\Theta_{t+1} = j | \Theta_t = i, \Theta'_t = i'\} \cdot \\ &\qquad \Pr\{\Theta'_{t+1} = j' | \Theta'_t = i', \Theta_t = i, \Theta_{t+1} = j\} && (5.48)\\ &= \Pr\{\Theta_{t+1} = j | \Theta_t = i\}\Pr\{\Theta'_{t+1} = j' | \Theta'_t = i'\} && (5.49)\\ &= Q(i,j)Q'(i',j'). && (5.50) \end{aligned}$$

Note that in each of the above derivations, we have used the fact that the original Markov chains for the signal and noise are statistically independent.

The output density associated with state (i,i') of the new Markov super-chain can be obtained, as before, by convolving the corresponding signal and noise densities $g_i(\cdot)$ and $g'_{i'}(\cdot)$. Performing this convolution yields the new pdf

$$h_{(i,i')}(z) = \sum_{j=1}^{M_i}\sum_{j'=1}^{M'_{i'}} \rho_{ij}\rho'_{i'j'}\mathcal{N}\left(z; \mu_{ij} + \mu'_{i'j'}, \sqrt{\sigma_{ij}^2 + \sigma'^2_{i'j'}}\right). \qquad (5.51)$$

This completes the parameter specification of the new HMM for the observed signal $\{Z_t\}$. Clearly, the required posterior probabilities $\Pr\{\Theta_t = i, \Theta'_t = i' | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\}$ and $\Pr\{\Phi_t = j, \Phi'_t = j' | \Theta_t = i, \Theta_t = i', Z_t = z_t\}$ can now be computed exactly as they were in the previously considered non-Gaussian-white-noise case (i.e., through the use of standard recursions on the forward and backward variables $\alpha_t(\cdot)$ and $\beta_t(\cdot)$, and through the application of Bayes' rule, respectively).

However, we once again must pay a substantial premium in computation, this time for the added capability of handling temporally dependent non-Gaussian noise. To determine how much additional computation is needed, let us first assume for convenience that $M_i = M$ for $i = 1, 2, \cdots, L$, and that $M'_i = M'$ for $i = 1, 2, \cdots, L'$. Also, let us recall from our earlier computational analysis that the required number of operations in the Gaussian-white-noise case was approximately cML^2 per sample. Now, since the number of states in the underlying Markov chain is LL', rather than L, and the number of components in each output pdf is MM', rather than M, the computational expense has grown to approximately $cMM'L^2L'^2$ operations per sample. Thus, for this case, we see that total computation increases by a factor of $M'L'^2$ over the original Gaussian-white-noise case, and by a factor of L'^2 over the non-Gaussian-white-noise case.

5.5.3 Separating Multiple Linearly Combined Non-Gaussian Signals

In the preceding sections, we have addressed the problem of estimating a stationary signal of interest that has been corrupted by additive stationary noise, and we have developed extensions of our basic estimation technique in order to deal with increasingly complex signal and noise waveforms. In this section, we extend our basic technique even further by demonstrating that it can be applied to the more general problem of signal separation, i.e., the problem in which multiple signals of interest have been added together and corrupted by noise, and in which each individual signal must be optimally recovered from a single finite-length observation.

We have seen in earlier derivations of our estimation technique that, under the HMM-based formulation in which each output density is a Gaussian mixture, the estimation of an entire waveform ultimately reduces to solving numerous scalar Gaussian estimation problems at each time t, and then nonlinearly combining these intermediate estimates (with appropriately defined posterior probabilities) to produce the final signal estimate at time t. Remarkably, this same basic sequence of operations (with only slight modifications) may also be used to solve the more general problem in which the values from many signals, rather than just one, have been additively combined with noise at each sample.

Before delving into a demonstration of an HMM-based signal separation technique, we first briefly summarize the main idea behind it. Note that this new nonlinear technique rests on the assumption that all of the signal and noise waveforms contained in the observation are mutually independent. With this assumption in mind, let us consider the problem of optimally extracting only one of the many signals of interest included in the observed sum. We can generate an estimate of the desired signal using the tools we have already developed, once we recognize that all other processes contained in the observation — both signal and noise — can be effectively lumped together and viewed as a single, monolithic noise process that is to be filtered out. Of course, the parameters that characterize this newly defined lumped noise process will vary according to the identity of the signal we are currently attempting to estimate. Nonetheless, once the appropriate changes have been made to the definition of the noise structure, the basic estimation technique can be applied exactly as before. In this way, each of the signals contained in the observation can be estimated in turn.

Our objective in the remainder of this section is to give a concrete example of a non-Gaussian signal separation problem, and to compare the performance of our HMM-based smoothing technique in this problem to the conventional Wiener smoothing technique. In particular, we shall consider an example in which the given observation is a sum of three independent waveforms, two of which are temporally dependent non-Gaussian signals of interest, and the other of which is white Gaussian noise.

Let us begin by specifying the statistical structure of the two signals of interest contained in the observation. The first of these signals, which we denote by $\{X_t\}$, is defined to be the output of a nonlinear autoregressive system driven by white noise. Specifically, we assume that $\{X_t\}$ obeys the difference equation[6]

$$X_{t+1} = 0.5X_t + \frac{25X_t}{1+X_t^2} + 8\cos(1.2t) + W_t, \tag{5.52}$$

where the elements of the sequence $\{W_t\}$ are i.i.d. Gaussian random variables, each having a mean of zero and a standard deviation of unity. A finite-length realization of $\{X_t\}$ is depicted in Figure 5-4(a). Also, in Figure 5-4(b), we show a scatter plot of pairs of consecutive samples $\{(x_{t-1}, x_t)\}$, which have been constructed directly from the realization shown in Figure 5-4(a). This scatter plot reveals the unusual nature of the probability distribution that characterizes the state variable of the nonlinear system described above.

The second signal of interest, which we denote by $\{Y_t\}$, is defined to be a discrete-time version of the classical telegraph signal, which switches back and forth between two distinct values according to a simple probabilistic rule. Specifically, we assume that the values taken by $\{Y_t\}$ are -20 and $+20$, and that consecutive samples of this signal obey the symmetric Markovian probability laws

$$\Pr\{Y_{t+1} = +20 | Y_t = +20\} = \Pr\{Y_{t+1} = -20 | Y_t = -20\} = 0.97 \tag{5.53}$$

and

$$\Pr\{Y_{t+1} = +20 | Y_t = -20\} = \Pr\{Y_{t+1} = -20 | Y_t = +20\} = 0.03 \tag{5.54}$$

for all integer values of t. For this example, we assume the telegraph signal is initialized at time $t = 0$ according to

$$\Pr\{Y_0 = +20\} = \Pr\{Y_0 = -20\} = 0.5. \tag{5.55}$$

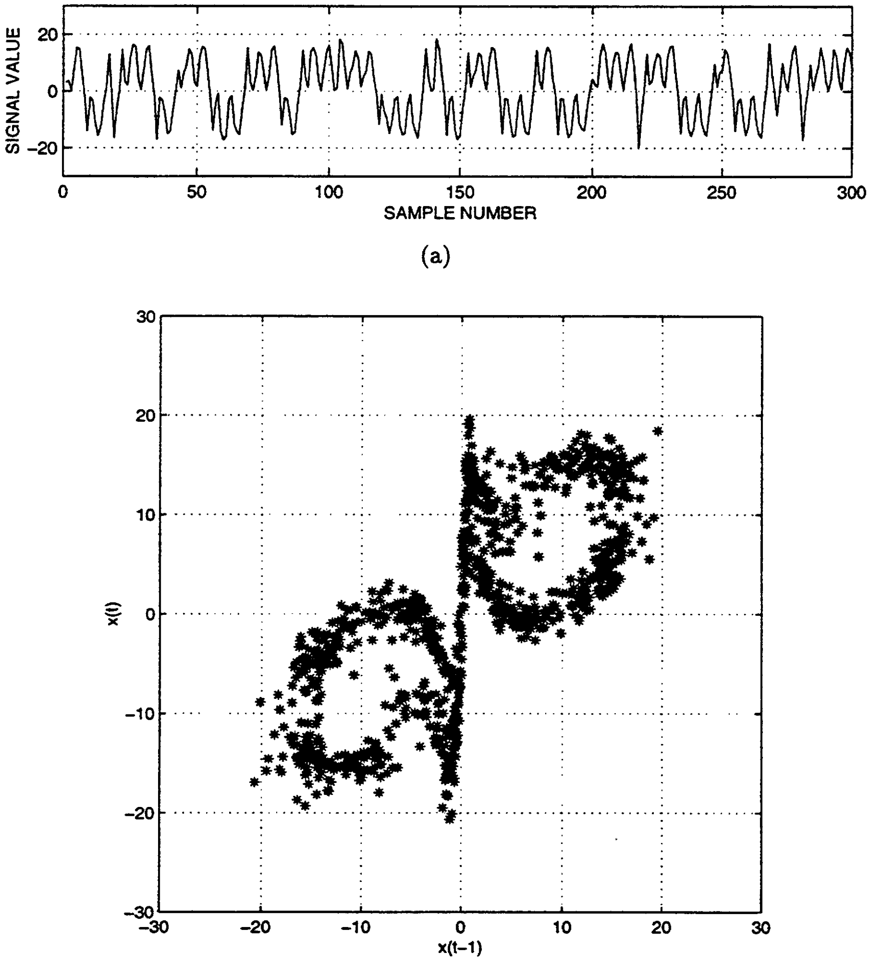
Finally, the corrupting noise sequence contained in the observation, which we denote by $\{V_t\}$, is assumed to consist of i.i.d. Gaussian random variables, each having a mean of zero and a standard deviation of 5.0.

Given these specifications for the signal and noise components, our problem is now to estimate — in the MMSE sense — the particular values taken by the random vectors $\mathbf{X}_{0:N-1}$ and $\mathbf{Y}_{0:N-1}$, based only the value of the observed vector $\mathbf{Z}_{0:N-1}$, which is defined by

$$\mathbf{Z}_{0:N-1} = \mathbf{X}_{0:N-1} + \mathbf{Y}_{0:N-1} + \mathbf{V}_{0:N-1}. \tag{5.56}$$

If we are to apply our HMM-based smoothing technique to solve this signal separation problem (at least in an approximate sense), we must first have appropriate models for each component of the observation. Fortunately, for the noise waveform $\{V_t\}$, an exact, degenerate HMM-based representation is readily available. In particular, the corrupting noise sequence can, in its present form, be viewed as the output of a one-state HMM, whose

[6] This particular random process has appeared frequently in the literature on non-Gaussian signal estimation. For example, it has been previously used in the work of Netto et al [131], Kitagawa [99], and Gordon et al [68].



(a)

(b)

Figure 5-4: Plots showing the temporal and statistical character of the non-Gaussian signal described in text: (a) 300-point realization of signal; (b) scatter plot of pairs $\{(x_{t-1}, x_t)\}$.

i	$P(i)$	$Q(i,\cdot)$	
1	0.50	0.97	0.03
2	0.50	0.03	0.97

i	$\mu(i)$	$\sigma(i)$	$\rho(i)$
1	-20.0	0.1	1.0
2	20.0	0.1	1.0

Table 5.6: Parameter definitions for 2-state HMM representation of the discrete-time telegraph signal $\{Y_t\}$ discussed in the text. From top to bottom: specification of initial state probabilities and state transition probabilities for Markov chain; specification of means, standard deviations, and weighting coefficients for the (one-component) Gaussian-mixture pdf associated with each state.

initial state probability and sole self-transition probability are both 1.0, and whose output density $g_1'(\cdot)$ is defined by

$$g_1'(v) = \mathcal{N}(v; 0, 5). \tag{5.57}$$

The discrete-time telegraph signal $\{Y_t\}$ can also be represented exactly by an HMM, provided that we allow Dirac delta functions in the definitions of the output densities. To see this, observe that the underlying Markov chain associated with this HMM could consist of two states, one for each possible value that can be taken by $\{Y_t\}$. The initial state probabilities and state transition probabilities for such a Markov chain can be inferred directly from the formulas given in (5.53), (5.54), and (5.55). The output densities for the two states, which we denote by $g_1(\cdot)$ and $g_2(\cdot)$, would then be defined as

$$g_1(y) = \delta(y - 20) \tag{5.58}$$

and

$$g_2(y) = \delta(y + 20), \tag{5.59}$$

where $\delta(\cdot)$ is the Dirac delta function. This definition causes some practical difficulty, however, for we can not evaluate densities such as those defined above during the implementation of our HMM-based estimation technique. Instead, we must settle for an approximation to a translated Dirac delta function, the most convenient of which is a Gaussian pdf having the same mean value (i.e., either +20 or −20), but with an extremely small standard deviation. In Table 5.6, we give a complete specification for the HMM used to approximate the signal $\{Y_t\}$ in this example.

To create an HMM-based representation for the more complicated non-Gaussian signal $\{X_t\}$, we must rely on the model-building methods developed in Chapter 4. For the purposes of this example, we chose to model $\{X_t\}$ using a 16-state HMM in which the output pdf associated with each state was a three-component Gaussian mixture. Furthermore, although it is clear from the autoregression in (5.52) that a scalar state variable would be sufficient to describe the state of the original nonlinear system at any time, we chose to use a two-dimensional state vector in order to improve the accuracy of the rather coarse finite-state approximation. For this reason, the 16 states of the underlying Markov chain actually represent 16 disjoint, collectively exhaustive regions in a two-dimensional state space. (Recall that such a space was depicted in Figure 5-4(b).) In Table 5.7, we give a complete specification of the HMM used to approximate the signal $\{X_t\}$ in this example.

Using the finite-state models just described for the signals and noise, we can now easily perform signal separation by applying our new nonlinear HMM-based estimation algorithm to a specific realization of the random vector $\mathbf{Z}_{0:N-1}$. In order to establish a useful point of reference by which we can assess the resulting estimation performance, we shall compare the results of our nonlinear algorithm to those of a conventional linear technique, namely the Wiener smoother. Although the Wiener smoother is not a globally optimal MMSE estimator for this problem (owing to the fact that the observation contains non-Gaussian signals), it is nonetheless the best possible linear estimator that we can use.

The Wiener smoother associated with each signal of interest can be implemented through straightforward matrix-vector multiplication. However, we must first know the values of the covariance matrices associated with the three constituent random vectors $\mathbf{X}_{0:N-1}$, $\mathbf{Y}_{0:N-1}$, and $\mathbf{V}_{0:N-1}$ making up the observation. Since each of these random vectors represents a section of a stationary signal, each associated covariance matrix possesses a Toeplitz structure; moreover, because the random vectors are all zero-mean, each covariance matrix is specified entirely by the first N values of the associated autocorrelation function (i.e., autocorrelation values ranging from the 0th lag up to and including the $(N-1)$th lag).

In light of the definitions given earlier, we see that the kth lag of the autocorrelation function for the noise waveform $\{V_t\}$ is given by

$$E\{V_t V_{t+k}\} = 25 \cdot \delta_{t,k}, \tag{5.60}$$

where $\delta_{t,k}$ is the Kronecker delta sequence (i.e., a sequence which has a value of unity if $k = t$, but otherwise is identically zero). It follows that the covariance matrix for the vector $\mathbf{V}_{0:N-1}$ is just a scaled version of the identity matrix, where the scale factor is the value of the noise variance at each sample.

The autocorrelation function of the discrete-time telegraph waveform $\{Y_t\}$ can also be obtained in closed form. In fact, it can be shown (after a significant amount of analysis of the structure of the underlying Markov chain) that the kth lag of the autocorrelation function for the telegraph waveform is given by

$$E\{Y_t Y_{t+k}\} = 0.94^{|k|} \cdot 400. \tag{5.61}$$

Using this formula, we can construct the covariance matrix for $\mathbf{Y}_{0:N-1}$ simply by repeating the value of the kth lag along both the kth sub-diagonal and the kth super-diagonal of the $N \times N$ matrix, for $k = 0, 1, \cdots, N-1$.

Unfortunately, the mathematical definition of the remaining non-Gaussian signal $\{X_t\}$

i	$P(i)$	$Q(i,1),\cdots,Q(i,8)$							
1	0.065	0.125	0.225	0.540	0.025	0	0	0	0.080
2	0.056	0	0	0	0.125	0.220	0.280	0.360	0
3	0.061	0	0	0.065	0.760	0	0	0.100	0.070
4	0.059	0	0	0	0.045	0.915	0.040	0	0
5	0.073	0	0	0	0	0.105	0.545	0.350	0
6	0.058	0.055	0	0	0	0	0	0.010	0.940
7	0.053	0.665	0	0.085	0	0	0	0	0.190
8	0.075	0.120	0.070	0.125	0	0	0	0	0
9	0.065	0.005	0	0	0	0	0	0	0
10	0.056	0	0	0	0	0	0	0	0
11	0.061	0	0	0	0	0	0	0	0
12	0.059	0	0	0	0	0	0	0	0
13	0.073	0	0	0	0	0	0	0	0
14	0.058	0	0	0	0	0	0	0	0
15	0.053	0.020	0.025	0.015	0	0	0	0	0
16	0.075	0.110	0.460	0.110	0	0	0	0	0

i	$Q(i,9),\cdots,Q(i,16)$							
1	0.005	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0.020	0.025	0.015	0	0	0	0	0
8	0.110	0.460	0.110	0	0	0	0	0
9	0.125	0.225	0.540	0.025	0	0	0	0.080
10	0	0	0	0.125	0.220	0.280	0.360	0
11	0	0	0.065	0.760	0	0	0.100	0.070
12	0	0	0	0.045	0.915	0.040	0	0
13	0	0	0	0	0.105	0.545	0.350	0
14	0.055	0	0	0	0	0	0.010	0.940
15	0.665	0	0.085	0	0	0	0	0.190
16	0.120	0.070	0.125	0	0	0	0	0

i	$\mu(i)$			$\sigma(i)$			$\rho(i)$		
1	5.60	3.75	1.76	0.75	0.91	0.91	0.24	0.30	0.46
2	16.51	12.52	14.20	1.67	0.45	0.93	0.34	0.22	0.44
3	9.88	8.24	11.09	0.77	0.92	0.64	0.31	0.45	0.23
4	14.21	16.34	15.02	1.86	0.91	1.08	0.21	0.13	0.67
5	16.45	12.52	14.62	1.10	1.27	1.07	0.17	0.27	0.56
6	7.02	6.72	5.09	1.40	1.44	1.02	0.25	0.23	0.52
7	2.55	1.29	3.38	0.90	1.01	0.60	0.37	0.35	0.27
8	-0.42	-0.12	-1.72	0.79	1.18	0.54	0.39	0.50	0.11
9	-5.60	-3.75	-1.76	0.75	0.91	0.91	0.24	0.30	0.46
10	-16.51	-12.52	-14.20	1.67	0.45	0.93	0.34	0.22	0.44
11	-9.88	-8.24	-11.09	0.77	0.92	0.64	0.31	0.45	0.23
12	-14.21	-16.34	-15.02	1.86	0.91	1.08	0.21	0.13	0.67
13	-16.45	-12.52	-14.62	1.10	1.27	1.07	0.17	0.27	0.56
14	-7.02	-6.72	-5.09	1.40	1.44	1.02	0.25	0.23	0.52
15	-2.55	-1.29	-3.38	0.90	1.01	0.60	0.37	0.35	0.27
16	0.42	0.12	1.72	0.79	1.18	0.54	0.39	0.50	0.11

Table 5.7: Parameter definitions for 16-state HMM representation of the non-Gaussian signal $\{X_t\}$ discussed in the text. From top to bottom: specification of initial state probabilities and state transition probabilities for Markov chain; specification of means, standard deviations, and weighting coefficients for Gaussian-mixture pdf associated with each state.

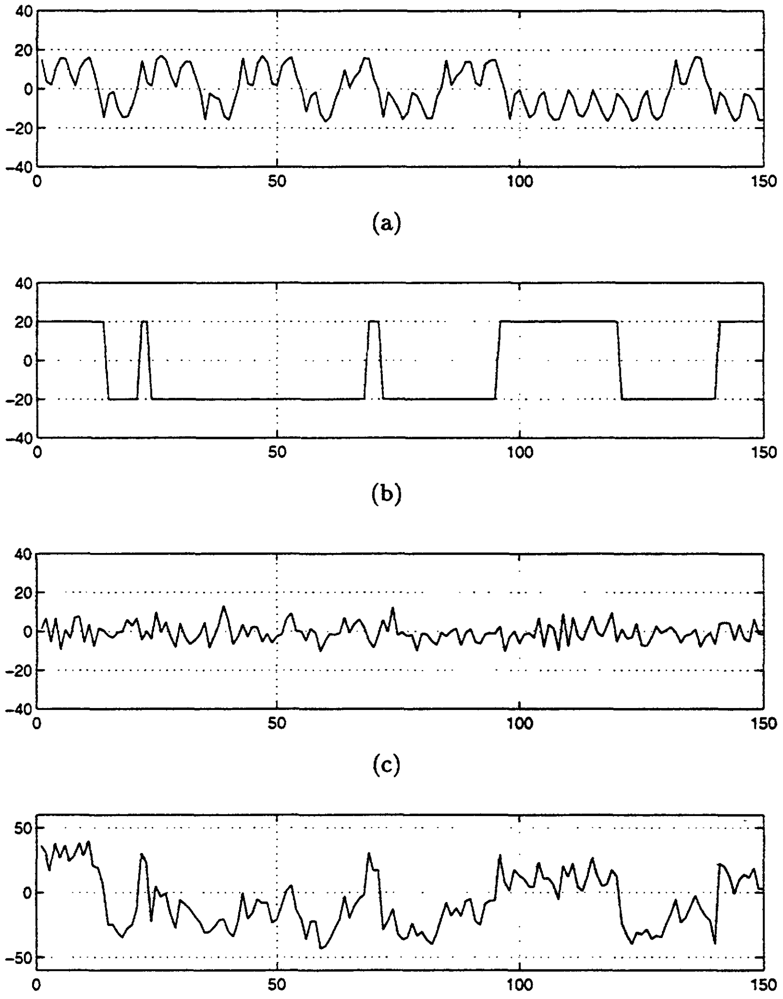
makes it extremely difficult to solve for the covariance matrix of $\mathbf{X}_{0:N-1}$ in closed form. Thus, for this signal we resort to the numerical technique of computing the sample autocorrelation function based on a very long realization of $\{X_t\}$ (specifically, a realization having a length of 100,000 samples); we can then construct the corresponding covariance matrix just as we did before, by specifying its diagonals one at a time based on the lags of the autocorrelation function.

Having defined all of the necessary elements for implementing both the Wiener smoother and the HMM-based smoother, let us now consider performing signal separation using a specific realization of the observed vector $\mathbf{Z}_{0:N-1}$. In Figures 5-5(a) through 5-5(d), we show, respectively, realizations of the individual vectors $\mathbf{X}_{0:N-1}$, $\mathbf{Y}_{0:N-1}$, and $\mathbf{V}_{0:N-1}$, and their sum $\mathbf{Z}_{0:N-1}$; here, we have arbitrarily chosen the observation length $N = 150$. When the Wiener smoothing technique is applied to the observed waveform shown in Figure 5-5(d), we obtain the estimated waveforms shown in Figure 5-6. Recall that, because there are three distinct random processes that make up the observation in this example, there are, accordingly, three distinct Wiener smoothers that must be applied to the observation in order to separate these processes. The smoothers that are designed to extract the vectors $\mathbf{X}_{0:N-1}$, $\mathbf{Y}_{0:N-1}$, and $\mathbf{V}_{0:N-1}$ generate the estimates shown in Figures 5-6(a), 5-6(b), and 5-6(c), respectively. The sum of these estimated waveforms is also shown in Figure 5-6(d).

Let us now compare these baseline estimates with the corresponding estimates produced by the HMM-based smoothing technique, which are shown in Figures 5-7(a) through 5-7(d). By visually comparing and contrasting the estimated waveforms displayed in Figures 5-6 and 5-7, along with their true original counterparts shown in Figure 5-5, we see that the results of the HMM-based smoothing technique appear to be superior to those of the Wiener smoothing technique. In a moment, we will provide quantitative evidence supporting this assertion. Interestingly, an attribute shared by both of the signal separation methods presented here is that the sum of the three estimated waveforms is always the same as the sum of three original waveforms. Hence, the plots appearing in Figures 5-5(d), 5-6(d), and 5-7(d) are actually identical. This property follows directly from the mathematical definitions of the estimates produced by each technique.

To obtain a more precise numerical characterization of the performance of each smoothing technique, we repeated the above signal separation experiment a total of 1000 times using randomly generated observations. On each trial, we recorded the error incurred by each smoothing technique after estimating each signal of interest from the observation. (Estimation of the noise waveform was considered unimportant, and hence the results for this waveform were not examined.) The measure of performance used on each trial, for each signal of interest, was the realized mean squared error (MSE) value, i.e., the arithmetic average of the 150 real numbers obtained by subtracting the actual waveform from the estimated waveform and squaring the resulting residual value at each time index. After all 1000 trials had been performed, we were left with a collection of 1000 such MSE scores for each of four separate cases, representing the results of applying each of the two smoothing techniques to extract each of the two signals of interest.

The sample mean and sample standard deviation of the MSE value for each possible case are displayed in Table 5.8. Observe that, when the non-Gaussian signal $\{X_t\}$ is being estimated, the HMM-based smoother yields, on the average, an MSE value that is nearly



(a)

(b)

(c)

(d)

Figure 5-5: Plots of constituent waveforms used for signal separation problem: (a) realization of non-Gaussian signal; (b) realization of discrete-time telegraph signal; (c) realization of white Gaussian noise; (d) superposition of waveforms shown in (a), (b) and (c).

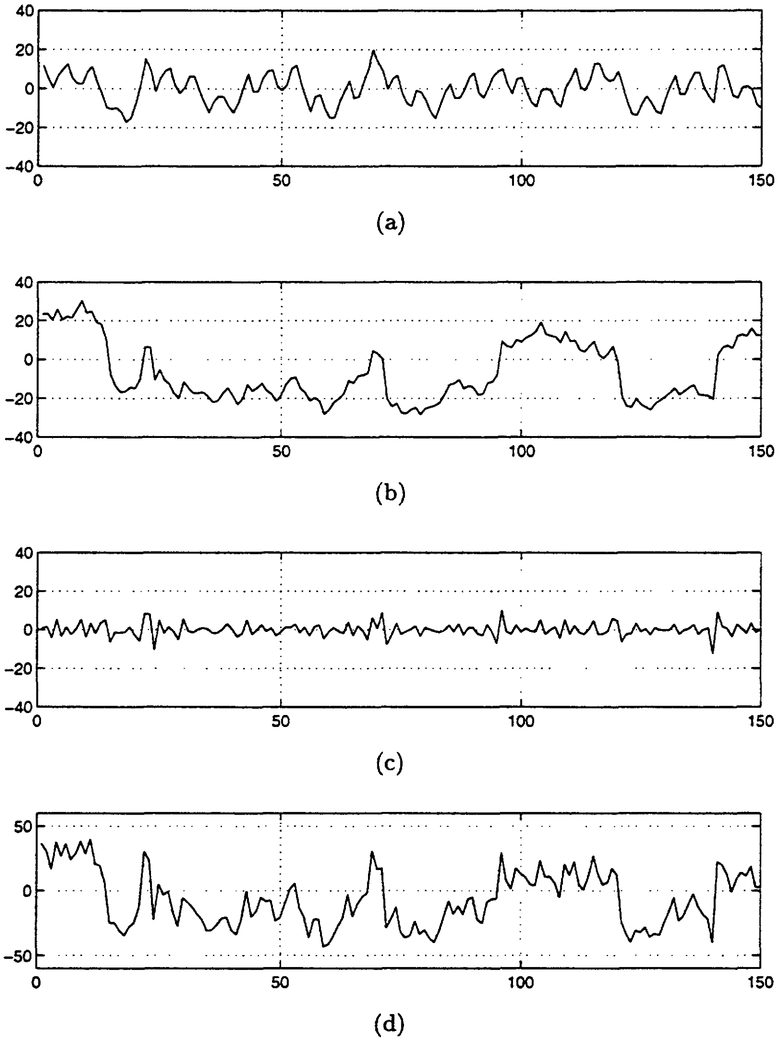
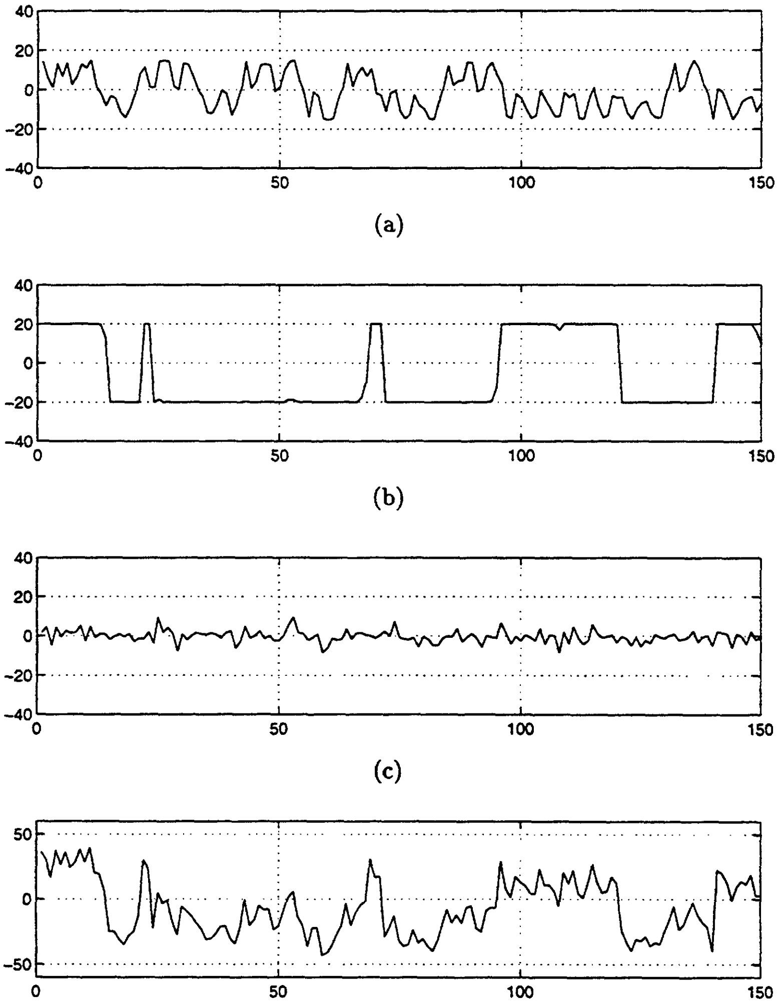


Figure 5-6: Plots of signal separation results using Wiener smoother: (a) estimate of non-Gaussian signal; (b) estimate of discrete-time telegraph signal; (c) estimate of white Gaussian noise; (d) superposition of waveforms shown in (a), (b) and (c).



(a)

(b)

(c)

(d)

Figure 5-7: Plots of signal separation results using HMM-based smoother: (a) estimate of non-Gaussian signal; (b) estimate of discrete-time telegraph signal; (c) estimate of white Gaussian noise; (d) superposition of waveforms shown in (a), (b) and (c).

ESTIMATION OF $\{X_t\}$ GIVEN $\{Z_t\}$		
Type of Estimator	*Sample Mean of MSE Value*	*Sample St. Dev. of MSE Value*
LINEAR	51.6	9.8
HMM-BASED	18.1	5.9

ESTIMATION OF $\{Y_t\}$ GIVEN $\{Z_t\}$		
Type of Estimator	*Sample Mean of MSE Value*	*Sample St. Dev. of MSE Value*
LINEAR	71.8	12.9
HMM-BASED	11.6	10.4

Table 5.8: Results of the 1000-trial experiment designed to compare the performance of the optimal linear estimator and the optimal HMM-based estimator in the signal separation problem. Sample means and sample standard deviations of the MSE value are shown for both estimation techniques for the non-Gaussian signal $\{X_t\}$ (top) and for the telegraph signal $\{Y_t\}$ (bottom).

three times smaller than the value given by the Wiener smoother. When the telegraph signal $\{Y_t\}$ is being estimated, the HMM-based smoother yields an MSE value that is more than six times smaller than the value given by the Wiener smoother. Moreover, in both of these cases, the standard deviation of the MSE value associated with the HMM-based method is lower than the standard deviation associated with the Wiener method. Thus, we see that the HMM-based smoother performs significantly better than the conventional Wiener smoother for the particular non-Gaussian signal separation problem considered here.

5.6 Discussion

5.6.1 Using Finite-State Models for Problems Other than Smoothing

Although we have focused exclusively in this chapter on the problem of signal smoothing, it should be clear that our fundamental approach of finite-state signal modeling can be applied to a variety of other signal processing problems as well. Specifically, some of the most obvious and immediate applications of our basic modeling paradigm would include other variations on the signal estimation problem, e.g., variations such as signal filtering and signal prediction. In this section, we draw heavily from the concepts developed earlier for the smoothing problem in order to explain how these alternative signal estimation problems can also be solved. We then turn our attention to an entirely different class of signal processing problems — namely problems in signal detection and classification (i.e., M-ary hypothesis testing) — and show that approximate solutions to these problems can also be constructed using the finite-state approach.

5.6.1.1 Extensions to Filtering and Prediction Problems

We begin by considering the problems of signal filtering and prediction. The filtering problem differs fundamentally from the smoothing problem in that we are allowed to use only those observations that are available at or before time t to produce an estimate of the true signal value at time t, i.e., we are prohibited from introducing a delay in order to use any additional observations that become available in the future. On the other hand, in the prediction problem, we use observations that are available at or before the current time t to estimate the true signal value at some future time $t+k$.

In either type of problem, HMM-based estimation techniques can still be employed successfully. Whenever such techniques are used, whether for filtering or prediction, the first and most important step to be performed is the calculation of the posterior pmf — based on all data observed up to time t — for the state variable of the underlying Markov chain at time t. This critical first step can be carried out by using a simple recursive procedure, whereby, at each time index, the previously computed values of the posterior state pmf become updated at the moment the current sample in the observed sequence becomes available. It is straightforward to show that a recursive procedure for accomplishing this task can be constructed directly from the forward recursion described in Section F.1. In fact, we note that the desired posterior probabilities in this case can actually be expressed in terms of the forward variable $\alpha_t(i)$ defined earlier, as shown by

$$\begin{aligned}\Pr\{\Theta_t = i|\mathbf{Z}_{0:t} = \mathbf{z}_{0:t}\} &= \frac{f_{\mathbf{Z}_{0:t},\Theta_t}(\mathbf{z}_{0:t}, \Theta_t = i)}{f_{\mathbf{Z}_{0:t}}(\mathbf{z}_{0:t})} &(5.62)\\ &= \frac{f_{\mathbf{Z}_{0:t},\Theta_t}(\mathbf{z}_{0:t}, \Theta_t = i)}{\sum_{j=1}^{L} f_{\mathbf{Z}_{0:t},\Theta_t}(\mathbf{z}_{0:t}, \Theta_t = j)} &(5.63)\\ &= \frac{\alpha_t(i)}{\sum_{j=1}^{L}\alpha_t(j)}. &(5.64)\end{aligned}$$

Once the current posterior pmf has been computed, the remaining steps in generating an estimate differ according to whether filtering or prediction is being performed. The steps involved in prediction are quite simple. In this case, we merely need to project the posterior state pmf for the current time t out to the future time $t+k$, so that it actually represents the state pmf at time $t+k$ based on all data observed up to time t. This projection is achieved by multiplying the current posterior state pmf (taken to be a row vector) by the kth-order state transition matrix (i.e., the matrix which is constructed by multiplying the ordinary state transition matrix by itself k times, and whose (i,j) entry represents the probability that the Markov chain will be in state j at time $t+k$ given that it is in state i at time t). Once this projection has been accomplished, the optimal prediction of the signal value at time $t+k$ is simply a weighted average of the mean values associated with the output densities of the HMM; the weighting coefficients used in this average are just the elements of the (projected) posterior state pmf at time $t+k$.

In the filtering problem, the remaining steps in generating a signal estimate are carried out exactly as they were in the smoothing problem. In particular, for each state of the Markov chain, the posterior sub-state probabilities at time t are first computed based on the value of the observation at time t. (Recall that each sub-state probability indicates the relative likelihood that a particular component of the Gaussian-mixture pdf associated with that state was active at time t, given that the Markov chain was actually in that state at time t.) Once these probabilities have been computed, a conditional mean value can be obtained for a given state of the Markov chain by taking a weighted average of the mean values associated with the conditional densities in the Gaussian mixture for that state; the weighting coefficients used in this average are simply the posterior sub-state probabilities. Finally, the optimal estimate of the signal value at time t is a weighted average of these resulting conditional mean values associated with the states of the Markov chain; the weighting coefficients used in this final average are just the elements of the posterior state pmf at time t.

5.6.1.2 Extension to Multi-Class Hypothesis Testing

To demonstrate that certain signal processing problems other than signal smoothing, filtering, or prediction can also be addressed using the finite-state modeling paradigm, let us now turn our attention to the problem of binary (or, in the more general case, M-ary) hypothesis testing. In the general version of this problem, the waveform we observe is known to be a realization from one of M distinct signal distributions or classes. Our goal in processing the observation is to determine the true class from which it came, based on our prior knowledge about each of the signal classes involved. A simple, classical example of M-ary hypothesis testing is the signal detection problem, in which we have only a signal-plus-noise class and a noise-only class, and we wish to determine whether the signal of interest is present or absent in the given observation.

It is well known that the optimal decision rule for the M-ary hypothesis testing problem — in the sense that it yields the smallest probability of making a classification error — is the maximum a posteriori (MAP) rule [15, 166, 215]. To express this rule precisely, let us denote the N-sample observed signal by $\mathbf{Z}_{0:N-1}$, the given realization of this signal by $\mathbf{z}_{0:N-1}$, the M classes themselves by $\{1, 2, \cdots, M\}$, and the hypothesis that class k is the true class of the observed signal by H_k. In addition, let $\hat{k}$ represent our estimate of the true class. Then, by the MAP rule, this estimate $\hat{k}$ is defined as

$$\hat{k} = \underset{k \in \{1,2,\cdots,M\}}{\arg\max} \; \log \Pr\{H_k | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\}, \tag{5.65}$$

That is, class $\hat{k}$ is, among all M classes, the one whose posterior probability is largest after we have observed the event $\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}$. The logarithm in the above expression (which is a monotonically increasing function and therefore has no effect on the argument of the maximization) has been introduced only to simplify later calculations.

We will find it convenient to re-express the above posterior probability (through an application of Bayes' rule) as

$$\Pr\{H_k | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\} = \frac{\Pr\{H_k\} f_{\mathbf{Z}_{0:N-1}|H_k}(\mathbf{z}_{0:N-1}|H_k)}{f_{\mathbf{Z}_{0:N-1}}(\mathbf{z}_{0:N-1})}, \tag{5.66}$$

where $\Pr\{H_k\}$ is the prior probability of the event H_k, $f_{\mathbf{Z}_{0:N-1}|H_k}(\cdot)$ is the conditional density of $\mathbf{Z}_{0:N-1}$ given that the event H_k has occurred, and $f_{\mathbf{Z}_{0:N-1}}(\cdot)$ is the unconditional density of $\mathbf{Z}_{0:N-1}$. Because the denominator in the above expression is a positive constant independent of k, we can ignore it when performing the maximization over all classes. With this modification, we can then express $\hat{k}$ equivalently as

$$\hat{k} = \underset{k\in\{1,2,\cdots,M\}}{\arg\max} \left\{\log \Pr\{H_k\} + \log f_{\mathbf{Z}_{0:N-1}|H_k}(\mathbf{z}_{0:N-1}|H_k)\right\}. \tag{5.67}$$

Since each prior probability $\Pr\{H_k\}$ is known, the only remaining quantity that must be computed is the value of the conditional density $f_{\mathbf{Z}_{0:N-1}|H_k}(\mathbf{z}_{0:N-1}|H_k)$ for each k.

We now invoke the assumption that the observed signal $\mathbf{Z}_{0:N-1}$ is (either exactly or approximately) the output of a known, unique finite-state HMM under each of the M hypotheses. With this assumption, we can use algorithms derived earlier to compute the associated forward recursion variable $\alpha_{kt}(\cdot)$ for each hypothesis H_k and for each time index t. This forward variable is important because it can be manipulated to give the density value we need, as shown by

$$\log f_{\mathbf{Z}_{0:N-1}|H_k}(\mathbf{z}_{0:N-1}|H_k) = \log \sum_{j=1}^{L} f_{\mathbf{Z}_{0:N-1},\Theta_{N-1}|H_k}(\mathbf{z}_{0:N-1}, \Theta_{N-1} = j|H_k) \tag{5.68}$$

$$= \log \sum_{j=1}^{L} \alpha_{k,N-1}(j). \tag{5.69}$$

As we discuss in Appendix F, however, the forward recursion variable cannot be evaluated directly on a computer (unless N is very small), because the arithmetic operations required to evaluate it at each time index will eventually exceed the dynamic range of essentially any machine without the use of a special scaling procedure. Ironically, it is also demonstrated in Appendix F that, as a by-product of the properly conditioned version of the forward recursion, we obtain a set of scaling coefficients that are in fact the key to evaluating the remaining terms needed in (5.67).

If we denote the set of scaling coefficients associated with hypothesis H_k by $\{c_{kt}\}_{t=0}^{N-1}$, then, based on arguments put forth in Appendix F, we see that the desired value of the conditional density of the observation can be computed in terms of these scaling coefficients as

$$\log f_{\mathbf{Z}_{0:N-1}|H_k}(\mathbf{z}_{0:N-1}|H_k) = \sum_{t=0}^{N-1} \log c_{kt}, \qquad k = 1, 2, \cdots, M. \tag{5.70}$$

With the ability to evaluate these remaining M terms, we can now easily solve the maximization in (5.67), and thus make efficient use of the MAP rule under the finite-state modeling paradigm. In many cases, using this technique could allow us to attain near-optimal M-ary signal classification performance, provided that we use signal models of sufficiently high fidelity.

5.6.2 HMM-Based Signal Estimation in Non-Stationary Noise

A potentially useful avenue for future research would be to explore the design of HMM-based signal estimation schemes for environments in which the noise is non-stationary. To develop this idea further, let us return briefly to the Gaussian estimation problem considered in Section 5.4 and make some additional observations concerning the results plotted in Figure 5-3(b). In particular, observe from this figure that for the two smallest tested values of SNR_{in}, the performance curves become relatively flat after only three states have been included in the signal model. This implies, at least for the Gaussian estimation problem considered in our example, that using an HMM containing any more than three states is wasteful in an environment where the SNR is low.

One way in which we might apply a conclusion of this kind is to incorporate it into an SNR-dependent metric for choosing the appropriate HMM order as part of the model design process. Alternatively, we could use it to develop an estimator that is capable of filtering out a white Gaussian noise process whose power level is changing over time. In this generalized version of the original estimation problem, we might not always insist on having the best available signal estimate; instead, we may prefer to have a reasonably good estimate which can be produced at a modest computational cost. The associated estimation algorithm would require access to multiple HMM-based representations for the signal, each having a unique number of states and hence a unique degree of fidelity. This estimator would use a predetermined rule for optimally trading off computation for performance as a function of the current SNR. Future work could be aimed at developing the overall estimation algorithm for this more complex situation. One must determine, for example, how to estimate the current noise level, how to decide whether to switch from the current signal model to a different model (as well as how often this switching decision must be made), and how to select the best model among all available models.

Chapter 6

Summary and Future Directions

6.1 Synopsis

The central goal of this thesis has been to develop a new statistical framework for analyzing and processing stationary non-Gaussian signals. A unifying theme of the concepts presented in the thesis has been our consideration of two fundamental inference problems that often arise in practical situations, namely (i) source identification (i.e., estimation of the parameters of a signal source based on a mathematical model of the source and an uncorrupted observation of the source output); and (ii) signal estimation (i.e., recovery of the signal values themselves based on complete parametric knowledge of the measurement model and a noisy observation of the signal). The results following from our analyses of these two inference problems constituted the technical core of the thesis.

The main body of technical material, which was presented in Chapters 2 through 5, was logically divided into two parts, according to the type of mathematical model that was used to define the structure of the source signal. The first part, which consisted solely of Chapter 2, dealt with the two basic inference problems under the assumption that the signal was produced by an ARGMIX source (i.e., an autoregressive LTI system driven by i.i.d. Gaussian-mixture noise). The second part, which consisted of Chapters 3, 4, and 5, developed an entirely new signal model in order to overcome some of the computational difficulties imposed by the ARGMIX assumption. In this part of the thesis we developed the notion that a stationary non-Gaussian signal could be approximated by a finite-state hidden Markov model (HMM); we then showed how such an approximation could be manipulated to produce accurate and efficient solutions to the two basic inference problems.

Through our investigation of these alternative signal models, we developed a new set of concepts and techniques for dealing with non-Gaussian problems, and we encountered a number of potentially rich areas for further exploration. In the remainder of this chapter, we provide a summary of the main contributions of the thesis and suggest several topics for future investigation. A number of open issues and potential research topics have already been identified in the individual chapter discussions and therefore will not be repeated here.

6.2 Summary of Thesis Contributions

6.2.1 Development of ARGMIX Parameter Identification Algorithm

We developed a new iterative technique for identifying the parameters of an ARGMIX process based on a finite-length realization of such a process. This technique, which we refer to as the EMAX algorithm, was derived using the generalized expectation-maximization principle. The strength of the EMAX algorithm lies in its ability to identify both the shape of the driving-noise pdf and the LTI system that gave rise the signal. We demonstrated in several numerical examples that the estimation performance of the EMAX algorithm was superior not only to that of traditional least-squares techniques, but also to that of other existing algorithms that are also based on the ARGMIX signal model. We also developed an alternative form of the EMAX algorithm to solve a restricted version of the original ARGMIX source identification problem. The assumptions of this restricted problem were more closely matched to those of the classical AR Gaussian source identification problem, in that the basic shape of the driving-noise pdf was assumed known but the scale of the pdf was assumed unknown. Although this alternative version of the EMAX algorithm required more a priori signal information than the original, it had better convergence properties because it was designed to operate on a likelihood function that had no singularities.

6.2.2 Formulation of HMM-Based Signal Approximation Concept

We formulated and developed the novel concept that an arbitrary stationary AR signal could be approximated as the output of a finite-state hidden Markov model. The HMM-based signal model was introduced as an alternative to the ARGMIX model to reduce the computational burden incurred under the ARGMIX assumption; a reduction in computation was deemed possible because an HMM has a simple probabilistic structure that can be specified using only a small number of parameters. To develop this new model, we considered the optimization problem in which exact knowledge of the true signal pdf is given and the best HMM-based approximation to this pdf is to be found. The optimization was carried out under the constraint that the states of the underlying Markov chain represent a collection of disjoint regions making up a partition of the original state space. Using the Kullback-Leibler distance as our figure of merit, we first derived optimal parameter values for the approximating HMM directly in terms of the true signal pdf, under the assumption that the state-space partition was fixed. We then showed that the best partition was the one that maximized the mutual information between state values of the underlying Markov chain at successive time steps. Although most of our initial analysis assumed that the true signal was a first-order AR process, we also showed that the same basic results applied to higher-order AR processes.

6.2.3 Development of HMM Parameter Identification Algorithm

We constructed a practical iterative algorithm for estimating the parameters of an optimal HMM-based approximation of a stationary AR process based only on a finite-length realization of such a process, rather than on a complete description of its pdf. This algorithm can thus be viewed as an approximate solution to the general AR source identification problem. The algorithm was configured to select a feasible initial partition of the original state space, then to iteratively adjust the region boundaries of the partition until the optimal partition is reached, and finally to compute the HMM parameter estimates based on the distribution of data points among the resulting regions. The basic ideas that guided the iterative search portion of the algorithm were based on the theoretical results derived in the HMM-based signal approximation problem. Although several techniques have been developed by other researchers for estimating the parameters of an HMM, these techniques are typically designed to optimize a likelihood-based criterion rather than a mutual information criterion; moreover, they are not equipped to handle the state-space partitioning constraint.

6.2.4 Development of HMM-Based Signal Estimation Techniques

We developed a collection of techniques for performing MMSE signal estimation based on the assumption that both the signal and noise processes are outputs of finite-state HMMs. These techniques also relied heavily on the assumption that the pdf associated with any state of either HMM is a Gaussian mixture. We began our development by constructing a smoothing algorithm for the simple case in which the signal and noise were, respectively, colored and white Gaussian processes. For this case, we also evaluated estimation performance as a function of the order of the HMM-based approximation and compared the results to those of the globally optimal Wiener smoother; we found that near-optimal performance could be achieved when this HMM contained only a small number of states. We then extended the basic smoothing algorithm so that it applied to the case in which both the signal and noise were allowed to be colored non-Gaussian processes. In addition, we indicated how similar algorithms could be developed for the problems of filtering and prediction. These HMM-based estimation algorithms are quite general and powerful signal processing tools; they consume only a modest amount of computation and can be applied in an extremely broad range of non-Gaussian problems.

6.3 Directions for Future Research

6.3.1 Consideration of More Realistic Inference Problems

Clearly, the problems of source identification and signal estimation, as they have been defined in the thesis, are idealized versions of more complicated problems encountered in practice. There are many practical situations, for example, in which we would like to identify the parameters of a signal, but we have only noisy observations of the signal available to carry out the identification procedure. On the other hand, there are also situations in which we would like to estimate a signal in additive noise, but we have only partial knowledge of the parametric measurement model. Both types of situations call for the solution of a joint inference problem involving aspects of both source identification and signal estimation. Consideration of such a problem was well beyond the scope of the thesis, mainly because it was too complex to make a convenient starting point for the development of an inference framework. Now that we have made some initial progress on the idealized inference problems, however, it appears that a logical direction for further investigation is to address more realistic versions of these problems using either of the two signal models we have introduced.

6.3.2 Streamlining of HMM-Based Signal Estimation Algorithm

An area in which future research will almost certainly be beneficial is the streamlining of computation in the basic HMM-based signal estimation algorithm described in Chapter 5. In several examples presented throughout the thesis, we have observed that an HMM-based approximation of a random process can typically be described using only a small number of non-zero state transition probabilities. That is, upon constructing an approximation with the method described in Chapter 4, we have found that a transition of the original state vector from, say, time t to time $t+1$ often begins and ends in adjacent regions (or in the same region) within the optimal state-space partition. Moreover, state-vector transitions between regions that are far apart tend to occur with either negligible or zero probability. However, since this special attribute of a typical state trajectory has not been exploited in the basic signal estimation algorithm, it is likely that the algorithm performs many unnecessary computations to produce the final signal estimate. With a modest amount of work, one could develop a more sophisticated algorithm which spends computation only to deal with state trajectories that have non-negligible probability. In certain cases where the state transition matrix of the approximating Markov chain is very sparse, it is conceivable that the computational cost of the algorithm could be reduced from $\mathcal{O}(L^2)$ to $\mathcal{O}(L)$.

6.3.3 Further Development of HMM-Based Approximation Concept

Although the notion of representing random signals approximately using finite-state HMMs appears to hold enormous potential, we have taken only the first steps toward exploiting this concept in the thesis. A number of theoretical issues must still be resolved before the HMM-based signal model can serve as a basis for routine signal processor design. It is easy to imagine a complex signal processing or decision system in which signals are optimally represented as finite-state HMMs, as we have discussed earlier. Within such a system, a particular signal may undergo a variety of known, well defined transformations, e.g., it may pass through a linear system, become corrupted by noise, or perhaps be subjected to a memoryless nonlinearity. For each stage of processing within the system, it would be useful to know precisely how to best represent the output of the transformation as an HMM when we are given an optimal representation of the input as an HMM. Moreover, in cases where two random processes become combined during the transformation (e.g., signal and noise), we would like to know how to jointly design the HMM-based representations of both processes so that overall system performance is optimized. (Recall that in Chapter 5, we merely combined the existing HMMs for signal and noise to obtain a new, more complex HMM for the observation; however, the two original HMMs had been optimized individually, rather than jointly.) If problems such as these can be solved through further research, then many functions carried out within a complex signal processing system could ultimately be cast in terms of optimal operations on HMMs. Because an HMM has a particularly simple stochastic structure, we expect that such operations would be fairly straightforward to derive.

6.3.4 Application of HMMs to Other Signal Processing Problems

Our investigation of the HMM-based approach to non-Gaussian inference problems was necessarily limited in scope; specifically, we restricted our attention to the two basic problems of source identification and signal estimation. Furthermore, for the signal estimation problem in particular, we focused almost exclusively on the development of a smoothing algorithm. As we pointed out in Chapter 5, however, filtering and prediction algorithms could be developed in a similar manner. We also provided a fairly detailed outline indicating how HMMs could be used to solve detection and classification problems efficiently. Still, there remain many signal processing problems in which the HMM paradigm could be successfully applied. Thus, another potentially fruitful direction for future work is to develop HMM-based solutions to problems such as deconvolution (in which the distorting system may be either known or unknown), joint detection and estimation, signal enhancement, signal quantization, or compression.

Appendix A

Notational Conventions and Abbreviations

The notational conventions and abbreviations used in the thesis are generally introduced and explained in detail as they are needed. For convenient reference, we summarize some of the most important symbols and abbreviations below. These definitions should be assumed to hold unless otherwise stated.

A.1 Abbreviations

AR	→	autoregressive
ARGMIX	→	autoregressive Gaussian-mixture
ASK	→	amplitude-shift keying
cdf	→	cumulative distribution function
EM	→	expectation-maximization
FIR	→	finite impulse response
GEM	→	generalized expectation-maximization
HMM	→	hidden Markov model
HOS	→	higher-order statistics
i.i.d.	→	independent and identically distributed
IIR	→	infinite impulse response
ISI	→	intersymbol interference
LRT	→	likelihood ratio test
LTI	→	linear and time invariant
ML	→	maximum likelihood

MAP	→	maximum a posteriori
MMSE	→	minimum mean squared error
MSE	→	mean squared error
pdf	→	probability density function
pmf	→	probability mass function
psd	→	power spectral density
SER	→	signal-to-error ratio
SNR	→	signal-to-noise ratio

A.2 Notational Conventions

$\mathbb{R}$	→	the set of real numbers
$\mathbb{R}^+$	→	the set of nonnegative real numbers
$\mathbb{R}^n$	→	the set of n-dimensional real-valued tuples
$\mathbb{Z}$	→	the set of integers
$\mathbb{Z}^+$	→	the set of nonnegative integers
$\varnothing$	→	the empty set
$\cup$	→	union operator for sets
$\cap$	→	intersection operator for sets
$\log x$	→	the natural logarithm of x
$\exp x$	→	the exponential e^x
$\Pr\{\mathcal{A}\}$	→	probability of the event $\mathcal{A}$
$\Pr\{\mathcal{A},\mathcal{B}\}$	→	joint probability of the events $\mathcal{A}$ and $\mathcal{B}$
$\Pr\{\mathcal{A}\|\mathcal{B}\}$	→	probability of the event $\mathcal{A}$ conditioned on the event $\mathcal{B}$
$E\{Y\}$	→	expected value of the random variable Y
$E\{Y\|Z=z\}$	→	expected value of Y conditioned on the event $Z=z$
$\hat{Y}$	→	estimate of the random variable Y
$\{Y_t\}$	→	discrete-time random process
$\{\tilde{Y}_t\}$	→	approximation of the random process $\{Y_t\}$
$\mathbf{Y}_{i:j}$	→	the vector $(Y_i, Y_{i+1}, \cdots, Y_j)$ if $i<j$ or the vector $(Y_i, Y_{i-1}, \cdots, Y_j)$ if $i>j$
$f_Y(\cdot)$	→	pdf of random variable Y

$f_Y(\cdot;\mathbf{\Psi})$ → pdf of Y which depends on the parameter vector $\mathbf{\Psi}$

$f_{Y,Z}(\cdot,\cdot)$ → joint pdf of random variables Y and Z

$f_{Y|Z}(\cdot|Z=z)$ → conditional pdf of Y given the event $Z=z$

$\mathcal{N}(\cdot;\mu,\sigma)$ → Gaussian pdf with mean μ and standard deviation σ

$\mathcal{D}(f_X,f_Y)$ → Kullback-Leibler distance between densities $f_X(\cdot)$ and $f_Y(\cdot)$

$I(X,Y)$ → mutual information between the random variables X and Y

$\max\limits_{x\in\mathcal{P}}\{h(x)\}$ → largest value of the function $h(\cdot)$ on the set $\mathcal{P}$

$\min\limits_{x\in\mathcal{P}}\{h(x)\}$ → smallest value of the function $h(\cdot)$ on the set $\mathcal{P}$

$\arg\max\limits_{x\in\mathcal{P}}\{h(x)\}$ → element in $\mathcal{P}$ yielding the largest value of $h(\cdot)$

$\arg\min\limits_{x\in\mathcal{P}}\{h(x)\}$ → element in $\mathcal{P}$ yielding the smallest value of $h(\cdot)$

A.3 Context-Specific Symbols

$\mathbf{X}_t$ → state vector of a dynamical system at time t

Y_t → source signal at time t

Z_t → observed signal at time t

W_t → driving noise of a dynamical system at time t

V_t → additive observation noise at time t

$\mathbf{\Psi}$ → parameter vector characterizing a pdf or a dynamical system

K → order of an autoregressive process

L → number of states in a Markov chain or an HMM

M → number of components in a Gaussian-mixture pdf

N → number of samples in a finite-length observation

Φ_t → component of a Gaussian mixture selected at time t

Θ_t → state variable for a Markov chain at time t

$P(i)$ → initial state probability $\Pr\{\Theta_0=i\}$

$Q(i,j)$ → state transition probability $\Pr\{\Theta_t=j|\Theta_{t-1}=i\}$

$R(i,j)$ → joint state probability $\Pr\{\Theta_{t-1}=i,\Theta_t=j\}$

$g_i(\cdot)$ → HMM output density $f_{\bar{Y}_t|\Theta_t}(\cdot|\Theta_t=i)$

$f_i(\cdot)$ → HMM output density $f_{\bar{\mathbf{X}}_t|\Theta_t}(\cdot|\Theta_t=i)$

Appendix B

Maximization of a Function Related to Cross-Entropy

In this appendix, we derive solutions to two closely related optimization problems that arise repeatedly throughout the thesis. One of these problems deals with finite-length tuples whose elements are positive real numbers, and the other deals with positive functions of a real variable. In the first optimization problem, we are given an M-dimensional tuple $\mathbf{a} = (a_1, a_2, \cdots, a_M)$ whose elements are all positive real numbers, and we seek the tuple $\mathbf{b}^* = (b_1^*, b_2^*, \cdots, b_M^*)$ specified implicitly through the maximization

$$\mathbf{b}^* = \arg\max_{\mathbf{b}\in\mathcal{B}} \sum_{k=1}^{M} a_k \log b_k, \tag{B.1}$$

where $\mathcal{B}$ represents the set of all tuples $\mathbf{b} = (b_1, b_2, \cdots, b_M)$ whose elements are positive and satisfy the constraint

$$\sum_{k=1}^{M} b_k = 1. \tag{B.2}$$

For this discrete case, we will prove that the elements $b_1^*, b_2^*, \cdots, b_M^*$ of the optimal tuple are given by

$$b_k^* = \frac{a_k}{\sum_{j=1}^{M} a_j}, \qquad k = 1, 2, \cdots, M. \tag{B.3}$$

In the second optimization problem, we are given a real-valued continuous function $a(\cdot)$ which is strictly positive on the open interval (x_1, x_2) and has the property that

$$0 < \int_{x_1}^{x_2} a(x)\, dx < \infty. \tag{B.4}$$

The function we seek, which we denote by $b^*(\cdot)$, is specified implicitly as the solution to the maximization problem

$$b^*(\cdot) = \arg\max_{b(\cdot)\in C} \int_{x_1}^{x_2} a(x) \log b(x)\, dx, \tag{B.5}$$

where C is the set of all real-valued, continuous, strictly positive functions $b(\cdot)$ defined on (x_1, x_2) that satisfy the constraint

$$\int_{x_1}^{x_2} b(x)\, dx = 1. \tag{B.6}$$

For this continuous case, we will prove that the maximizing function $b^*(\cdot)$ is given by

$$b^*(x) = \frac{a(x)}{\int_{x_1}^{x_2} a(u)\, du}, \qquad x_1 < x < x_2. \tag{B.7}$$

Before proceeding with our proofs, we remark that, although both of the assertions above are made with the assumptions of strict positivity on the variables involved, analogous proofs can easily be constructed when the variables are taken to be merely nonnegative.

Let us denote by $\tilde{\mathbf{a}} = (\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_M)$ the normalized version of the given tuple $\mathbf{a}$, so that the elements of $\tilde{\mathbf{a}}$ are defined by

$$\tilde{a}_k = \frac{a_k}{\sum_{j=1}^{M} a_j}, \qquad k = 1, 2, \cdots, M. \tag{B.8}$$

Clearly, we have that $\tilde{\mathbf{a}} \in \mathcal{B}$. Our strategy in proving that $\tilde{\mathbf{a}}$ is the unique solution to the maximization taken in (B.1) will be to show that

$$\sum_{k=1}^{M} a_k \log b_k \leq \sum_{k=1}^{M} a_k \log \tilde{a}_k \tag{B.9}$$

for all $\mathbf{b} \in \mathcal{B}$, and furthermore that equality holds in this expression if and only if $\mathbf{b} = \tilde{\mathbf{a}}$.

A proof of the above inequality can be developed by using certain key properties of the function $g(x) = x \log x$, which is defined for $x > 0$. Consider the second-order Taylor series expansion of $g(\cdot)$ about the point x_0, given by

$$g(x) = g(x_0) + g'(x_0)(x - x_0) + \tfrac{1}{2} g''(x^*)(x - x_0)^2, \tag{B.10}$$

where x^* is a number lying between x and x_0 whose value is dependent on both x and x_0 (although we have not indicated this dependence explicitly) and whose existence is guaranteed by Taylor's theorem [160]. Observe that because

$$g'(x) = \frac{d}{dx}(x \log x) = 1 + \log x \tag{B.11}$$

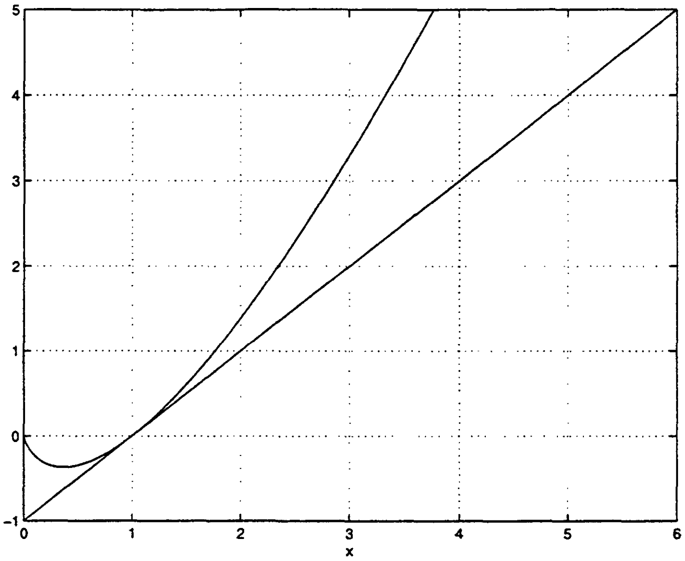


Figure B-1: Plots of $g(x) = x \log x$ (upper curve) and $h(x) = x - 1$.

and

$$g''(x) = \frac{d}{dx}(1 + \log x) = \frac{1}{x}, \tag{B.12}$$

we have that $g''(x) > 0$ whenever $x > 0$. This implies that the final term $\frac{1}{2}g''(x^*)(x - x_0)^2$ in the Taylor series expansion given above is always nonnegative, and in fact is zero only in the case when $x = x_0$. If we now rewrite the Taylor series expansion for the particular value $x_0 = 1$, making use of the facts that $g(1) = 0$, $g'(1) = 1$, and $g''(x^*) = \epsilon > 0$, we obtain

$$x \log x = (x - 1) + \tfrac{1}{2}\epsilon(x - 1)^2. \tag{B.13}$$

From this representation we conclude that

$$x \log x \geq x - 1 \tag{B.14}$$

with equality if and only if $x = 1$. This inequality provides a foundation for the proof of (B.9). In Figure B-1, we show plots of the functions $g(x) = x \log x$ and $h(x) = x - 1$ in the vicinity of the point $x_0 = 1$.

If we now replace the positive variable x in (B.14) with the positive ratio $\tilde{a}_k / b_k$, we obtain the expression

$$\frac{\tilde{a}_k}{b_k} \log \frac{\tilde{a}_k}{b_k} \geq \frac{\tilde{a}_k}{b_k} - 1, \tag{B.15}$$

or equivalently, after multiplying both sides by b_k,

$$\tilde{a}_k \log \frac{b_k}{\tilde{a}_k} \geq \tilde{a}_k - b_k, \tag{B.16}$$

which holds with equality if and only if $b_k = \tilde{a}_k$. This last expression actually represents not one but M distinct inequalities, one for each value of the tuple index k. The left-hand and right-hand sides of these M inequalities may then be separately summed to yield

$$\begin{aligned} \sum_{k=1}^{M} \tilde{a}_k \log \frac{b_k}{\tilde{a}_k} &\geq \sum_{k=1}^{M} b_k - \sum_{k=1}^{M} \tilde{a}_k & \text{(B.17)} \\ &= 1 - 1 & \text{(B.18)} \\ &= 0, & \text{(B.19)} \end{aligned}$$

which holds with equality if and only if all of the constituent inequalities hold with equality, i.e., if and only if $b_k = \tilde{a}_k$ for $k = 1, 2, \cdots, M$. If in (B.19) we write the logarithm of the ratio as a difference of logarithms, we obtain

$$\sum_{k=1}^{M} \tilde{a}_k \log b_k \leq \sum_{k=1}^{M} \tilde{a}_k \log \tilde{a}_k. \tag{B.20}$$

Finally, upon multiplying both sides of this expression by the positive quantity $\sum_{j=1}^{M} a_k$, which removes the normalization from the coefficients $\tilde{a}_k$ in each summation, we obtain

$$\sum_{k=1}^{M} a_k \log b_k \leq \sum_{k=1}^{M} a_k \log \tilde{a}_k, \tag{B.21}$$

which is what we wished to show.

To prove the analogous result in the continuous case, we need to revisit the original Taylor series expansion given in (B.13). In particular, from (B.13) we have

$$\frac{\tilde{a}(x)}{b(x)} \log \frac{\tilde{a}(x)}{b(x)} = \left(\frac{\tilde{a}(x)}{b(x)} - 1 \right) + \tfrac{1}{2}\epsilon(x) \left(\frac{\tilde{a}(x)}{b(x)} - 1 \right)^2, \tag{B.22}$$

where $\tilde{a}(\cdot)$ is the normalized function given by

$$\tilde{a}(x) = \frac{a(x)}{\int_{x_1}^{x_2} a(u)\, du}, \qquad x_1 < x < x_2, \tag{B.23}$$

and $\epsilon(x)$ is a positive function whose existence is guaranteed by Taylor's theorem. Upon multiplying both sides of (B.22) by $b(x)$ and then integrating from x_1 to x_2, we obtain

$$\int_{x_1}^{x_2} \tilde{a}(x) \log \frac{\tilde{a}(x)}{b(x)}\, dx = \int_{x_1}^{x_2} \tilde{a}(x)\, dx - \int_{x_1}^{x_2} b(x)\, dx + \int_{x_1}^{x_2} \tfrac{1}{2}\epsilon(x) b(x) \left(\frac{\tilde{a}(x)}{b(x)} - 1 \right)^2 dx$$

$$= 1 - 1 + \int_{x_1}^{x_2} \tfrac{1}{2}\epsilon(x)b(x)\left(\frac{\tilde{a}(x)}{b(x)} - 1\right)^2 dx \tag{B.24}$$

$$= \int_{x_1}^{x_2} \tfrac{1}{2}\epsilon(x)b(x)\left(\frac{\tilde{a}(x)}{b(x)} - 1\right)^2 dx, \tag{B.25}$$

or, after a bit of straightforward algebraic rearrangement,

$$\int_{x_1}^{x_2} a(x)\log\tilde{a}(x)\,dx = \int_{x_1}^{x_2} a(x)\log b(x)\,dx + A\int_{x_1}^{x_2} \tfrac{1}{2}\epsilon(x)b(x)\left(\frac{\tilde{a}(x)}{b(x)} - 1\right)^2 dx, \tag{B.26}$$

where $A = \int_{x_1}^{x_2} a(u)\,du$ is the positive normalizing term taken from $\tilde{a}(\cdot)$. Note that the second term on the right-hand side of this last expression is always nonnegative. Moreover, owing to the continuity of $\tilde{a}(\cdot)$ and $b(\cdot)$, this term is zero if and only if $\tilde{a}(x) = b(x)$ for all $x \in (x_1, x_2)$. This observation gives the desired result that

$$\int_{x_1}^{x_2} a(x)\log b(x)\,dx \leq \int_{x_1}^{x_2} a(x)\log\tilde{a}(x)\,dx, \tag{B.27}$$

with equality if and only if $\tilde{a}(x) = b(x)$.

Appendix C

Computer Implementation of the EMAX Algorithm

The following source code listing, which is written in the MATLAB programming language, represents one implementation of the EMAX algorithm derived in Chapter 2.

```
function [mu,sig,rho,a] = EMAX(y,mu0,sig0,rho0,a0,tol,n_iter)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%   Description of variables
%%   ------------------------
%%
%%   Input arguments:
%%
%%   y ........ vector of observations to be processed
%%   mu0 ...... vector of initial values for means
%%   sig0 ..... vector of initial values for standard deviations
%%   rho0 ..... vector of initial values for weighting coefficients
%%   a0 ....... vector of initial values for autoregressive parameters
%%   tol ...... numerical tolerance for terminating algorithm
%%   n_iter ... number of iterations for coordinate ascent
%%
%%   Output arguments:
%%
%%   mu ...... vector of final estimates for means
%%   sig ..... vector of final estimates for standard deviations
%%   rho ..... vector of final estimates for weighting coefficients
%%   a ....... vector of final estimates for autoregressive parameters
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%   Compute dimension of autoregression vector, number of components
%%   in Gaussian-mixture pdf, and number of input observations.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
K = length(a0);
M = length(mu0);
N = length(y);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%   Construct convolution matrix from input data for efficient
%%   implementation of FIR filtering operation, and modify observation
%%   sequence accordingly.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

H = toeplitz(y(K:1:N-1),y(K:-1:1));
y = y(K+1:N);
N = length(y);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%   Initialize parameter values and create parameter vector psi.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a = a0;
mu = mu0;
sig = sig0;
rho = rho0;
psi = [a' mu' sig' rho'];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%   Begin loop to iterate formulas for EMAX algorithm until
%%   convergence is obtained.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

err = tol + 1;
while (err > tol),

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%   Process observations with inverse filter and make array of
%%   mean-removed residual sequences for all classes.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  z = y - H*a;
  zmu = z*ones(1,M)-ones(N,1)*mu';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%   Compute posterior class probabilities.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  P = ones(N,1)*((rho./sig)') .* ...
      exp(-0.5*(zmu.^2).*(ones(N,1)*(1./sig').^2));
  P = P./(sum(P')'*ones(1,M));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%   Update estimates for means, standard deviations, and AR
%%   parameters via coordinate ascent by iterating update formulas
%%   n_iter times.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  for k = 1:n_iter,
    z = y - H*a;
    mu = ((sum(P.*(z*ones(1,M))))./sum(P))';
    zmu = z*ones(1,M)-ones(N,1)*mu';
    sig = sqrt(diag(zmu'*(P.*zmu))./sum(P)');
    wgt = sum((P.*(ones(N,1)*((1./sig').^2)))')';
    u = sum(((P.*(y*ones(1,M)-ones(N,1)*mu'))./(ones(N,1)*((sig').^2)))')';
    a = inv((H'.*(ones(K,1)*wgt'))*H)*H'*u;
  end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%   Update estimates for weighting coefficients.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  rho = mean(P)';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%   Reassign parameter vector psi and compute Euclidean distance
%%   between this value and the previous value.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  psi_old = psi;
  psi = [a' mu' sig' rho'];
  err = norm(psi_old - psi);
end
```

Appendix D

Relationship of Kullback-Leibler Distance to Other Metrics

In this appendix, we demonstrate that the Kullback-Leibler distance is closely related to two other important statistical measures, namely probability of detection and log-likelihood. A discussion of these relationships gives us a better intuitive understanding — from the points of view of classical detection theory and parameter estimation — about the distributional properties that are actually measured by the Kullback-Leibler distance.

D.1 Connection with Probability of Detection

To describe the nature of the link between Kullback-Leibler distance and probability of detection, let us first recall our earlier discussion from Section 3.2.1 in which we cast the problem of quality assessment in terms of a particular type of game played with an independent observer. In this game, the observer was furnished with descriptions of two densities, and his goal was to correctly guess which of them gave rise to a given set of realizations. We, on the other hand, were able to select one of the two densities in advance from a given class $\mathcal{F}$; our goal was to choose a density that would confuse the observer more often than would any other density in $\mathcal{F}$. Because we knew that the observer would always use a statistical test that maximized the probability of a correct decision, we argued that the optimal density in $\mathcal{F}$ was that which yielded the minimum possible value of this probability.

Recall that one of the two components making up the probability of a correct decision is the probability of detection, or P_d, which is given by

$$\begin{aligned} P_d &= \Pr\{\text{Declare } H_1 | H_1 \text{ true}\} && \text{(D.1)} \\ &= \Pr\{\ell(\mathbf{Z}_{0:N-1}) \geq 0 | H_1 \text{ true}\} && \text{(D.2)} \\ &= \int_0^\infty f_{\ell|H_1}(\xi | H_1 \text{ true})\, d\xi. && \text{(D.3)} \end{aligned}$$

This component is a particularly important one to examine because it involves hypothesis H_1, which is the only hypothesis that ever occurs in an actual signal processing situation.

(That is, any observations that must be processed in a realistic scenario are precisely those that arise from the true source, and not from some approximation to the true source.) It may be just as reasonable, therefore, to define the quality of our approximation exclusively in terms of P_d as it was to define it in terms of the probability of a correct decision.

Note that P_d can be viewed as the area under the tail of the conditional density of $\ell(\mathbf{Z}_{0:N-1})$ given H_1. We can learn more about the specific shape of this distributional tail by rewriting the discrimination statistic $\ell(\mathbf{Z}_{0:N-1})$ as

$$\begin{aligned}
\ell(\mathbf{Z}_{0:N-1}) &= \log \frac{f_{\mathbf{Y}_{0:N-1}}(\mathbf{Z}_{0:N-1})}{f_{\tilde{\mathbf{Y}}_{0:N-1}}(\mathbf{Z}_{0:N-1})} && \text{(D.4)}\\
&= \log \frac{\prod_{t=0}^{N-1} f_Y(Z_t)}{\prod_{t=0}^{N-1} f_{\tilde{Y}}(Z_t)} && \text{(D.5)}\\
&= \log \prod_{t=0}^{N-1} \frac{f_Y(Z_t)}{f_{\tilde{Y}}(Z_t)} && \text{(D.6)}\\
&= \sum_{t=0}^{N-1} \log \frac{f_Y(Z_t)}{f_{\tilde{Y}}(Z_t)} && \text{(D.7)}\\
&= \sum_{t=0}^{N-1} \ell_t, && \text{(D.8)}
\end{aligned}$$

where we have used the definition

$$\ell_t = \log \frac{f_Y(Z_t)}{f_{\tilde{Y}}(Z_t)}. \tag{D.9}$$

We can easily see from this re-expression that $\ell(\mathbf{Z}_{0:N-1})$ is merely the sum of N i.i.d. of random variables $\{\ell_t\}_{t=0}^{N-1}$, which have been derived from the original set of i.i.d. observations $\{Z_t\}_{t=0}^{N-1}$. Therefore, if we assume that the number of observations, N, is very large, then we have by the Central Limit Theorem that $\ell(\mathbf{Z}_{0:N-1})$ behaves approximately like a Gaussian random variable [42, 55, 57].

Because $\ell(\mathbf{Z}_{0:N-1})$ is ultimately compared to a threshold of zero in the optimal test, we now introduce a more convenient, normalized version of this statistic given by

$$\tilde{\ell}(\mathbf{Z}_{0:N-1}) = \frac{1}{N}\ell(\mathbf{Z}_{0:N-1}) = \frac{1}{N}\sum_{t=1}^{N} \ell_t. \tag{D.10}$$

Since this normalized statistic will also be approximately Gaussian for large N, we write

$$f_{\tilde{\ell}|H_1}(\xi | H_1 \text{ true}) \approx \mathcal{N}(\xi; \mu, \sigma), \tag{D.11}$$

where the mean and standard deviation parameters, μ and σ, are understood to be dependent on the particular density $f_{\tilde{Y}}(\cdot)$ that is selected from $\mathcal{F}$.

To simplify the remaining exposition, let us now suppose that, as we vary $f_{\tilde{Y}}(\cdot)$ over the entire set $\mathcal{F}$, the range of values taken by σ is extremely small in comparison to the range of values taken by μ, so that σ can be considered essentially constant. Then, if we wish to confuse the observer by minimizing his probability of detection (i.e., by putting as little area as possible under the positive tail of $f_{\tilde{\ell}|H_1}(\cdot)$), we need only choose the density in $\mathcal{F}$ associated with the smallest value of μ. But observe that the value of μ is given by

$$\begin{aligned}
\mu &= E\{\tilde{\ell}(\mathbf{Z}_{0:N-1})|H_1 \text{ true}\} && \text{(D.12)}\\
&= E\left\{\frac{1}{N}\sum_{t=0}^{N-1}\ell_t \,\middle|\, H_1 \text{ true}\right\} && \text{(D.13)}\\
&= E\{\ell_0|H_1 \text{ true}\} && \text{(D.14)}\\
&= E\left\{\log\frac{f_Y(Z_0)}{f_{\tilde{Y}}(Z_0)} \,\middle|\, H_1 \text{ true}\right\} && \text{(D.15)}\\
&= \int_{-\infty}^{\infty} f_Y(y)\log\frac{f_Y(y)}{f_{\tilde{Y}}(y)}\,dy && \text{(D.16)}\\
&= D(f_Y, f_{\tilde{Y}}). && \text{(D.17)}
\end{aligned}$$

In other words, μ is precisely the Kullback-Leibler distance between the true and approximate densities. We conclude, therefore, under the assumptions stated above, that as the number of observations becomes very large, minimizing the Kullback-Leibler distance between f_Y and $f_{\tilde{Y}}$ is approximately equivalent to minimizing the observer's probability of detection.

D.2 Connection with Log-Likelihood

We now show that there is a direct connection between Kullback-Leibler distance and log-likelihood, in the sense that minimizing the Kullback-Leibler distance to obtain the best parametric description of a pdf is equivalent to maximizing the log-likelihood function in a closely related problem. Before describing the relationship between these two measures more precisely, let us first consider the two optimization problems that link them. These problems may be stated as follows:

(i) *(Minimization of Kullback-Leibler Distance.)* Let Y be a discrete-valued random variable which is distributed on the finite set $\{1, 2, \cdots, L\}$ according to the pmf $f_Y(\cdot)$. Assume that the pmf values $\{f_Y(j)\}_{j=1}^{L}$ are given. Let $\mathcal{F}$ be a parameterized set of pmfs defined by $\mathcal{F} = \{f_{\tilde{Y}}(\cdot; \Psi) \mid \Psi \in \mathcal{P}\}$, where $\mathcal{P}$ represents a collection of admissible parameter values. Determine the set of all parameter values $\Psi_{\text{KL}} \in \mathcal{P}$ that satisfy

$$\Psi_{\text{KL}} = \arg\min_{\Psi\in\mathcal{P}} \sum_{j=1}^{L} f_Y(j)\log\frac{f_Y(j)}{f_{\tilde{Y}}(j;\Psi)}. \qquad \text{(D.18)}$$

(ii) *(Maximization of Likelihood Function.)* Let $\{Y_t\}_{t=0}^{N-1}$ be a set of i.i.d. discrete-valued random variables, each distributed on the finite set $\{1, 2, \cdots, L\}$ according to the pmf

$f_Y(\cdot)$, and let $\{y_t\}_{t=0}^{N-1}$ be a corresponding set of realizations of these random variables. Assume that the pmf values $\{f_Y(j)\}_{j=1}^{L}$ are unknown, but that the parameterized collection of pmfs $\mathcal{F} = \{f_{\tilde{Y}}(\cdot;\boldsymbol{\Psi}) \mid \boldsymbol{\Psi} \in \mathcal{P}\}$ is hypothesized (possibly incorrectly) to contain $f_Y(\cdot)$. Determine the set of all parameter values $\boldsymbol{\Psi}_{\text{ML}} \in \mathcal{P}$ that satisfy

$$\boldsymbol{\Psi}_{\text{ML}} = \arg\max_{\boldsymbol{\Psi}\in\mathcal{P}} \prod_{t=0}^{N-1} f_{\tilde{Y}}(y_t;\boldsymbol{\Psi}) \tag{D.19}$$

We will demonstrate that, as $N \to \infty$ in problem (ii), the sets $\{\boldsymbol{\Psi}_{\text{ML}}\}$ and $\{\boldsymbol{\Psi}_{\text{KL}}\}$ become identical, i.e., any parameter value that asymptotically maximizes the hypothesized likelihood function also minimizes the Kullback-Leibler distance between the hypothesized pmf and the true pmf, and vice versa.

To establish this result, it will be convenient to introduce a function known as the empirical pmf (also sometimes termed the *type* [41]), which we denote by $\hat{f}_Y(\cdot)$. In the context of problem (ii) above, the empirical pmf is defined in terms of a particular realization $\mathbf{y}_{0:N-1}$ of the random vector $\mathbf{Y}_{0:N-1}$ according to the formula

$$\hat{f}_Y(j) = \frac{1}{N}\sum_{t=0}^{N-1} \delta_{j,y_t} \qquad j = 1, 2, \cdots, L, \tag{D.20}$$

where $\delta_{j,k}$ is the Kronecker delta function, defined by

$$\delta_{j,k} = \begin{cases} 1 & \text{if } j = k, \\ 0 & \text{otherwise,} \end{cases} \tag{D.21}$$

The empirical pmf may be viewed alternatively as a histogram having L distinct bins (one for each of the L symbols that could occur within the realization) whose bin totals have been normalized so that they sum to one; in other words, the empirical pmf merely provides a record of the relative frequency of occurrence of each symbol.

We first seek to establish that the maximum likelihood parameter estimation problem given in (D.19) explicitly involves the empirical pmf. Observe that by taking the logarithm of the right-hand side of (D.19) and then pre-multiplying by a factor of $1/N$, we arrive at an equivalent expression of the maximum-likelihood problem given by

$$\hat{\boldsymbol{\Psi}} = \arg\max_{\boldsymbol{\Psi}\in\mathcal{P}} \frac{1}{N}\sum_{t=0}^{N-1} \log f_{\tilde{Y}}(y_t;\boldsymbol{\Psi}). \tag{D.22}$$

Now, since any given observation y_t must assume exactly one of L values, each of the N terms in the above summation can be placed into one of L categories according to its value. After all of the terms have been categorized in this way, the jth category would then contain a total of $\sum_{t=0}^{N-1} \delta_{j,y_t}$ terms. This suggests that we can rewrite the above log-likelihood function as

$$
\begin{aligned}
\frac{1}{N}\sum_{t=0}^{N-1}\log f_{\tilde{Y}}(y_t;\boldsymbol{\Psi}) &= \sum_{j=1}^{L}\left(\frac{1}{N}\sum_{t=0}^{N-1}\delta_{j,y_t}\right)\log f_{\tilde{Y}}(j;\boldsymbol{\Psi}) && \text{(D.23)}\\
&= \sum_{j=1}^{L}\hat{f}_Y(j)\log f_{\tilde{Y}}(j;\boldsymbol{\Psi}). && \text{(D.24)}
\end{aligned}
$$

Therefore, we can express (D.19) equivalently as

$$
\hat{\boldsymbol{\Psi}} = \arg\max_{\boldsymbol{\Psi}\in\mathcal{P}}\sum_{j=1}^{L}\hat{f}_Y(j)\log f_{\tilde{Y}}(j;\boldsymbol{\Psi}). \tag{D.25}
$$

As an intermediate step, let us now examine the Kullback-Leibler distance between the empirical pmf $\hat{f}_Y(\cdot)$ and the hypothesized pmf $f_{\tilde{Y}}(\cdot;\boldsymbol{\Psi})$, which is given by

$$
\begin{aligned}
D(\hat{f}_Y, f_{\tilde{Y}}) &= \sum_{j=1}^{L}\hat{f}_Y(j)\log\frac{\hat{f}_Y(j)}{f_{\tilde{Y}}(j;\boldsymbol{\Psi})} && \text{(D.26)}\\
&= \sum_{j=1}^{L}\hat{f}_Y(j)\log\hat{f}_Y(j) - \sum_{j=1}^{L}\hat{f}_Y(j)\log f_{\tilde{Y}}(j;\boldsymbol{\Psi}) && \text{(D.27)}
\end{aligned}
$$

In this last equality, we note that the first term is a constant and is therefore entirely independent of the parameter $\boldsymbol{\Psi}$. Thus, in order to minimize the Kullback-Leibler distance given above, we may simply ignore the first term in (D.27) and then attempt to maximize the second term (after dropping the leading minus sign). But this once again yields the rule

$$
\hat{\boldsymbol{\Psi}} = \arg\max_{\boldsymbol{\Psi}\in\mathcal{P}}\sum_{j=1}^{L}\hat{f}_Y(j)\log f_{\tilde{Y}}(j;\boldsymbol{\Psi}), \tag{D.28}
$$

which is identical to (D.25). We have therefore shown that maximizing the log-likelihood function in (D.25) leads to precisely the same result as minimizing the Kullback-Leibler distance between the empirical pmf $\hat{f}_Y(\cdot)$ and the hypothesized pmf $f_{\tilde{Y}}(\cdot;\boldsymbol{\Psi})$.

But, in the limit as $N\to\infty$, we have that the empirical pmf $\hat{f}_Y(\cdot)$ converges in probability to the true distribution $f_Y(\cdot)$, as shown by

$$
\begin{aligned}
&\lim_{N\to\infty}E\left\{\left(\hat{f}_Y(j)-f_Y(j)\right)^2\right\}\\
&\quad= \lim_{N\to\infty}E\left\{\hat{f}_Y^2(j)-2\hat{f}_Y(j)f_Y(j)+f_Y^2(j)\right\}\\
&\quad= \lim_{N\to\infty}\left[E\left\{\left(\frac{1}{N}\sum_{t=0}^{N-1}\delta_{j,Y_t}\right)^2\right\}-2E\left\{\frac{1}{N}\sum_{t=0}^{N-1}\delta_{j,Y_t}\right\}f_Y(j)+f_Y^2(j)\right]
\end{aligned}
$$

$$\begin{aligned}
&= \lim_{N\to\infty}\left[\frac{1}{N^2}\sum_{t=0}^{N-1}\sum_{s=0}^{N-1} E\left\{\delta_{j,Y_t}\cdot\delta_{j,Y_s}\right\} - 2f_Y^2(j) + f_Y^2(j)\right] \\
&= \lim_{N\to\infty}\left[\frac{1}{N^2}\sum_{t}\sum_{s\neq t} E\left\{\delta_{j,Y_t}\right\} E\left\{\delta_{j,Y_s}\right\} + \frac{1}{N^2}\sum_{t} E\left\{\delta_{j,Y_t}^2\right\} - f_Y^2(j)\right] \\
&= \lim_{N\to\infty}\left[\frac{N^2-N}{N^2} f_Y^2(j) - \frac{1}{N} f_Y(j) - f_Y^2(j)\right] \\
&= \lim_{N\to\infty}\left[\frac{1}{N} f_Y(j)(1 - f_Y(j))\right] \\
&= 0.
\end{aligned} \tag{D.29}$$

We conclude, therefore, that choosing a parameter value that minimizes the Kullback-Leibler distance between the hypothesized pmf and the true pmf is equivalent to choosing a parameter value that maximizes the hypothesized likelihood function formed from an infinite number of independent realizations from the true pmf. This close connection between estimates formed by minimizing Kullback-Leibler distance and by maximizing the likelihood function has been recognized previously by several authors, including Kullback [106], Kriz and Talacko [103], Hartigan [73], and Akaike [7].

Appendix E

A Gradient-Descent Technique for Evaluating HMM Parameters

Throughout much of Chapter 3, we focused exclusively on finding an abstract, theoretical solution to our problem of finite-state signal approximation. In this appendix, we develop a simple gradient-based algorithm to implement our solution. The goal of this algorithm is to produce explicit numerical values for the parameters of the best HMM-based representation of an arbitrary stationary first-order AR process $\{Y_t\}$. We shall assume throughout the appendix that we are given the true bivariate signal pdf $f_{Y_0,Y_1}(\cdot)$, which, as we have demonstrated in Section 3.3, summarizes all information relevant to the search for the best HMM. Such an assumption entails no loss in generality, since this bivariate pdf can, at least in principle, be derived from a complete specification of the original AR process. Because the vector of breakpoints $\mathbf{d} = (d_0, d_1, \cdots, d_L)$ will ultimately determine the parameter values of the approximating HMM, we concentrate on finding the best value of $\mathbf{d}$ based on the given function $f_{Y_0,Y_1}(\cdot)$.

E.1 Formulation of the Optimization Problem

We begin by reviewing the basic components involved in the finite-state approximation problem, and by introducing a slight variation of our previously established notation for these components. Recall that the joint probability mass function characterizing the pair of successive state variables (Θ_0, Θ_1) in the approximating Markov chain is defined by the formula

$$R(i,j;\mathbf{d}) = \int_{d_{j-1}}^{d_j} \int_{d_{i-1}}^{d_i} f_{Y_0,Y_1}(y_0,y_1)\, dy_0\, dy_1, \tag{E.1}$$

where we have now shown an explicit dependence on the vector of breakpoints $\mathbf{d}$. In addition, the marginal pmf for the single state variable Θ_0 is given by

$$P(i;\mathbf{d}) = \int_{d_{i-1}}^{d_i} f_{Y_0}(y_0)\, dy_0, \tag{E.2}$$

or equivalently, via the joint pmf $R(\cdot;\mathbf{d})$, by the expression

$$P(i;\mathbf{d}) = \sum_{j=1}^{L} R(i,j;\mathbf{d}) = \sum_{j=1}^{L} R(j,i;\mathbf{d}). \tag{E.3}$$

Our objective is to find the particular value of $\mathbf{d}$ that maximize the mutual information between the state variables Θ_0 and Θ_1. The mutual information corresponding to a given value of $\mathbf{d}$ is defined by

$$I(\mathbf{d}) = \sum_{i=1}^{L}\sum_{j=1}^{L} R(i,j;\mathbf{d}) \log \frac{R(i,j;\mathbf{d})}{P(i;\mathbf{d})P(j;\mathbf{d})}. \tag{E.4}$$

Thus, we wish to solve the maximization problem

$$\mathbf{d}^* = \arg\max_{\mathbf{d}\in\mathcal{D}} I(\mathbf{d}), \tag{E.5}$$

where $\mathcal{D}$ can be viewed as a subset of $\mathbb{R}^{L+1}$ that enforces the strict ordering constraint

$$-\infty = d_0 < d_1 < d_2 < \cdots < d_L = \infty. \tag{E.6}$$

Once the optimal breakpoint vector $\mathbf{d}^*$ has been obtained, we can easily compute the critical joint state probabilities $\{R(i,j;\mathbf{d}^*)\}_{i,j=1}^{L}$ by numerically integrating the pdf $f_{Y_0,Y_1}(\cdot)$ over appropriate rectangular regions in $\mathbb{R}^2$, as suggested by (E.1). Similarly, the remaining HMM parameter values can also be easily computed.

E.2 Finding an Optimal Solution with a Classical Hill-Climbing Algorithm

Because the maximization problem in (E.5) cannot be solved in closed form, we will pursue a simple iterative, hill-climbing procedure based on the principle of steepest ascent. Such a procedure generates a sequence of breakpoint vector values $\{\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \mathbf{d}^{(2)}, \cdots\}$ according to the recursive formula

$$\mathbf{d}^{(s+1)} = \mathbf{d}^{(s)} + \lambda^{(s)}\nabla I(\mathbf{d}^{(s)}), \tag{E.7}$$

where s is the iteration index, ∇I is the mutual information gradient vector defined by

$$\nabla I(\mathbf{d}) = \left(\frac{\partial I}{\partial d_0}, \frac{\partial I}{\partial d_1}, \cdots, \frac{\partial I}{\partial d_L}\right), \tag{E.8}$$

and $\lambda^{(s)}$ is a real number chosen such that

$$\lambda_{(s)} = \arg\max_{\lambda\in\mathbb{R}} \left\{ I\left(\mathbf{d}^{(s)} + \lambda\nabla I(\mathbf{d}^{(s)})\right)\right\}. \tag{E.9}$$

The core of the steepest-ascent algorithm therefore consists of two steps: (i) calculation of the gradient vector $\nabla I(\mathbf{d}^{(s)})$; and (ii) solution of the univariate maximization problem in (E.9) (often referred to as the line search portion of the algorithm) to obtain the proper scalar multiplier $\lambda^{(s)}$.

We remark that it is sometimes difficult — through purely analytical means — to obtain an exact, globally optimal solution during the line search in step (ii). This should not be surprising, since the original $L+1$-dimensional maximization of the same objective function was sufficiently complicated that it required a numerical optimization procedure in the first place. In many such cases, it is desirable to abandon an intensive search for the optimal solution in favor of an alternative method that yields a somewhat coarse, approximate solution but consumes far less computation. In the present case, it is reasonable (from the standpoint of minimizing computational expense) to choose a pseudo-optimal value of $\lambda^{(s)}$ by searching over a small, finite set of candidate values $\{\lambda_1^{(s)}, \lambda_2^{(s)}, \cdots, \lambda_J^{(s)}\}$, which may be allowed to change at each iteration; hence, the modified line search will take the form

$$\lambda_{(s)} = \underset{\lambda \in \{\lambda_1, \lambda_2, \cdots, \lambda_J\}}{\arg\max} \left\{ I\left(\mathbf{d}^{(s)} + \lambda \nabla I(\mathbf{d}^{(s)})\right) \right\}. \tag{E.10}$$

A number of methods are available for determining which candidate values should be in the above search set, as well as for determining the appropriate cardinality of this set (see, for example, [101, 112, 202]).

To complete the description of the steepest-ascent algorithm, we need to include suitable procedures for both initialization and termination of the recursion in (E.7). To initialize the algorithm, we must assign a reasonable value to the vector $\mathbf{d}^{(0)}$. Of course, since the first and last elements of this vector are, as always, constrained by the equations $d_0^{(0)} = -\infty$ and $d_L^{(0)} = +\infty$, we need not be concerned with finding values for these elements. However, for the remaining elements $d_1^{(0)}, d_2^{(0)}, \cdots, d_{L-1}^{(0)}$, we can use the values that result from imposing the simultaneous conditions

$$\int_{-\infty}^{d_1^{(0)}} f_{Y_0}(y)\, dy = \int_{d_1^{(0)}}^{d_2^{(0)}} f_{Y_0}(y)\, dy = \cdots \int_{d_{L-1}^{(0)}}^{\infty} f_{Y_0}(y)\, dy = \frac{1}{L}, \tag{E.11}$$

so that all intervals initially have equal probability. This method of assigning starting breakpoint values tends to work well in practice. To terminate the algorithm, we can simply iterate the recursion in (E.7) until we reach a point at which the magnitude of the applied perturbation $\lambda^{(s)} \nabla I(\mathbf{d}^{(s)})$ is below some prespecified threshold value.

E.3 Derivation of the Mutual Information Gradient

The only ingredient that is missing from the above description of the steepest-ascent algorithm is a formula for the gradient of the mutual information with respect to the breakpoint vector $\mathbf{d}$. Because this quantity is an essential part of the algorithm, and because the mathematical expression for it is rather involved, we devote this subsection to a derivation of

∇I.

We begin by applying basic principles of differential calculus to the expression in (E.4). Specifically, after some calculation we find that the partial derivative of $I(\mathbf{d})$ with respect to the element d_k is given by

$$\begin{aligned}\frac{\partial I(\mathbf{d})}{\partial d_k} = \sum_{i=1}^{L}\sum_{j=1}^{L}\Bigg[&\frac{\partial R(i,j;\mathbf{d})}{\partial d_k}\left(1+\log\frac{R(i,j;\mathbf{d})}{P(i;\mathbf{d})P(j;\mathbf{d})}\right)\\ &-\frac{R(i,j;\mathbf{d})}{P(i;\mathbf{d})P(j;\mathbf{d})}\left(\frac{\partial P(i;\mathbf{d})}{\partial d_k}P(j;\mathbf{d})+P(i;\mathbf{d})\frac{\partial P(j;\mathbf{d})}{\partial d_k}\right)\Bigg].\end{aligned} \tag{E.12}$$

While this expression is also true for the special index values $k = 0$ and $k = L$, in these cases we have that

$$\frac{\partial I(\mathbf{d})}{\partial d_0} = \frac{\partial I(\mathbf{d})}{\partial d_L} = 0, \tag{E.13}$$

owing to the fact d_0 and d_L are constants; thus, in the following derivation we shall focus only on the intermediate index values $k = 1, 2, \cdots, L-1$. We note in addition that the right-hand side of (E.12) involves derivatives of both of the probability mass functions R and P. However, since we have already established the identities

$$\frac{\partial P(i;\mathbf{d})}{\partial d_k} = \sum_{j=1}^{L}\frac{\partial R(i,j;\mathbf{d})}{\partial d_k} = \sum_{j=1}^{L}\frac{\partial R(j,i;\mathbf{d})}{\partial d_k}, \tag{E.14}$$

we know that the derivatives of I actually depend on the derivatives of R alone; therefore, in our remaining calculations, it suffices to focus our attention only on the function R.

From (E.1) we see that the functional dependence of R on the elements of $\mathbf{d}$ is expressed implicitly through the upper and lower limits of a definite double integral. Because of this unusual implicit dependence, the calculation of a partial derivative such as $\partial R(i,j;\mathbf{d})/\partial d_k$ will require some rather careful bookkeeping. We begin by making some simplifying observations. First, note that each double integral in (E.1) is evaluated over a rectangular region of the form $[d_{i-1}, d_i] \times [d_{j-1}, d_j]$ within the coordinate plane, and furthermore that there are a total of L^2 such regions making up the entire plane. When the value of a single breakpoint d_k is modified slightly, only certain of these regions are affected, and therefore only certain partial derivatives with respect to d_k are nonzero. The relevant question is whether the perturbed breakpoint d_k coincides with at least one of the breakpoints $d_{i-1}, d_i, d_{j-1}, d_j$, which define a particular rectangular region.

For any given region, exactly one of the following four cases will be true regarding the relationship between the breakpoint indices i, j, and k:

(a) $k \notin \{i-1, i\}$ and $k \notin \{j-1, j\}$

(b) $k \notin \{i-1, i\}$ and $k \in \{j-1, j\}$

(c) $k \in \{i-1, i\}$ and $k \notin \{j-1, j\}$

(d) $k \in \{i-1, i\}$ and $k \in \{j-1, j\}$

These four cases are depicted in Figures E-1(a) through E-1(d), respectively. The heavy lines shown in the horizontal and vertical dimensions in each figure correspond to the breakpoint d_k that is being perturbed. All of the figures depict the same partitioning of the coordinate plane into L^2 rectangles, but each figure highlights a different subset of these L^2 rectangles through special shading. For example, the shaded rectangles shown in Figures E-1(a) are those whose associated index sets $\{i-1, i\}$ and $\{j-1, j\}$ satisfy condition (a) from the above list. Similarly, the index sets associated with the shaded rectangles in Figures E-1(b) through E-1(d) satisfy conditions (b) through (d), respectively.

Observe that none of the shaded rectangles shown in Figure E-1(a) is affected by a change in d_k; hence, we have that

$$\frac{\partial R(i, j; \mathbf{d})}{\partial d_k} = 0 \tag{E.15}$$

whenever condition (a) is true. In Figure E-1(b), however, each of the shaded rectangles will be affected in some way by a change in d_k. In particular, the rectangles situated below the heavy horizontal line (i.e., those that have d_k as an upper limit), will increase in area if d_k increases; on the other hand, the rectangles situated above heavy line will decrease in area if d_k increases. More specifically, if we change the value of d_k by a very small amount to the new value $d_k + \Delta$, then the value of the double integral over a rectangle just below the line — say, the rectangle associated with the index set $\{i-1, i\}$ on the horizontal axis — will change by approximately the amount $\Delta \int_{d_{i-1}}^{d_i} f_{Y_0,Y_1}(y_0, d_k)\, dy_0$. Moreover, an equal and opposite change will occur in the value of the double integral over the corresponding rectangle just above the line. Thus, in the limit as $\Delta \to 0$ we can write

$$\frac{\partial R(i, j; \mathbf{d})}{\partial d_k} = [\delta_{k,j} - \delta_{k,j-1}] \int_{d_{i-1}}^{d_i} f_{Y_0,Y_1}(y_0, d_k)\, dy_0, \tag{E.16}$$

where $\delta_{k,j}$ is the Kronecker delta function defined by

$$\delta_{k,j} = \begin{cases} 1, & \text{if } k = j, \\ 0, & \text{if } k \neq j. \end{cases} \tag{E.17}$$

An analogous argument holds for condition (c), which is depicted in Figure E-1(c). When this condition is true, we have that

$$\frac{\partial R(i, j; \mathbf{d})}{\partial d_k} = [\delta_{k,i} - \delta_{k,i-1}] \int_{d_{j-1}}^{d_j} f_{Y_0,Y_1}(d_k, y_1)\, dy_1, \tag{E.18}$$

Finally, when condition (d) is true, each of the shaded rectangles shown in Figure E-1(d) undergoes two different kinds of changes when d_k changes, one in the vertical dimension and one in the horizontal dimension. The overall affect on the value of the double integral

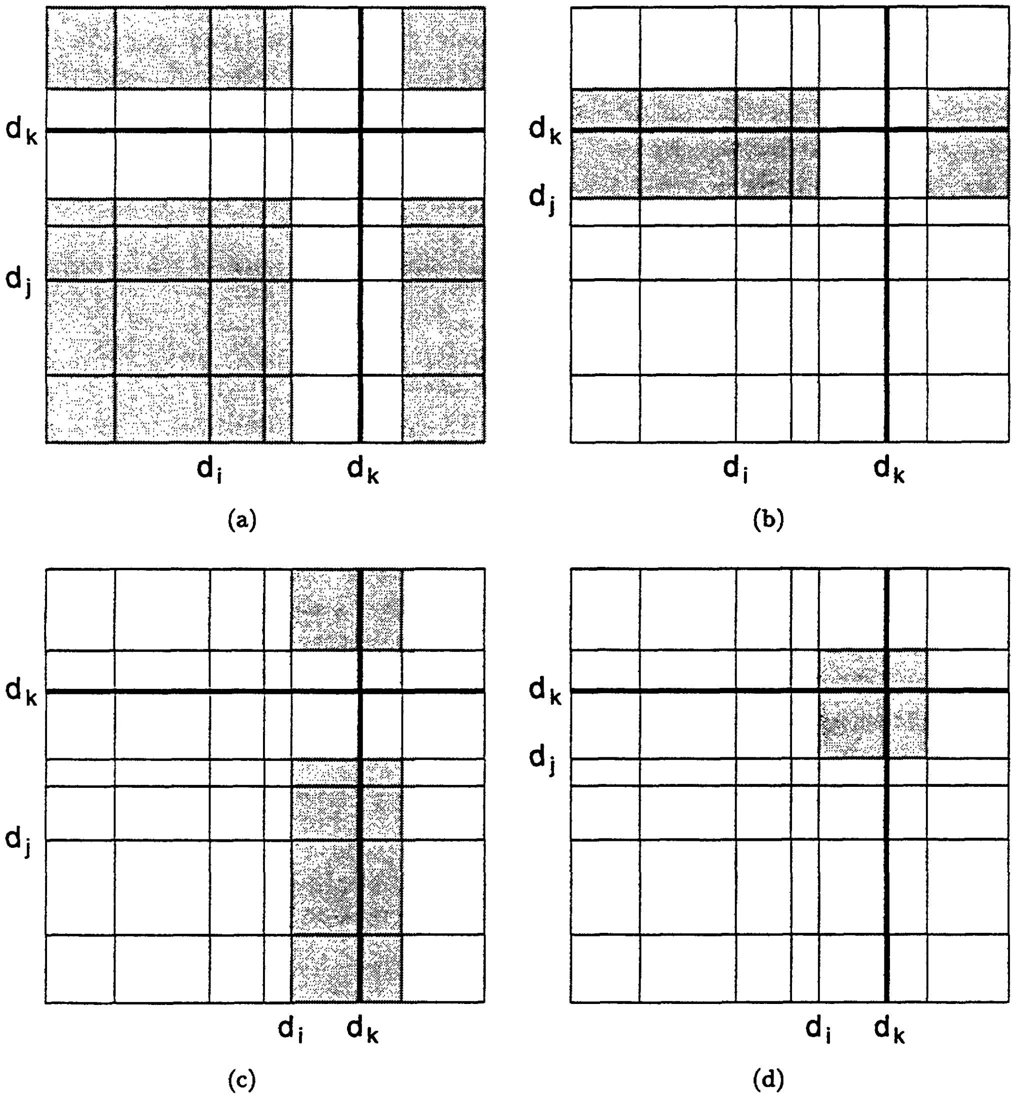


Figure E-1: Depiction of four separate cases encountered in calculation of the gradient vector. Shaded rectangular regions undergo the following types of changes when the value of d_k is perturbed: (a) no change at all; (b) change in the vertical dimension only; (c) change in the horizontal dimension only; (d) change in both the vertical and horizontal dimensions.

over each rectangle is simply the sum of these two individual changes. Thus we can write

$$\begin{aligned}\frac{\partial R(i,j;\mathbf{d})}{\partial d_k} &= [\delta_{k,j} - \delta_{k,j-1}] \int_{d_{i-1}}^{d_i} f_{Y_0,Y_1}(y_0, d_k)\, dy_0 \\ &+ \ [\delta_{k,i} - \delta_{k,i-1}] \int_{d_{j-1}}^{d_j} f_{Y_0,Y_1}(d_k, y_1)\, dy_1.\end{aligned} \tag{E.19}$$

In fact, though it may not be evident at first, this last expression concisely represents each of the conditions (a) through (d). A careful inspection reveals that (E.19) is valid regardless of whether d_k is an upper or lower limit in the inner integral, an upper or lower limit in the outer integral, or not a limit at all. Although applying the formula in (E.19) may require the use of a standard numerical integration procedure, the formula itself now allows us to easily evaluate all other derivatives needed during the operation of our steepest-ascent algorithm, namely those expressed in (E.14) and (E.12).

Appendix F

An Algorithm for Computing Posterior HMM State Probabilities

During our analysis of the signal estimation problem in Chapter 5, we demonstrated that, if the signal and noise are independent, additively combined processes, and if each is the output of a finite-state HMM whose state densities are Gaussian mixtures, then the observation is also the output of an HMM of this same type (albeit one with many more parameters). The special HMM-based structure of the observed signal gives rise to an extremely efficient recursive algorithm for computing the posterior probabilities associated with the states of the underlying Markov chain [19, 47, 78, 154, 155]. In this appendix, we shall construct the overall algorithm in three distinct steps. First, we develop a recursion that runs forward in time, accounting for all past observations at any given time by computing the quantity $f_{\mathbf{Z}_{0:t},\Theta_t}(\mathbf{z}_{0:t}, \Theta_t = i)$. Next, we develop a complementary recursion that runs backward in time, accounting for all future observations at any time by computing the quantity $f_{\mathbf{Z}_{t+1:N-1}|\Theta_t}(\mathbf{z}_{t+1:N-1}|\Theta_t = i)$. Finally, we combine the results of the forward and backward recursions to get the value of the desired posterior state probabilities $\Pr\{\Theta_t = i|\mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\}$. After considering each of these steps in turn, we then describe a special numerical conditioning procedure that must be incorporated into the recursive algorithm when it is implemented on a digital computer.

F.1 Developing the Forward Recursion

We begin by deriving the recursion that runs forward in time. Let us define the forward variable $\alpha_t(i)$ as

$$\alpha_t(i) = f_{\mathbf{Z}_{0:t},\Theta_t}(\mathbf{z}_{0:t}, \Theta_t = i), \tag{F.1}$$

where it is understood that the time index t lies in the set $\{0, 1, \cdots, N-1\}$ and the state index i lies in the set $\{1, 2, \cdots, L\}$. To develop a recursive procedure for computing all of the values $\alpha_t(i)$, we require a method for starting the recursion (the initialization step) and a method for carrying the recursion forward in time (the induction step). The initialization step follows directly from quantities we already know; in particular, we have

$$\alpha_0(i) = f_{Z_0,\Theta_0}(z_0, \Theta_0 = i) = P(i)h_i(z_0), \qquad i = 1, 2, \cdots, L. \tag{F.2}$$

The induction step requires a little more work. Let us assume that the values $\{\alpha_s(j)\}$ of the forward variable have already been computed for the time indices $s = 0, 1, \cdots, t$ and for the state values $j = 1, 2, \cdots, L$. We wish to relate these known values to the new, as yet unknown value $\alpha_{t+1}(i)$. Using the definition of $\alpha_{t+1}(i)$, we may write

$$\begin{aligned}
\alpha_{t+1}(i) &= f_{\mathbf{Z}_{0:t+1},\Theta_{t+1}}(\mathbf{z}_{0:t+1}, \Theta_{t+1} = i) && \text{(F.3)}\\
&= \sum_{j=1}^{L} f_{\mathbf{Z}_{0:t+1},\Theta_t,\Theta_{t+1}}(\mathbf{z}_{0:t+1}, \Theta_t = j, \Theta_{t+1} = i) && \text{(F.4)}\\
&= \sum_{j=1}^{L} f_{\mathbf{Z}_{0:t},Z_{t+1},\Theta_t,\Theta_{t+1}}(\mathbf{z}_{0:t}, z_{t+1}, \Theta_t = j, \Theta_{t+1} = i) && \text{(F.5)}\\
&= \sum_{j=1}^{L} \Pr\{\Theta_{t+1} = i | \Theta_t = j\} \cdot \\
&\qquad f_{Z_{t+1}|\Theta_{t+1}}(z_{t+1} | \Theta_{t+1} = i) f_{\mathbf{Z}_{0:t},\Theta_t}(\mathbf{z}_{0:t}, \Theta_t = j) && \text{(F.6)}\\
&= \left[\sum_{j=1}^{L} \alpha_t(j) Q(j,i)\right] h_i(z_{t+1}). && \text{(F.7)}
\end{aligned}$$

This is the desired recursive relationship. In Figure F-1, we give a graphical depiction of the computations that are required to generate the future forward variable value $\alpha_{t+1}(i)$ from the L currently available values $\{\alpha_t(j)\}_{j=1}^{L}$. The figure depicts a trellis whose dimensions are state and time. A feasible trajectory of the state variable over the entire length of the observation can be envisioned on this trellis as a polygonal path that intersects exactly one state node at each time index between 0 and $N - 1$.

The figure shows the path segments that could lead to state i at time $t + 1$ from the L possible states at the immediately preceding time t. Recall that the quantity $\alpha_t(j)$ is the joint probability that (i) the vector $\mathbf{z}_{0:t}$ was observed, and (ii) the state at time t was j. This means we can interpret the product $\alpha_t(j)Q(j,i)$ as the joint probability that (i) $\mathbf{z}_{0:t}$ was observed, and (ii) state i was reached at time $t + 1$ by way of state j. If we then add together all of the products of this form (i.e., the products for all possible values of j, holding fixed the time value t and state value i) we obtain the joint probability that (i) the vector $\mathbf{z}_{0:t}$ was observed, and (ii) the state at time $t + 1$ was i. Once this probability has been computed, we see that the new quantity $\alpha_{t+1}(i)$ can be evaluated by multiplying the summed quantity by the output pdf value $h_i(z_{t+1})$; this accounts for the fact that z_{t+1} was observed while in state i. An analogous computation is carried out for each possible state value i at time $t + 1$. To keep the recursion moving forward, we then repeat this same overall sequence of computations at the subsequent time index.

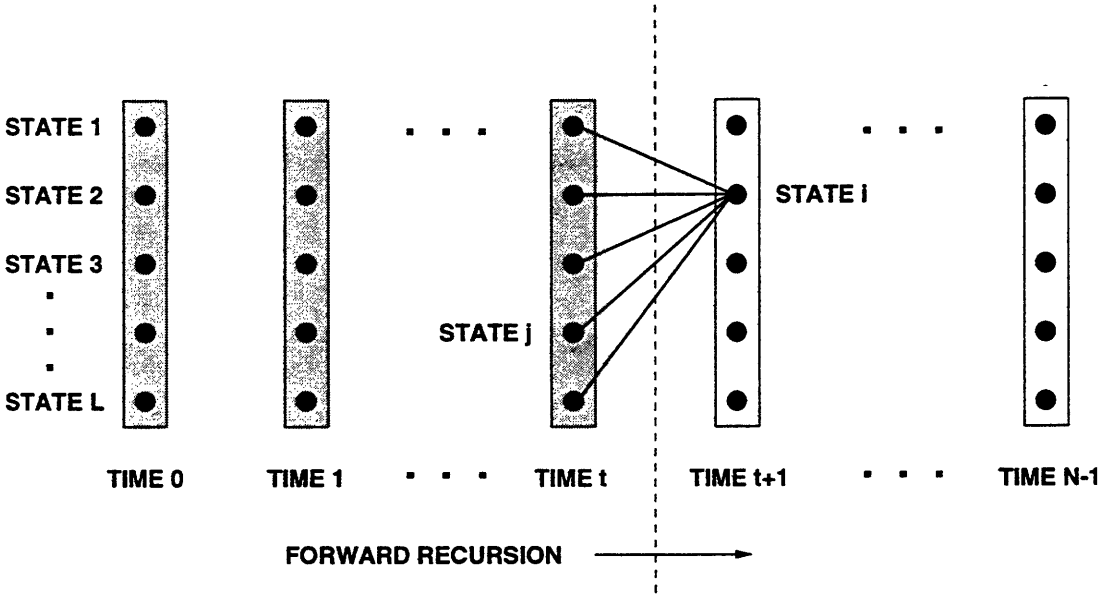


Figure F-1: Illustration of the forward recursive procedure used to compute the new quantity $\alpha_{t+1}(i)$ from the L values $\{\alpha_t(j)\}_{j=1}^{L}$ just computed. Recursion is depicted on an $L \times N$ trellis whose nodes represent possible states of the underlying Markov chain at each time index. Shaded sets of nodes are those for which forward variable has already been evaluated. Vertical dashed line indicates current stage of computation.

We summarize the forward recursion with the following two formulas:

INITIALIZATION

$$\alpha_0(i) = P(i)h_i(z_0), \qquad i = 1, 2, \cdots, L \tag{F.8}$$

INDUCTION

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{L} \alpha_t(j)Q(j,i)\right] h_i(z_{t+1}), \qquad \begin{array}{l} i = 1, 2, \cdots, L; \\ t = 0, 1, \cdots, N-2. \end{array} \tag{F.9}$$

F.2 Developing the Backward Recursion

We now derive the recursion that runs backward in time. For this procedure, we define the backward variable $\beta_t(i)$ as

$$\beta_t(i) = f_{\mathbf{z}_{t+1:N-1}|\Theta_t}(\mathbf{z}_{t+1:N-1}|\Theta_t = i), \tag{F.10}$$

which holds when the state index i is contained in $\{1, 2, \cdots, L\}$ and the time index t is contained in $\{0, 1, \cdots, N-2\}$. For the final time index $t = N-1$, we no longer use the definition in (F.10), but instead arbitrarily specify that

$$\beta_{N-1}(i) = 1, \qquad i = 1, 2, \cdots, L. \tag{F.11}$$

Once this initialization is done, we can develop the recursion for $\beta_t(i)$ just as we did for $\alpha_t(i)$, now progressing backward in time rather than forward. Specifically, we can write

$$\begin{aligned}
\beta_t(i) &= f_{\mathbf{Z}_{t+1:N-1}|\Theta_t}(\mathbf{z}_{t+1:N-1}|\Theta_t = i) && \text{(F.12)}\\
&= \sum_{j=1}^{L} f_{\mathbf{Z}_{t+1:N-1},\Theta_{t+1}|\Theta_t}(\mathbf{z}_{t+1:N-1}, \Theta_{t+1} = j|\Theta_t = i) && \text{(F.13)}\\
&= \sum_{j=1}^{L} f_{Z_{t+1},\mathbf{Z}_{t+2:N-1},\Theta_{t+1}|\Theta_t}(z_{t+1}, \mathbf{z}_{t+2:N-1}, \Theta_{t+1} = j|\Theta_t = i) && \text{(F.14)}\\
&= \sum_{j=1}^{L} \Pr\{\Theta_{t+1} = j|\Theta_t = i\} f_{Z_{t+1}|\Theta_{t+1}}(z_{t+1}|\Theta_{t+1} = j) \cdot \\
&\qquad f_{\mathbf{Z}_{t+2:N-1}|\Theta_{t+1}}(\mathbf{z}_{t+2:N-1}|\Theta_{t+1} = j) && \text{(F.15)}\\
&= \sum_{j=1}^{L} Q(i,j) h_j(z_{t+1}) \beta_{t+1}(j). && \text{(F.16)}
\end{aligned}$$

This last formula is the desired recursive relationship. In Figure F-2, we give a graphical depiction (analogous to the one given in Figure F-1) of the computations that are required to generate the backward variable value $\beta_t(i)$ from the L values $\{\beta_{t+1}(j)\}_{j=1}^{L}$ just computed. The figure shows the path segments leading from state i at time t to the L possible states that could be reached at time $t+1$. Recall that the quantity $\beta_{t+1}(j)$ is the probability that the vector $\mathbf{z}_{t+2:N-1}$ was observed given that the state at time $t+1$ was j. This means that we can interpret the product $h_j(z_{t+1})\beta_{t+1}(j)$ as the probability that $\mathbf{z}_{t+1:N-1}$ was observed given that the state at time $t+1$ was j. Furthermore, we can interpret the product $Q(i,j)h_j(z_{t+1})\beta_{t+1}(j)$ as the joint probability, conditioned on the event that the state at time t was i, that (i) $\mathbf{z}_{t+1:N-1}$ was observed; and (ii) j was reached at time $t+1$ by way of state i. If we then add together the products of this form for all j, we obtain probability that the vector $\mathbf{z}_{t+1:N-1}$ was observed given that the state at time t was i, which is simply $\beta_t(i)$. An analogous computation is carried out for each possible state value i at time t. To keep the recursion moving backward, we then repeat this same overall sequence of computations at the immediately preceding time index.

We summarize the backward recursion with the following two formulas:

INITIALIZATION

$$\beta_{N-1}(i) = 1, \qquad i = 1, 2, \cdots, L \tag{F.17}$$

INDUCTION

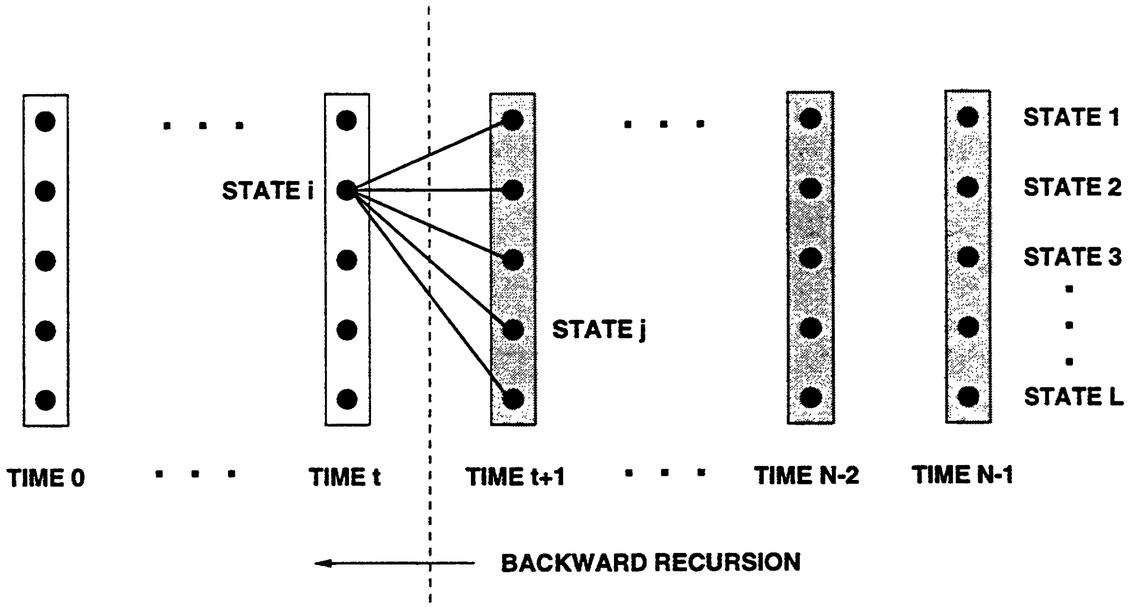


Figure F-2: Illustration of the backward recursive procedure used to compute the new quantity $\beta_t(i)$ from the L values $\{\beta_{t+1}(j)\}_{j=1}^{L}$ just computed. Recursion is depicted on an $L \times N$ trellis whose nodes represent possible states of the underlying Markov chain at each time index. Shaded sets of nodes are those for which backward variable has already been evaluated. Vertical dashed line indicates current stage of computation.

$$\beta_t(i) = \sum_{j=1}^{L} \beta_{t+1}(j) Q(i,j) h_j(z_{t+1}), \qquad \begin{array}{l} i = 1, 2, \cdots, L; \\ t = N-2, N-3 \cdots, 0. \end{array} \tag{F.18}$$

F.3 Combining the Forward and Backward Results

Once the forward and backward recursions have been applied to the observed sequence, we can easily combine the results generated by these recursions to obtain the desired HMM state probabilities. Let us introduce the more concise notation $\gamma_t(i)$ to represent the probability that the Markov chain is in state i at time t based on the value of the observation $\mathbf{z}_{0:N-1}$. Observe that, from the definition of conditional probability, we can immediately write

$$\begin{aligned} \gamma_t(i) &= \Pr\{\Theta_t = i | \mathbf{Z}_{0:N-1} = \mathbf{z}_{0:N-1}\} && \text{(F.19)} \\ &= \frac{f_{\mathbf{Z}_{0:N-1},\Theta_t}(\mathbf{z}_{0:N-1}, \Theta_t = i)}{f_{\mathbf{Z}_{0:N-1}}(\mathbf{z}_{0:N-1})} && \text{(F.20)} \\ &= \frac{f_{\mathbf{Z}_{0:N-1},\Theta_t}(\mathbf{z}_{0:N-1}, \Theta_t = i)}{\sum_{j=1}^{L} f_{\mathbf{Z}_{0:N-1},\Theta_t}(\mathbf{z}_{0:N-1}, \Theta_t = j)}. && \text{(F.21)} \end{aligned}$$

Furthermore, since we have the natural decomposition

$$\begin{aligned} &f_{\mathbf{Z}_{0:N-1},\Theta_t}(\mathbf{z}_{0:N-1},\Theta_t = i) \\ &\qquad = f_{\mathbf{Z}_{0:t},\Theta_t}(\mathbf{z}_{0:t},\Theta_t = i) \cdot f_{\mathbf{Z}_{t+1:N-1}|\Theta_t}(\mathbf{z}_{t+1:N-1}|\Theta_t = i) && \text{(F.22)} \\ &\qquad = \alpha_t(i)\beta_t(i), && \text{(F.23)} \end{aligned}$$

we obtain the simplified expression

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{L} \alpha_t(j)\beta_t(j)}, \tag{F.24}$$

which reveals explicitly the functional dependence of the posterior state probabilities on the output from the recursive procedures derived earlier.

F.4 Conditioning the HMM-Based Recursions

Unfortunately, the recursive formulas for the forward and backward variables that we derived in Sections F.1 and F.2 will produce distorted results when implemented directly on a digital computer, even for moderate values of the observation length N (e.g., values of approximately 100 or more). To understand why this is true, let us first re-examine the mathematical structure of the forward variable $\alpha_t(\cdot)$. By unraveling the recursive definition of this variable at each successive time index, starting at time 0, we can determine its composition explicitly at a general time t. Assuming that our HMM consists of L states, and that a particular realization of the HMM output is represented by $\mathbf{z}_{0:N-1}$, we can write the following non-recursive formulas for the forward variable:

$$\begin{aligned} \alpha_0(i_0) &= P(i_0)h_{i_0}(z_0) && \text{(F.25)} \\ \alpha_1(i_1) &= \sum_{i_0=1}^{L} \alpha_0(i_0)Q(i_0,i_1)h_{i_1}(z_1) \\ &= \sum_{i_0=1}^{L} P(i_0)Q(i_0,i_1)h_{i_0}(z_0)h_{i_1}(z_1) && \text{(F.26)} \\ \alpha_2(i_2) &= \sum_{i_1=1}^{L} \alpha_1(i_1)Q(i_1,i_2)h_{i_2}(z_2) \\ &= \sum_{i_1=1}^{L}\sum_{i_0=1}^{L} P(i_0)Q(i_0,i_1)Q(i_1,i_2)h_{i_0}(z_0)h_{i_1}(z_1)h_{i_2}(z_2) && \text{(F.27)} \\ &\vdots \end{aligned}$$

$$\begin{aligned}\alpha_{t+1}(i_{t+1}) &= \sum_{i_{t+1}=1}^{L} \alpha_t(i_t)Q(i_t, i_{t+1})h_{i_{t+1}}(z_{t+1}) \\ &= \sum_{i_t=1}^{L} \sum_{i_{t-1}=1}^{L} \cdots \sum_{i_0=1}^{L} P(i_0) \prod_{k=1}^{t} Q(i_k, i_{k+1}) \prod_{k=0}^{t+1} h_{i_k}(z_k) \qquad \text{(F.28)}\end{aligned}$$

From these expressions, we can see that the forward variable is made up of products of probabilities (which are, by definition, confined to the interval $[0, 1]$) as well as products of pdf values (which may exceed unity, but are usually much closer to zero). Because these terms are typically quite small, and because the number of terms in each product grows linearly with t, the products themselves eventually tend toward zero at an exponential rate. Furthermore, the summing of many such vanishing products (which is done to produce the final value of the forward variable) has virtually no mitigating effect on this exponential decay in magnitude. Thus, when the recursive algorithm from Section F.1 is implemented in its original form, and t is allowed to become sufficiently large, the dynamic range required in the computation of $\alpha_t(\cdot)$ cannot be accommodated by the finite-length registers in a typical computer, even if the computation is performed using double-precision arithmetic.

To avoid the problem of eventual register underflow, we must somehow numerically condition the forward variable so that all computations performed for $t = 0, 1, \cdots, N-1$ occur well within the dynamic range of the computer. We can achieve the desired conditioning by multiplying the forward variable at each time index by an appropriately chosen time-dependent scale factor. As we will see, such a scaling procedure essentially creates a normalized version of the original set of forward variables at each time index, but nonetheless allows us to compute the new set of scaled forward variables recursively, as we did originally.

To describe the scaling procedure in detail, we first introduce the notation $\alpha'_t(\cdot)$ to represent the scaled forward variable at time t. The values assumed by this variable are propagated forward in time using a method similar to that used for the original variable $\alpha_t(\cdot)$, except that now the recursive step consists of two distinct parts. In the first part, the scaled variable from the previous time, $\alpha'_{t-1}(\cdot)$, is projected ahead to time t through the usual inductive formula

$$\alpha''_t(j) = \sum_{i=1}^{L} \alpha'_{t-1}(i)Q(i,j)h_j(z_t), \qquad j = 1, 2, \cdots, L, \qquad \text{(F.29)}$$

where the new variable $\alpha''_t(\cdot)$ has been introduced to represent the preliminary, unscaled result of this transformation. In the second part of the recursive step, the unscaled variable $\alpha''_t(\cdot)$ is transformed back into its properly scaled counterpart according to the formula

$$\alpha'_t(i) = c_t\alpha''_t(i), \qquad i = 1, 2, \cdots, L, \qquad \text{(F.30)}$$

where c_t is the normalization factor given by

$$c_t = \frac{1}{\sum_{j=1}^{L} \alpha_t''(j)}. \tag{F.31}$$

Note from (F.30) that at time t, the same scale factor c_t is applied to each of the L terms $\{\alpha_t''(i)\}_{i=1}^{L}$; thus, although the scale factor does indeed vary with the time index t, it is completely independent of the state index i. The recursive procedure for the scaled forward variable is initialized by the equations

$$\alpha_0''(i) = \alpha_0(i) \qquad i = 1, 2, \cdots, L \tag{F.32}$$

$$c_0 = \frac{1}{\sum_{j=1}^{L} \alpha_0''(j)} \tag{F.33}$$

$$\alpha_0'(i) = c_0 \alpha_0''(i) \qquad i = 1, 2, \cdots, L. \tag{F.34}$$

Once this initialization is performed at $t = 0$, the recursion then consists of a repeated application of the pair of operations given in (F.29) and (F.30) for $t = 1, 2, \cdots, N-1$.

With the new recursion now fully specified, let us attempt to express our scaled forward variable $\alpha_t'(\cdot)$ in terms of the original forward variable $\alpha_t(\cdot)$ at an arbitrary time t. We claim that the relationship between these two variables is given by

$$\alpha_t'(i) = \left[\prod_{s=0}^{t} c_s\right] \alpha_t(i). \tag{F.35}$$

From the initialization formulas in (F.34), it is easy to verify that this relationship holds at time 0. Proceeding inductively, then, let us assume that it also holds for each time up to and including time $t-1$. Then using (F.29) and (F.30), in addition to the original recursive formula for $\alpha_t(\cdot)$, at time t we may write

$$\begin{aligned} \alpha_t'(j) &= c_t \alpha_t''(j) && \text{(F.36)} \\ &= c_t \sum_{i=1}^{L} \alpha_{t-1}'(i) Q(i,j) h_j(z_t) && \text{(F.37)} \\ &= c_t \sum_{i=1}^{L} \left[\prod_{s=0}^{t-1} c_s\right] \alpha_{t-1}(i) Q(i,j) h_j(z_t) && \text{(F.38)} \\ &= \left[\prod_{s=0}^{t} c_s\right] \sum_{i=1}^{L} \alpha_{t-1}(i) Q(i,j) h_j(z_t) && \text{(F.39)} \\ &= \left[\prod_{s=0}^{t} c_s\right] \alpha_t(i), && \text{(F.40)} \end{aligned}$$

which proves the claim. Now, by substituting (F.35) into the recursive formula for $\alpha_t'(\cdot)$, we can derive a more direct expression of the relationship between the scaled and original forward variables that does not involve the scale factors $\{c_s\}$. In particular, we have

$$\alpha'_t(j) = c_t \sum_{i=1}^{L} \alpha'_{t-1}(i) Q(i,j) h_j(z_t) \tag{F.41}$$

$$= \frac{\sum_{i=1}^{L} \alpha'_{t-1}(i) Q(i,j) h_j(z_t)}{\sum_{k=1}^{L} \sum_{i=1}^{L} \alpha'_{t-1}(i) Q(i,k) h_k(z_t)} \tag{F.42}$$

$$= \frac{\sum_{i=1}^{L} \left[\prod_{s=0}^{t-1} c_s\right] \alpha_{t-1}(i) Q(i,j) h_j(z_t)}{\sum_{k=1}^{L} \sum_{i=1}^{L} \left[\prod_{s=0}^{t-1} c_s\right] \alpha_{t-1}(i) Q(i,k) h_k(z_t)} \tag{F.43}$$

$$= \frac{\alpha_t(j)}{\sum_{k=1}^{L} \alpha_t(k)}. \tag{F.44}$$

From this last equation, we see that the new scaled forward variable $\alpha'_t(j)$ is truly just a normalized version of the original forward variable $\alpha_t(j)$ at each time t, even though this may not be immediately evident from the definition of the new variable.

Thus far, we have discussed only the scaling procedure that applies to the forward recursion. A similar scaling procedure must also be employed during the computation of the backward variable $\beta_t(\cdot)$, since this variable also exhibits an exponential decay in magnitude when the original recursive formula is used. However, we need not calculate a new set of scale factors for the new backward variable; instead, we can simply re-use the ones that were generated at the corresponding times during the forward recursion. Although this method does not guarantee that the scale factor applied at time t will restore the sum of the backward variables to unity at time t, it nonetheless yields accurate final results, since the magnitudes of the forward and backward variables are comparable. Moreover, this strategy of using common scale factors for the two sets of variables has the obvious advantage of reducing the overall computational expense associated with the HMM-based estimation algorithm.

The new backward recursion is defined in a similar manner to the forward recursion described above. In particular, once again we use the two basic formulas

$$\beta''_t(j) = \sum_{i=1}^{L} \beta'_{t+1}(i) Q(j,i) h_i(z_{t+1}), \qquad j = 1, 2, \cdots, L \tag{F.45}$$

and

$$\beta'_t(i) = c_t \beta''_t(i), \qquad i = 1, 2, \cdots, L, \tag{F.46}$$

where the first formula is used to project the scaled variable values back from time $t+1$ to time t, and the second is used to adjust the magnitude of these projected values. This new recursion is initialized by the equation

$$\beta'_{N-1}(i) = c_{N-1} \beta_{N-1}(i) \qquad i = 1, 2, \cdots, L. \tag{F.47}$$

After this initialization is performed at $t = N-1$, the recursion then consists of a repeated application of the pair of operations given in (F.45) and (F.46) for $t = N-2, N-3, \cdots, 0$.

By using an inductive argument analogous to that used for the forward variable, we can show (although the proof will be omitted) that the scaled backward variable can be expressed in terms of the original backward variable at time t as

$$\beta_t'(i) = \left[\prod_{s=t+1}^{N-1} c_s\right] \beta_t(i). \tag{F.48}$$

Furthermore, by combining this identity with its counterpart from the forward recursion (given in (F.35)), we find that the product of a scaled forward variable and a scaled backward variable, taken at time t and for state i, is given by

$$\alpha_t'(i)\beta_t'(i) = \left[\prod_{s=0}^{t} c_s\right] \alpha_t(i) \left[\prod_{s=t+1}^{N-1} c_s\right] \beta_t(i) \tag{F.49}$$

$$= \left[\prod_{s=0}^{N-1} c_s\right] \alpha_t(i)\beta_t(i). \tag{F.50}$$

This is a significant equation from the standpoint of calculating the posterior state probability $\gamma_t(i)$, for we can immediately use it to write

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{L} \alpha_t(j)\beta_t(j)} \tag{F.51}$$

$$= \frac{\left[\prod_{s=0}^{N-1} c_s\right] \alpha_t(i)\beta_t(i)}{\sum_{j=1}^{L} \left[\prod_{s=0}^{N-1} c_s\right] \alpha_t(j)\beta_t(j)} \tag{F.52}$$

$$= \frac{\alpha_t'(i)\beta_t'(i)}{\sum_{j=1}^{L} \alpha_t'(j)\beta_t'(j)}. \tag{F.53}$$

This last equation implies that we can operate on the new scaled forward and backward variables in exactly the same way that we operated on the original variables when computing the value of $\gamma_t(i)$, a critical quantity in the HMM-based estimation algorithm. This is quite convenient, for it means that we will not incur a premium in computational cost to recover certain key quantities needed for signal estimation, despite the fact that the underlying recursive procedure has been modified significantly by the numerical conditioning strategy.

Another desirable property of our new scaling procedure is that the coefficients $\{c_t\}$ generated during the forward recursion can be used to compute the log-likelihood value $\log f_{\mathbf{z}_{0:N-1}}(\mathbf{z}_{0:N-1})$, which is very useful in problems such as signal detection and signal classification (see, for example, Section 5.6.1.2). If the scaling procedure were not required at all (i.e., if all computations from the original recursions could somehow be performed with infinite precision arithmetic), then this log-likelihood value could be computed as

$$\log f_{\mathbf{Z}_{0:N-1}}(\mathbf{z}_{0:N-1}) = \log \sum_{i=1}^{L} f_{\mathbf{Z}_{0:N-1},\Theta_{N-1}}(\mathbf{z}_{0:N-1}, \Theta_{N-1} = i) \tag{F.54}$$

$$= \log \sum_{i=1}^{L} \alpha_{N-1}(i), \tag{F.55}$$

where in the latter equation we have merely exploited the definition of the original forward variable at time $N-1$. As we know from our earlier discussion, the summation appearing in this expression cannot be evaluated directly. However, we can still obtain the desired log-likelihood value by using the fact that

$$\sum_{i=1}^{L} \alpha_{N-1}(i) = \frac{1}{\prod_{t=0}^{N-1} c_t}, \tag{F.56}$$

which follows easily from the previously derived identities (F.35) and (F.44). Of course, this last equation can be rewritten as

$$f_{\mathbf{Z}_{0:N-1}}(\mathbf{z}_{0:N-1}) = \frac{1}{\prod_{t=0}^{N-1} c_t}, \tag{F.57}$$

or equivalently, after taking logarithms of both sides, as

$$\log f_{\mathbf{Z}_{0:N-1}}(\mathbf{z}_{0:N-1}) = -\sum_{t=0}^{N-1} \log c_t. \tag{F.58}$$

This demonstrates that the scale factors by themselves constitute an extremely valuable by-product of the new forward recursion, since they can be used to compute the log-likelihood value of the observation.

Appendix G

Using a Gaussian Mixture to Approximate a Nonlinear Estimator

In this appendix, we demonstrate that using a low-order Gaussian-mixture approximation in place of the true signal pdf can lead to the development of a nearly optimal, computationally efficient estimation scheme. We examine a simple but illustrative non-Gaussian signal estimation problem. In particular, we consider a problem in which we are allowed to observe only the sum of two statistically independent, scalar-valued random variables, each having a pdf that is precisely known. One of these random variables is assumed to have a non-Gaussian pdf and is designated as a signal variable; the other is assumed to have a Gaussian pdf and is designated as a noise variable. On the basis of our single observation, we wish to generate an MMSE estimate of the value assumed by the signal variable.

We begin our discussion with mathematical description of the elements of the estimation problem, and we then immediately introduce a Gaussian-mixture approximation for the non-Gaussian signal pdf. Using this pdf approximation, we in turn develop an approximation for the true globally optimal processor that should be applied to the observation. We are able to decompose the approximate processor into a collection of linear terms, and we can therefore easily make predictions concerning how the optimal processor behaves when the observation lies in various ranges. We show that using such an approximation can provide a deeper, more intuitive understanding of the structure of the optimal estimator. Finally, we derive the exact mathematical form for the optimal processor and show that this estimator does indeed behave as we had predicted.

G.1 Observation Model and Problem Statement

Let Y and V be statistically independent random variables representing, respectively, the signal and noise components of our scalar-valued observation Z, which is defined in the usual way as

$$Z = Y + V. \tag{G.1}$$

Suppose further that the signal variable Y is characterized by a zero-mean Laplacian pdf, defined by

$$f_Y(y) = \frac{1}{2\beta} \exp\left\{-\frac{|y|}{\beta}\right\}, \qquad -\infty < y < \infty, \tag{G.2}$$

and that the noise variable V has a zero-mean Gaussian pdf, which is given by

$$f_V(v) = \frac{1}{\sqrt{2\pi}\sigma_V} \exp\left\{-\frac{v^2}{2\sigma_V^2}\right\}, \qquad -\infty < v < \infty. \tag{G.3}$$

For the sake of concreteness, we assume throughout the following discussion that the pdf scale parameters associated with the signal and noise variables (i.e., β and σ_V, respectively) take the values $\beta = 2$ and $\sigma_V = 3$. (Our choice for the value of β implies that the standard deviation for the signal is given by $\sigma_Y = \sqrt{2}\beta \approx 2.83$; hence, the signal-to-noise ratio for this problem is approximately 0 dB.)

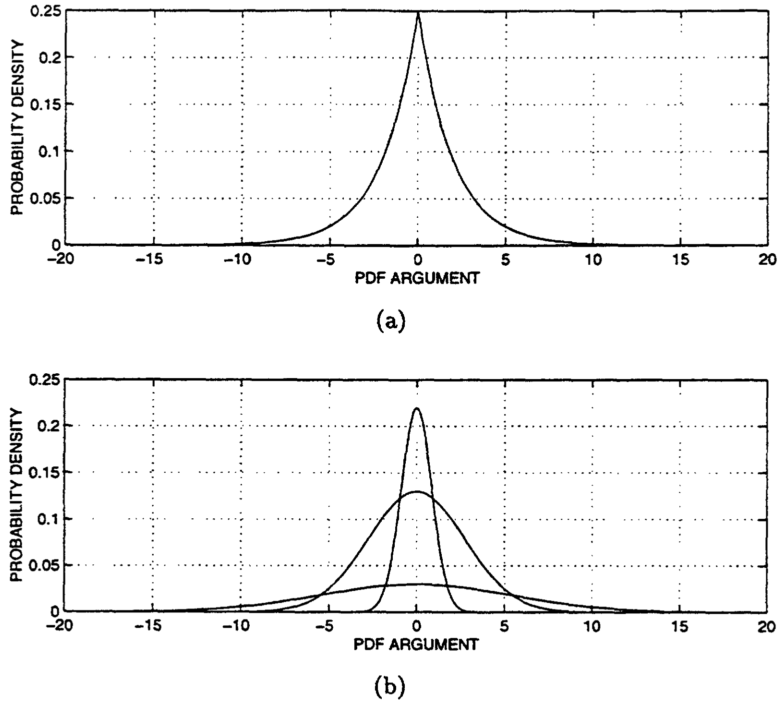
Ultimately, we would like to determine the explicit form for the function $\hat{y}(z) = E\{Y|Z = z\}$, which is known to yield the globally optimal estimate (in the MMSE sense) of the signal value y based on our observation of the event $Z = z$. However, the fact that the signal variable Y is Laplacian makes the estimation problem considerably more complicated than it would be if Y were Gaussian. As we will soon discover (following a rather involved derivation), the optimal data processor for this problem is indeed nonlinear and has a complex mathematical structure. Before delving into the details of this derivation, however, let us first try to predict the basic form for this optimal processor by reasoning — in an approximate sense — about what action this processor should perform over various ranges of the input value z.

G.2 Using the Gaussian-Mixture Approximation

We can develop our intuitive understanding of the optimal data processor by introducing an approximation for the Laplacian signal pdf itself. Recall from the example we presented in Chapter 2 — specifically, the example from our discussion on the source identification problem involving Laplacian-distributed driving noise for an AR process — that we have already obtained such an approximation in the form of a three-component Gaussian-mixture pdf. In fact, numerous approximations of this kind were generated in that example, one for each of the experimental trials that was performed. For the purposes of the present example, we have arbitrarily selected one of these approximations, specified by the collection of parameter values

$$(\mu_1, \sigma_1, \rho_1) = (0.0, 0.87, 0.28) \tag{G.4}$$

$$(\mu_2, \sigma_2, \rho_2) = (0.0, 2.76, 0.61) \tag{G.5}$$



(a)

(b)

Figure G-1: (a) True Laplacian density with scale parameter $\beta = 2$; (b) Individual Gaussian densities making up approximating Gaussian mixture.

$$(\mu_3, \sigma_3, \rho_3) = (0.0, 5.47, 0.11). \tag{G.6}$$

In Figures G-1(a) and G-1(b), we show, respectively, a plot of the original Laplacian density and the three individual Gaussian densities that make up the Gaussian-mixture approximation. (The densities shown in Figure G-1(b) were scaled in such a way that they could be conveniently superimposed on a single plot. As depicted in this figure, the scaled densities appear in approximately the same relative proportions as they would if each original pdf had been multiplied by its associated weighting coefficient ρ_i.)

Now that we have a simple and reasonably accurate Gaussian-mixture approximation for the pdf of the signal variable, we would like to develop an optimal estimator for the signal value based on the assumption that our approximation is exact. Recall that we have already carried out such a development in Chapter 5. The key technique used in that development was to condition the problem on each of the three possible choices for the purely Gaussian pdf that could give rise to the signal value. Under each condition, the optimal processor is a Wiener smoother, since the noise is also Gaussian. For this scalar case, implementing the Wiener smoother for each condition is carried out by multiplying the measurement by an appropriately chosen positive coefficient. The overall approximate processor is therefore given by

$$\hat{y}(z) = \sum_{i=1}^{3} \Pr\{\Phi = i|Z = z\} \left(\frac{\sigma_i^2}{\sigma_i^2 + \sigma_V^2} \right) z. \tag{G.7}$$

Although the basic components of this processor are linear terms, the overall processor itself is clearly nonlinear, since the posterior probability $\Pr\{\Phi = i|Z = z\}$ multiplying the ith linear processor is actually a nonlinear function of the observed data z.

All three linear processors included in (G.7) are plotted in Figure G-2. Based on the form of (G.7), we see that the slope of each line is determined by the relationship between the standard deviation of its associated signal component, σ_i, and the standard deviation of the noise, σ_V.[1] Note that each line shown in the figure consists of a dotted portion and a solid portion. Each line appears solid throughout the region where the corresponding Gaussian component of the signal pdf has non-negligible posterior probability; hence, in regions where the line is dotted, the effect of the linear processor can essentially be ignored. This suggests that the overall data processor operates in three separate regimes, depending on whether the magnitude of the observation z is small, intermediate, or large. These regimes are labeled in Figure G-2 as Regimes 1, 2, and 3, respectively.

Let us now try to predict what the processor will do in each of its three regimes. In Regime 1, while it is true that any of the three Gaussian components of the signal pdf could have produced the signal value, clearly the first two components (i.e., those with the smallest standard deviations) account for most of the probability. We expect that the processor will be approximately linear throughout this regime and will be situated somewhere between the two lines of smallest slope depicted in the figure. On the other hand, in Regime 3, it is very unlikely that any but the third Gaussian component of the signal pdf (i.e., that with the largest standard deviation) could have produced the signal value. In this case the processor will also be approximately linear, but it will now virtually coincide with the line of largest slope. Finally, in Regime 2, the processor will once again be approximately linear, but it will be structured in such a way that it connects the other linear functions from Regimes 1 and 3. In fact, as we can see from Figure G-3, this is precisely how the nonlinear processor given in (G.7) appears when plotted.

G.3 Derivation of Globally Optimal Processor

To assess the accuracy of our prediction, let us now derive the functional form of the actual optimal processor $E\{Y|Z = z\}$ for this non-Gaussian problem. We begin by going back to

[1] In general, the slope of each line can range from zero to unity. If σ_i is extremely small relative to σ_V, then the slope of the corresponding line will be near its minimum value of zero; this represents the case in which there is almost no information to be gained from a single observation of the corrupted signal value. As the value of σ_i increases relative to σ_V, the corresponding line becomes steeper, and hence more weight is given to the observation. Finally, if σ_i is extremely large relative to σ_V, the slope of the corresponding line approaches its maximum value of unity; in this case, the observation is rich in information and, accordingly, the optimal data processor is approximately the identity function.

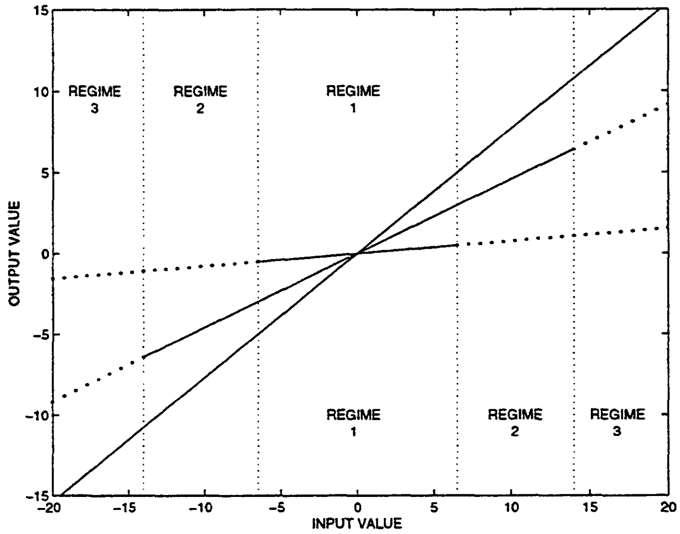


Figure G-2: Optimal linear processors associated with the three Gaussian densities in approximating mixture. Solid portion of each line indicates region where the corresponding density has non-negligible probability.

the definition of conditional mean, from which we can write

$$E\{Y|Z=z\} = \int_{-\infty}^{\infty} y f_{Y|Z}(y|Z=z)dy \tag{G.8}$$

$$= \frac{\int_{-\infty}^{\infty} y f_{Y,Z}(y,z)dy}{\int_{-\infty}^{\infty} f_{Y,Z}(y,z)dy}, \tag{G.9}$$

where in the latter step we have used the fact that the conditional pdf of Y, given that $Z=z$, is simply a normalized version of the joint pdf of Y and Z when this joint pdf is viewed as a function of y alone (i.e., when z is assumed fixed). From the definitions given in (G.2) and (G.3), we have that

$$f_{Y,Z}(y,z) = f_Y(y) f_{Z|Y}(z|Y=y) \tag{G.10}$$

$$= f_Y(y) f_V(z-y) \tag{G.11}$$

$$= \frac{1}{2\beta\sqrt{2\pi}\sigma_V} \exp\left\{-\frac{|y|}{\beta}\right\} \exp\left\{-\frac{(z-y)^2}{2\sigma_V^2}\right\} \tag{G.12}$$

$$= \frac{e^{-z^2/2\sigma_V^2}}{2\beta\sqrt{2\pi}\sigma_V} \exp\left\{-\frac{y^2}{2\sigma_V^2} + \frac{zy}{\sigma_V^2} - \frac{|y|}{\beta}\right\}. \tag{G.13}$$

If we now substitute this expression into (G.9) and then cancel like terms from the numerator and denominator, we obtain the equivalent estimation formula

$$E\{Y|Z=z\} = \frac{\int_{-\infty}^{\infty} y \exp\left(-\frac{y^2}{2\sigma_V^2} + \frac{zy}{\sigma_V^2} - \frac{|y|}{\beta}\right) dy}{\int_{-\infty}^{\infty} \exp\left(-\frac{y^2}{2\sigma_V^2} + \frac{zy}{\sigma_V^2} - \frac{|y|}{\beta}\right) dy}. \tag{G.14}$$

Now, by selecting appropriate limits of integration and introducing a simple change of variables, we can rewrite the above expression so that terms of the form $|y|$ never appear. The new formula is given by

$$E\{Y|Z=z\}$$
$$= \frac{\int_0^{\infty} y \exp\left\{-\frac{y^2}{2\sigma_V^2} - \left(\frac{\sigma_V^2-\beta z}{\beta\sigma_V^2}\right) y\right\} dy - \int_0^{\infty} y \exp\left\{-\frac{y^2}{2\sigma_V^2} - \left(\frac{\sigma_V^2+\beta z}{\beta\sigma_V^2}\right) y\right\} dy}{\int_0^{\infty} \exp\left\{-\frac{y^2}{2\sigma_V^2} - \left(\frac{\sigma_V^2-\beta z}{\beta\sigma_V^2}\right) y\right\} dy + \int_0^{\infty} \exp\left\{-\frac{y^2}{2\sigma_V^2} - \left(\frac{\sigma_V^2+\beta z}{\beta\sigma_V^2}\right) y\right\} dy}. \tag{G.15}$$

Each integral appearing in this expression can easily be evaluated using standard integral tables [69]. From the tables, we have, for given constants A and B, that

$$\int_0^{\infty} y \exp\left\{-\frac{y^2}{4A} - By\right\} dy = 2A - 2AB\sqrt{\pi A} \cdot \exp(AB^2) \cdot \operatorname{erfc}(\sqrt{A}B) \tag{G.16}$$

$$\int_0^{\infty} \exp\left\{-\frac{y^2}{4A} - By\right\} dy = \sqrt{\pi A} \cdot \exp(AB^2) \cdot \operatorname{erfc}(\sqrt{A}B), \tag{G.17}$$

where erfc$(\cdot)$ is the complementary error function given by

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt. \tag{G.18}$$

For each of the four definite integrals appearing in (G.15), the constants A and B are easy to identify. Upon substituting the integral values back into (G.15), we find, after a considerable amount algebraic manipulation, that the optimal data processor in this case reduces to

$$E\{Y|Z=z\} = \frac{(z - \sigma_V^2/\beta)G(z) + (z + \sigma_V^2/\beta)H(z)}{G(z) + H(z)} \tag{G.19}$$

$$= \frac{z(G(z) + H(z)) - (\sigma_V^2/\beta)(G(z) - H(z))}{G(z) + H(z)} \tag{G.20}$$

$$= z - \frac{\sigma_V^2}{\beta}\left[\frac{G(z) - H(z)}{G(z) + H(z)}\right], \tag{G.21}$$

where $G(\cdot)$ and $H(\cdot)$ are the rather complicated nonlinear functions defined by

$$G(z) = \exp\left\{\left(\frac{\sigma_V^2 - \beta z}{\sqrt{2}\beta\sigma}\right)^2\right\} \operatorname{erfc}\left(\frac{\sigma_V^2 - \beta z}{\sqrt{2}\beta\sigma}\right) \tag{G.22}$$

and

$$H(z) = \exp\left\{\left(\frac{\sigma_V^2 + \beta z}{\sqrt{2}\beta\sigma}\right)^2\right\} \operatorname{erfc}\left(\frac{\sigma_V^2 + \beta z}{\sqrt{2}\beta\sigma}\right). \tag{G.23}$$

It can be proven without too much difficulty (although the details will be omitted here) that the following three limits hold for the ratio $[G(z) - H(z)]/[G(z) + H(z)]$:

$$\lim_{z \to -\infty} \frac{G(z) - H(z)}{G(z) + H(z)} = -1 \tag{G.24}$$

$$\lim_{z \to 0} \frac{G(z) - H(z)}{G(z) + H(z)} = 0 \tag{G.25}$$

$$\lim_{z \to \infty} \frac{G(z) - H(z)}{G(z) + H(z)} = +1. \tag{G.26}$$

From these three facts alone, we can begin to envision the form that the optimal processor must possess. Specifically, using (G.21), we have that the optimal processor is zero at $z = 0$, approaches the positively biased linear function $z + (\sigma_V^2/\beta)$ as $z \to -\infty$, and approaches the negatively biased linear function $z - (\sigma_V^2/\beta)$ as $z \to +\infty$. A plot of the true optimal processor is shown as the solid curve in Figure G-4. Also shown in this figure is our approximation to the optimal processor that results from using our three-component Gaussian mixture specified in G.6.

Because the optimal processor and its approximation are so close in this example, they yield nearly identical mean squared error values of about 4.07. Since the Laplacian pdf represents a deviation from a Gaussian pdf that is not very severe, a linear processor would probably be adequate in this case. A logical choice for a linear estimator would be the optimal processor associated with a Gaussian signal pdf whose standard deviation is the same as that of the Laplacian pdf, namely $\sigma_Y = \sqrt{2}\beta = 2\sqrt{2}$; applying such a processor in this example yields a slightly higher MSE value of about 4.24.

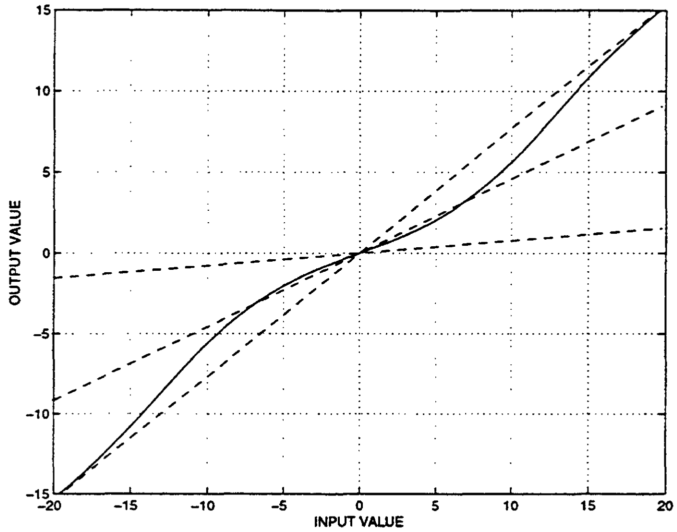


Figure G-3: Linear data processors (dashed curves) associated with individual components of the Gaussian mixture and the resulting approximate nonlinear data processor (solid curve) associated with overall mixture.

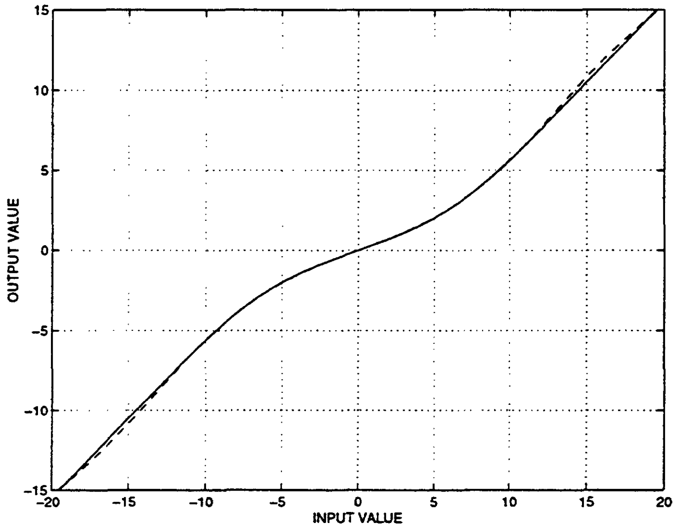


Figure G-4: Optimal processor (solid curve) and approximate processor (dashed curve) for estimating a Laplacian random variable in additive Gaussian noise.

Bibliography

[1] G. A. Ackerson and K. S. Fu, "On state estimation in switching environments," *IEEE Transactions on Automatic Control*, vol. AC–15, pp. 10–17, February 1970.

[2] L. V. Ahlfors, *Complex Analysis.* New York: McGraw-Hill, 3rd ed., 1979.

[3] M. Aitkin and G. Tunnicliffe-Wilson, "Mixture models, outliers, and the EM algorithm," *Technometrics*, vol. 22, pp. 325–332, 1980.

[4] H. Akaike, "Statistical predictor identification," *Annals of the Institute of Statistical Mathematics*, vol. 22, pp. 203–217, 1970.

[5] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Proceedings of the 2nd International Symposium on Information Theory*, (Budapest, Hungary), pp. 267–281, 1973.

[6] H. Akaike, "A new look at statistical model identification," *IEEE Transactions on Automatic Control*, vol. AC–19, pp. 716–723, December 1974.

[7] H. Akaike, "Information measures and model selection," *Bulletin of the International Statistical Institute*, vol. 50, pp. 277–290, 1983.

[8] H. Akashi and H. Kumamoto, "Random sampling approach to state estimation in switching environments," *Automatica*, vol. 13, pp. 429–434, July 1977.

[9] S. M. Ali and S. D. Silvey, "A general class of coefficients of divergence of one distribution from another," *Journal of the Royal Statistical Society*, vol. 28, pp. 131–142, 1966.

[10] D. L. Alspach, *A Bayesian Approximation Technique for Estimation and Control of Time-Discrete Stochastic Systems.* PhD thesis, University of California, San Diego, CA, 1970.

[11] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximation," *IEEE Transactions on Automatic Control*, vol. AC–17, pp. 439–448, 1972.

[12] D. L. Alspach, "Gaussian sum approximations in nonlinear filtering and control," *Information Sciences*, vol. 7, pp. 271–290, 1974.

[13] S. Amari, *Differential-Geometrical Methods in Statistics.* New York: Springer-Verlag, 1990.

[14] B. D. O. Anderson and J. B. Moore, *Optimal Filtering.* Englewood Cliffs, NJ: Prentice-Hall, 1979.

[15] T. W. Anderson, *Introduction to Multivariate Statistical Analysis.* New York: John Wiley & Sons, 2nd ed., 1984.

[16] M. Athans, R. P. Wishner, and A. Bertolini, "Suboptimal state estimation for continuous-time nonlinear systems for discrete noisy measurements," *IEEE Transactions on Automatic Control*, vol. AC–13, pp. 504–514, 1968.

[17] M. Avriel, *Nonlinear Programming: Analysis and Methods.* Englewood Cliffs, NJ: Prentice-Hall, 1976.

[18] Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Transactions on Automatic Control*, vol. AC–23, pp. 618–626, August 1978.

[19] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Annals of Mathematical Statistics*, vol. 37, pp. 1554–1563, 1966.

[20] L. E. Baum and J. A. Egon, "An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology," *Bulletin of the American Meteorological Society*, vol. 73, pp. 360–363, 1967.

[21] L. E. Baum and G. R. Sell, "Growth functions for transformations on manifolds," *Pacific Journal of Mathematics*, vol. 27, no. 2, pp. 211–227, 1968.

[22] 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 of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[23] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1–8, 1972.

[24] C. Beightler, D. Phillips, and D. Wilde, *Foundations of Optimization.* Englewood Cliffs, NJ: Prentice-Hall, 1979.

[25] R. E. Bellman, *Dynamic Programming.* Princeton, NJ: Princeton University Press, 1957.

[26] S. L. Bernstein, "Long-range communication at extremely low frequency," *Proceedings of the IEEE*, vol. 62, pp. 292–312, March 1974.

[27] U. N. Bhat, *Elements of Applied Stochastic Processes.* New York: John Wiley & Sons, 1972.

[28] P. Billingsley, "Statistical methods in Markov chains," *Annals of Mathematical Statistics*, vol. 32, pp. 12–40, 1961.

[29] M. Bouvet and S. C. Schwartz, "Comparison of adaptive and robust receivers for signal detection in ambient underwater noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-37, pp. 621–626, May 1989.

[30] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control.* San Francisco: Holden-Day, 1970.

[31] J. M. H. Bruckner and R. W. Scott, "State estimation for systems with multiple operational modes," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 201–204, 1972.

[32] R. S. Bucy, "Bayes theorem and digital realization for nonlinear filters," *Journal of Astronautical Science*, vol. 17, pp. 80–94, 1969.

[33] R. S. Bucy, "Linear and nonlinear filtering," *Proceedings of the IEEE*, vol. 58, pp. 854–864, 1970.

[34] R. S. Bucy and K. D. Senne, "Digital synthesis of non-linear filters," *Automatica*, vol. 7, pp. 287–298, 1971.

[35] J. P. Burg, *Maximum Entropy Spectral Analysis.* PhD thesis, Stanford University, Stanford, CA, 1975.

[36] P. J. Buxbaum and R. A. Haddad, "Recursive optimal estimation for a class of non-gaussian processes," in *Proceedings of the Symposium on Computer Processing in Communications*, (Polytechnic Institute of Brooklyn), April 1969.

[37] M. J. Caputi and R. L. Moose, "A modified Gaussian sum approach to estimation of non-Gaussian signals," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-29, pp. 446–451, 1993.

[38] B. P. Carlin, N. G. Polson, and D. S. Stoffer, "A Monte Carlo approach to nonnormal and nonlinear state-space modeling," *Journal of the American Statistical Association*, vol. 87, pp. 493–500, 1992.

[39] J. L. Center, "Practical nonlinear filtering of discrete observations by generalized least-squares approximation of the conditional probability distribution," in *Proceedings of the 2nd Symposium on Nonlinear Identification*, (San Diego, CA), pp. 88–89, 1971.

[40] J. L. Center, *Practical Nonlinear Filtering Based on Generalized Least-Squares Approximation of the Conditional Probability Distribution.* PhD thesis, Washington University, St. Louis, MO, 1972.

[41] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: John Wiley & Sons, 1991.

[42] H. Cramér, *Mathematical Methods of Statistics.* Princeton, NJ: Princeton University Press, 1946.

[43] I. Csiszár, "Information type measures of difference of probability distributions and indirect observations," *Studia Scientiarum Mathematicarum Hungarica*, vol. 2, pp. 229–318, 1967.

[44] M. H. A. Davis, *Linear Estimation and Stochastic Control.* New York: John Wiley & Sons, 1977.

[45] N. E. Day, "Estimating the components of a mixture of normal distributions," *Biometrika*, vol. 56, pp. 463–474, 1969.

[46] R. J. P. De Figueiredo and Y. G. Jan, "Spline filters," in *Proceedings of the 2nd Symposium on Nonlinear Estimation Theory and its applications*, (San Diego, CA), pp. 127–141, 1971.

[47] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals.* New York: Macmillan, 1993.

[48] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.

[49] N. R. Draper and H. Smith, *Applied Regression Analysis.* New York: John Wiley & Sons, 1950.

[50] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: John Wiley & Sons, 1973.

[51] J. Durbin, "The fitting of time series models," *Review of the Institute for International Statistics*, vol. 28, pp. 233–244, 1960.

[52] R. F. Dwyer, "Fram II single channel ambient noise statistics," Tech. Rep. 6583, NUSC, New London, CT, November 1981.

[53] Y. Ephraim, A. Dembo, and L. Rabiner, "A minimum discrimination information approach for hidden Markov modeling," in *Proceedings of the 1987 International Conference on Acoustics, Speech, and Signal Processing*, (Dallas, TX), April 1987.

[54] B. S. Everitt and D. J. Hand, *Finite Mixture Distributions.* London: Chapman and Hall, 1981.

[55] W. Feller, *Introduction to Probability Theory and its Applications, Volume I.* New York: John Wiley & Sons, 3rd ed., 1968.

[56] T. S. Ferguson, *Mathematical Statistics: A Decision Theoretic Approach.* New York: Academic Press, 1967.

[57] M. Fisz, *Probability Theory and Mathematical Statistics.* New York: John Wiley & Sons, 1963.

[58] R. Fletcher, *Practical Methods of Optimization.* New York: John Wiley & Sons, 2nd ed., 1987.

[59] B. Friedlander and B. Porat, "Asymptotically optimal estimation of MA and ARMA parameters of non-Gaussian processes from high-order moments," *IEEE Transactions on Automatic Control*, vol. AC–35, pp. 27–35, 1990.

[60] K. Fukunaga and T. E. Flick, "Estimation of parameters of a Gaussian mixture using the method of moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI–5, pp. 410–416, 1983.

[61] K. Fukunaga and R. R. Hayes, "The reduced Parzen classifier," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI–11, pp. 423–425, 1989.

[62] A. R. Gallant, *Nonlinear Statistical Models.* New York: John Wiley & Sons, 1987.

[63] A. Gelb, *Applied Optimal Estimation.* Cambridge, MA: MIT Press, 1974.

[64] G. B. Giannakis, "Cumulants: a powerful tool in signal processing," *Proceedings of the IEEE*, vol. 75, pp. 1333–1334, September 1987.

[65] G. B. Giannakis and J. M. Mendel, "Identification of non-minimum phase systems using higher order statistics," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 360–377, March 1989.

[66] G. B. Giannakis and A. Swami, "On estimating noncausal nonminimum phase ARMA models of non-Gaussian processes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP–38, pp. 478–495, March 1990.

[67] G. B. Giannakis and M. K. Tsatsanis, "A unifying maximum-likelihood view of cumulant and polyspectral measures for non-Gaussian signal classification and estimation," *IEEE Transactions on Information Theory*, vol. IT–38, pp. 386–406, March 1992.

[68] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "A novel approach to nonlinear, non-Gaussian state estimation," *IEE Proceedings, Part F*, vol. 140, pp. 107–113, 1993.

[69] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products.* New York: Academic Press, 5th ed., 1994.

[70] C. W. J. Granger and P. Newbold, *Forecasting Economic Time Series.* Orlando: Academic Press, 2nd ed., 1986.

[71] I. A. Gura and L. J. Henrikson, "A unified approach to nonlinear estimation," *Journal of Astronautical Science*, vol. 16, pp. 68–78, 1969.

[72] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions.* New York: John Wiley & Sons, 1986.

[73] J. A. Hartigan, "The likelihood principle and invariance principles," *Journal of the Royal Statistical Society, Series B*, vol. 29, pp. 533–539, 1967.

[74] A. C. Harvey, *Forecasting, Structural Times Series Models and the Kalman Filter.* Cambridge: Cambridge University Press, 1989.

[75] C. Hecht, "Digital realization of nonlinear filters," in *Proceedings of the Second Symposium on Nonlinear Estimation Theory and its Application*, pp. 152–158, North Hollywood: Western Periodicals, 1972.

[76] C. W. Helstrom, *Statistical Theory of Signal Detection.* Oxford: Pergamon Press, 2nd ed., 1968.

[77] D. W. Hosmer, "A comparison of iterative maximum likelihood estimates of the parameters of a mixture of two normal distributions under three different types of sample," *Biometrics*, vol. 29, pp. 761–770, 1973.

[78] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden Markov Models for Speech Recognition.* Edinburgh: Edinburgh University Press, 1990.

[79] P. J. Huber, *Robust Statistics.* New York: John Wiley & Sons, 1981.

[80] L. Izzo and L. Paura, "Error probability for fading CPSK signals in Gaussian and impulsive atmospheric noise environments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES–17, pp. 719–722, September 1981.

[81] L. Izzo, L. Panico, and L. Paura, "Error rates for fading NCFSK signals in an additive mixture of impulsive and Gaussian noise," *IEEE Transactions on Communcations*, vol. COM–30, pp. 2434–2438, November 1982.

[82] A. H. Jazwinski, "Filtering for nonlinear dynamical systems," *IEEE Transactions on Automatic Control*, vol. AC–11, pp. 765–766, 1966.

[83] A. H. Jazwinski, *Stochastic Processes and Filtering Theory.* New York: Academic Press, 1970.

[84] F. Jelinek, *Statistical Methods for Speech Recognition.* Cambridge, MA: MIT Press, 1997.

[85] B.-H. Juang and L. R. Rabiner, "Mixture autoregressive hidden Markov models for speech signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, pp. 1404–1413, 1985.

[86] B.-H. Juang and L. R. Rabiner, "A probabilistic distance measure for hidden Markov models," *AT&T Technical Journal*, vol. 64, pp. 391–408, February 1985.

[87] T. Kailath, "A view of three decades of linear filtering theory," *IEEE Transactions on Information Theory*, vol. IT–20, pp. 146–181, March 1974.

[88] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering (ASME), Series D*, vol. 82, pp. 35–45, March 1960.

[89] R. E. Kalman and R. Bucy, "New results in filtering and prediction theory," *Journal of Basic Engineering (ASME), Series D*, vol. 83, pp. 95–108, March 1961.

[90] S. A. Kassam and H. V. Poor, "Robust techniques for signal processing: a survey," *Proceedings of the IEEE*, vol. 73, pp. 433–481, 1985.

[91] S. A. Kassam, *Signal Detection in Non-Gaussian Noise.* New York: Springer-Verlag, 1988.

[92] S. M. Kay, *Modern Spectral Estimation: Theory and Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[93] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[94] D. Kazakos and P. Papantoni-Kazakos, *Detection and Estimation.* New York: Computer Science Press, 1990.

[95] J. G. Kemeny and J. L. Snell, *Finite Markov Chains.* Princeton, NJ: Van Nostrand, 1960.

[96] M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics, Volume I.* New York: Hafner Publishing Company, 1958.

[97] D. H. Kil and F. B. Shin, *Pattern Recognition and Prediction with Applications to Signal Characterization.* Woodbury, NY: AIP Press, 1996.

[98] W. Kizner, "Optimal nonlinear estimation based on orthogonal expansions," Tech. Rep. 32–1366, Jet Propulsion Laboratory, Pasadena, CA, 1969.

[99] G. Kitagawa, "Non-Gaussian state-space modeling of nonstationary time series (with discussion)," *Journal of the American Statistical Association*, vol. 82, pp. 1032–1063, 1987.

[100] W. C. Knight, R. G. Pridham, and S. M. Kay, "Digital signal processing for sonar," *Proceedings of the IEEE*, vol. 69, pp. 1451–1506, November 1981.

[101] J. Kowalik and M. R. Osborne, *Methods for Unconstrained Optimization Problems.* New York: Elsevier, 1968.

[102] S. C. Kramer and H. W. Sorenson, "Recursive Bayesian estimation using piecewise constant approximations," *Automatica*, vol. 24, pp. 789–801, 1988.

[103] T. A. Kriz and J. V. Talacko, "Equivalence of the maximum likelihood estimator to the minimum entropy estimator," *Trabajos de Estadística*, vol. 19, pp. 55–65, 1968.

[104] R. A. Kronmal and M. E. Tarter, "The estimation of probability densities and cumulatives by Fourier series methods," *Journal of the American Statistical Association*, vol. 63, pp. 925–952, 1968.

[105] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, pp. 79–86, March 1951.

[106] S. Kullback, *Information Theory and Statistics.* New York: John Wiley & Sons, 1959.

[107] R. M. Lerner, "Design of signals," in *Lectures on Communication System Theory*, pp. 243–277, New York: McGraw-Hill, 1961.

[108] N. Levinson, "The wiener RMS (root mean square) error criterion in filter design and prediction," *Journal of Mathematical Physics*, vol. 25, pp. 261–278, 1947.

[109] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell System Technical Journal*, vol. 62, pp. 1035–1075, April 1983.

[110] K. S. Lii and M. Rosenblatt, "Deconvolution and estimation of transfer function phase and coefficients for non-gaussian linear processes," *Annals of Statistics*, vol. 10, pp. 1195–1208, 1982.

[111] L. A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Transactions on Information Theory*, vol. IT–28, pp. 729–734, September 1982.

[112] D. G. Luenberger, *Linear and Nonlinear Programming.* Reading, MA: Addison-Wesley, 1984.

[113] F. W. Machell, C. S. Penrod, and G. E. Ellis, "Statistical characteristics of ocean acoustic noise processes," in *Topics in Non-Gaussian Signal Processing*, pp. 29–57, New York: Springer-Verlag, 1989.

[114] P. Mandl, "Estimation and control in Markov chains," *Advances in Applied Probability*, vol. 6, pp. 40–60, 1974.

[115] S. L. Marple, *Digital Spectral Analysis with Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[116] R. D. Martin and S. C. Schwartz, "Robust detection of a known signal in nearly Gaussian noise," *IEEE Transactions on Information Theory*, vol. IT–17, pp. 50–56, 1971.

[117] R. D. Martin, "Robust methods for time series," in *Applied Time Series II*, New York: Academic Press, 1981.

[118] R. D. Martin and D. J. Thomson, "Robust resistant spectrum estimation," *Proceedings of the IEEE*, vol. 70, pp. 1097–1115, 1982.

[119] C. J. Masreliez and R. D. Martin, "Robust Bayesian estimation for the linear model and robustifying the Kalman filter," *IEEE Transactions on Automatic Control*, vol. AC–22, pp. 361–371, 1977.

[120] W. B. McCain and C. D. McGillem, "Performance improvement of DPLLs in non-Gaussian noise using robust estimators," *IEEE Transactions on Communications*, vol. COM–35, no. 11, pp. 1207–1216, 1987.

[121] G. J. McLachlan and K. E. Basford, *Mixture Models.* New York: Marcel Dekker, 1988.

[122] R. J. Meinhold and N. D. Singpurwalla, "Robustification of Kalman filter models," *Journal of the American Statistical Association*, vol. 84, pp. 479–486, 1989.

[123] J. M. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications," *Proceedings of the IEEE*, vol. 79, pp. 278–305, March 1991.

[124] P. Mertz, "Model of impulsive noise for data transmission," *IRE Transactions on Communications Systems*, vol. 9, pp. 130–137, June 1961.

[125] D. Middleton, *An Introduction to Statistical Communication Theory.* New York: McGraw-Hill, 1960.

[126] D. Middleton, "Statistical-physical models of electromagnetic interference," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC–19, no. 3, pp. 106–127, 1977.

[127] D. Middleton, "Procedures for determining the parameters of the first-order canonical models of class A and class B electromagnetic interference," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC–21, no. 3, pp. 190–208, 1979.

[128] D. C. Montgomery, L. A. Johnson, and J. S. Gardiner, *Forecasting and Time Series Analysis.* New York: McGraw-Hill, 1990.

[129] V. K. Murthy, "Estimation of probability density," *Annals of Mathematical Statistics*, vol. 36, pp. 1027–1031, June 1965.

[130] C. Myers, A. Singer, F. Shin, and E. Church, "Modeling chaotic systems with hidden Markov models," in *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, (San Francisco), March 1992.

[131] A. Netto, M. L. Gimeno, and M. J. Mendes, "A new spline algorithm for nonlinear filtering of discrete-time systems," in *Proceedings of the 4th IFAC Symposium on Identification and System Parameter Estimation*, (Tbilisi, USSR), pp. 2123–2130, 1978.

[132] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society of London*, vol. 231, pp. 289–337, 1933.

[133] C. L. Nikias and M. R. Raghuveer, "Bispectrum estimation: a digital signal processing framework," *Proceedings of the IEEE*, vol. 75, pp. 869–891, July 1987.

[134] C. L. Nikias and J. M. Mendel, "Signal processing with higher-order spectra," *IEEE Signal Processing Magazine*, vol. 10, pp. 10–37, July 1993.

[135] C. L. Nikias and A. P. Petropulu, *Higher-Order Spectral Analysis: A Nonlinear Signal Processing Framework*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[136] P. L. Odell and J. P. Basu, "Concerning several methods for estimating crop acreages using remote sensing data," *Communications in Statistics: Theory and Methods*, vol. 5, pp. 1091–1114, 1976.

[137] R. Ohap and A. R. Stubberud, "A technique for estimating the state of a nonlinear system," *IEEE Transactions on Automatic Control*, vol. AC–10, pp. 150–155, April 1965.

[138] T. Orchard and M. A. Woodbury, "A missing information principle: theory and applications," in *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 697–715, 1987.

[139] T. Parsons, *Voice and Speech Processing*. New York: McGraw-Hill, 1986.

[140] E. Parzen, "On the estimation of a probability density function and mode," *Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.

[141] K. Pearson, "The method of moments and the method of maximum likelihood," *Biometrika*, vol. 28, pp. 34–59, 1936.

[142] B. C. Peters and H. F. Walker, "An iterative procedure for obtaining maximum-likelihood estimates of the parameters for a mixture of normal distributions," *SIAM Journal of Applied Mathematics*, vol. 35, pp. 362–378, 1978.

[143] T. Petrie, "Probabilistic functions of finite state Markov chains," *Annals of Mathematical Statistics*, vol. 40, pp. 97–115, 1969.

[144] J. Picone, "Continuous speech recognition using HMMs," *IEEE Signal Processing Magazine*, vol. 7, pp. 26–41, 1990.

[145] S. S. Pillai and M. Harisankar, "Simulated performance of a DS spread-spectrum system in impulsive atmospheric noise," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC–29, pp. 80–82, February 1987.

[146] B. T. Poljak and Y. Z. Tsypkin, "Robust identification," *Automatica*, vol. 16, pp. 53–63, 1980.

[147] H. V. Poor, *An Introduction to Signal Detection and Estimation.* New York: Spinger-Verlag, 1994.

[148] B. Porat and B. Friedlander, "Performance analysis of parameter estimation algorithms based on high-order moments," *International Journal of Adaptive Control and Signal Processing*, vol. 3, pp. 191–229, 1989.

[149] B. Porat, *Digital Processing of Random Signals: Theory and Methods.* Englewood Cliffs, NJ: Prentice-Hall, 1994.

[150] A. B. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proceedings of the 1982 International Conference on Acoustics, Speech, and Signal Processing*, (Paris), May 1982.

[151] A. B. Poritz, "Hidden Markov models: a guided tour," in *Proceedings of the 1988 International Conference on Acoustics, Speech, and Signal Processing*, (New York), April 1988.

[152] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction.* New York: Springer-Verlag, 1988.

[153] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals.* Englewood Cliffs, NJ: Prentice-Hall, 1978.

[154] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–285, 1989.

[155] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[156] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood, and the EM algorithm," *SIAM Review*, vol. 26, April 1984.

[157] M. D. Richard, *Estimation and Detection with Chaotic Systems.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1994.

[158] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[159] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *Annals of Mathematical Statistics*, vol. 27, pp. 832–837, 1956.

[160] W. Rudin, *Principles of Mathematical Analysis.* New York: McGraw-Hill, 1964.

[161] W. Rudin, *Real and Complex Analysis.* New York: McGraw-Hill, 1974.

[162] P. A. Rund, "Extensions of estimation methods using the EM algorithm," *Journal of Econometrics*, vol. 49, pp. 305–341, 1991.

[163] E. B. Saff and A. D. Snider, *Fundamentals of Complex Analysis for Mathematics, Science, and Engineering.* Englewood Cliffs, NJ: Prentice-Hall, 2nd ed., 1993.

[164] A. P. Sage and J. L. Melsa, *Estimation Theory with Applications to Communications and Control.* New York: McGraw-Hill, 1971.

[165] Y. Sawaragi, T. Katayama, and S. Fujishige, "Sequential state estimation with interrupted observations," *Information and Control*, vol. 21, pp. 56–64, 1972.

[166] L. L. Scharf, *Statistical Signal Processing.* Reading, MA: Addison-Wesley, 1991.

[167] E. J. Schlossmacher, "An interactive technique for absolute deviations curve fitting," *Journal of the American Statistical Association*, vol. 68, December 1973.

[168] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[169] S. C. Schwartz and J. B. Thomas, "Detection in a non-Gaussian environment," in *Statistical Signal Processing*, New York: Marcel Dekker, 1984.

[170] D. W. Scott and J. R. Thompson, "Probability density estimation in higher dimensions," in *Computer Science and Statistics*, pp. 173–179, Amsterdam: North-Holland, 1983.

[171] D. Sengupta and S. Kay, "Efficient estimation of parameters for non-Gaussian autoregressive processes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, June 1989.

[172] K. D. Senne and R. S. Bucy, "Digital realization of optimal discrete-time nonlinear filters," in *Proceedings of the 4th Annual Princeton Conference on Systems Science*, pp. 280–284, 1970.

[173] J. Seo, S. Cho, and K. Feher, "Impact of non-Gaussian impulsive noise on the performance of high-level QAM," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC–31, pp. 177–180, May 1989.

[174] M. P. Shinde and S. N. Gupta, "Signal detection in the presence of atmospheric noise in tropics," *IEEE Transactions on Communications*, vol. COM–22, pp. 1055–1063, August 1974.

[175] J. E. Shore and R. W. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Transactions on Information Theory*, vol. IT–26, pp. 26–37, January 1980.

[176] J. E. Shore and R. W. Johnson, "Properties of cross-entropy minimization," *IEEE Transactions on Information Theory*, vol. IT–27, pp. 472–482, July 1981.

[177] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *Journal of Time Series Analysis*, vol. 3, pp. 253–264, 1982.

[178] B. W. Silverman, *Density Estimation for Statistics and Data Analysis.* London: Chapman and Hall, 1986.

[179] R. A. Singer and R. G. Sea, "Derivation and evaluation of improved tracking filters for use in dense multi-target environments," *IEEE Transactions on Information Theory*, vol. IT–20, pp. 423–431, 1974.

[180] A. C. Singer, G. W. Wornell, and A. V. Oppenheim, "Nonlinear autoregressive modeling and estimation in the presence of noise," *Digital Signal Processing*, vol. 4, pp. 207–221, 1994.

[181] H. W. Sorenson, "Kalman filtering techniques," in *Advances in Control Systems, Volume 3*, New York: Academic Press, 1966.

[182] H. W. Sorenson and A. R. Stubberud, "Recursive filtering for systems with small but nonnegligible nonlinearities," *International Journal of Control*, vol. 7, pp. 271–280, 1968.

[183] H. W. Sorenson and A. R. Stubberud, "Nonlinear filtering by approximation of the a posteriori density," *International Journal of Control*, vol. 8, pp. 33–51, 1968.

[184] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, pp. 465–479, July 1971.

[185] H. W. Sorenson, "Estimation for dynamic systems: a perspective," in *Proceedings of the Fourth Symposium on Nonlinear Estimation Theory and its Application*, pp. 291–318, North Hollywood: Western Periodicals, 1972.

[186] H. W. Sorenson, "On the development of practical nonlinear filters," *Information Sciences*, vol. 7, pp. 253–270, 1974.

[187] H. W. Sorenson, *Parameter Estimation: Principles and Problems.* New York: Marcel Dekker, 1980.

[188] H. W. Sorenson, ed., *Kalman Filtering: Theory and Application.* Piscataway, NJ: IEEE Press, 1985.

[189] H. W. Sorenson, "Recursive estimation for nonlinear dynamic systems," in *Bayesian Analysis of Time Series and Dynamic Models*, pp. 127–165, New York: Marcel Dekker, 1988.

[190] K. Srinivasan, "State estimation by orthogonal expansion of probability distributions," *IEEE Transactions on Automatic Control*, vol. AC–15, pp. 3–10, 1970.

[191] R. Sundberg, "An iterative method for solution of the likelihood equations for incomplete data from exponential families," *Communications in Statistics: Simulation and Computation*, vol. 5, pp. 55–64, 1976.

[192] H. Tanizaki and R. S. Mariano, "Prediction, filtering, and smoothing in nonlinear and nonnormal cases using Monte Carlo integration," *Journal of Applied Econometrics*, vol. 9, no. 2, pp. 163–179, 1994.

[193] H. Tanizaki, *Nonlinear Filters: Estimation and Applications.* Berlin: Springer-Verlag, 2nd ed., 1996.

[194] H. Tanizaki, "Non-linear and non-normal filter based on Monte-Carlo technique," *Systems Analysis, Modelling, Simulation*, vol. 27, no. 4, 1997.

[195] M. E. Tarter and R. A. Kronmal, "An introduction to the implementation and theory of nonparametric density estimation," *American Statistician*, vol. 30, pp. 105–112, 1976.

[196] S. J. Taylor, *Modelling Financial Time Series.* New York: John Wiley & Sons, 1986.

[197] C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1992.

[198] J. R. Thompson, *Empirical Model Building.* New York: John Wiley & Sons, 1989.

[199] J. R. Thompson and R. A. Tapia, *Nonparametric Function Estimation, Modeling, and Simulation.* Philadelphia: Society for Industrial and Applied Mathematics, 1990.

[200] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions.* New York: John Wiley & Sons, 1985.

[201] H. Tong, *Nonlinear Time Series: A Dynamical System Approach.* New York: Oxford University Press, 1990.

[202] J. F. Traub, *Iterative Methods for the Solution of Equations.* Englewood Cliffs, NJ: Prentice-Hall, 1964.

[203] Y. Z. Tsypkin, "Use of the stochastic approximation method in estimating unknown distribution densities from observations," *Automation and Remote Control*, vol. 27, pp. 432–434, March 1966.

[204] J. K. Tugnait and A. H. Haddad, "A detection-estimation scheme for state estimation in switching environments," *Automatica*, vol. 15, pp. 477–481, 1979.

[205] J. K. Tugnait, "Detection and estimation for abruptly changing systems," *Automatica*, vol. 18, pp. 607–615, 1982.

[206] J. K. Tugnait, "Identification of nonminimum phase linear stochastic systems," *Automatica*, vol. 22, pp. 454–464, 1986.

[207] J. K. Tugnait, "Identification of linear stochastic systems via second- and fourth-order cumulant matching," *IEEE Transactions on Information Theory*, vol. IT–33, pp. 393–407, 1987.

[208] J. K. Tugnait, "Fitting noncausal autoregressive signal plus noise models to noisy non-Gaussian linear processes," *IEEE Transactions on Automatic Control*, vol. AC–32, pp. 547–552, 1987.

[209] J. K. Tugnait, "Recovering the poles from third-order cumulants of system output," *IEEE Transactions on Automatic Control*, vol. AC–34, pp. 1085–1089, 1989.

[210] J. K. Tugnait, "Consistent parameter estimation for noncausal autoregressive models via higher-order statistics," *Automatica*, vol. 26, pp. 51–61, January 1990.

[211] J. K. Tugnait, "Approaches to FIR system identification with noisy data using higher order statistics," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP–38, pp. 1307–1317, 1990.

[212] R. Urick, *Principles of Underwater Sound for Engineers.* New York: McGraw-Hill, 1967.

[213] I. Vajda, *Theory of Statistical Inference and Information.* Dordrecht: Kluwer Academic Publishers, 1989.

[214] J. Van Ryzin, "A histogram method of density estimation," *Communications in Statistics*, vol. 2, pp. 493–506, 1973.

[215] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I.* New York: John Wiley & Sons, 1968.

[216] S. M. Verbout, J. M. Ooi, J. T. Ludwig, and A. V. Oppenheim, "Parameter estimation for autoregressive Gaussian-mixture processes: the EMAX algorithm," in *Proceedings of the 1997 International Conference on Acoustics, Speech, and Signal Processing*, (Munich), April 1997.

[217] S. M. Verbout, J. M. Ooi, J. T. Ludwig, and A. V. Oppenheim, "Parameter estimation for autoregressive Gaussian-mixture processes: the EMAX algorithm," *IEEE Transactions on Signal Processing*, vol. 46, pp. 2744–2756, October 1998.

[218] T. J. Wagner, "Nonparametric estimates of probability densities," *IEEE Transactions on Information Theory*, vol. IT–21, pp. 438–440, 1975.

[219] A. T. Walden, "Non-gaussian reflectivity, entropy, and deconvolution," *Geophysics*, vol. 51, pp. 123–137, 1986.

[220] G. R. Walsh, *Methods of Optimization.* New York: John Wiley & Sons, 1975.

[221] A. H. Wang and R. L. Klein, "Implementation of nonlinear estimators using monospline," in *Proceedings of the 13th IEEE Conference on Decision and Control*, pp. 1305–1307, 1976.

[222] A. H. Wang and R. L. Klein, "Optimal quadrature formula for nonlinear estimators," *Information Sciences*, vol. 16, pp. 169–184, 1978.

[223] G. S. Watson and M. R. Leadbetter, "On the estimation of a probability density," *Annals of Mathematical Statistics*, vol. 34, pp. 480–491, 1963.

[224] G. S. Watson, "Density estimation by orthogonal series," *Annals of Mathematical Statistics*, vol. 40, pp. 1496–1498, 1969.

[225] E. J. Wegman and J. G. Smith, eds., *Statistical Signal Processing.* New York: Marcel Dekker, 1984.

[226] E. J. Wegman, S. C. Schwartz, and J. B. Thomas, eds., *Topics in Non-Gaussian Signal Processing.* New York: Springer-Verlag, 1989.

[227] A. Wernersson, "On Bayesian estimators for discrete time linear with Markovian parameters," in *Proceedings of the Sixth Symposium on Nonlinear Estimation and its Applications*, (San Diego, CA), September 1975.

[228] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series.* New York: John Wiley & sons, 1949.

[229] R. P. Wishner, J. A. Tabaczynski, and M. Athans, "A comparison of three non-linear filters," *Automatica*, vol. 5, pp. 487–496, 1969.

[230] J. S. Wit, "Advances in anti-submarine warfare," *Scientific American*, vol. 244, no. 2, pp. 31–41, 1981.

[231] C. F. J. Wu, "On the convergence properties of the EM algorithm," *Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[232] Y. Zhao, X. Zhuang, and S.-J. Ting, "Gaussian mixture density modeling of non-Gaussian source for autoregressive process," *IEEE Transactions on Signal Processing*, vol. 43, April 1995.

[233] M. A. Zissman, "Automatic language identification using Gaussian mixtures and hidden Markov models," in *Proceedings of the 1993 International Conference on Acoustics, Speech, and Signal Processing*, (Minneapolis, MN), 1993.