Springer Series in Statistics

Olivier Cappé
Eric Moulines
Tobias Rydén

Inference in Hidden Markov Models

With 78 Illustrations

Springer

Olivier Cappé
CNRS LTCI
GET / Télécom Paris
46 rue Barrault
75634 Paris cedex 13
France
cappe@tsi.enst.fr

Eric Moulines
CNRS LTCI
GET / Télécom Paris
46 rue Barrault
75634 Paris cedex 13
France
moulines@tsi.enst.fr

Tobias Rydén
Centre for Mathematical Sciences
Lund University
Box 118
221 00 Lund
Sweden
tobias@maths.lth.se

Library of Congress Control Number: 2005923551

ISBN-10: 0-387-40264-0 Printed on acid-free paper.
ISBN-13: 978-0387-40264-2

Printed in the United States of America. (MVY)

9 8 7 6 5 4 3 2 1

springeronline.com

Preface

Hidden Markov models—most often abbreviated to the acronym "HMMs"—are one of the most successful statistical modelling ideas that have came up in the last forty years: the use of hidden (or unobservable) states makes the model generic enough to handle a variety of complex real-world time series, while the relatively simple prior dependence structure (the "Markov" bit) still allows for the use of efficient computational procedures. Our goal with this book is to present a reasonably complete picture of statistical inference for HMMs, from the simplest finite-valued models, which were already studied in the 1960's, to recent topics like computational aspects of models with continuous state space, asymptotics of maximum likelihood, Bayesian computation and model selection, and all this illustrated with relevant running examples. We want to stress at this point that by using the term *hidden Markov model* we do not limit ourselves to models with finite state space (for the hidden Markov chain), but also include models with continuous state space; such models are often referred to as *state-space models* in the literature.

We build on the considerable developments that have taken place during the past ten years, both at the foundational level (asymptotics of maximum likelihood estimates, order estimation, etc.) and at the computational level (variable dimension simulation, simulation-based optimization, etc.), to present an up-to-date picture of the field that is self-contained from a theoretical point of view and self-sufficient from a methodological point of view. We therefore expect that the book will appeal to academic researchers in the field of HMMs, in particular PhD students working on related topics, by summing up the results obtained so far and presenting some new ideas. We hope that it will similarly interest practitioners and researchers from other fields by leading them through the computational steps required for making inference in HMMs and/or providing them with the relevant underlying statistical theory.

The book starts with an introductory chapter which explains, in simple terms, what an HMM is, and it contains many examples of the use of HMMs in fields ranging from biology to telecommunications and finance. This chapter also describes various extension of HMMs, like models with autoregression or hierarchical HMMs. Chapter 2 defines some basic concepts like transition kernels and Markov chains. The remainder of the book is divided into three parts: *State Inference*, *Parameter Inference* and *Background and Complements*; there are also three appendices.

Part I of the book covers inference for the unobserved state process. We start in Chapter 3 by defining smoothing, filtering and predictive distributions and describe the forward-backward decomposition and the corresponding recursions. We do this in a general framework with no assumption on finiteness of the hidden state space. The special cases of HMMs with finite state space and Gaussian linear state-space models are detailed in Chapter 5. Chapter 3 also introduces the idea that the conditional distribution of the hidden Markov chain, given the observations, is Markov too, although non-homogeneous, for both ordinary and time-reversed index orderings. As a result, two alternative algorithms for smoothing are obtained. A major theme of Part I is simulation-based methods for state inference; Chapter 6 is a brief introduction to Monte Carlo simulation, and to Markov chain Monte Carlo and its applications to HMMs in particular, while Chapters 7 and 8 describe, starting from scratch, so-called sequential Monte Carlo (SMC) methods for approximating filtering and smoothing distributions in HMMs with continuous state space. Chapter 9 is devoted to asymptotic analysis of SMC algorithms. More specialized topics of Part I include recursive computation of expectations of functions with respect to smoothed distributions of the hidden chain (Section 4.1), SMC approximations of such expectations (Section 8.3) and mixing properties of the conditional distribution of the hidden chain (Section 4.3). Variants of the basic HMM structure like models with autoregression and hierarchical HMMs are considered in Sections 4.2, 6.3.2 and 8.2.

Part II of the book deals with inference for model parameters, mostly from the maximum likelihood and Bayesian points of views. Chapter 10 describes the expectation-maximization (EM) algorithm in detail, as well as its implementation for HMMs with finite state space and Gaussian linear state-space models. This chapter also discusses likelihood maximization using gradient-based optimization routines. HMMs with continuous state space do not generally admit exact implementation of EM, but require simulation-based methods. Chapter 11 covers various Monte Carlo algorithms like Monte Carlo EM, stochastic gradient algorithms and stochastic approximation EM. In addition to providing the algorithms and illustrative examples, it also contains an in-depth analysis of their convergence properties. Chapter 12 gives an overview of the framework for asymptotic analysis of the maximum likelihood estimator, with some applications like asymptotics of likelihood-based tests. Chapter 13 is about Bayesian inference for HMMs, with the focus being on models with finite state space. It covers so-called reversible jump MCMC algorithms for choosing between models of different dimensionality, and contains detailed examples illustrating these as well as simpler algorithms. It also contains a section on multiple imputation algorithms for global maximization of the posterior density.

Part III of the book contains a chapter on discrete and general Markov chains, summarizing some of the most important concepts and results and applying them to HMMs. The other chapter of this part focuses on order estimation for HMMs with both finite state space and finite output alphabet; in particular it describes how concepts from information theory are useful for elaborating on this subject.

Various parts of the book require different amounts of, and also different kinds of, prior knowledge from the reader. Generally we assume familiarity with probability and statistical estimation at the levels of Feller (1971) and Bickel and Doksum (1977), respectively. Some prior knowledge of Markov chains (discrete and/or general) is very helpful, although Part III does contain a primer on the topic; this chapter should however be considered more a brush-up than a comprehensive treatise of the subject. A reader with that knowledge will be able to understand most parts of the book. Chapter 13 on Bayesian estimation features a brief introduction to the subject in general but, again, some previous experience with Bayesian statistics will undoubtedly be of great help. The more theoretical parts of the book (Section 4.3, Chapter 9, Sections 11.2–11.3, Chapter 12, Sections 14.2–14.3 and Chapter 15) require knowledge of probability theory at the measure-theoretic level for a full understanding, even though most of the results as such can be understood without it.

There is no need to read the book in linear order, from cover to cover. Indeed, this is probably the wrong way to read it! Rather we encourage the reader to first go through the more algorithmic parts of the book, to get an overall view of the subject, and then, if desired, later return to the theoretical parts for a fuller understanding. Readers with particular topics in mind may of course be even more selective. A reader interested in the EM algorithm, for instance, could start with Chapter 1, have a look at Chapter 2, and then proceed to Chapter 3 before reading about the EM algorithm in Chapter 10. Similarly a reader interested in simulation-based techniques could go to Chapter 6 directly, perhaps after reading some of the introductory parts, or even directly to Section 6.3 if he/she is already familiar with MCMC methods. Each of the two chapters entitled "Advanced Topics in..." (Chapters 4 and 8) is really composed of three disconnected complements to Chapters 3 and 7, respectively. As such, the sections that compose Chapters 4 and 8 may be read independently of one another. Most chapters end with a section entitled "Complements" whose reading is not required for understanding other parts of the book—most often, this section mostly contains bibliographical notes—although in some chapters (9 and 11 in particular) it also features elements needed to prove the results stated in the main text.

Even in a book of this size, it is impossible to include all aspects of hidden Markov models. We have focused on the use of HMMs to model long, potentially stationary, time series; we call such models *ergodic HMMs*. In other applications, for instance speech recognition or protein alignment, HMMs are used to represent short variable-length sequences; such models are often called *left-to-right HMMs* and are hardly mentioned in this book. Having said that we stress that the computational tools for both classes of HMMs are virtually the same. There are also a number of generalizations of HMMs which we do not consider. In Markov random fields, as used in image processing applications, the Markov chain is replaced by a graph of dependency which may be represented as a two-dimensional regular lattice. The numerical techniques that can be used for inference in hidden Markov random fields are similar to some of the methods studied in this book but the statistical side is very different. Bayesian networks are even more general since the dependency structure is allowed to take any form represented by a (directed or undirected) graph. We do not consider Bayesian networks in their generality although some of the concepts developed in the Bayesian networks literature (the graph representation, the sum-product algorithm) are used. Continuous-time HMMs may also be seen as a further generalization of the models considered in this book. Some of these "continuous-time HMMs", and in particular partially observed diffusion models used in mathematical finance, have recently received considerable attention. We however decided this topic to be outside the range of the book; furthermore, the stochastic calculus tools needed for studying these continuous-time models are not appropriate for our purpose.

We acknowledge the help of Stéphane Boucheron, Randal Douc, Gersende Fort, Elisabeth Gassiat, Christian P. Robert, and Philippe Soulier, who participated in the writing of the text and contributed the two chapters that compose Part III (see next page for details of the contributions). We are also indebted to them for suggesting various forms of improvement in the notations, layout, etc., as well as helping us track typos and errors. We thank François Le Gland and Catherine Matias for participating in the early stages of this book project. We are grateful to Christophe Andrieu, Søren Asmussen, Arnaud Doucet, Hans Künsch, Steve Levinson, Ya'acov Ritov and Mike Titterington, who provided various helpful inputs and comments. Finally, we thank John Kimmel of Springer for his support and enduring patience.

Paris, France
& Lund, Sweden
March 2005

Olivier Cappé
Eric Moulines
Tobias Rydén

Contributors

We are grateful to

Randal Douc
Ecole Polytechnique
Christian P. Robert
CREST INSEE & Université Paris-Dauphine

for their contributions to Chapters 9 (Randal) and 6, 7, and 13 (Christian) as well as for their help in proofreading these and other parts of the book

Chapter 14 was written by

Gersende Fort
CNRS & LMC-IMAG
Philippe Soulier
Université Paris-Nanterre

with Eric Moulines

Chapter 15 was written by

Stéphane Boucheron
Université Paris VII-Denis Diderot
Elisabeth Gassiat
Université d'Orsay, Paris-Sud

Contents

Part I State Inference

Part II Parameter Inference

Part III Background and Complements

Part IV Appendices

1

Introduction

1.1 What Is a Hidden Markov Model?

A *hidden Markov model* (abbreviated HMM) is, loosely speaking, a Markov chain observed in noise. Indeed, the model comprises a Markov chain, which we will denote by $\{X_k\}_{k\geq 0}$, where k is an integer index. This Markov chain is often assumed to take values in a finite set, but we will not make this restriction in general, thus allowing for a quite arbitrary state space. Now, the Markov chain is *hidden*, that is, it is not observable. What is available to the observer is another stochastic process $\{Y_k\}_{k\geq 0}$, linked to the Markov chain in that X_k governs the distribution of the corresponding Y_k. For instance, Y_k may have a normal distribution, the mean and variance of which is determined by X_k, or Y_k may have a Poisson distribution whose mean is determined by X_k. The underlying Markov chain $\{X_k\}$ is sometimes called the *regime*, or *state*. All statistical inference, even on the Markov chain itself, has to be done in terms of $\{Y_k\}$ only, as $\{X_k\}$ is not observed. There is also a further assumption on the relation between the Markov chain and the observable process, saying that X_k must be the only variable of the Markov chain that affects the distribution of Y_k. This is expressed more precisely in the following formal definition.

A hidden Markov model is a bivariate discrete time process $\{X_k, Y_k\}_{k\geq 0}$, where $\{X_k\}$ is a Markov chain and, conditional on $\{X_k\}$, $\{Y_k\}$ is a sequence of independent random variables such that the conditional distribution of Y_k only depends on X_k. We will denote the state space of the Markov chain $\{X_k\}$ by X and the set in which $\{Y_k\}$ takes its values by Y.

The dependence structure of an HMM can be represented by a *graphical model* as in Figure 1.1. Representations of this sort use a directed graph without loops to describe dependence structures among random variables. The nodes (circles) in the graph correspond to the random variables, and the edges (arrows) represent the structure of the joint probability distribution, with the interpretation that the latter may be factored as a product of the conditional distributions of each node given its "parent" nodes (those that are directly

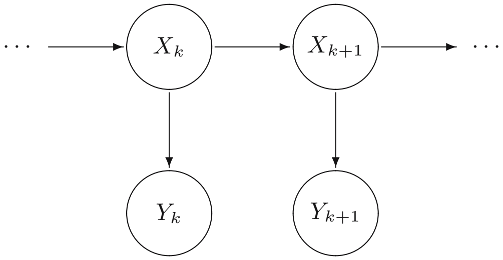

Fig. 1.1. Graphical representation of the dependence structure of a hidden Markov model, where $\{Y_k\}$ is the observable process and $\{X_k\}$ is the hidden chain.

connected to it by an arrow). Figure 1.1 thus implies that the distribution of a variable X_{k+1} conditional on the history of the process, $X_0, \ldots, X_k$, is determined by the value taken by the preceding one, X_k; this is called the *Markov property*. Likewise, the distribution of Y_k conditionally on the past observations $Y_0, \ldots, Y_{k-1}$ and the past values of the state, $X_0, \ldots, X_k$, is determined by X_k only (this is exactly the definition we made above). We shall not go into details about graphical models, but just sometimes use them as an intuitive means of illustrating various kinds of dependence. The interested reader is referred to, for example, Jensen (1996) or Jordan (2004) for introductory texts and to Lauritzen (1996), Cowell *et al.* (1999), or Jordan (1999) for in-depth coverage. Throughout the book, we will assume that each HMM is *homogeneous*, by which we mean that the Markov chain $\{X_k\}$ is homogeneous (its transition kernel does not depend on the time index k), and that the conditional law of Y_k given X_k does not depend on k either. In order to keep this introductory discussion simple, we do not embark into precise mathematical definitions of Markov chain concepts such as transition kernels for instance. The formalization of several of the ideas that are first reviewed on intuitive grounds here will be the topic of the first part of the book (Section 2.1).

As mentioned above, of the two processes $\{X_k\}$ and $\{Y_k\}$, only $\{Y_k\}$ is actually observed, whence inference on the parameters of the model must be achieved using $\{Y_k\}$ only. The other topic of interest is of course inference on the unobserved $\{X_k\}$: given a model and some observations, can we estimate the unobservable sequence of states? As we shall see later in the book, these two major statistical objectives are indeed strongly connected. Models that comprise unobserved random variables, as HMMs do, are called *latent variable models*, *missing data models*, or also *models with incomplete data*, where the latent variable refers to the unobservable random quantities.

Let us already at this point give a simple and illustrative example of an HMM. Suppose that $\{X_k\}$ is a Markov chain with state space $\{0, 1\}$ and that Y_k, conditional on $X_k = i$, has a Gaussian $\mathrm{N}(\mu_i, \sigma_i^2)$ distribution. In other words, the value of the regime governs the mean and variance of the Gaussian distribution from which we then draw the output. This model illustrates a common feature of HMMs considered in this book, namely that the conditional distributions of Y_k given X_k all belong to a single parametric family, with parameters indexed by X_k. In this case, it is the Gaussian family of distributions, but one may of course also consider the Gamma family, the Poisson family, etc. A meaningful observation, in the current example, is that the marginal distribution of $\{Y_k\}$ is that of a mixture of two Gaussian distributions. Hence we may also view HMMs as an extension of independent mixture models, including some degree of dependence between observations.

Indeed, even though the Y-variables are conditionally independent given $\{X_k\}$, $\{Y_k\}$ is not an independent sequence because of the dependence in $\{X_k\}$. In fact, $\{Y_k\}$ is not a Markov chain either: the joint process $\{X_k, Y_k\}$ is of course a Markov chain, but the observable process $\{Y_k\}$ does not have the loss of memory property of Markov chains, in the sense that the conditional distribution of Y_k given $Y_0, \ldots, Y_{k-1}$ generally depends on all the conditioning variables. As we shall see in Chapter 2, however, the dependence in the sequence $\{Y_k\}$ (defined in a suitable sense) is not stronger than that in $\{X_k\}$. This is a general observation that is valid not only for the current example.

Another view is to consider HMMs as an extension of Markov chains, in which the observation $\{Y_k\}$ of the state $\{X_k\}$ is distorted or blurred in some manner that includes some additional, independent randomness. In the previous example, the distortion is simply caused by additive Gaussian noise, as we may write this model as $Y_k = \mu_{X_k} + \sigma_{X_k} V_k$, where $\{V_k\}_{k\geq 0}$ is an i.i.d. (independent and identically distributed) sequence of standard Gaussian random variables. We could even proceed one step further by deriving a similar functional representation for the unobservable sequence of states. More precisely, if $\{U_k\}_{k\geq 0}$ denotes an i.i.d. sequence of of uniform random variables on the interval $[0, 1]$, we can define recursively $X_1, X_2, \ldots$ by the equation

$$X_{k+1} = \mathbb{1}(U_k \leq p_{X_k})$$

where p_0 and p_1 are defined respectively by $p_i = \mathrm{P}(X_{k+1} = 1 \,|\, X_k = i)$ (for $i = 0$ and 1). Such a representation of a Markov chain is usually referred to as a *stochastically recursive sequence* (and sometimes abbreviated to SRS) (Borovkov, 1998). An alternative view consists in regarding $\mathbb{1}(U_k \leq p.)$ as a random function (here on $\{0, 1\}$), hence the name *iterated random functions* also used to refer to the above representation of a Markov chain (Diaconis and Freedman, 1999). Our simple example is by no means a singular case and, in great generality, any HMM may be equivalently defined through a functional representation known as a (general) *state-space model*,

$$X_{k+1} = a(X_k, U_k) \;, \tag{1.1}$$

$$Y_k = b(X_k, V_k) \;, \tag{1.2}$$

where $\{U_k\}_{k\geq 0}$ and $\{V_k\}_{k\geq 0}$ are mutually independent i.i.d. sequences of random variables that are independent of X_0, and a and b are measurable functions. The first equation is known as the state or dynamic equation, whereas the second one is the observation equation. These two equations correspond to a recursive, generative form of the model, as opposed to our initial exposition, which focused on the specification of the joint probability distribution of the variables. Which view is most natural and fruitful typically depends on what the HMM is intended to model and for what purpose it is used (see the examples section below).

In the times series literature, the term "state-space model" is usually reserved for models in which a and b are linear functions and the sequences $\{U_k\}$, $\{V_k\}$, and X_0 are jointly Gaussian (Anderson and Moore, 1979; Brockwell and Davis, 1991; Kailath *et al.*, 2000). In this book, we reverse the perspective and refer to the family of models defined by (1.1) as (general) state-space models. The linear Gaussian sub-family of models will be covered in some detail, notably in Chapter 5, but is clearly not the main focus of this book. Similarly, in the classical HMM literature like the tutorial by Rabiner (1989) or the books by Elliott *et al.* (1995) and MacDonald and Zucchini (1997), it is tacitly assumed that the denomination "hidden Markov model" implies a finite state space X. This is a very important case indeed, but in this book we will treat more general state spaces as well. In our view, the terms "hidden Markov model" and "state-space model" refer to the same type of objects, although we will reserve the latter for describing the functional representation of the model given by (1.1).

1.2 Beyond Hidden Markov Models

The original works on (finite state space) hidden Markov models, as well as most of the theory regarding Gaussian linear state-space models, date back to the 1960s. Since then, the practical success of these models in several distinct application domains has generated an ever-increasing interest in HMMs and a similarly increasing number of new models based on HMMs. Several of these extensions of the basic HMM structure are, to some extent, also covered in this book.

A first simple extension is when the hidden state sequence $\{X_k\}_{k\geq 0}$ is a dth order Markov process, that is, when the conditional distribution of X_k given past values X_ℓ (with $0 \leq \ell < k$) depends on the d-tuple $X_{k-d}, X_{k-d+1}, \ldots, X_{k-1}$. At least conceptually this is not a very significant step, as we can fall back to the standard HMM setup by redefining the state to be the vector $(X_{k-d+1}, \ldots, X_k)$, which has Markovian evolution. Another variation consists in allowing for non-homogeneous transitions of the hidden chain or for non-homogeneous observation distributions. By this we mean that the distribution of X_k given X_{k-1}, or that of Y_k given X_k, can be allowed to depend on the index k. As we shall see in the second part of this book, non-homogeneous models lead to identical methods as far as state inference, i.e., inference about the hidden chain $\{X_k\}$, is concerned (except for the need to index conditional distributions with k).

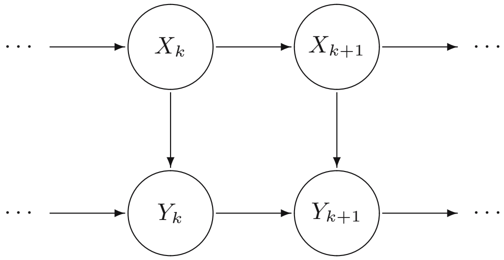

Fig. 1.2. Graphical representation of the dependence structure of a Markov-switching model, where $\{Y_k\}$ is the observable process and $\{X_k\}$ is the hidden chain.

Markov-switching models perhaps constitute the most significant generalization of HMMs. In such models, the conditional distribution of Y_{k+1}, given all past variables, depends not only on X_{k+1} but also on Y_k (and possibly more lagged Y-variables). Thus, conditional on the state sequence $\{X_k\}_{k\geq 0}$, $\{Y_k\}_{k\geq 0}$ forms a (non-homogeneous) Markov chain. Graphically, this is represented as in Figure 1.2. In state-space form, a Markov-switching model may be written as

$$X_{k+1} = a(X_k, U_k) \; , \tag{1.3}$$

$$Y_{k+1} = b(X_{k+1}, Y_k, V_{k+1}) \; . \tag{1.4}$$

The terminology regarding these models is not fully standardized and the term *Markov jump systems* is also used, at least in cases where the (hidden) state space is finite.

Markov-switching models have much in common with basic HMMs. In particular, virtually identical computational machinery may be used for both models. The statistical analysis of Markov-switching models is however much more intricate than for HMMs due to the fact that the properties of the observed process $\{Y_k\}$ are not directly controlled by those of the unobservable chain $\{X_k\}$ (as is the case in HMMs; see the details in Chapter 4). In particular, $\{Y_k\}$ is an infinite memory process whose dependence may be stronger than that of $\{X_k\}$ and it may even be the case that no stationary solution $\{Y_k\}_{k\geq 0}$ to (1.3)–(1.4) exists.

A final observation is that the computational tools pertaining to posterior inference, and in particular the smoothing equations of Chapter 3, hold in even greater generality. One could for example simply assume that $\{X_k, Y_k\}_{k\geq 0}$ jointly forms a Markov process, only a part $\{Y_k\}_{k\geq 0}$ of which is actually observed. We shall see however in the third part of the book that all statistical statements that we can currently make about the properties of estimators of the parameters of HMMs heavily rely on the fact that $\{X_k\}_{k\geq 0}$ is a Markov chain, and even more crucially, a uniformly ergodic Markov chain (see Chapter 4). For more general models such as partially observed Markov processes, it is not yet clear what type of (not overly restrictive and reasonably general) conditions are required to guarantee that reasonable estimators (such as the maximum likelihood estimator for instance) are well behaved.

1.3 Examples

HMMs and their generalizations are nowadays used in many different areas. The (partial) bibliography by Cappé (2001b) (which contains more than 360 references for the period 1990–2000) gives an idea of the reach of the domain. Several specialized books are available that largely cover applications of HMMs to some specific areas such as speech recognition (Rabiner and Juang, 1993; Jelinek, 1997), econometrics (Hamilton, 1989; Kim and Nelson, 1999), computational biology (Durbin *et al.*, 1998; Koski, 2001), or computer vision (Bunke and Caelli, 2001). We shall of course not try to compete with these in fully describing real-world applications of HMMs. We will however consider throughout the book a number of prototype HMMs (used in some of these applications) in order illustrate the variety of situations: finite-valued state space (DNA or protein sequencing), binary Markov chain observed in Gaussian noise (ion channel), non-linear Gaussian state-space model (stochastic volatility), conditionally Gaussian state-space model (deconvolution), etc.

It should be stressed that the idea one has about the nature of the hidden Markov chain $\{X_k\}$ may be quite different from one case to another. In some cases it does have a well-defined physical meaning, whereas in other cases it is conceptually more diffuse, and in yet other cases the Markov chain may be completely fictitious and the probabilistic structure of the HMM is then used only as a tool for modeling dependence in data. These differences are illustrated in the examples below.

1.3.1 Finite Hidden Markov Models

In a *finite hidden Markov model*, both the state space X of the hidden Markov chain and the set Y in which the output lies are finite. We will generally assume that these sets are $\{1, 2, \ldots, r\}$ and $\{1, 2, \ldots, s\}$, respectively. The HMM is then characterized by the transition probabilities $q_{ij} = \mathrm{P}(X_{k+1} = j \mid X_k = i)$ of the Markov chain and the conditional probabilities $g_{ij} = \mathrm{P}(Y_k = j \mid X_k = i)$.

Example 1.3.1 (Gilbert-Elliott Channel Model). The Gilbert-Elliott channel model, after Gilbert (1960) and Elliott (1963), is used in information theory to model the occurrence of transmission errors in some digital communication channels. Interestingly, this is a pre-HMM hidden Markov model, as it predates the seminal papers by Baum and his colleagues who introduced the term *hidden Markov model*.

In digital communications, all signals to be transmitted are first digitized and then transformed, a step known as *source coding*. After this preprocessing, one can safely assume that the bits that represent the signal to be transmitted form an i.i.d. sequence of fair Bernoulli draws (Cover and Thomas, 1991). We will denote by $\{B_k\}_{k \geq 0}$ the sequence of bits at the input of the transmission system.

Abstracted high-level models of how this sequence of bits may get distorted during the transmission are useful for devising efficient reception schemes and deriving performance bounds. The simplest model is the *(memoryless) binary symmetric channel* in which it is assumed that each bit may be randomly flipped by an independent error sequence,

$$Y_k = B_k \oplus V_k \ , \tag{1.5}$$

where $\{Y_k\}_{k \geq 0}$ are the observations and $\{V_k\}_{k \geq 0}$ is an i.i.d. Bernoulli sequence with $\mathrm{P}(V_k = 1) = q$, and $\oplus$ denotes modulo-two addition. Hence, the received bit is equal to the input bit B_k if $V_k = 0$; otherwise $Y_k \neq B_k$ and an error occurs.

The more realistic Gilbert-Elliott channel model postulates that errors tend to be more bursty than predicted by the memoryless channel. In this model, the channel regime is modeled as a two-state Markov chain $\{S_k\}_{k \geq 0}$, which represents low and high error conditions, respectively. The transition matrix of this chain is determined by the switching probabilities $p_0 = \mathrm{P}(S_{k+1} = 1 \,|\, S_k = 0)$ (transition into the high error regime) and $p_1 = \mathrm{P}(S_{k+1} = 0 \,|\, S_k = 1)$ (transition into the low error regime). In each regime, the model acts like the memoryless symmetric channel with error probabilities $q_0 = \mathrm{P}(Y_k \neq B_k \,|\, S_k = 0)$ and $q_1 = \mathrm{P}(Y_k \neq B_k \,|\, S_k = 1)$, where $q_0 < q_1$.

To recover the HMM framework, define the hidden state sequence as the joint process that collates the emitted bits and the sequence of regimes, $X_k = (B_k, S_k)$. This is a four-state Markov chain with transition matrix

	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(1-p_0)/2$	$p_0/2$	$(1-p_0)/2$	$p_0/2$
$(0,1)$	$p_1/2$	$(1-p_1)/2$	$p_1/2$	$(1-p_1)/2$
$(1,0)$	$(1-p_0)/2$	$p_0/2$	$(1-p_0)/2$	$p_0/2$
$(1,1)$	$p_1/2$	$(1-p_1)/2$	$p_1/2$	$(1-p_1)/2$

Neither the emitted bit B_k nor the channel regime S_k is observed directly, but the model asserts that conditionally on $\{X_k\}_{k \geq 0}$, the observations are independent Bernoulli draws with

$$\mathrm{P}(Y_k = b \,|\, B_k = b, S_k = s) = 1 - q_s \ .$$

■

Example 1.3.2 (Channel Coding and Transmission Over Memoryless Discrete Channel). We will consider in this example another elementary example of the use of HMMs, also drawn from the digital communication world. Assume we are willing to transmit a message encoded as a sequence $\{b_0, \ldots, b_m\}$ of bits, where $b_i \in \{0, 1\}$ are the bits and m is the length of the message. We wish to transmit this message over a channel, which will typically affect the transmitted message by introducing (at random) errors.

To go further, we need to have an abstract model for the channel. In this example, we will consider *discrete* channels, that is, the channel's inputs and outputs are assumed to belong to finite alphabets: $\{i_1, \ldots, i_q\}$ for the inputs and $\{o_1, \ldots, o_l\}$ for the outputs. In this book, we will most often consider binary channels only; then the inputs and the outputs of the transmission channel are bits, $q = l = 2$ and $\{i_1, i_2\} = \{o_1, o_2\} = \{0, 1\}$. A transmission channel is said to be *memoryless* if the probability of the channel's output $Y_{0:n} = y_{0:n}$ conditional on its input sequence $S_{0:n} = s_{0:n}$ factorizes as

$$\mathrm{P}(Y_{0:n} \,|\, S_{0:n}) = \prod_{i=0}^{n} \mathrm{P}(Y_i \,|\, S_i) \ .$$

In words, conditional on the input sequence $S_{0:n}$, the channel outputs are conditionally independent. The transition probabilities of the discrete memoryless channel are defined by a transition kernel $R : \{i_1, \ldots, i_q\} \times \{o_1, \ldots, o_l\} \to [0, 1]$, where for $i = 1, \ldots, q$ and $j = 1, \ldots, l$,

$$R(i_i, o_j) = \mathrm{P}(Y_0 = o_j \,|\, S_0 = i_i) \ . \tag{1.6}$$

The most classical example of a discrete memoryless channel is the *binary symmetric channel* (BSC) with binary input and binary output, for which $R(0, 1) = R(1, 0) = \varepsilon$ with $\varepsilon \in [0, 1]$. In words, every time a bit $S_k = 0$ or $S_k = 1$ is sent across the BSC, the output is also a bit $Y_k = \{0, 1\}$, which differs from the input bit with probability ε; that is, the error probability is $\mathrm{P}(Y_k \neq O_k) = \varepsilon$. As described in Example 1.3.1, the output of a binary symmetric channel can be modeled as a noisy version of the input sequence, $Y_k = S_k \oplus V_k$, where $\oplus$ is the modulo-two addition and $\{V_k\}_{k \geq 0}$ is an independent and identically distributed sequence of bits, independent of the input sequence $\{X_k\}_{k \geq 0}$ and with $\mathrm{P}\{V_k = 0\} = 1 - \varepsilon$. If we wish to transmit a message $S_{0:m} = b_{0:m}$ over a BSC without coding, the probability of getting an error will be

$$\mathrm{P}(Y_{0:m} \neq b_{0:m} \,|\, S_{0:m} = b_{0:m}) = \\ 1 - \mathrm{P}(Y_{0:m} = b_{0:m} \,|\, S_{0:m} = b_{0:m}) = 1 - (1 - \varepsilon)^m \ .$$

Therefore, as m becomes large, with probability close to 1, at least one bit of the message will be incorrectly received, which calls for practical solution. Channel coding is a viable method to increase reliability, but at the expense of reduced information rate. Increased reliability is achieved by adding redundancy to the information symbol vector, resulting in a longer coded vector of symbols that are distinguishable at the output of the channel. There are many ways to construct codes, and we consider in this example only a very elementary example of a rate 1/2 convolutional coder with memory length 2.

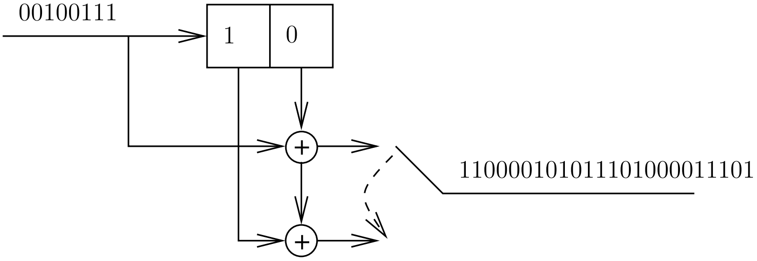

Fig. 1.3. Rate 1/2 convolutional code with memory length 2.

The rate 1/2 means that a message of length m will be transformed into a message of length $2m$, that is, we will send $2m$ bits over the transmission channel in order to introduce some kind of redundancy to increase our chance of getting an error-free message. The principle of this convolutional coder is depicted in Figure 1.3.

Because the memory length is 2, there are 4 different states and the behavior of this convolutional encoder can be captured as 4-state machine, where the state alphabet is $\mathsf{X} = \{(0,0),(0,1),(1,0),(1,1)\}$. Denote by X_k the value of the state at time k, $X_k = (X_{k,1}, X_{k,2}) \in \mathsf{X}$. Upon the arrival of the bit B_{k+1}, the state is transformed to

$$X_{k+1} = (X_{k+1,1}, X_{k+1,2}) = (B_{k+1}, X_{k,1}) \ .$$

In the engineering literature, X_k is said to be a shift register. If the sequence $\{B_k\}_{k\geq 0}$ of input bits is i.i.d. with probability $\mathrm{P}(B_k = 1) = p$, then $\{X_k\}_{k\geq 0}$ is a Markov chain with transition probabilities

$$\begin{aligned}
&\mathrm{P}[X_{k+1} = (1,1) \mid X_k = (1,0)] = \mathrm{P}[X_{k+1} = (1,1) \mid X_k = (1,1)] = p \ ,\\
&\mathrm{P}[X_{k+1} = (1,0) \mid X_k = (0,1)] = \mathrm{P}[X_{k+1} = (1,0) \mid X_k = (0,0)] = p \ ,\\
&\mathrm{P}[X_{k+1} = (0,1) \mid X_k = (1,0)] = \mathrm{P}[X_{k+1} = (0,1) \mid X_k = (1,1)] = 1-p \ ,\\
&\mathrm{P}[X_{k+1} = (0,0) \mid X_k = (0,1)] = \mathrm{P}[X_{k+1} = (0,0) \mid X_k = (0,0)] = 1-p \ ,
\end{aligned}$$

all other transition probabilities being zero. To each input bit, the convolutional encoder generates two outputs according to

$$S_k = (S_{k,1}, S_{k,2}) = (B_k \oplus X_{k,2}, B_k \oplus X_{k,2} \oplus X_{k,1}) \ .$$

These encoded bits, referred to as *symbols*, are then sent on the transmission channel. A graphical interpretation of the problem is quite useful. A convolutional encoder (or, more generally, a finite state Markovian machine) can be represented by a state transition diagram of the type in Figure 1.4. The nodes are the states and the branches represent transitions having non-zero probability. If we index the states with both the time index k and state index m, we get the *trellis diagram* of Figure 1.4. The trellis diagram shows the time

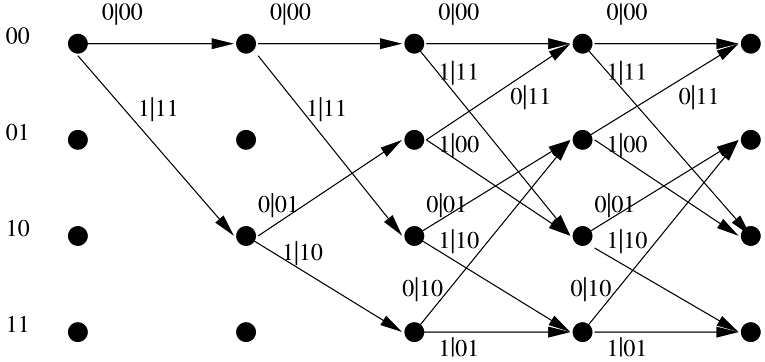

Fig. 1.4. Trellis representation of rate 1/2 convolutional code with memory length 2.

progression of the state sequences. For every state sequence, there is a unique path through the trellis diagram and *vice versa*.

More generally, the channel encoder is a finite state machine that transforms a message encoded as a finite stream of bits into an output sequence whose length is increased by a multiplicative factor that is the inverse of the rate of the encoder. If the input bits are i.i.d., the state sequence of this finite state machine is a finite state Markov chain. The m distinct states of the Markov source are $\{t_1, \ldots, t_m\}$. The outputs of this finite state machine is a sequence S_k with values in a finite alphabet $\{o_1, \ldots, o_q\}$. The state transitions of the Markov source are governed by the transition probabilities $p(i,j) = \mathrm{P}(X_n = t_j \mid X_{n-1} = t_i)$ and the output of the finite-state machine by the probabilities $q(i;j,k) = \mathrm{P}(S_n = o_i \mid X_n = t_j, X_{n-1} = t_k)$.

The Markov source always starts from the same initial state, $X_0 = t_1$ say, and produces an output sequence $S_{0:n} = (S_0, S_1, \ldots, S_n)$ ending in the terminal state $X_n = t_1$. $S_{0:n}$ is the input to a noisy *discrete memoryless channel* whose output is the sequence $Y_{0:n} = (Y_0, \ldots, Y_n)$. This discrete memoryless channel is also governed by transition probabilities (1.6). It is easy to recognize the general set-up of hidden Markov models, which are an extremely useful and popular tool in the digital communication community.

The objective of the decoder is to examine $Y_{0:n}$ and estimate the *a posteriori probability* of the states and transitions of the Markov source, i.e., the conditional probabilities $\mathrm{P}(X_k = t_i \mid Y_{0:n})$ and $\mathrm{P}(X_k = t_i, X_{k+1} = t_j \mid Y_{0:n})$.

■

Example 1.3.3 (HMM in Biology). Another example featuring finite HMMs is stochastic modeling of biological sequences. This is certainly one of the most successful examples of applications of HMM methodology in recent years. There are several different uses of HMMs in this context (see Churchill, 1992; Durbin *et al.*, 1998; Koski, 2001; Baldi and Brunak, 2001, for further references and details), and we only briefly describe the application of HMMs to gene finding in DNA, or more generally, functional annotation of sequenced genomes.

In their genetic material, all living organisms carry a blueprint of the molecules they need for the complex task of living. This genetic material is (usually) stored in the form of DNA—short for deoxyribonucleic acid—sequences. The DNA is not actually a sequence, but a long, chain-like molecule that can be specified uniquely by listing the sequence of amine bases from which it is composed. This process is known as *sequencing* and is a challenge on its own, although the number of complete sequenced genomes is growing at an impressive rate since the early 1990s. This motivates the abstract view of DNA as a sequence over a four-letter alphabet A, C, G, and T (for adenine, cytosine, guanine, and thymine—the four possible instantiations of the amine base).

The role of DNA is as a storage medium for information about the individual molecules needed in the biochemical processes of the organism. A region of the DNA that encodes a single functional molecule is referred to as a gene. Unfortunately, there is no easy way to discriminate coding regions (those that correspond to genes) from non-coding ones. In addition, the dimension of the problem is enormous as typical bacterial genomes can be millions of bases long with the number of genes to be located ranging from a few hundreds to a few thousands.

The simplistic approach to this problem (Churchill, 1992) consists in modeling the observed sequence of bases $\{Y_k\}_{k\geq 0} \in \{A, C, G, T\}$ by a two-state hidden Markov model such that the non-observable state is binary-valued with one state corresponding to non-coding regions and the other one to coding regions. In the simplest form of the model, the conditional distribution of Y_k given X_k is simply parameterized by the vector of probabilities of observing A, C, G, or T when in the coding and non-coding states, respectively. Despite its deceptive simplicity, the results obtained by estimating the parameters of this basic two-state finite HMM on actual genome sequences and then determining the smoothed estimate of the state sequence X_k (using techniques to be discussed in Chapter 3) were sufficiently promising to generate an important research effort in this direction.

The basic strategy described above has been improved during the years to incorporate more and more of the knowledge accumulated about the behavior of actual genome sequences—see Krogh *et al.* (1994), Burges and Karlin (1997), Kukashin and Borodovsky (1998), Jarner *et al.* (2001) and references therein. A very important fact, for instance, is that in coding regions the DNA is structured into *codons*, which are composed of three successive symbols in our A, C, G, T alphabet. This property can be accommodated by using higher order HMMs in which the distribution of Y_k does not only depend on the current state X_k but also on the previous two observations Y_{k-1} and Y_{k-2}. Another option consists in using non-homogeneous models such that the distribution of Y_k does not only depend on the current state X_k but also on the value of the index k modulo 3. In addition, some particular sub-sequences have a specific function, at least when they occur in a coding region (there are start and end codons for instance). Needless to say, enlarging the state space X to add specific states corresponding to those well identified functional sub-sequences is essential. Finally and most importantly, the functional description of the DNA sequence is certainly not restricted to just the coding/non-coding dichotomy, and most models use many more hidden states to differentiate between several distinct functional regions in the genome sequence. ■

Example 1.3.4 (Capture-Recapture). Capture-recapture models are often used in the study of populations with unknown sizes as in surveys, census undercount, animal abundance evaluation, and software debugging to name a few of their numerous applications. To set up the model in its original framework, we consider here the setting examined in Dupuis (1995) of a population of lizards (*Lacerta vivipara*) that move between three spatially connected zones, denoted 1, 2, and 3, the focus being on modeling these moves. For a given lizard, the sequence of the zones where it stays can be modeled as a Markov chain with transition matrix Q. This model still pertains to HMMs as, at a given time, most lizards are not observed: this is therefore a partly hidden Markov model. To draw inference on the matrix Q, the capture-recapture experiment is run as follows. At time $k = 0$, a (random) number of lizards are captured, marked, and released. This operation is repeated at times $k = 1, \ldots, n$ by tagging the newly captured animals and by recording at each capture the position (zone) of the recaptured animals. Therefore, the model consists of a series of capture events and positions (conditional on a capture) of $n+1$ cohorts of animals marked at times $k = 0, \ldots, n$. To account for open populations (as lizards can either die or leave the region of observation for good), a fourth state is usually added to the three spatial zones. It is denoted † (*dagger*) and, from the point of view of the underlying Markov chain, it is an absorbing state while, from the point of view of the HMM, it is always hidden.[1]

The observations may thus be summarized by the series $\{Y_{km}\}_{0 \le k \le n}$ of capture histories that indicate, for each lizard at least captured once (m being the lizard index), in which zone it was at each of the times it was captured. We may for instance record

$$\{y_{km}\}_{0 \le k \le n} = (0, \ldots, 0, 1, 1, 2, 0, 2, 0, 0, 3, 0, 0, 0, 1, 0, \ldots, 0) ,$$

where 0 means that the lizard was not captured at that particular time index. To each such observed sequence, there corresponds a (partially) hidden sequence $\{X_{km}\}_{0 \le k \le n}$ of lizard locations, for instance

$$\{x_{km}\}_{0 \le k \le n} = (1, \ldots, 2, \mathbf{1}, \mathbf{1}, \mathbf{2}, 2, \mathbf{2}, 3, 2, \mathbf{3}, 3, 2, 2, \mathbf{1}, \dagger, \ldots, \dagger)$$

[1] One could argue that lizards may also enter the population, either by migration or by birth. The latter reason is easily accounted for, as the age of the lizard can be assessed at the first capture. The former reason is real but will be ignored.

which indicates that the animal disappeared right after the last capture (where the values that are deterministically known from the observations have been stressed in bold).

The purposes in running capture-recapture experiments are often twofold: first, inference can be drawn on the size of the whole population based on the recapture history as in the Darroch model (Castledine, 1981; Seber, 1983), and, second, features of the population can be estimated from the captured animals, like capture and movement probabilities. ■

1.3.2 Normal Hidden Markov Models

By a *normal hidden Markov model* we mean an HMM in which the conditional distribution of Y_k given X_k is Gaussian. In many applications, the state space is finite, and we will then assume it is $\{1, 2, \ldots, r\}$. In this case, given $X_k = i$, $Y_k \sim \mathrm{N}(\mu_i, \sigma_i^2)$, so that the marginal distribution of Y_k is a finite mixture of normals.

Example 1.3.5 (Ion Channel Modeling). A cell, for example in the human body, needs to exchange various kinds of ions (sodium, potassium, etc.) with its surrounding for its metabolism and for purposes of chemical communication. The cell membrane itself is impermeable to such ions but contains so-called ion channels, each tailored for a particular kind of ion, to let ions pass through. Such a channel is really a large molecule, a protein, that may assume different configurations, or states. In some states, the channel allows ions to flow through—the channel is open—whereas in other states ions cannot pass—the channel is closed. A flow of ions is a transportation of electrical charge, hence an electric current (of the order of picoamperes). In other words, each state of the channel is characterized by a certain conductance level. These levels may correspond to a fully open channel, a closed channel, or something in between. The current through the channel can be measured using special probes (this is by no means trivial!), with the result being a time series that switches between different levels as the channel reconfigures. In this context, the main motivation is to study the characteristics of the dynamic of these ion channels, which is only partly understood, based on sampled measurements.

In the basic model, the channel current is simply assumed to be corrupted by additive white (i.i.d.) Gaussian measurement noise. If the state of the ion channel is modeled as a Markov chain, the measured time series becomes an HMM with conditionally Gaussian output and with the variances σ_i^2 not depending on i. A limitation of this basic model is that if each physical configuration of the channel (say closed) corresponds to a single state of the underlying Markov chain, we are implicitly assuming that each visit to this state has a duration drawn from a geometric distribution. A work-around that makes it possible to keep the HMM framework consists in modeling each physical configuration by a compound of distinct states of the underlying Markov chain, which are constrained to have a common conditional Gaussian output distribution. Depending on the exact transition matrix of the hidden chain, the durations spent in a given physical configuration can be modeled by negative binomial, mixtures of geometric or more complicated discrete distributions.

Further reading on ion-channel modeling can be found, for example, in Ball and Rice (1992) for basic references and Ball *et al.* (1999) and Hodgson (1998) for more advanced statistical approaches. ■

Example 1.3.6 (Speech Recognition). As yet another example of normal HMMs, we consider applications to speech recognition, which was the first area where HMMs were used extensively, starting in the early 1980s. The basic task is to, from a recording of a person's voice (or in real time, on-line), automatically determine what he or she said.

To do that, the recorded and sampled speech signal is slotted into short sections (also called frames), typically representing about 20 milliseconds of the original signal. Each section is then analyzed separately to produce a set of coefficients that represent the estimated power spectral density of the signal in the frame. This preprocessing results in a discrete-time multivariate time series of spectral coefficients. For a given word to be recognized (imagine, for simplicity, that speakers only pronounce single words), the length of the series of vectors resulting from this preprocessing is not determined beforehand but depends on the time taken for the speaker to utter the word. A primary requirement on the model is thus to cope with the time alignment problem so as to be able to compare multivariate sequences of unequal lengths.

In this application, the hidden Markov chain corresponds to sub-elements of the utterance that are expected to have comparable spectral characteristics. In particular, we may view each word as a sequence of phonemes (for instance, red: [r-e-d]; class: [k-l-a:-s]). The state of the Markov chain is then the hypothetical phoneme that is currently being uttered at a given time slot. Thus, for a word with three phonemes, like "red" for example, the state of the Markov chain may evolve according to Figure 1.5. Note that as opposed to Figures 1.1 and 1.2, Figure 1.5 is an automaton description of the Markov chain that indicates where the chain may jump to given its current state. Each arrow thus represents a possible transition that is associated with a non-zero transition probability. In this book, we shall use double circles for the nodes of such automata, as in Figure 1.5, to distinguish them from graphical models. We see that each state corresponding to a phoneme has a transition back

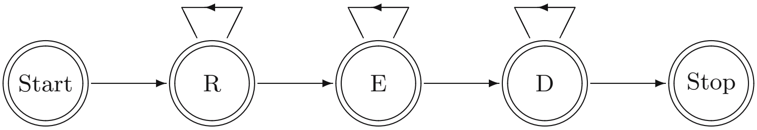

Fig. 1.5. Automaton representation of the Markov chain structure of an HMM for recognizing the word "red".

to itself, that is, a loop; this is to allow the phoneme to last for as long as the recording of it does. The purposes of the initial state *Start* and terminal state *Stop* is simply to have well-defined starts and terminations of the Markov chain; the stop state may be thought of as an absorbing state with no associated observation.

The observation vectors associated with a particular (unobservable) state are assumed to be independent and are assigned a multivariate distribution, most often a mixture of Gaussian distributions. The variability induced by this distribution is used to model spectral variability within and between speakers. The actual speech recognition is realized by running the recorded word as input to several different HMMs, each representing a particular word, and selecting the one that assigns the largest likelihood to the observed sequence. In a prior training phase, the parameters of each word model have been estimated using a large number of recorded utterances of the word. Note that the association of the states of the hidden chain with the phonemes in Figure 1.5 is more a conceptual view than an actual description of what the model does. In practice, the recognition performance of HMM-based speech recognition engines is far better than their efficiency at segmenting words into phonemes.

Further reading on speech recognition using HMMs can be found in the books by Rabiner and Juang (1993) and Jelinek (1997). The famous tutorial by Rabiner (1989) gives a more condensed description of the basic model, and Young (1996) provides an overview of current large-scale speech recognition systems. ■

1.3.3 Gaussian Linear State-Space Models

The standard state-space model that we shall most often employ in this book takes the form

$$X_{k+1} = AX_k + RU_k \ , \tag{1.7}$$

$$Y_k = BX_k + SV_k \ , \tag{1.8}$$

where

- $\{U_k\}_{k\geq 0}$, called the *state* or *process noise*, and $\{V_k\}_{k\geq 0}$, called the *measurement noise*, are independent standard (multivariate) Gaussian white noise (sequences of i.i.d. multidimensional Gaussian random variables with zero mean and identity covariance matrices);
- The initial condition X_0 is Gaussian with mean μ_ν and covariance Γ_ν and is uncorrelated with the processes $\{U_k\}$ and $\{V_k\}$;
- The *state transition matrix* A, the *measurement transition matrix* B, the square-root of the state noise covariance R, and the square-root of the measurement noise covariance S are known matrices with appropriate dimensions.