Ramon van Handel

Hidden Markov Models

Lecture Notes

This version: July 28, 2008

Contents

The following chapters are not (yet?) written. If time permits, we may cover one or more of these topics at the end of the course.

1

Hidden Markov Models

1.1 Markov Processes

Consider an E-valued stochastic process $(X_k)_{k\geq 0}$, i.e., each X_k is an E-valued random variable on a common underlying probability space $(\Omega, \mathcal{G}, \mathbf{P})$ where E is some measure space. We think of X_k as the state of a model at time k: for example, X_k could represent the price of a stock at time k (set $E = \mathbb{R}_+$), the position and momentum of a particle at time k (set $E = \mathbb{R}^3 \times \mathbb{R}^3$), or the operating status of an industrial process (set $E = \{\text{working}, \text{defective}\}$). We will refer to E as the *state space* of the process $(X_k)_{k\geq 0}$.

The process $(X_k)_{k\geq 0}$ is said to possess the *Markov property* if

$$\mathbf{P}(X_{k+1} \in A | X_0, \ldots, X_k) = \mathbf{P}(X_{k+1} \in A | X_k) \quad \text{for all } A, k.$$

In words, the Markov property guarantees that the future evolution of the process depends only on its present state, and not on its past history.

Markov processes are ubiquitous in stochastic modeling, and for good reasons. On the one hand, many models are naturally expected to be Markovian. For example, the basic laws of physics guarantee that the motion of a particle in a (small) time step is determined only by its present position and velocity; it does not matter how it ended up in this situation. On the other hand, the simple structure of Markov processes allow us to develop powerful mathematical techniques and computational algorithms which would be intractable without the Markov property. It is therefore very desirable in practice to build stochastic models which possess the Markov property.

Almost everything we will encounter in this course relies on the Markov property on some level, and this explains two of the three words in the title of these notes. In this section we recall some basic facts about Markov processes.

The transition kernel

For a succinct description of the Markov property of a stochastic process we will need the notion of a *transition kernel*.

Definition 1.1. *A* kernel *from a measurable space $(E, \mathcal{E})$ to a measurable space $(F, \mathcal{F})$ is a map $P : E \times \mathcal{F} \to \mathbb{R}_+$ such that*

1. *for every $x \in E$, the map $A \mapsto P(x, A)$ is a measure on F; and*
2. *for every $A \in \mathcal{F}$, the map $x \mapsto P(x, A)$ is measurable.*

If $P(x, F) = 1$ for every $x \in E$, the kernel P is called a transition kernel.

Let us now rephrase the definition of a Markov process. We will call the stochastic process $(X_k)_{k\geq 0}$ on the state space $(E, \mathcal{E})$ a *homogeneous Markov process* if there exists a transition kernel P from E to itself such that

$$\mathbf{P}(X_{k+1} \in A | X_0, \ldots, X_k) = P(X_k, A) \quad \text{for all } A, k.$$

Think of $P(x, A)$ as the probability that the process will be in the set $A \subset E$ in the next time step, when it is currently in the state $x \in E$. 'Homogeneous' refers to the fact that this probability is the same at every time k.

Example 1.2. Let ξ_k, $k \geq 1$ be an i.i.d. sequence of real-valued random variables with law μ, and define recursively the E-valued random variables

$$X_0 = z, \qquad X_{k+1} = f(X_k, \xi_{k+1}) \quad (k \geq 0),$$

where $f : E \times \mathbb{R} \to E$ is a measurable function and $z \in E$. Then $(X_k)_{k\geq 0}$ is a homogeneous Markov process on the state space $(E, \mathcal{E})$ with transition kernel

$$P(x, A) = \int I_A(f(x, z))\, \mu(dz), \qquad x \in E,\ A \in \mathcal{E}.$$

Indeed, note that ξ_{k+1} is independent of $X_0, \ldots, X_k$, so

$$\begin{aligned}
\mathbf{P}(X_{k+1} \in A | X_0, \ldots, X_k) &= \mathbf{E}(I_A(X_{k+1}) | X_0, \ldots, X_k) \\
&= \mathbf{E}(I_A(F(X_k, \xi_{k+1})) | X_0, \ldots, X_k) \\
&= \mathbf{E}(I_A(F(x, \xi_{k+1})) | X_0, \ldots, X_k)|_{x=X_k} \\
&= \mathbf{E}(I_A(F(x, \xi_{k+1})))|_{x=X_k} = P(X_k, A).
\end{aligned}$$

That P is indeed a kernel is easily verified (use Fubini's theorem).

When a Markov process is not homogeneous, we need to introduce a different transition kernel for every time k.

Definition 1.3. *A stochastic process $(X_k)_{k\geq 0}$ on the state space $(E, \mathcal{E})$ is called an* inhomogeneous Markov process *if there exists for every time $k \geq 0$ a transition kernel $P_k : E \times \mathcal{E} \to [0, 1]$ such that*

$$\mathbf{P}(X_{k+1} \in A | X_0, \ldots, X_k) = P_k(X_k, A) \quad \text{for every } k \geq 0,\ A \in \mathcal{E}.$$

If we can choose a single transition kernel $P = P_k$ for all k, then the process is called a homogeneous Markov process. *The probability measure μ on E defined as $\mu(A) = \mathbf{P}(X_0 \in A)$ is called the* initial measure *of $(X_k)_{k\geq 0}$.*

For simplicity we will typically work with homogeneous Markov processes, though most of the theory that we are about to develop in the following chapters does not rely on it. When not specified explicitly, we will always assume a Markov process to be homogeneous.

Remark 1.4. Under an extremely mild technical condition (that E is a Borel space—this is the case in all our examples), this definition of an inhomogeneous Markov process is equivalent to the definition of the Markov property given at the beginning of the chapter. See, e.g., [Kal02, theorem 6.3].

Finite dimensional distributions

Let $(X_k)_{k\geq 0}$ be a Markov process on the state space $(E, \mathcal{E})$ with transition kernel P and initial measure μ. What can we say about the law of this process?

Lemma 1.5. *Let $(X_k)_{k\geq 0}$ be a Markov process on E with transition kernel P and initial measure μ. Then for any bounded measurable $f : E^{k+1} \to \mathbb{R}$*

$$\mathbf{E}(f(X_0, \ldots, X_k)) = \int f(x_0, \ldots, x_k)\, P(x_{k-1}, dx_k) \cdots P(x_0, dx_1)\, \mu(dx_0).$$

Evidently the initial law and transition kernel completely determine the finite dimensional distributions, hence the law, of the Markov process $(X_k)_{k\geq 0}$.

Proof. It suffices to prove the result for functions of the form $f(x_0, \ldots, x_k) = f_0(x_0) \cdots f_k(x_k)$ (use the monotone class theorem). Note that

$$\begin{aligned}
\mathbf{E}(f_0(X_0) \cdots f_k(X_k)) &= \mathbf{E}(f_0(X_0) \cdots f_{k-1}(X_{k-1})\, \mathbf{E}(f_k(X_k) | X_0, \ldots, X_{k-1})) \\
&= \mathbf{E}\left(f_0(X_0) \cdots f_{k-1}(X_{k-1}) \int f_k(x_k)\, P(X_{k-1}, dx_k) \right) \\
&= \mathbf{E}\left(f_0(X_0) \cdots f_{k-2}(X_{k-2}) \times \right. \\
&\qquad \left. \mathbf{E}\left(f_{k-1}(X_{k-1}) \int f_k(x_k)\, P(X_{k-1}, dx_k) \,\middle|\, X_0, \ldots, X_{k-2} \right) \right) \\
&= \mathbf{E}\left(f_0(X_0) \cdots f_{k-2}(X_{k-2}) \times \right. \\
&\qquad \left. \int f_{k-1}(x_{k-1})\, f_k(x_k)\, P(x_{k-1}, dx_k)\, P(X_{k-2}, dx_{k-1}) \right) \\
&\qquad \cdots \\
&= \mathbf{E}\left(f_0(X_0) \int f_1(x_1) \cdots f_k(x_k)\, P(x_{k-1}, dx_k) \cdots P(X_0, dx_1) \right) \\
&= \int f_0(x_0) \cdots f_k(x_k)\, P(x_{k-1}, dx_k) \cdots P(x_0, dx_1)\, \mu(dx_0).
\end{aligned}$$

The proof is complete. □

Let us introduce some common notation. For any bounded measurable function $f : E \to \mathbb{R}$, we define the function $Pf : E \to \mathbb{R}$ by setting

$$Pf(x) = \int f(z)\, P(x, dz), \qquad x \in E.$$

Note that for a Markov process $(X_k)_{k\geq 0}$ with transition kernel P, we have

$$\mathbf{E}(f(X_{k+1})|X_0, \ldots, X_k) = Pf(X_k).$$

Now define recursively, for $n \geq 1$, the functions $P^n f = PP^{n-1} f$ ($P^0 f = f$). By repeated conditioning, it follows easily that

$$\begin{aligned}
\mathbf{E}(f(X_{k+n})|X_0, \ldots, X_k) &= \mathbf{E}(\mathbf{E}(f(X_{k+n})|X_0, \ldots, X_{k+n-1})|X_0, \ldots, X_k) \\
&= \mathbf{E}(Pf(X_{k+n-1})|X_0, \ldots, X_k) \\
&= \mathbf{E}(\mathbf{E}(Pf(X_{k+n-1})|X_0, \ldots, X_{k+n-2})|X_0, \ldots, X_k) \\
&= \mathbf{E}(P^2 f(X_{k+n-2})|X_0, \ldots, X_k) \qquad (\cdots) \\
&= \mathbf{E}(P^n f(X_k)|X_0, \ldots, X_k) \\
&= P^n f(X_k).
\end{aligned}$$

Similarly, let ρ be a measure on E. Define the measure ρP on E as

$$\rho P(A) = \int P(x, A)\, \rho(dx), \qquad A \in \mathcal{E},$$

and, for $n \geq 1$, the measures $\rho P^n = \rho P^{n-1} P$ ($\rho P^0 = \rho$). Then for a Markov process $(X_k)_{k\geq 0}$ with transition kernel P and initial measure μ, lemma 1.5 shows that $\mathbf{P}(X_k \in A) = \mu P^k(A)$ for all $A \in \mathcal{E}$, i.e., μP^k is the law of X_k.

Finally, we will frequently use the following fact: for any function f

$$\int f(x)\, \mu P(dx) = \int Pf(x)\, \mu(dx),$$

i.e., the maps $\mu \mapsto \mu P$ and $f \mapsto Pf$ are dual to each other.

1.2 Hidden Markov Models

In the broadest sense of the word, a hidden Markov model is a Markov process that is split into two components: an *observable* component and an *unobservable* or 'hidden' component. That is, a hidden Markov model is a Markov process $(X_k, Y_k)_{k\geq 0}$ on the state space $E \times F$, where we presume that we have a means of observing Y_k, but not X_k. Adopting terminology from signal processing, we will typically refer to the unobserved component X_k as the *signal process* and E as the *signal state space*, while the observed component Y_k is called the *observation process* and F is the *observation state space*.

Hidden Markov models appear in a wide variety of applications. To fix some ideas one might distinguish between two main classes of applications, though many applications fall somewhere in between.

On the one hand, hidden Markov models naturally describe a setting where a stochastic system is observed through noisy measurements. For example, in communications theory, one might think of X_k as a (random) signal to be transmitted through a communications channel. As the channel is noisy, the receiver observes a corrupted version Y_k of the original signal, and he might want to reconstruct as well as is possible the original signal from the noisy observations. This is the origin of the signal/observation process terminology.

On the other hand, it may be the process Y_k which is ultimately of interest, while the X_k represents the influence on Y_k of certain unobservable external factors. For example, one might think of Y_k as the market price of stock, where X_k is an unobserved economic factor process which influences the fluctuations of the stock price. We are ultimately interested in modeling the observed stock price fluctuations, not in the unobservable factor process, but by including the latter one might well be able to build a model which more faithfully reflects the statistical properties of the observed stock prices. It should be noted that even though $(X_k, Y_k)_{k\geq 0}$ is Markov, typically the observed component $(Y_k)_{k\geq 0}$ will not be Markov itself. Hidden Markov models can thus be used to model non-Markov behavior (e.g., of the stock price), while retaining many of the mathematical and computational advantages of the Markov setting.

This course is an introduction to some of the basic mathematical, statistical and computational methods for hidden Markov models. To set the stage for the rest of the course, we will describe in the next two sections a number of representative examples of hidden Markov models in applications taken from a variety of fields, and we will introduce the basic questions that will be tackled in the remainder of the course. Before we do this, however, we must give a precise definition of the class of models which we will be considering.

Definition and elementary properties

The broadest notion of a hidden Markov model, as outlined above, is a little too general to lead to a fruitful theory. Throughout this course, and in much of the literature, the term *hidden Markov model* is used to denote a Markov process $(X_k, Y_k)_{k\geq 0}$ with two essential restrictions:

- the signal $(X_k)_{k\geq 0}$ is itself a Markov process; and
- the observation Y_k is a noisy functional of X_k only (in a sense to be made precise shortly).

As we will see in the next section, there is a wide variety of applications that fit within this framework.

Definition 1.6. *A stochastic process $(X_k, Y_k)_{k\geq 0}$ on the product state space $(E\times F, \mathcal{E}\otimes\mathcal{F})$ is called a* hidden Markov model *if there exist transition kernels $P : E \times \mathcal{E} \to [0,1]$ and $\Phi : E \times \mathcal{F} \to [0,1]$ such that*

$$\mathbf{E}(g(X_{k+1}, Y_{k+1})|X_0, Y_0, \ldots, X_k, Y_k) = \int g(x, y)\, \Phi(x, dy)\, P(X_k, dx),$$

and a probability measure μ *on* E *such that*

$$\mathbf{E}(g(X_0, Y_0)) = \int g(x, y)\, \Phi(x, dy)\, \mu(dx),$$

for every bounded measurable function $g : E \times F \to \mathbb{R}$. *In this setting* μ *is called the* initial measure, P *the* transition kernel, *and* Φ *the* observation kernel *of the hidden Markov model* $(X_k, Y_k)_{k\geq 0}$.

Comparing with definition 1.3, it is immediately clear that $(X_k, Y_k)_{k\geq 0}$ and $(X_k)_{k\geq 0}$ are both (homogeneous) Markov processes. To illustrate the structure of the observations $(Y_k)_{k\geq 0}$, we consider a canonical example.

Example 1.7. Let α_k, $k \geq 1$ and β_k, $k \geq 0$ be independent i.i.d. sequences of real-valued random variables with laws α and β, respectively. Define

$$\begin{array}{ll} X_0 = z, & X_k = f(X_{k-1}, \alpha_k), \\ Y_0 = h(X_0, \beta_0), & Y_k = h(X_k, \beta_k) \end{array} \qquad (k \geq 1),$$

where $f : E \times \mathbb{R} \to E$ and $h : E \times \mathbb{R} \to F$ are measurable functions and $z \in E$. Then $(X_k, Y_k)_{k\geq 0}$ is a hidden Markov model with transition kernel

$$P(x, A) = \int I_A(f(x, z))\, \alpha(dz),$$

observation kernel

$$\Phi(x, B) = \int I_B(h(x, z))\, \beta(dz),$$

and initial measure δ_z. Indeed, as β_{k+1} is independent of $X_{0,\ldots,k+1}$ and $Y_{0,\ldots,k}$,

$$\begin{aligned} &\mathbf{E}(g(X_{k+1}, Y_{k+1})|X_0, Y_0, \ldots, X_k, Y_k) \\ &\quad = \mathbf{E}(g(X_{k+1}, h(X_{k+1}, \beta_{k+1}))|X_0, Y_0, \ldots, X_k, Y_k) \\ &\quad = \mathbf{E}(\mathbf{E}(g(X_{k+1}, h(X_{k+1}, \beta_{k+1}))|X_{0,\ldots,k+1}, Y_{0,\ldots,k})|X_0, Y_0, \ldots, X_k, Y_k) \\ &\quad = \mathbf{E}(\mathbf{E}(g(x, h(x, \beta_{k+1})))|_{x=X_{k+1}}|X_0, Y_0, \ldots, X_k, Y_k) \\ &\quad = \mathbf{E}\left(\int g(X_{k+1}, y)\, \Phi(X_{k+1}, dy) \middle| X_0, Y_0, \ldots, X_k, Y_k\right) \\ &\quad = \int g(x, y)\, \Phi(x, dy)\, P(X_k, dx). \end{aligned}$$

The corresponding expression for $E(g(X_0, Y_0))$ follows similarly.

In this example, it is immediately clear in which sense Y_k is a noisy functional of X_k only: indeed, Y_k is a function of X_k and a noise variable β_k which is independent of the noise corrupting the remaining observations Y_ℓ, $\ell \neq k$. If the observation $(Y_k)_{k\geq 0}$ represents a signal $(X_k)_{k\geq 0}$ transmitted through a noisy communications channel, this basic property corresponds to the idea that the communications channel is *memoryless*. A more formal expression of the elementary properties of our hidden Markov models is given as follows.

Lemma 1.8. *Let $(X_k, Y_k)_{k\geq 0}$ be a hidden Markov model on $(E \times F, \mathcal{E} \otimes \mathcal{F})$ with transition kernel P, observation kernel Φ, and initial measure μ. Then*

1. *$(X_k, Y_k)_{k\geq 0}$ is a Markov process;*
2. *$(X_k)_{k\geq 0}$ is Markov with transition kernel P and initial measure μ; and*
3. *$Y_0, \ldots, Y_k$ are conditionally independent given $X_0, \ldots, X_k$:*

$$\mathbf{P}(Y_0 \in A_0, \ldots, Y_k \in A_k | X_0, \ldots, X_k) = \Phi(X_0, A_0) \cdots \Phi(X_k, A_k).$$

Moreover, the finite dimensional distributions of $(X_k, Y_k)_{k\geq 0}$ are given by

$$\mathbf{E}(f(X_0, Y_0, \ldots, X_k, Y_k)) = \int f(x_0, y_0, \ldots, x_k, y_k) \times \Phi(x_k, dy_k)\, P(x_{k-1}, dx_k) \cdots \Phi(x_1, dy_1)\, P(x_0, dx_1)\, \Phi(x_0, dy_0)\, \mu(dx_0).$$

Proof. This can be read off directly from definition 1.6 and lemma 1.5. □

Nondegeneracy

In addition to the general requirements of definition 1.6, we will frequently impose a stronger assumption on the structure of the observations $(Y_k)_{k\geq 0}$.

Definition 1.9. *Let $(X_k, Y_k)_{k\geq 0}$ be a hidden Markov model on $(E \times F, \mathcal{E} \otimes \mathcal{F})$ with observation kernel Φ. The model is said to have* nondegenerate observations *if the observation kernel is of the form*

$$\Phi(x, B) = \int I_B(z)\, \Upsilon(x, z)\, \varphi(dz), \qquad x \in E,\ B \in \mathcal{F},$$

where $\Upsilon : E \times F \to]0, \infty[$ is a strictly positive measurable function and φ is a probability measure on F. The function Υ is called the observation density.

Let us attempt to explain the relevance of this assumption. Much of this course is concerned with problems where we try to infer something about the unobserved process $(X_k)_{k\geq 0}$ from observations of the observed process $(Y_k)_{k\geq 0}$. We will therefore develop techniques which take as input an observation time series $y_0, \ldots, y_k$ and which output certain conclusions about the unobserved process. We would like these techniques to be 'nondegenerate' in the sense that they can be applied even if the input time series $y_0, \ldots, y_k$ does not precisely match the mathematical model that we have assumed. If this is not the case, there would be little hope that such techniques could be applied to real-world data. Without additional assumptions, however, the general definition 1.6 can lead to models where inference becomes problematic. To make this point, let us consider a particularly extreme example.

Example 1.10. Let $E = F = \mathbb{R}$. Let ρ_k, $k \geq 0$ be an i.i.d. sequence of random variables whose law ρ is supported on the integers $\mathbb{Z}$, and let ρ'_k, $k \geq 0$ be

an i.i.d. sequence of random variables whose law is supported on the positive integers $\mathbb{N}$. We now define $(X_k, Y_k)_{k\geq 0}$ recursively as

$$X_0 = Y_0 = 0, \qquad X_k = X_{k-1} + \rho_k/\rho'_k, \quad Y_k = X_k \quad (k \geq 1).$$

This clearly defines a hidden Markov model in the sense of definition 1.6.

Now suppose that we observe a sequence of observations $y_0, \ldots, y_k$ that are generated by this model. Then it must be the case that the differences $y_n - y_{n-1}$ are rational numbers for every n, as this is true with probability one by construction. However, if in practice the signal X_n is perturbed by even the slightest amount, then a real-world sample of the observation time series $y_0, \ldots, y_k$ would no longer satisfy this property. An inference procedure based on our hidden Markov model would be at a loss as to how to deal with this observation sequence—after all, according to our model, what we have observed is technically impossible. We therefore run into trouble, as even the smallest of modeling errors can give rise to observation time series for which our inference techniques do not make mathematical sense.

This example is, of course, highly contrived. However, it highlights the fact that applying definition 1.6 without further assumptions can lead to models which are problematic to deal with. Indeed, most of the techniques that we will develop in the following chapters can not be applied to this model.

As it turns out, the nondegeneracy assumption effectively rules out this problem. The reason is that when the observation kernel Φ satisfies definition 1.9, any property of a finite number of observations $Y_0, \ldots, Y_k$ which holds with unit probability must do so *for every choice of transition kernel P and initial measure μ* (problem 1.4). As a consequence, if $y_0, \ldots, y_k$ is a valid observation sample path for some model for the signal $(X_k)_{k\geq 0}$, then this observed path is valid for any signal model. This does not mean, of course, that our inference procedures will not be sensitive to (even small) modeling errors; however, definition 1.9 guarantees enough nondegeneracy so that our inference procedures will be at least mathematically well defined.

A typical example which *does* satisfy the nondegeneracy assumption is:

Example 1.11. Let $F = \mathbb{R}$, and consider an observation model of the form

$$Y_k = h(X_k) + \xi_k \quad (k \geq 0),$$

where $h : E \to \mathbb{R}$ is measurable and ξ_k, $k \geq 0$ are i.i.d. $N(0,1)$. Then

$$\Phi(x, B) = \int I_B(z) \, \frac{e^{-(z-h(x))^2/2}}{\sqrt{2\pi}} \, dz,$$

which certainly satisfies the requirement of definition 1.9.

The above discussion was intended to provide some intuition for the nondegeneracy assumption. Its mathematical consequences will be obvious, however, when we start developing the basic theory in the following chapter.

On our assumptions

Throughout most of this course, we will develop techniques which apply to hidden Markov models in the sense of definition 1.6 that satisfy the nondegeneracy assumption of definition 1.9. That is not to say that models in which some of our assumptions do not hold are not encountered in applications, nor that such models are necessarily intractable. In many cases more general models can be treated, either by modifying the techniques which we will develop here or through other methods that we will not cover.

Fortunately, our assumptions are general enough to cover a wide range of applications, which can all be treated using a common set of techniques to be developed in the following chapters. For conceptual, mathematical and notational simplicity, and as one can only cover so much in one semester, we will from now on stay within this framework without further apology.

1.3 Examples

To motivate our mathematical definitions, we will now describe briefly some sample applications taken from various fields. Note that

- all examples are hidden Markov models in the sense of definition 1.6; and
- all examples satisfy the nondegeneracy assumption of definition 1.9.

These examples are not the most sophisticated possible, but they show that many interesting models fit within our framework. As we progress throughout the course, you may want to go back on occasion and think about how the various techniques apply to the examples in this section.

Example 1.12 (Financial time series). The simplest model of financial time series S_k, such as the market price of stock, is of the Black-Scholes form

$$S_k = \exp(\mu - \sigma^2/2 + \sigma\,\xi_k)\,S_{k-1},$$

where $\xi_k \sim N(0,1)$ are i.i.d., $\sigma \in \mathbb{R}$ is the volatility, and $\mu \in \mathbb{R}$ is the rate of return (indeed, note that $\mathbf{E}(S_k/S_{k-1}) = e^{\mu}$). High volatility means that the stock prices exhibit large random fluctuations, while high return rate means that the value of the stock increases rapidly on average.

A simple model of this type can work reasonably well on short time scales, but on longer time scales real-world stock prices exhibit properties that can not be reproduced by this model, e.g., stock prices are often observed to have non-Markovian properties. Intuitively, one might expect that this is the case because μ and σ depend on various external (economical, political, environmental) factors which are not constant on longer time scales. To incorporate this idea we can allow the volatility and/or return rates to fluctuate; for this purpose, we introduce a Markov process X_k (independent of ξ_k) and set

$$S_k = \exp(\mu(X_k) - \sigma(X_k)^2/2 + \sigma(X_k)\,\xi_k)\,S_{k-1},$$

where now μ and σ are suitably chosen functions. If we choose as our observation process the log-returns $Y_k = \log(S_k/S_{k-1})$, then $(X_k, Y_k)_{k\geq 0}$ is a hidden Markov model. By tuning the dynamics of X_k appropriately, one can obtain a stock price model that is more realistic than the Black-Scholes model.

One common choice for X_k is a real-valued recursion of the form

$$X_k = \alpha(X_{k-1}) + \beta(X_{k-1})\,\eta_k,$$

where η_k are i.i.d. If μ is constant and only the volatility σ depends on X_k, this is a typical example of a *stochastic volatility model*. A different type of model is obtained if we let X_k be a Markov process on a finite state space. Each state represents a particular 'regime': for example, the demand for a certain product might be well described as being either low or high, and the statistics of the resulting price fluctuations depend on which regime we are presently in. This type of model is called a *regime switching model*.

Note that typically only stock prices are observable to investors—even if the economic factor process X_k has some real-world significance (rather than serving as a mathematical tool to model non-Markov time series), such underlying economic factors are typcially not disclosed to the public. Therefore any modeling, inference, pricing, or investment decisions must be based on observations of the price process S_k (equivalently, Y_k) only. The purpose of the theory of hidden Markov models is to provide us with the necessary tools.

Example 1.13 (Bioinformatics). Genetic information is encoded in DNA, a long polymer found in almost all living systems which consists of a linear sequence of base pairs A, C, G, T (i.e., genetic code is a very long word in a four letter alphabet). An impressive effort in molecular biology has led to the sequencing of an enormous amount of genetic information; for example, the ordering of base pairs of almost the entire human genome has been documented by the Human Genome Project. As the genetic code plays a major role in the inner workings of the living cell, the decoding of this information ought to lead to significant scientific and medical advances.

However, the interpretation of genetic data is a highly nontrivial task. For example, one encounters the following problem. The genetic code consists of *coding* and *non-coding* regions. Coding regions directly encode the structure of proteins, which are produced in the cell by an intricate process which begins by transcribing the relevant portion of the DNA strand. Non-coding regions, however, do not directly encode molecular structure, but may serve to regulate when and how much of the protein will be produced (other 'junk DNA' non-coding regions have no known purpose). In order to interpret the genetic code, we must therefore first separate out the coding and non-coding regions. Unfortunately, there is no clear signature for when a coding region starts or ends, so that typically this identification must be done by statistical methods.

The use of hidden Markov models has been remarkably successful in approaching this problem. The simplest approach is as follows. The time parameter k represents the position along the DNA strand. The signal process X_k is a Markov process on $E = \{0, 1\}$: the kth base pair is in a coding region if $X_k = 1$, and in a non-coding region otherwise. The observation process Y_k has the four-letter state space $F = \{A, C, G, T\}$, so that Y_k represents the type of the kth base pair. The transition and observation kernels P, Φ are estimated from the sequence data. Once this is done, we can run a reverse estimation procedure to determine which regions of a DNA sequence are coding or non-coding. This approach is rather naive, yet it already gives surprisingly good results: evidently coding and non-coding regions a characterized by different relative frequencies for each of the base pairs. The approach can be improved by choosing a more sophisticated underlying hidden Markov model.

Example 1.14 (Change detection). A classical problem of sequential analysis is the detection of an abrupt change in the distribution of a noisy time series. For example, consider a chemical plant which produces independent batches of a certain product. Though each batch will have a slightly different concentration of the desired product, its distribution is such that majority of batches falls within an acceptable tolerance range (the remaining batches must be discarded). However, if a problem occurs somewhere in the plant (e.g., the stirring mechanism gets stuck), then the output distribution changes such that a larger fraction of the batches must be discarded.

A simple model for this problem is obtained as follows. Let X_k be a $\{0, 1\}$-valued Markov chain. The 0 state denotes that the process is broken, while 1 denotes normal operation; we presume that $X_0 = 1$, and that once the system breaks it can not fix itself, i.e., $P(0, \{1\}) = 0$. The observation Y_k is obtained by specifying the observation kernel Φ, such that $\Phi(1, \cdot)$ is the distribution of output concentrations under normal operation and $\Phi(0, \cdot)$ is the output distribution when the process is broken. Ultimately we would like to detect when the system breaks so that it can be repaired. As we only have at our disposal the observed output concentrations in the previous batches, an unusually large number of discarded batches can mean that the process is broken, but it can also just be a random fluctuation in the output concentrations. There is therefore always a probability of false alarm, which we would like to minimize as interrupting production for repair is costly. On the other hand, if we keep observing more and more discarded batches then the probability of false alarm is very small, but we now obtain a large delay between the occurence of the fault and its repair. The tradeoff between detection delay and false alarm probability is characteristic of this type of problem.

Variants of the change detection problem appear in many applications, including the detection of the onset of a computer network (DoS) attack from network traffic data, or detecting when an economic bubble bursts from stock price data. Another variant is the setting where different types of faults can occur; here the goal is to detect both when the fault occurs and its type.

Example 1.15 (Communications). We are interested in modeling the transmission of a digital message, i.e., a sequence of $\{0,1\}$-valued random variables B_k, $k \geq 0$ called *bits*, over a noisy channel. We suppose that the message B_k can be modelled a Markov process on the state space $E = \{0,1\}$.

What does a bit look like when it is transmitted? A classic channel model is one where the output bit Y_k equals the input bit B_k with some probability $p \in]0,1[$, and is flipped from the input bit with probability $1-p$. To model this, we introduce another sequence of i.i.d. $\{0,1\}$-valued random variables ξ_k with $\mathbf{P}(\xi_k = 0) = p$. Then the hidden Markov model

$$X_k = B_k, \qquad Y_k = (1-\xi_k)\, B_k + \xi_k\, (1-B_k)$$

describes the basic binary symmetric channel model. In order to counteract the corruption of bits, one typically does some *encoding* before transmitting the message over the noisy channel. This introduces some redundancy, which makes it more likely that the message will be decoded correctly on the other end. Encoding can be added to our hidden Markov model at the expense of a more complicated signal model. For example, hidden Markov models for convolutional codes are commonly applied in telecommunications.

In a different setting, you might imagine that the bit B_k is transmitted by maintaining a voltage B_k over a noisy satellite link. In this case, the corrupting noise is typically taken to be Gaussian, i.e., we set $Y_k = \alpha\, B_k + \xi_k$, where ξ_k, $k \geq 0$ are now i.i.d. $N(\mu, \sigma^2)$ and $\alpha \in \mathbb{R}$ is a gain coefficient. More realistic, however, would be to let α fluctuate in time in order to take into account the varying atmospheric conditions, which we model as a Markov process W_k. Let η_k, $k \geq 0$ be a sequence of i.i.d. random variables, and set

$$X_k = (B_k, W_k), \qquad W_k = f(W_{k-1}, \eta_k), \qquad Y_k = W_k\, B_k + \xi_k.$$

A channel model of this type is called a fading channel.

Ultimately, the goal of the receiver is to infer the original message B_k from the noisy observations Y_k. If we were to transmit a real-valued (analog) signal S_k through a noisy channel, instead of the digital signal B_k, this becomes a *signal processing* task of denoising the corrupted signal.

Example 1.16 (Target tracking). In various applications one is interested in tracking a moving object using noisy sensor data. Consider an object that is moving randomly in the plane: its two position components might evolve as

$$X_k^1 = X_{k-1}^1 + \xi_k^1 + \alpha^1(U_k), \qquad X_k^2 = X_{k-1}^2 + \xi_k^2 + \alpha^2(U_k),$$

where $\alpha(U_k)$ is the base velocity of the target (possibly controlled by some external process U_k), while ξ_k, $k \geq 1$ are i.i.d. and correspond to random velocity perturbations. By choosing U_k to be, e.g., a finite state Markov process, one can model a target which tries to confuse us by randomly switching its velocity in different preset directions (think of tracking the position of a fighter jet). The case $\alpha = 0$ could be used to model a large molecule which is

moving around diffusively in a thin layer of liquid (single molecule tracking for biological or polymer dynamics studies).

The noisy observations of the object to be tracked typically take the form

$$Y_k = h(X_k) + \eta_k,$$

where η_k, $k \geq 0$ are i.i.d. and h is the *observation function*. The function h can be quite nonlinear. For example, if we track the location of a jet from a fixed position on the ground, one might imagine a situation where we can only observe the direction of the line of sight between the sensor and the jet, and not the distance between the sensor and the jet. In this setting, called bearings-only tracking, one would have $h(X_k^1, X_k^2) = \arctan(X_k^2/X_k^1)$. The goal is then to track as well as possible the position of the object given any prior knowledge of its position and the observed sensor data.

There are many variations on this problem in applications such as positioning, navigation, robotics, etc. The problem obtains an additional dimension if we introduce control into the picture: e.g., the sensor might be itself mounted on another jet plane, and we might want to develop a pursuit strategy so that our trajectory intersects as closely as possible the trajectory of the other plane at a fixed time in the future. As our strategy can only depend on the observed sensor data, it is not surprising that tracking plays an important role.

Example 1.17 (Speech recognition). One of the oldest applications of hidden Markov models is automatic speech recognition. This approach turns out to be extremely successful, and almost all modern speech recoginition systems are based on hidden Markov model techniques. Let us briefly discuss the simplest type of speech recognition: the problem of isolated word recognition. In this setting our goal is to determine, on the basis of an audio recording of a human voice, which of a finite set of allowed words was spoken.

The basic idea is to use maximum likelihood estimation to solve this problem; in principle this has nothing to do with hidden Markov models. To account for the variability of human speech, the audio signal corresponding to each word is modeled as a stochastic process. Denote by $\mathbf{P}^i$ the law of the audio signal $Y_0, \ldots, Y_N$ corresponding to the ith word, and let us suppose that $\mathbf{P}^i$ is absolutely continuous with respect to some reference measure $\mathbf{Q}$ for every i. Once we are given an actual recorded signal $y_0, \ldots, y_N$, the most likely spoken word is given by the maximum likelihood estimate $\operatorname{argmax}_i \frac{d\mathbf{P}^i}{d\mathbf{Q}}(y_1, \ldots, y_N)$.

The problem is, of course, what model one should use for the laws $\mathbf{P}^i$. It is here that hidden Markov models enter the picture. The audio signal of a given word is represented as the observed component Y_k of a hidden Markov model. The unobserved component X_k is a finite state Markov process, where each state corresponds to a consecutive sound in the word of interest (e.g., for the word 'quick' one could choose $E = \{k1, w, i, k2\}$). The idea is that each sound will give rise to an audio sequence with roughly i.i.d. spectral content, but that the length of each sound within the word will vary from recording to recording. Typically $(Y_k)_{k\geq 0}$ does not represent the raw audio data (which is

highly oscillatory and not well suited for direct use); instead, the raw audio is chopped into fixed size frames ($\sim 50\,\text{ms}$ each), and each Y_k represents the dominant spectral components of the corresponding frame.

Speech recognition now proceeds as follows. First, the system is trained: a speaker provides voice samples for each allowed word, and these are used to estimate the transition and observation kernels P and Φ for the corresponding hidden Markov model. Once the training is complete, speech recognition can be performed using the maximum likelihood approach. In all cases preprocessing of the raw audio ('feature analysis') is first performed to extract the spectral information that is modeled by the hidden Markov models.

1.4 What Is This Course About?

This is not a course about stochastic modeling; it is our purpose to develop in the following chapters the basic mathematical and statistical techniques that are fudamental to the theory of hidden Markov models. Before we embark on this journey in earnest, let us give a brief overview of coming attractions. The examples in the previous section will serve as motivation.

Estimation

Suppose that we have somehow managed to obtain a hidden Markov model (i.e., the kernels P and Φ are given). As only the observations $(Y_k)_{k\geq 0}$ are observable in the real world, an important problem is to develop techniques which estimate the unobserved signal component $(X_k)_{k\geq 0}$ on the basis of an observed trajectory $y_0, y_1, \ldots$ of the observation process.

There are three elementary estimation problems. In the first problem, we observe a finite number of observations $Y_0, \ldots, Y_N$, and we wish to estimate the corresponding signal trajectory $X_1, \ldots, X_N$. To this end, we will show how to compute the conditional expectations

$$\mathbf{E}(f(X_k)|Y_0, \ldots, Y_N), \qquad 0 \leq k \leq N,$$

for any function f. This is called the *smoothing* problem. For example, one might apply this method to decode a (digital or analog) message transmitted through a noisy communication channel, or to segment a DNA strand into coding and non-coding regions on the basis of a given base pair sequence.

Still fixing the observation sequence $Y_0, \ldots, Y_N$, we sometimes wish to estimate also the future evolution of the signal

$$\mathbf{E}(f(X_k)|Y_0, \ldots, Y_N), \qquad k \geq N.$$

This is known as the *prediction* problem. For example, one might try to apply this technique to the pursuit problem, where we must decide what action to

take presently on the basis of the available observations in order to intercept a moving target at some predetermined future time.

The most common scenario is one where we wish to estimate the present value of the signal, given all available observations to date. In other words, in this case the observation sequence is not fixed, but we obtain an new observation in every time step. The computation of the conditional expectations

$$\mathbf{E}(f(X_k)|Y_0, \ldots, Y_k), \qquad k \geq 0$$

is called the *filtering* problem. This is precisely what is of interest, e.g., in the target tracking problem. In a sense, it turns out that the filtering problem is particularly fundamental: its solution is a necessary step in many of the techniques that we will discuss, including smoothing and prediction.

Our solutions of the filtering, smoothing and prediction problems will be *recursive* in nature. In particular, the solution of the filtering problem is such that the filtered estimates at time $k+1$ can be computed from the filtered estimates at time k and the new observation Y_k only. This is of course a manifestation of the Markov nature of our models, and is computationally very convenient. In certain cases—particularly when the signal state space E is a finite set—these recursions can be implemented directly as a computer algorithm. In more complicated cases this will no longer be tractable; however, we will develop an efficient and computationally tractable Monte Carlo algorithm to approximate the conditional estimates, and we will prove theorems that quantify the resulting approximation error.

Inference

In the above estimation problems, we presumed that the underlying hidden Markov model is already known. However, in many applications it is initially far from clear how to design the transition and observation kernels P and Φ and the initial measure μ. This is particularly true in applications such as financial time series models, DNA sequence segmentation and speech recognition, where the design of a hidden Markov model for which the observation process possesses the desired statistical properties is an important component of the problem. It is therefore essential to develop statistical inference techniques which allow us to design and calibrate our hidden Markov model to match observed real-world data.

It should be noted that in this setting we may not have much, if any, a priori knowledge of the structure of the unobserved process. In particular, the unobserved process can typically not be observed in real life even for modeling purposes. This distinguishes what we are trying to achieve from, e.g., supervised learning problems, where estimators are constructed on the basis of a training set in which both the observed and unobserved components are available. In our setting, the only data on which inference may be based are given time series of the observation process. (Of course, even if the structure

of the unobserved process is fairly well known, the calibration of parameter values on the basis of observed time series is often of interest).

In statistical inference problems we will typically consider a parametrized family of transition and observation kernels P^θ, Φ^θ and initial measures μ^θ, where the parameter θ takes values in some class of models $\theta \in \Theta$. Our goal is to select a suitable $\theta^\star \in \Theta$ so that the resulting observation process $(Y_k)_{k\geq 0}$ reproduces the statistical properties of a given training sequence $y_1, \ldots, y_N$. We will approach this problem through *maximum likelihood* estimation. Moreover, we will develop an iterative algorithm—the *EM algorithm*—in order to compute the maximum likelihood estimate in a tractable manner.

When the signal state space E is a finite set, the transition kernel P is a matrix and the initial measure μ is a vector. In this case it becomes feasible to estimate the entire signal model P, μ as it is defined by a finite number of parameters—there is no need to restrict to some subclass Θ (though the latter might be preferable if the cardinality of E is large). Applying the EM algorithm in this setting provides an ideal tool for speech recognition or sequence analysis problems, as no assumptions need to be imposed on the signal model except that the cardinality of E is fixed at the outset.

Even if we believe that a signal state space of finite cardinality suffices, however, it may not always be clear what cardinality to choose. For example, consider the stock price model with regime switching. The stock price dynamics might very well be excellently modeled by choosing a finite number of regimes, but it is often not clear at the outset how many regimes to choose to obtain a good model. This is known as the *model order estimation* problem, and we will develop some techniques to solve it.

Decision

Beside the design and calibration of the hidden Markov model and estimation of the unobserved signal, various applications require us to make certain decisions in order to achieve a particular objective. For example, in the stock market model we might wish to decide how to invest our capital in order to maximize our ultimate wealth; in the pursuit problem, we wish to decide how to navigate our plane in order to intercept the target; and in the change detection problem, we wish to decide when to interrupt production in order to make repairs. What all these problems have in common is that we are able to base our decisions only on the observation process Y_k, as we do not have access to the unobserved signal X_k. In the language of stochastic control, these are control problems with *partial observations*.

It turns out that the filtering problem plays a fundamental role in partially observed decision problems. By reformulating these problems in terms of the filter, we will find that they can be tackled using standard techniques from optimal control and optimal stopping theory. Alternatively, sub-optimal schemes may be much simpler to implement, particularly in complex systems, and still lead to acceptable (and even near-optimal) performance.

Problems

1.1. Finite State Markov Chains
Let E be a finite set, e.g., $E = \{1, \dots, n\}$. Measures on E and functions $f : E \to \mathbb{R}$ can be represented as n-dimensional vectors in an elementary fashion. Let $(X_k)_{k\geq 0}$ be a Markov process with state space E: such a process is called a *(finite state) Markov chain*. Show that the definitions and expressions in section 1.1 reduce to the notion of a Markov chain as you encountered it in your introduction to stochastic processes course.

1.2. Time Series
There are many standard time series models that are used in the literature. One common choice is the real-valued AR(p) model defined by the recursion

$$\tilde{X}_n = \sum_{k=1}^{p} a_k \tilde{X}_{n-k} + \varepsilon_n \qquad (n \geq p)$$

with the initial condition $\tilde{X}_0 = \cdots = \tilde{X}_{p-1} = 0$, where a_k are real-valued coefficients and ε_k are i.i.d. random variables.
(a) An AR(p) process is not Markov. Show that it can nonetheless be represented as a Markov process by enlarging the state space. (Hint: prove that the process $X_n = (\tilde{X}_n, \dots, \tilde{X}_{n+p-1})$, $n \geq 0$ is Markov.)
A different time series model, which is popular in econometric applications, is the nonlinear ARCH(p) model defined as

$$\tilde{X}_n = a_0 + \sum_{k=1}^{p} a_k \tilde{Z}_{n-k}^2, \qquad \tilde{Z}_n = \sqrt{\tilde{X}_n}\,\varepsilon_n \qquad (n \geq p)$$

where a_k are nonnegative constants and ε_k are i.i.d. random variables.
(b) Repeat part (a) for the ARCH(p) model.

1.3. DNA Sequence Alignment I ([Kro98])
DNA sequences encode genetic information in four letters A, C, G, T. DNA code is much more sloppy than human language, however, and the manner in which the same feature is encoded in different species or individuals can vary significantly. For example, the following five strings might encode the same feature: $ACAATG$, $AGAATC$, $ACACAGC$, $ACCGATC$, $TCAATGATC$. To exhibit their common pattern, let us align them (by hand) as follows:

$l1$	$l2$	$l3$	li	li	li	$l4$	$l5$	$l6$
A	C	A	$-$	$-$	$-$	A	T	G
A	G	A	$-$	$-$	$-$	A	T	C
A	C	A	C	$-$	$-$	A	G	C
A	C	C	G	$-$	$-$	A	T	C
T	C	A	A	T	G	A	T	C

Evidently the 'base' pattern $ACAATC$ varies in two ways: individual pattern symbols $l1$–$l6$ may be mutated in a fraction of the instances, and arbitrary extra symbols li may be inserted in the middle of the pattern.
(a) Model the above pattern as a hidden Markov model. Hint: as in speech recognition, use $F = \{A, C, G, T\}$ and $E = \{l1, \ldots, l6, li, le\}$ where le is the terminal state $P(l6, \{le\}) = P(le, \{le\}) = 1$. You may assume that $\Phi(le, \{y\}) = 1/4$ for all $y \in F$, i.e., the pattern is followed a random sequence of symbols. Read off the remaining probabilities $P(x, \{x'\})$ and $\Phi(x, \{y\})$.
(b) Suppose we are given a sequence $y_0, \ldots, y_k$ of symbols ($y_i \in F$). Write a computer program that computes $\mathbf{P}(Y_0 = y_0, \ldots, Y_k = y_k)$.
(c) Given a symbol sequence $y_0, \ldots, y_k$ that is not in our training set, we can use your program from part (b) to determine whether or not the string likely matches the pattern. To this end, we will 'score' a sequence $y_0, \ldots, y_k$ by computing the relative likelihood that it comes from our hidden Markov model versus a random sequence of symbols:

$$\text{score}(y_0, \ldots, y_k) = \frac{\mathbf{P}(Y_0 = y_0, \ldots, Y_k = y_k)}{(1/4)^{k+1}}.$$

Compute the scores of each of our training sequences and experiment with various mutations and insertions in the 'base' sequence. Also try some strings which are very unlike the 'base' sequence.
(d) A high score (at least > 1) in the previous part indicates that the string matches our pattern. Adapt your computer program to compute also

$$(\hat{x}_0, \ldots, \hat{x}_k) = \operatorname*{argmax}_{x_0, \ldots, x_k \in E} \mathbf{P}(X_0 = x_0, \ldots, X_k = x_k, Y_0 = y_0, \ldots, Y_k = y_k).$$

Experiment with the training sequences and with various mutations and insertions in the 'base' sequence, and show that your program allows us to automate the sequence alignment procedure which we previously did by hand (i.e., inserting the right number of dashes in the table above).

Remark 1.18. The DNA pattern in the previous problem is exceedingly simple. In realistic sequence alignment problems, both the 'base' pattern and the inserted 'junk' regions are typically much longer, and the naive computation of the relevant quantities becomes computationally expensive. In chapter 3, we will develop *recursive* algorithms which allow us to compute these quantities in a very efficient manner, even for very long sequences.

1.4. Fix signal and observation state spaces E and F, let P and P' be two transition kernels and let μ and μ' be two initial measures on E. Let Φ be an observation kernel which satisfies the nondegeneracy assumption (definition 1.9). Prove that a hidden Markov model with initial law μ, transition kernel P and observation kernel Φ on the one hand, and a hidden Markov model with initial law μ', transition kernel P' and observation kernel Φ on the other hand, give rise to observations $(Y_k)_{k \leq n}$ whose laws are absolutely continuous. (Beware: in general, the claim is only true on a finite horizon $n < \infty$.)

Notes

This course presumes an elementary knowledge of (measure-theoretic) probability theory. There are very many excellent textbooks on probability. We will on occasion refer to the wonderful reference book of Kallenberg [Kal02] or to the textbook by Shiryaev [Shi96] for basic probabilistic facts.

An excellent text on Markov chains in general state spaces is Revuz [Rev75]. The more recent text of Meyn and Tweedie [MT93], which emphasizes various notions of geometric ergodicity and coupling (see chapter 5), is often cited. An well known introductory text at the undergraduate level (mostly in a finite state space) is Norris [Nor98].

The theory of hidden Markov models is treated in detail in the recent monograph by Cappé, Moulines and Rydén [CMR05], while Ephraim and Merhav [EM02] have written a well known review of the subject with many references to the literature. Many of the topics that we will encounter in this course can be found in these references in much greater detail. Elliott, Aggoun and Moore [EAM95] has a more control-theoretic flavor.

A large number of applications of hidden Markov models can be found in the literature. The following is by no means a comprehensive list of references; it can only serve as an entry point. A *Google Scholar* search will reveal many more applications in your favorite area of interest.

Some of the earliest and most successful applications are in the field of speech and handwriting recognition; the tutorial paper by Rabiner [Rab89] has been very influential in popularizing these ideas. Some applications to communication and information theory are reviewed in Ephraim and Merhav [EM02] and in Kailath and Poor [KP98]. Applications to navigation and tracking are very old, see, e.g., the book by Bucy and Joseph [BJ87]. More recent tracking applications include navigation by GPS [CDMS97]; see also Bar-Shalom et al. [BLK01]. Optimal changepoint detection and sequential hypothesis testing are developed by Shiryaev [Shi73], while a general text on changepoint detection and applications is Basseville and Nikiforov [BN93]. Applications in bioinformatics are described in the book by Koski [Kos01]. Various statistical applications are described in MacDonald and Zucchini [MZ97]. Applications to financial economics are described in Bhar and Hamori [BH04]. Some applications to mathematical finance can be found in the collection [ME07] and in [She02, SH04]. Note that financial models are often in continuous time; hidden Markov models in continuous time is the topic of chapters 10–13.

2

Filtering, Smoothing, Prediction

2.1 Conditional Distributions

The purpose of this chapter is to solve (at least in principle) the filtering, smoothing and prediction problems introduced in section 1.4: given a hidden Markov model $(X_k, Y_k)_{k\geq 0}$, we are interested in computing conditional expectations of the form $\mathbf{E}(f(X_n)|Y_0, \ldots, Y_k)$ for all functions f. In other words, we are interested in computing the conditional distributions

$$\mathbf{P}(X_n \in \cdot\,|Y_0, \ldots, Y_k).$$

Before we turn to this problem in the setting of hidden Markov models, we recall in this section how conditional distributions may be computed in a general setting. First, however, we briefly discuss the following question: in what sense can the conditional distribution be thought of as an estimator?

Conditional distributions and estimation

Let X be a real-valued random variable and let Y be a B-valued random variable on some probability space $(\Omega, \mathcal{G}, \mathbf{P})$ and state space $(B, \mathcal{B})$. We suppose that we can observe Y but not X, and we would like to estimate X. In our hidden Markov model, we could choose, e.g., $X = f(X_n)$ for some $n \geq 0$ and $f : E \to \mathbb{R}$, and $Y = (Y_0, \ldots, Y_k)$ for some $k \geq 0$.

What does it mean to estimate a random variable X? What we seek is a function $g(Y)$ of the observed variables only, such that $g(Y)$ is close to X in a certain sense. For example, we can try to find such a function g that minimizes the mean square estimation error $\mathbf{E}((X - g'(Y))^2)$. As it turns out, this is precisely the conditional expectation.

Lemma 2.1. *Suppose that* $\mathbf{E}(X^2) < \infty$. *Then* $g(Y) = \mathbf{E}(X|Y)$ *satisfies*

$$g = \operatorname*{argmin}_{g'} \mathbf{E}((X - g'(Y))^2).$$

Proof. By construction $\mathbf{E}(X|Y)$ is a function of Y, and $\mathbf{E}((X - \mathbf{E}(X|Y))^2) \leq 2\,\mathbf{E}(X^2) < \infty$. It remains to prove that for any other function $g'(Y)$ we have

$$\mathbf{E}((X - \mathbf{E}(X|Y))^2) \leq \mathbf{E}((X - g'(Y))^2).$$

Let us write $G = \mathbf{E}(X|Y)$ and $G' = g'(Y)$. Note that

$$\begin{aligned}
\mathbf{E}((X-G)^2) &= \mathbf{E}((X - G' + G' - G)^2) \\
&= \mathbf{E}((X-G')^2) + \mathbf{E}((G'-G)^2) + 2\,\mathbf{E}((X-G')(G'-G)) \\
&= \mathbf{E}((X-G')^2) + \mathbf{E}((G'-G)^2) + 2\,\mathbf{E}(\mathbf{E}((X-G')(G'-G)|Y)) \\
&= \mathbf{E}((X-G')^2) - \mathbf{E}((G'-G)^2) \\
&\leq \mathbf{E}((X-G')^2).
\end{aligned}$$

The proof is complete. □

By computing the conditional expectation, we therefore find the *least mean square estimate* of the unobserved variable X given the observed variable Y.

However, what if we are interested in finding an estimator with a different error criterion? For example, we might wish to minimize $\mathbf{E}(|X - g'(Y)|)$ or, more generally, $\mathbf{E}(H(X - g'(Y)))$ for some some loss function H. To tackle this problem, we need the notion of a conditional distribution.

Definition 2.2. *Let X be an $(E, \mathcal{E})$-valued random variable and let Y be a $(B, \mathcal{B})$-valued random variable on a probability space $(\Omega, \mathcal{G}, \mathbf{P})$. A transition kernel $P_{X|Y} : B \times \mathcal{E} \to [0, 1]$ which satisfies*

$$\int f(x)\, P_{X|Y}(Y, dx) = \mathbf{E}(f(X)|Y)$$

for every bounded measurable function $f : E \to \mathbb{R}$ is called the conditional distribution *(or* regular conditional probability*) of X given Y.*

This idea is likely familiar: intuitively $P_{X|Y}(y, A) = \mathbf{P}(X \in A | Y = y)$.

Remark 2.3. Existence and uniqueness of conditional distributions is guaranteed under the mild technical condition that E is a Borel space, as is the case in all our examples [Kal02, theorem 6.3]. We will shortly see, however, that the nondegeneracy assumption allows us to construct the conditional distributions explicitly. We therefore will not need this general fact.

Returning to our estimation problem, we now claim that we can solve the optimal estimation problem of minimizing $\mathbf{E}(H(X - g'(Y)))$ for some some loss function H in two steps. First, we compute the conditional distribution $P_{X|Y}$. The optimal estimate $g(y)$ is then obtained simply by minimizing the expected loss with respect to the conditional distribution $P_{X|Y}(y, \cdot)$.

Lemma 2.4. *Let $H : \mathbb{R} \to [0, \infty[$ be a given loss function, X be a real-valued random variable with $\mathbf{E}(H(X)) < \infty$, and Y be a $(B, \mathcal{B})$-valued random variable. Suppose there is a measurable function $g : B \to \mathbb{R}$ such that*

$$g(y) = \operatorname*{argmin}_{\hat{x} \in \mathbb{R}} \int H(x - \hat{x})\, P_{X|Y}(y, dx) \quad \text{for all } y \in B',$$

where $B' \in \mathcal{B}$ satisfies $\mathbf{P}(Y \in B') = 1$. Then g minimizes $\mathbf{E}(H(X - g'(Y)))$.

Proof. Note that by construction

$$\int H(x - g(Y))\, P_{X|Y}(Y, dx) \leq \int H(x - g'(Y))\, P_{X|Y}(Y, dx) \quad \text{a.s.}$$

for any measurable function g'. Therefore

$$\begin{aligned} \mathbf{E}(H(X - g(Y))) &= \mathbf{E}\left[\int H(x - g(Y))\, P_{X|Y}(Y, dx)\right] \\ &\leq \mathbf{E}\left[\int H(x - g'(Y))\, P_{X|Y}(Y, dx)\right] = \mathbf{E}(H(X - g'(Y))). \end{aligned}$$

Setting $g' = 0$, we find that $\mathbf{E}(H(X - g(Y))) \leq \mathbf{E}(H(X)) < \infty$. Therefore g does indeed minimize $\mathbf{E}(H(X - g'(Y)))$, and the proof is complete. □

If the loss function H is convex this approach is always successful. A nice discussion along these lines and many further details can be found in [BH85].

Example 2.5. For the square loss $H(x) = x^2$, we have already seen that the best estimator of X given Y is the *conditional mean* $\mathrm{mean}(P_{X|Y}) = \mathbf{E}(X|Y)$. By lemma 2.4, the best estimator for the deviation loss $H(x) = |x|$ is the *conditional median* $\mathrm{med}(P_{X|Y})$ (note that the latter need not be unique).

Example 2.6. Suppose that the random variable X takes a finite number of values $\{x_1, \ldots, x_n\}$, and choose the loss function

$$H(x) = \begin{cases} 0 & x = 0, \\ 1 & x \neq 0. \end{cases}$$

In other words, we wish to choose an estimator g in order to maximize the probability $\mathbf{P}(X = g'(Y))$. Then by lemma 2.4 we should choose

$$g(y) = x_i \qquad \text{whenever} \qquad \mathbf{P}_{X|Y}(y, X = x_i) = \max_{j=1,\ldots,n} \mathbf{P}_{X|Y}(y, X = x_j).$$

This is called the *maximum a posteriori (MAP) estimate* of X given Y.

To conclude, we have seen that once the conditional distribution of X given Y has been computed, the solution of the optimal estimation problem for any loss function H reduces to a deterministic minimization problem. We can therefore restrict our attention without any loss of generality to the computation of the conditional distribution $P_{X|Y}$.

The Bayes formula

Given two random variables X and Y, how does one compute the conditional distribution $P_{X|Y}$? This turns out to be particularly straightforward if the law of Y is nondegenerate (compare with definition 1.9). The following result is one of the many forms of the *Bayes formula*.

Theorem 2.7 (Bayes formula). *Let X be an $(E, \mathcal{E})$-valued random variable and let Y be a $(B, \mathcal{B})$-valued random variable on a probability space $(\Omega, \mathcal{G}, \mathbf{P})$. Suppose that there exists a measurable function $\gamma : E \times B \to]0, \infty[$, a probability measure μ_X on E, and a probability measure μ_Y on B, such that*

$$\mathbf{E}(f(X,Y)) = \int f(x,y)\,\gamma(x,y)\,\mu_X(dx)\,\mu_Y(dy)$$

for every bounded measurable function f. Then

$$P_{X|Y}(y,A) = \frac{\int I_A(x)\,\gamma(x,y)\,\mu_X(dx)}{\int \gamma(x,y)\,\mu_X(dx)} \quad \text{for all } A \in \mathcal{E},\ y \in B$$

is the conditional distribution of X given Y.

Proof. By definition 2.2, we need to verify that for every $A \in \mathcal{E}$ we have $P_{X|Y}(Y,A) = \mathbf{P}(X \in A|Y)$. Equivalently, using the definition of the conditional expectation, we need to verify that we have $\mathbf{E}(P_{X|Y}(Y,A)\,I_C(Y)) = \mathbf{E}(I_A(X)\,I_C(Y))$ for every $A \in \mathcal{E}$ and $C \in \mathcal{B}$. But note that

$$\begin{aligned}
\mathbf{E}(P_{X|Y}(Y,A)\,I_C(Y)) &= \mathbf{E}\left[\frac{\int I_A(x')\,I_C(Y)\,\gamma(x',Y)\,\mu_X(dx')}{\int \gamma(x',Y)\,\mu_X(dx')}\right] \\
&= \int \frac{\int I_A(x')\,I_C(y)\,\gamma(x',y)\,\mu_X(dx')}{\int \gamma(x',y)\,\mu_X(dx')}\,\gamma(x,y)\,\mu_X(dx)\,\mu_Y(dy) \\
&= \int I_A(x')\,I_C(y)\,\gamma(x',y)\,\mu_X(dx')\,\mu_Y(dy) \\
&= \mathbf{E}(I_A(X)\,I_C(Y)).
\end{aligned}$$

The proof is complete. □

2.2 Filtering, Smoothing, and Prediction Recursions

Throughout this section, let $(X_k, Y_k)_{k\geq 0}$ be a hidden Markov model with signal state space $(E, \mathcal{E})$, observation state space $(F, \mathcal{F})$, transition kernel P, observation kernel Φ, and initial measure μ (definition 1.6). We also presume that the observations are nondegenerate, i.e., that Φ possesses an observation density Υ with respect to a reference measure φ (definition 1.9).

Our goal is to compute the conditional distributions

$$\pi_{k|n} = P_{X_k|Y_0,\ldots,Y_n}, \qquad k,n \geq 0.$$

We distinguish between three cases. The goal of the *filtering* problem is to compute $\pi_{k|k}$ for $k \geq 0$; for notational simplicity, we define the *filtering distributions* $\pi_k = \pi_{k|k}$. Similarly, the goal of the *smoothing* problem is to compute the *smoothing distributions* $\pi_{k|n}$ for $k < n$, while the goal of the *prediction* problem is to compute *prediction distributions* $\pi_{k|n}$ for $k > n$. As we will see, a key feature of our computations is that they can be performed *recursively.*

Filtering

Using lemma 1.8, we easily find the finite dimensional distributions

$$\begin{aligned}\mathbf{E}(f(X_0,Y_0,\ldots,X_k,Y_k)) = \int f(x_0,y_0,\ldots,x_k,y_k)\,\Upsilon(x_0,y_0)\cdots\Upsilon(x_k,y_k)\\ \times\,\varphi(dy_0)\cdots\varphi(dy_k)\,P(x_{k-1},dx_k)\cdots P(x_0,dx_1)\,\mu(dx_0)\end{aligned}$$

of our hidden Markov model. To compute the filtering distributions, we will combine this expression with the Bayes formula.

Definition 2.8. *For every time $k \geq 0$, the* unnormalized filtering distribution *σ_k is the kernel $\sigma_k : F^{k+1} \times \mathcal{E} \to \mathbb{R}_+$ defined as*

$$\begin{aligned}&\sigma_k(y_0,\ldots,y_k,A) =\\ &\qquad\int I_A(x_k)\,\Upsilon(x_0,y_0)\cdots\Upsilon(x_k,y_k)\,P(x_{k-1},dx_k)\cdots P(x_0,dx_1)\,\mu(dx_0)\end{aligned}$$

for all $y_0,\ldots,y_k \in F$ and $A \in \mathcal{E}$.

Note that the kernel σ_k is not necessarily a transition kernel, i.e., it is typically the case that $\sigma_k(y_0,\ldots,y_k,E) \neq 1$. However, its normalization coincides precisely with the filtering distribution π_k.

Theorem 2.9 (Unnormalized filtering recursion). *The filtering distribution π_k can be computed as*

$$\pi_k(y_0,\ldots,y_k,A) = \frac{\sigma_k(y_0,\ldots,y_k,A)}{\sigma_k(y_0,\ldots,y_k,E)}$$

for every $A \in \mathcal{E}$ and $y_0,\ldots,y_k \in F$. Moreover, the unnormalized filtering distributions σ_k can be computed recursively according to

$$\sigma_k(y_0,\ldots,y_k,A) = \int I_A(x)\,\Upsilon(x,y_k)\,P(x',dx)\,\sigma_{k-1}(y_0,\ldots,y_{k-1},dx')$$

with the initial condition

$$\sigma_0(y_0,A) = \int I_A(x)\,\Upsilon(x,y_0)\,\mu(dx).$$

Proof. Define the probability measure μ_Y on F^{k+1} as the product measure

$$\mu_Y(dy_0, \ldots, dy_k) = \varphi(dy_0) \cdots \varphi(dy_k).$$

Similarly, we define the probability measure μ_X on E^{k+1} as

$$\mu_X(dx_0, \ldots, dx_k) = P(x_{k-1}, dx_k) \cdots P(x_0, dx_1)\, \mu(dx_0),$$

and we define the function

$$\gamma(x_0, \ldots, x_k, y_0, \ldots, y_k) = \Upsilon(x_0, y_0) \cdots \Upsilon(x_k, y_k).$$

Then by the Bayes formula (theorem 2.7), we have

$$\int f(x_0, \ldots, x_k)\, P_{X_0,\ldots,X_k|Y_0,\ldots,Y_k}(y_0, \ldots, y_k, dx_0, \ldots, dx_k) = \frac{\int f(x_0, \ldots, x_k)\, \gamma(x_0, \ldots, x_k, y_0, \ldots, y_k)\, \mu_X(dx_0, \ldots, dx_k)}{\int \gamma(x_0, \ldots, x_k, y_0, \ldots, y_k)\, \mu_X(dx_0, \ldots, dx_k)}.$$

Therefore, the first statement follows from the fact that

$$\begin{aligned}
\int f(x)\, \pi_k(y_0, \ldots, y_k, dx) &= \int f(x_k)\, P_{X_k|Y_0,\ldots,Y_k}(y_0, \ldots, y_k, dx_k) \\
&= \int f(x_k)\, P_{X_0,\ldots,X_k|Y_0,\ldots,Y_k}(y_0, \ldots, y_k, dx_0, \ldots, dx_k) \\
&= \frac{\int f(x_k)\, \gamma(x_0, \ldots, x_k, y_0, \ldots, y_k)\, \mu_X(dx_0, \ldots, dx_k)}{\int \gamma(x_0, \ldots, x_k, y_0, \ldots, y_k)\, \mu_X(dx_0, \ldots, dx_k)} \\
&= \frac{\int f(x_k)\, \sigma_k(y_0, \ldots, y_k, dx_k)}{\int \sigma_k(y_0, \ldots, y_k, dx_k)}.
\end{aligned}$$

The recursion for σ_k is easily verified by inspection. □

Rather than computing σ_k recursively, and subsequently normalizing to obtain π_k, we may compute the filtering distributions π_k directly.

Corollary 2.10 (Filtering recursion). *The filtering distributions π_k can be computed recursively according to*

$$\pi_k(y_0, \ldots, y_k, A) = \frac{\int I_A(x)\, \Upsilon(x, y_k)\, P(x', dx)\, \pi_{k-1}(y_0, \ldots, y_{k-1}, dx')}{\int \Upsilon(x, y_k)\, P(x', dx)\, \pi_{k-1}(y_0, \ldots, y_{k-1}, dx')}$$

with the initial condition

$$\pi_0(y_0, A) = \frac{\int I_A(x)\, \Upsilon(x, y_0)\, \mu(dx)}{\int \Upsilon(x, y_0)\, \mu(dx)}.$$

Proof. This follows immediately from the previous theorem. □

The recursive nature of the filtering problem is computationally very convenient: to compute the filtered estimate π_k, we only need to know the filtered estimate in the previous time step π_{k-1} and the new observation y_k obtained in the present time step. In particular, we do not need to remember the entire observation history $y_0, \ldots, y_{k-1}$ as long as we are interested in the filter only.

Smoothing

To find the smoothing distributions $\pi_{k|n}$ $(k < n)$, we once again appeal to the Bayes formula. We will see that the computation splits into two parts: the observations $Y_0, \ldots, Y_k$ and $Y_{k+1}, \ldots, Y_n$ enter the problem in a different way.

Definition 2.11. *For every $0 \leq k < n$, the* unnormalized smoothing density $\beta_{k|n}$ *is the function $\beta_{k|n} : E \times F^{n-k} \to]0, \infty[$ defined as*

$$\beta_{k|n}(x_k, y_{k+1}, \ldots, y_n) = \int \Upsilon(x_{k+1}, y_{k+1}) \cdots \Upsilon(x_n, y_n)\, P(x_{n-1}, dx_n) \cdots P(x_k, dx_{k+1})$$

for all $y_{k+1}, \ldots, y_n \in F$ and $x_k \in E$.

The Bayes formula allows us to prove the following.

Theorem 2.12 (Unnormalized smoothing recursion). *The smoothing distribution $\pi_{k|n}$ $(k < n)$ can be computed as*

$$\pi_{k|n}(y_0, \ldots, y_n, A) = \frac{\int I_A(x)\, \beta_{k|n}(x, y_{k+1}, \ldots, y_n)\, \sigma_k(y_0, \ldots, y_k, dx)}{\int \beta_{k|n}(x, y_{k+1}, \ldots, y_n)\, \sigma_k(y_0, \ldots, y_k, dx)}$$

for every $A \in \mathcal{E}$ and $y_0, \ldots, y_n \in F$. Moreover, the unnormalized smoothing densities $\beta_{k|n}$ can be computed by the backward recursion

$$\beta_{k|n}(x, y_{k+1}, \ldots, y_n) = \int \beta_{k+1|n}(x', y_{k+2}, \ldots, y_n)\, \Upsilon(x', y_{k+1})\, P(x, dx')$$

with the terminal condition $\beta_{n|n} = 1$.

Proof. Using the same notation as in the proof of theorem 2.9

$$\begin{aligned}
\int f(x)\, \pi_{k|n}(y_0, \ldots, y_n, dx) &= \int f(x_k)\, P_{X_k|Y_0,\ldots,Y_n}(y_0, \ldots, y_n, dx_k) \\
&= \int f(x_k)\, P_{X_0,\ldots,X_n|Y_0,\ldots,Y_n}(y_0, \ldots, y_n, dx_0, \ldots, dx_n) \\
&= \frac{\int f(x_k)\, \gamma(x_0, \ldots, x_n, y_0, \ldots, y_n)\, \mu_X(dx_0, \ldots, dx_n)}{\int \gamma(x_0, \ldots, x_n, y_0, \ldots, y_n)\, \mu_X(dx_0, \ldots, dx_n)} \\
&= \frac{\int f(x_k)\, \beta_{k|n}(x_k, y_{k+1}, \ldots, y_n)\, \sigma_k(y_0, \ldots, y_k, dx_k)}{\int \beta_{k|n}(x_k, y_{k+1}, \ldots, y_n)\, \sigma_k(y_0, \ldots, y_k, dx_k)}.
\end{aligned}$$

The recursion for $\beta_{k|n}$ is easily verified by inspection. □

As in the filtering problem, we can also obtain a normalized version of the backward smoothing recursion. This is sometimes computationally more conveninent. Note, however, that the filtering distributions appear in the normalized smoothing recursion: in order to use it, we must first make a *forward* (in time) pass through the observation data to compute the filtering distributions, and then a *backward* pass to compute the smoothing densities. This is sometimes called the *forward-backward algorithm*.

Corollary 2.13 (Smoothing recursion). *Define for $k < n$ the function $\bar{\beta}_{k|n} : E \times F^{n+1} \to]0, \infty[$ through the backward recursion*

$$\bar{\beta}_{k|n}(x, y_0, \ldots, y_n) = \frac{\int \bar{\beta}_{k+1|n}(x', y_0, \ldots, y_n)\, \Upsilon(x', y_{k+1})\, P(x, dx')}{\int \Upsilon(x', y_{k+1})\, P(x, dx')\, \pi_k(y_0, \ldots, y_k, dx)}$$

with terminal condition $\bar{\beta}_{n|n} = 1$. Then for any $k < n$

$$\pi_{k|n}(y_0, \ldots, y_n, A) = \int I_A(x)\, \bar{\beta}_{k|n}(x, y_0, \ldots, y_n)\, \pi_k(y_0, \ldots, y_k, dx)$$

for every $A \in \mathcal{E}$ and $y_0, \ldots, y_n \in F$.

Proof. From the unnormalized smoothing recursion, we can read off that

$$\bar{\beta}_{k|n}(x, y_0, \ldots, y_n) = \frac{\int \bar{\beta}_{k+1|n}(x', y_0, \ldots, y_n)\, \Upsilon(x', y_{k+1})\, P(x, dx')}{\int \bar{\beta}_{k+1|n}(x', y_0, \ldots, y_n)\, \Upsilon(x', y_{k+1})\, P(x, dx')\, \pi_k(y_0, \ldots, y_k, dx)}$$

with $\bar{\beta}_{n|n} = 1$. It therefore suffices to prove that for $k < n$

$$\int \bar{\beta}_{k+1|n}(x', y_0, \ldots, y_n)\, \Upsilon(x', y_{k+1})\, P(x, dx')\, \pi_k(y_0, \ldots, y_k, dx) = \int \Upsilon(x', y_{k+1})\, P(x, dx')\, \pi_k(y_0, \ldots, y_k, dx).$$

But using the normalized filtering recursion (corollary 2.10), we find

$$\frac{\int \bar{\beta}_{k+1|n}(x', y_0, \ldots, y_n)\, \Upsilon(x', y_{k+1})\, P(x, dx')\, \pi_k(y_0, \ldots, y_k, dx)}{\int \Upsilon(x', y_{k+1})\, P(x, dx')\, \pi_k(y_0, \ldots, y_k, dx)} = \int \bar{\beta}_{k+1|n}(x', y_0, \ldots, y_n)\, \pi_{k+1}(y_0, \ldots, y_{k+1}, dx') = 1$$

by construction. This completes the proof. □

Prediction

Prediction, i.e., the computation of $\pi_{k|n}$ for $k > n$, is the simplest of our estimation problems. The following theorem can be proved using the Bayes formula, but a direct proof is simple and illuminating.

Theorem 2.14 (Prediction recursion). *The prediction distribution $\pi_{k|n}$ $(k > n)$ can be computed recursively as*

$$\pi_{k|n}(y_0, \ldots, y_n, A) = \int I_A(x)\, P(x', dx)\, \pi_{k-1|n}(y_0, \ldots, y_n, dx')$$

for every $A \in \mathcal{E}$ and $y_0, \ldots, y_n \in F$, with the initial condition $\pi_{n|n} = \pi_n$.

Proof. By the tower property of the conditional expectation, we have

$$\mathbf{E}(f(X_k)|Y_0,\ldots,Y_n) = \mathbf{E}(\mathbf{E}(f(X_k)|X_0,Y_0,\ldots,X_n,Y_n)|Y_0,\ldots,Y_n)$$

for $k > n$. But using the Markov property of the signal, we have

$$\mathbf{E}(f(X_k)|X_0,Y_0,\ldots,X_n,Y_n) = P^{k-n}f(X_n).$$

Therefore $\mathbf{E}(f(X_k)|Y_0,\ldots,Y_n) = \mathbf{E}(P^{k-n}f(X_n)|Y_0,\ldots,Y_n)$ or, equivalently,

$$\int f(x)\,\pi_{k|n}(y_0,\ldots,y_n,dx) = \int P^{k-n}f(x)\,\pi_n(y_0,\ldots,y_n,dx).$$

for every bounded measurable function f. The recursion for $\pi_{k|n}$ can now be read off directly from this expression. □

We now make a simple observation: by corollary 2.10, the filter π_{k+1} can be naturally expressed in terms of the one step predictor $\pi_{k+1|k}$:

$$\pi_{k+1}(y_0,\ldots,y_{k+1},A) = \frac{\int I_A(x)\,\Upsilon(x,y_{k+1})\,\pi_{k+1|k}(y_0,\ldots,y_k,dx)}{\int \Upsilon(x,y_{k+1})\,\pi_{k+1|k}(y_0,\ldots,y_k,dx)}.$$

The filter recursion is therefore frequently interpreted as a two step procedure:

$$\pi_k \xrightarrow{\text{prediction}} \pi_{k+1|k} \xrightarrow{\text{correction}} \pi_{k+1}.$$

We will see this idea again in the chapter 4.

2.3 Implementation

In principle, the filtering, smoothing and prediction recursions obtained in the previous section provide a complete solution to these problems. However, in practice, these results may not be of immediate use. Indeed, these are recursions for probability measures and functions on the signal state space E: such objects are typically infinite dimensional, in which case one can not in general perform these computations on a computer without further approximation. The question then becomes how to apply these mathematical techniques, either exactly or approximately, to real-world problems.

Considering first the problem of approximate implementation, one might try the standard numerical technique of approximating continuous objects by their values on a discrete grid. Though this approach is sometimes successful in low dimensional problems, it suffers from the same problem that was famously formulated by Bellman many decades ago: the *curse of dimensionality*. The problem is that in the signal state space $E = \mathbb{R}^p$, the computational complexity of a grid method that achieves a fixed approximation error is typically of order $e^{\beta p}$ for some $\beta > 0$, i.e., the computational complexity of the

algorithm grows very rapidly with the state space dimension. Such techniques are therefore typically intractable in dimensions higher than $p = 2$ or 3. A more detailed analysis of this phenomenon can be found in remark 4.6. To mitigate the problem, we will develop in chapter 4 an approximate filtering algorithm which uses *random sampling* rather than gridding to discretize the problem. This technique is flexible and easily implemented, and it manages to avoid many (but not all) the problems of grid based algorithms.

Particularly in complex models, approximate implementation of the filter is the best one can hope for. However, there are two cases where the recursions obtained in this chapter can be implemented exactly.

The first is the case where the signal state space is a *finite set*, say $E = \{1, \dots, n\}$, so that measures and functions on E can be represented as n-dimensional vectors (problem 1.1). This means that the recursions obtained in this chapter can be expressed in terms matrix multiplication, which is easily implemented exactly as a computer algorithm. Though this setting is a special case of our general theory, it plays a particularly important role in applications: on the one hand there are may applications which can reasonably be modeled on a finite signal state space (see, e.g., the examples in section 1.3); on the other hand, the estimation theory for this class of models can be implemented exactly as a computer algorithm, which leads to tractable and powerful techniques that can be applied successfully to real data. We will develop this special setting in detail in chapter 3, including several new techniques that are of specific interest in a finite state space.

The other special case where exact computation is possible is the class of *linear Gaussian state space models* where $E = \mathbb{R}^p$, $F = \mathbb{R}^q$, and

$$X_k = a + AX_{k-1} + B\xi_k, \qquad Y_k = c + CX_k + D\eta_k.$$

We must assume, moreover, that ξ_k, $k \geq 1$ are i.i.d. $N(0, \mathrm{Id}_p)$, η_k, $k \geq 0$ are i.i.d. $N(0, \mathrm{Id}_q)$, and $X_0 \sim N(\mu_0, P_0)$. As the signal state space is continuous, the filtering, smoothing and prediction recursions will in fact be infinite dimensional. However, what happens in this special case is that as all the noise is Gaussian and all the operations are linear, every conditional distribution in this model is also Gaussian (problem 2.5). But the family of Gaussian distributions on $\mathbb{R}^p$ is a finite dimensional subset of the space of all probability measures on $\mathbb{R}^p$: a Gaussian distribution is completely characterized by its mean vector and covariance matrix. Therefore the filtering, smoothing and prediction recursions are really finite dimensional recursions in disguise, which can again be implemented efficiently as a computer algorithm. For the filtering problem, this leads to the famous *Kalman filter*.

Linear Gaussian models are ubiquitous in the engineering literature, at least partly due to their tractability. They exhibit rather special structure and properties, however, and the techniques which are introduced for general hidden Markov models are not always the best or most natural methods to deal with linear systems (this is in contrast to the theory for finite state models, which bears much resemblance to the general theory of hidden Markov models

and provides a host of excellent examples for the latter). For this reason, though they will make an occasional appearance, we will not spend much time on linear systems in this course. Of course, many of the techniques which will be discussed in this course can be applied to linear systems; for example, problem 2.5 below asks you to derive the Kalman filtering recursion from the general theory in the previous section. For a thorough introduction to linear estimation theory we refer, however, to the textbook [KSH00].

Remark 2.15. In the linear Gaussian case, what evidently happens is that the infinite dimensional recursions have finite dimensional invariant sets, so that the recursion can be represented in finite dimensional form. One might wonder whether there are other filtering models which have this highly desirable property. Unfortunately, it turns out that linear Gaussian models are rather special in this regard: typically finite dimensional invariant sets do not exist [Saw81]. Though one can construct examples of nonlinear filtering problems which have a finite-dimensional realization, these are almost always ad-hoc and appear rarely if ever in applications. In nonlinear continuous models, exact computations are therefore essentially always hopeless.

However, if the nonlinear model is linear to good approximation, then applying techniques for linear systems can be successful in practice. A common ad-hoc approach in engineering is to linearize nonlinear dynamics so that the Kalman filter can be applied locally; this is known as the *extended Kalman filter*. Unfortunately, the performance of this method is often poor, and it is very difficult to prove anything about it (but see [Pic91]). In any case, as we are interested in general hidden Markov models, such methods are out of place in this course and we will not go any further in this direction.

Problems

2.1. Best Linear Estimate
Let X, Y be real-valued random variables with finite mean and variance. Recall that the conditional expectation $\mathbf{E}(X|Y)$ is the optimal least squares estimate.
(a) Suppose that are are only interested in *linear* estimates, i.e., we seek an estimate of X of the form $\hat{X} = aY + b$ for some (non-random) constants $a, b \in \mathbb{R}$. Assume that Y has nonzero variance $\mathrm{var}(Y) > 0$. Show that

$$\hat{X} = \mathbf{E}(X) + \frac{\mathrm{cov}(X,Y)}{\mathrm{var}(Y)}\,(Y - \mathbf{E}(Y))$$

minimizes $\mathbf{E}((X - \hat{X})^2)$ over the class of all linear estimates. $\hat{X}$ is called the *best linear estimate* of X given Y.
(b) Provide an example where $\mathbf{E}((X - \mathbf{E}(X|Y))^2) < \mathbf{E}((X - \hat{X})^2)$. Evidently nonlinear estimates do indeed (typically) perform better than linear estimates.

2.2. Prove that the quantity

$$\int \beta_{k|n}(x, y_{k+1}, \ldots, y_n)\, \sigma_k(y_0, \ldots, y_k, dx),$$

which appears in the denominator of the expression in theorem 2.12, does not depend on k (and therefore equals $\sigma_n(y_0, \ldots, y_n, E)$).

2.3. Delayed Observations
Suppose that the observations Y_k are defined with one time step delay: $Y_0 = 0$ and $Y_k = H(X_{k-1}, \eta_k)$ for $k \geq 1$. The resulting model is strictly speaking not a hidden Markov model in the sense of chapter 1 (where $Y_k = H(X_k, \eta_k)$), but the resulting theory is almost identical. Modify the filtering, smoothing and prediction recursions developed in this chapter to this setting.

2.4. Path Estimation
In this problem, we investigate the conditional distribution $P_{X_0,\ldots,X_n|Y_0,\ldots,Y_n}$ of the entire signal path $X_0, \ldots, X_n$ given the observations $Y_0, \ldots, Y_n$.
(a) Show that the signal $(X_k)_{0\leq k\leq n}$ is a *nonhomogeneous* Markov process under the conditional distribution $P_{X_0,\ldots,X_n|Y_0,\ldots,Y_n}(y_0, \ldots, y_n, \cdot\,)$.
(b) The initial measure is obviously $\pi_{0|n}$. Give an explicit expression for the transition kernels of this nonhomogeneous Markov process using theorem 2.12.

2.5. Linear Gaussian Models
If the signal state space E is not finite, the filtering recursion can typically not be computed in a finite dimensional form. One of the very few exceptions is the linear Gaussian case. In this setting $E = \mathbb{R}^p$, $F = \mathbb{R}^q$, and

$$X_k = a + AX_{k-1} + B\xi_k, \qquad Y_k = c + CX_k + D\eta_k,$$

where A and B are $p \times p$ matrices, C is a $q \times p$ matrix, D is a $q \times q$ matrix, and $a \in \mathbb{R}^p$, $c \in \mathbb{R}^q$. Moreover, we assume that ξ_k, $k \geq 1$ are i.i.d. $N(0, \mathrm{Id}_p)$, that η_k, $k \geq 0$ are i.i.d. $N(0, \mathrm{Id}_q)$, and that $X_0 \sim N(\mu_0, P_0)$. In order to ensure the nondegeneracy assumption, we will assume that D is invertible.
(a) Show that the conditional distributions $\pi_{n|k}$ are Gaussian for every n, k.
(b) Denote by $\hat{X}_k$ and $\hat{P}_k$ the mean vector and covariance matrix of the filter conditional distribution π_k. Find a recursion for $(\hat{X}_k, \hat{P}_k)$ in terms of $(\hat{X}_{k-1}, \hat{P}_{k-1})$ and Y_k using the general filtering recursion in theorem 2.9. You may use the following matrix identity (assuming all inverses exist):

$$(\Sigma^{-1} + C^*(DD^*)^{-1}C)^{-1} = \Sigma - \Sigma C^*(DD^* + C\Sigma C^*)^{-1}C\Sigma.$$

The recursion for $(\hat{X}_k, \hat{P}_k)$ is called the *Kalman filter*.
(c) Find prediction and smoothing counterparts of the recursion in part (b).

Notes

The contents of this chapter are very well known. The filtering, smoothing and prediction problems have their origin in the work of Wiener, who was interested in stationary processes. In the more general setting of hidden Markov models, many of these ideas date back to the seminal work of Stratonovich, Kalman, Shiryaev, Baum, Petrie and others in the early 1960s.

When the signal state space is not finite and the hidden Markov model is not of the linear-Gaussian type, the filtering, smoothing and prediction recursions developed in this chapter can typically only be implemented in an approximate sense. Many such approximation have been suggested in the literature. One of the most succesful approximation methods, the Monte Carlo interacting particle filters, is discussed in chapter 4. What follows is a (highly incomplete) list of references to various other methods.

- *Extended Kalman filters* are based on local linearization of the hidden Markov model, after which the Kalman filter is applied; there are also other variations on this theme. See, e.g., [Jaz70, BLK01].
- *Truncated filters:* in certain problems the exact filter is a mixture of a finite number of simple distributions, but the number of distributions in the mixture increases in every time step. In this case, the exact filter may be approximated by 'culling' the least likely elements of the mixture in every time step to obtain a mixture of fixed size. See [BBS88, BLK01, GC03].
- *Projection filters:* here the exact filtering algorithm is constrained to remain in a fixed parametric family of distributions by 'projecting' the filter dynamics. See [BHL99, BP03].
- *Markov chain approximation:* here a finite grid is fixed in the signal state space, and the true signal process is approximated by a finite state Markov chain on this grid. The exact filter is then approximated by the filter corresponding to this finite state Markov chain. See [KD01]. How to choose a good grid is an interesting problem in itself; see [PP05].
- *Basis function expansions:* here the filter distribution is expanded in a suitable basis, and the number of basis elements is truncated in each time step. See, e.g., [Jaz70, LMR97].
- *Small noise approximations:* when the signal to noise ratio of the observations is very high, certain simple algorithms can be shown to be approximately optimal. See [Pic86] (and [Pic91] for related results).

Note that some of these papers deal with the continuous time setting.

Though the Kalman filter falls within our framework, the theory of linear estimation has a lot of special structure and is best studied as a separate topic. As a starting point, see the textbook by Kailath, Sayed and Hassibi [KSH00].

3

Finite State Space

3.1 Finite State Filtering, Smoothing, Prediction

In the previous chapter we worked out the filtering, smoothing and prediction recursions for a general hidden Markov model. In this chapter we will specialize and extend these results to an important special case: the setting where the signal state space E is a finite set. On the one hand, such models appear in many applications and therefore merit some additional attention; on the other hand, this setting is particularly convenient as the techniques developed in the previous chapter are computationally tractable without approximation.

Throughout this chapter, we consider a hidden Markov model $(X_k, Y_k)_{k\geq 0}$ on the state space $E \times F$, where the signal state space E is a finite set of cardinality $d < \infty$. Without loss of generality, we will label the elements of E as $E = \{1, \ldots, d\}$. The transition kernel, observation kernel and initial measure are denoted P, Φ, and μ, as usual. We also presume that the observations are nondegenerate, i.e., that Φ possesses a positive observation density $\Upsilon : E \times F \to]0, \infty[$ with respect to a reference probability measure φ on F.

In the finite state setting, it is convenient to think of functions and measures as vectors and of kernels as matrices (recall problem 1.1). To see this, note that a function $f : E \to \mathbb{R}$ is completely determined by the vector $\boldsymbol{f} = (f(1), \ldots, f(d))^* \in \mathbb{R}^d$ ($\boldsymbol{v}^*$, $\boldsymbol{M}^*$ denote the transpose of a vector $\boldsymbol{v}$ or matrix $\boldsymbol{M}$). Similarly, a measure μ on E is completely determined by the vector $\boldsymbol{\mu} = (\mu(\{1\}), \ldots, \mu(\{d\}))^* \in \mathbb{R}^d$: indeed,

$$\int f(x)\, \mu(dx) = \sum_{i=1}^{d} f(i)\, \mu(\{i\}) = \boldsymbol{\mu}^* \boldsymbol{f} = \boldsymbol{f}^* \boldsymbol{\mu} \qquad \text{for any } f : E \to \mathbb{R}.$$

The transition kernel P is naturally represented by a matrix $\boldsymbol{P}$ with matrix elements $\boldsymbol{P}_{ij} = P(i, \{j\})$. To see this, note that

$$Pf(i) = \sum_{j=1}^{d} P(i, \{j\})\, f(j) = (\boldsymbol{P}\boldsymbol{f})_i,$$

while

$$\mu P(\{j\}) = \sum_{i=1}^{d} \mu(\{i\})\, P(i,\{j\}) = (\boldsymbol{\mu}^* \boldsymbol{P})_j = (\boldsymbol{P}^* \boldsymbol{\mu})_j.$$

Finally, we will represent the observation density Υ as follows: for every $y \in F$, we define the diagonal matrix $\boldsymbol{\Upsilon}(y)$ with nonzero elements $(\boldsymbol{\Upsilon}(y))_{ii} = \Upsilon(i,y)$. The convenience of this definition will become evident presently.

With our new vector-matrix notation in hand, we can proceed to reformulate the results of the previous chapter. Note that we are doing nothing other than rewriting these results in a new notation: nonetheless, the vector-matrix notation leads immediately to a computational algorithm.

Remark 3.1. In the following, we will fix an observation sequence $(y_k)_{k\geq 0}$; we can therefore drop the dependence of σ_k, $\pi_{k|n}$, etc., on the observation sequence, which will considerably simplify our notation. For example: rather than writing $\sigma_k(y_0,\ldots,y_k,dx)$, we will simply write $\sigma_k(dx)$.

Let us begin by reformulating the unnormalized filtering recursion. As with any measure, we can represent the unnormalized filter by a vector $\boldsymbol{\sigma}_k = (\sigma_k(\{1\}),\ldots,\sigma_k(\{d\}))^*$. Then we immediately read off from theorem 2.9:

$$\boldsymbol{\sigma}_0 = \boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}, \qquad \boldsymbol{\sigma}_k = \boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\sigma}_{k-1} \quad (k \geq 1).$$

Denote by $\mathbf{1} \in \mathbb{R}^d$ the vector of ones $(1,\ldots,1)^*$ (i.e., $\mathbf{1}$ represents the constant function $f(x) = 1$). Representing the normalized filter π_k as a vector $\boldsymbol{\pi}_k$, we then find that $\boldsymbol{\pi}_k = \boldsymbol{\sigma}_k / \mathbf{1}^*\boldsymbol{\sigma}_k$. However, by corollary 2.10, the normalized filter can also be computed directly through the normalized recursion

$$\boldsymbol{\pi}_0 = \frac{\boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}}{\mathbf{1}^*\boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}}, \qquad \boldsymbol{\pi}_k = \frac{\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\pi}_{k-1}}{\mathbf{1}^*\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\pi}_{k-1}} \quad (k \geq 1).$$

Let us now turn to the smoothing problem. Dropping again the dependence on the observations, the unnormalized smoothing densities $\beta_{k|n}$ can be represented as vectors $\boldsymbol{\beta}_{k|n} = (\beta_{k|n}(1),\ldots,\beta_{k|n}(d))^*$. By theorem 2.12,

$$\boldsymbol{\beta}_{n|n} = \mathbf{1}, \qquad \boldsymbol{\beta}_{k|n} = \boldsymbol{P}\boldsymbol{\Upsilon}(y_{k+1})\boldsymbol{\beta}_{k+1|n} \quad (k < n).$$

The smoothing distributions can then be computed in various ways:

$$\boldsymbol{\pi}_{k|n} = \frac{\operatorname{diag}(\boldsymbol{\beta}_{k|n})\,\boldsymbol{\sigma}_k}{\boldsymbol{\beta}_{k|n}^*\boldsymbol{\sigma}_k} = \frac{\operatorname{diag}(\boldsymbol{\beta}_{k|n})\,\boldsymbol{\pi}_k}{\boldsymbol{\beta}_{k|n}^*\boldsymbol{\pi}_k} = \frac{\operatorname{diag}(\boldsymbol{\beta}_{k|n})\,\boldsymbol{\sigma}_k}{\mathbf{1}^*\boldsymbol{\sigma}_n},$$

where the second equality is trivial and the third equality follows from problem 2.2. On the other hand, we may also compute the normalized smoothing densities $\bar{\beta}_{k|n}$, represented as vectors $\bar{\boldsymbol{\beta}}_{k|n}$, as

$$\bar{\boldsymbol{\beta}}_{n|n} = \mathbf{1}, \qquad \bar{\boldsymbol{\beta}}_{k|n} = \frac{\boldsymbol{P}\boldsymbol{\Upsilon}(y_{k+1})\bar{\boldsymbol{\beta}}_{k+1|n}}{\mathbf{1}^*\boldsymbol{\Upsilon}(y_{k+1})\boldsymbol{P}^*\boldsymbol{\pi}_k} \quad (k < n),$$

Algorithm 3.1: Forward-Backward Algorithm

$\boldsymbol{\pi}_0 \leftarrow \boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}/\mathbf{1}^*\boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}$;
for *k=1,...,n* **do**
  $\tilde{\boldsymbol{\pi}}_k \leftarrow \boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\pi}_{k-1}$;
  $c_k \leftarrow \mathbf{1}^*\tilde{\boldsymbol{\pi}}_k$;
  $\boldsymbol{\pi}_k \leftarrow \tilde{\boldsymbol{\pi}}_k/c_k$;
end
$\bar{\boldsymbol{\beta}}_{n|n} \leftarrow \mathbf{1}$;
for *k=1,...,n* **do**
  $\bar{\boldsymbol{\beta}}_{n-k|n} \leftarrow \boldsymbol{P}\boldsymbol{\Upsilon}(y_{n-k+1})\bar{\boldsymbol{\beta}}_{n-k+1|n}/c_{n-k+1}$;
  $\boldsymbol{\pi}_{n-k|n} \leftarrow \mathrm{diag}(\bar{\boldsymbol{\beta}}_{n-k|n})\,\boldsymbol{\pi}_{n-k}$;
end

in which case we simply obtain $\boldsymbol{\pi}_{k|n} = \mathrm{diag}(\bar{\boldsymbol{\beta}}_{k|n})\,\boldsymbol{\pi}_k$. Finally, the vector form of the prediction recursion follows immediately from theorem 2.14:

$$\boldsymbol{\pi}_{n|n} = \boldsymbol{\pi}_n, \qquad \boldsymbol{\pi}_{k+1|n} = \boldsymbol{P}^*\boldsymbol{\pi}_{k|n} \quad (k \geq n).$$

Each of these recursions can be implemented efficiently on a computer. For example, an efficient way to compute the filtering and smoothing distributions is the *forward-backward* algorithm 3.1 which makes two passes through the observation data: a *forward* pass to compute the filtering distributions, and a *backward* pass to compute the smoothing densities.

We have obtained various forms of the filtering and smoothing recursions—both normalized and unnormalized. Which form should we use? For computational purposes, the normalized recursions are typically preferable. The reason is that in the unnormalized recursions, the normalization has the tendency to grow or shrink very rapidly in time. This will get us into big trouble when, sometimes after only a few time steps, the elements of the unnormalized filtering/smoothing quantities come close to or exceed machine precision. The normalized recursions keep the various computed quantities in a reasonable range, so that this problem is generally avoided.

3.2 Transition Counting and Occupation Times

In this section we are going to discuss some new estimation problems in the the finite state setting. The first problem is that of estimating the *occupation time* of each state $i = 1, \ldots, d$, i.e., we wish to estimate the number of times that the signal was in the state i before time n:

$$\omega_n^i(Y_0, \ldots, Y_n) = \mathbf{E}(\#\{\ell < n : X_\ell = i\}|Y_0, \ldots, Y_n).$$

The second problem that we will consider is estimation of the *transition count* between each pair of states (i, j), i.e., we wish to estimate the number of times that the signal jumped from state i to state j before time n:

$$\tau_n^{ij}(Y_0,\dots,Y_n) = \mathbf{E}(\#\{\ell < n : X_\ell = i \text{ and } X_{\ell+1} = j\}|Y_0,\dots,Y_n).$$

Though one could come up with similar problems in more general hidden Markov models, these problems are particularly natural in the finite state setting; solving them is good practice in working with the theory of the previous chapter. More importantly, however, it turns out that these two quantities are of central importance in the statistical inference problem of learning the transition probabilities $\boldsymbol{P}$ from training data, as we will see in chapter 6. We had therefore better make sure that we are able to compute them.

Forward-Backward approach

Let us begin by considering the expected occupations times ω_n^i. To compute this quantity, let us express the occupation time of state i as follows:

$$\#\{\ell < n : X_\ell = i\} = \sum_{\ell=0}^{n-1} I_i(X_\ell).$$

By the linearity of the conditional expectation, we obtain

$$\omega_n^i(Y_0,\dots,Y_n) = \sum_{\ell=0}^{n-1} \mathbf{P}(X_\ell = i|Y_0,\dots,Y_n) = \sum_{\ell=0}^{n-1} \pi_{\ell|n}(Y_0,\dots,Y_n,\{i\}).$$

To compute this quantity, we can therefore simply apply the forward-backward algorithm 3.1 of the previous section: once $\boldsymbol{\pi}_{k|n}$ have been computed for $k = 0,\dots,n-1$, we obtain directly $\omega_n^i = (\boldsymbol{\pi}_{0|n} + \cdots + \boldsymbol{\pi}_{n-1|n})_i$.

The expected transition counts τ_n^{ij} are a little more involved. We begin, in analogy with our approach to the occupation times, by noting that

$$\#\{\ell < n : X_\ell = i \text{ and } X_{\ell+1} = j\} = \sum_{\ell=0}^{n-1} I_i(X_\ell) I_j(X_{\ell+1}).$$

We therefore find that

$$\tau_n^{ij}(Y_0,\dots,Y_n) = \sum_{\ell=0}^{n-1} \mathbf{P}(X_\ell = i \text{ and } X_{\ell+1} = j|Y_0,\dots,Y_n).$$

In order to compute this quantity, we need to find a way to compute the *bivariate smoothing distributions* $\pi_{\ell,\ell+1|n} = P_{X_\ell,X_{\ell+1}|Y_0,\dots,Y_n}$.

Theorem 3.2 (Bivariate smoothing recursion). *The bivariate smoothing distributions $\pi_{\ell,\ell+1|n}$ ($\ell \le n-1$) can be computed as*

$$\pi_{\ell,\ell+1|n}(A \times B) = \frac{\int I_A(x_\ell)\, I_B(x_{\ell+1})\, \beta_{\ell+1|n}(x_{\ell+1})\, \Upsilon(x_{\ell+1}, y_{\ell+1})\, P(x_\ell, dx_{\ell+1})\, \sigma_\ell(dx_\ell)}{\int \beta_{\ell+1|n}(x_{\ell+1})\, \sigma_{\ell+1}(dx_{\ell+1})},$$

where we have dropped the dependence on $y_0, \ldots, y_n$ for notational convenience. Moreover, if we define recursively

$$\tilde{\beta}_{k|n}(x, y_0, \ldots, y_n) = \frac{\int \tilde{\beta}_{k+1|n}(x', y_0, \ldots, y_n)\, \Upsilon(x', y_{k+1})\, P(x, dx')}{\int \Upsilon(x', y_k)\, P(x, dx')\, \pi_{k-1}(y_0, \ldots, y_{k-1}, dx)}$$

with the terminal condition

$$\tilde{\beta}_{n|n}(x, y_0, \ldots, y_n) = \frac{1}{\int \Upsilon(x', y_n)\, P(x, dx')\, \pi_{n-1}(y_0, \ldots, y_{n-1}, dx)},$$

then we can write the bivariate smoothing distribution in normalized form

$$\pi_{\ell,\ell+1|n}(A \times B) = \int I_A(x_\ell)\, I_B(x_{\ell+1})\, \tilde{\beta}_{\ell+1|n}(x_{\ell+1})\, \Upsilon(x_{\ell+1}, y_{\ell+1})\, P(x_\ell, dx_{\ell+1})\, \pi_\ell(dx_\ell).$$

Proof. Up to you: Problem 3.1. □

Returning to the finite state setting, let us represent the bivariate smoothing distribution $\pi_{\ell,\ell+1|n}$ as a matrix $\boldsymbol{\pi}_{\ell,\ell+1|n}$ with matrix elements defined as $(\boldsymbol{\pi}_{\ell,\ell+1|n})_{ij} = \pi_{\ell,\ell+1|n}(\{i\} \times \{j\})$. Note that, by construction,

$$(\boldsymbol{\pi}_{\ell|n})_i = \sum_{j=1}^{d} (\boldsymbol{\pi}_{\ell,\ell+1|n})_{ij} = \boldsymbol{\pi}_{\ell,\ell+1|n}\mathbf{1} = \sum_{j=1}^{d} (\boldsymbol{\pi}_{\ell-1,\ell|n})_{ji} = \boldsymbol{\pi}^*_{\ell-1,\ell|n}\mathbf{1}.$$

Using problem 3.1(b), we may compute the bivariate smoothing distributions using the forward-backward algorithm 3.1. However, theorem 3.2 suggests that when we are interested in the bivariate distributions, it is convenient to modify the algorithm so that it computes the renormalized smoothing densities $\tilde{\boldsymbol{\beta}}_{k|n}$ rather than the smoothing densities $\bar{\boldsymbol{\beta}}_{k|n}$ of corollary 2.13. This gives the *Baum-Welch algorithm*, which is summarized as algorithm 3.2.

Finally, once we have run the Baum-Welch algorithm, we may evidently compute immediately the occupation times and transition counts:

$$\omega_n^i = \sum_{\ell=0}^{n} (\boldsymbol{\pi}_{\ell|n})_i, \qquad \tau_n^{ij} = \sum_{\ell=0}^{n-1} (\boldsymbol{\pi}_{\ell,\ell+1|n})_{ij}.$$

Alternatively, note that $\omega_n^i = \sum_{j=1}^{d} \tau_n^{ij}$, so we need not even compute $\boldsymbol{\pi}_{\ell|n}$.

Recursive approach

The Baum-Welch algorithm is of the forward-backward type: first, a forward pass is made through the observations to compute the filtering distributions; then, a backward pass is used to compute the bivariate smoothing distributions. Once the latter have been obtained, we may compute the transition

Algorithm 3.2: Baum-Welch Algorithm

$c_0 \leftarrow \mathbf{1}^* \boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}$;
$\boldsymbol{\pi}_0 \leftarrow \boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}/c_0$;
for *k=1,...,n* **do**
  $\tilde{\boldsymbol{\pi}}_k \leftarrow \boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\pi}_{k-1}$;
  $c_k \leftarrow \mathbf{1}^*\tilde{\boldsymbol{\pi}}_k$;
  $\boldsymbol{\pi}_k \leftarrow \tilde{\boldsymbol{\pi}}_k/c_k$;
end
$\tilde{\boldsymbol{\beta}}_{n|n} \leftarrow \mathbf{1}/c_n$;
for *k=1,...,n* **do**
  $\tilde{\boldsymbol{\beta}}_{n-k|n} \leftarrow \boldsymbol{P}\boldsymbol{\Upsilon}(y_{n-k+1})\tilde{\boldsymbol{\beta}}_{n-k+1|n}/c_{n-k}$;
  $\boldsymbol{\pi}_{n-k,n-k+1|n} \leftarrow \mathrm{diag}(\boldsymbol{\pi}_{n-k})\boldsymbol{P}\boldsymbol{\Upsilon}(y_{n-k+1})\mathrm{diag}(\tilde{\boldsymbol{\beta}}_{n-k+1|n})$;
  $\boldsymbol{\pi}_{n-k|n} \leftarrow \boldsymbol{\pi}_{n-k,n-k+1|n}\mathbf{1}$;
end

counts and occupation times by summing the smoothing distributions, as explained above. Note that the backward pass requires us to store both the entire observation history $y_0, \ldots, y_n$ and filter history $\boldsymbol{\pi}_0, \ldots, \boldsymbol{\pi}_n$ in memory; this is usually not a problem in off-line data analysis, but can become prohibitive if we have very long time series or if the estimation is performed on-line.

We are now going to develop a different method to compute the transition counts and occupation times which requires only a forward pass and no backward pass. This can have significant advantages; in particular, we do not need to store the observation history and filter history in memory, but instead the estimates are updated recursively in each time step using the new observation only (as in the filtering recursion). This approach also has some significant drawbacks, however. A brief discussion of the difference between the two approaches can be found at the end of the section.

Let us concentrate on the transition counts τ_n^{ij}; as noted above, we may obtain the occupation times ω_n^i by summing τ_n^{ij} over j. The idea is to introduce an auxiliary estimator of the following form:

$$(\boldsymbol{\tau}_n^{ij}(Y_0, \ldots, Y_n))_r = \mathbf{E}(I_r(X_n)\,\#\{\ell < n : X_\ell = i \text{ and } X_{\ell+1} = j\}|Y_0, \ldots, Y_n).$$

Given $\boldsymbol{\tau}_n^{ij}$, we can clearly compute the transition counts as $\tau_n^{ij} = \mathbf{1}^*\boldsymbol{\tau}_n^{ij}$. The key point is that unlike τ_n^{ij}, the auxiliary estimator $\boldsymbol{\tau}_n^{ij}$ can be computed recursively: this eliminates the need for a backward pass.

Theorem 3.3 (Transition count recursion). *The auxiliary estimator $\boldsymbol{\tau}_n^{ij}$ ($n \geq 0$) can be recursively computed as follows:*

$$\boldsymbol{\tau}_0^{ij} = \mathbf{0}, \qquad \boldsymbol{\tau}_k^{ij} = \frac{\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\tau}_{k-1}^{ij} + \boldsymbol{I}_j\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{I}_i\boldsymbol{\pi}_{k-1}}{\mathbf{1}^*\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\pi}_{k-1}} \quad (k \geq 1),$$

where $\boldsymbol{I}_i$ is the diagonal matrix whose single nonzero entry is $(\boldsymbol{I}_i)_{ii} = 1$, and $\mathbf{0}$ is the origin in $\mathbb{R}^d$ ($\mathbf{0}_i = 0$ for $i = 1, \ldots, d$).

Proof. We begin by writing

$$I_r(X_k)\,\#\{\ell < k : X_\ell = i \text{ and } X_{\ell+1} = j\} = I_r(X_k)\sum_{\ell=0}^{k-1} I_i(X_\ell)I_j(X_{\ell+1})$$
$$= I_r(X_k)\sum_{\ell=0}^{k-2} I_i(X_\ell)I_j(X_{\ell+1}) + \delta_{jr}\, I_i(X_{k-1})I_j(X_k).$$

It follows directly from theorem 3.2 that

$$\delta_{jr}\,\mathbf{E}(I_i(X_{k-1})I_j(X_k)|Y_0,\ldots,Y_k) = \frac{(\boldsymbol{I}_j\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{I}_i\boldsymbol{\pi}_{k-1})_r}{\mathbf{1}^*\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\pi}_{k-1}}.$$

It remains to deal with the first term. To this end, we return to the Bayes formula in the previous chapter, which states that

$$\int I_r(x_k)\sum_{\ell=0}^{k-2} I_i(x_\ell)I_j(x_{\ell+1})\,P_{X_0,\ldots,X_k|Y_0,\ldots,Y_k}(dx_0,\ldots,dx_k) =$$
$$\frac{\int I_r(x_k)\sum_{\ell=0}^{k-2} I_i(x_\ell)I_j(x_{\ell+1})\,\gamma_k(x_0,\ldots,x_k)\,\mu_k(dx_0,\ldots,dx_k)}{\int \gamma_k(x_0,\ldots,x_k)\,\mu_k(dx_0,\ldots,dx_k)},$$

where we have defined the functions $\gamma_k(x_0,\ldots,x_k) = \Upsilon(x_0,y_0)\cdots\Upsilon(x_k,y_k)$ and $\mu_k(dx_0,\ldots,dx_k) = P(x_{k-1},dx_k)\cdots P(x_0,dx_1)\mu(dx_0)$. Define

$$A_r(x_{k-1}) = \int I_r(x_k)\,\Upsilon(x_k,y_k)\,P(x_{k-1},dx_k).$$

Then we evidently have

$$\int I_r(x_k)\sum_{\ell=0}^{k-2} I_i(x_\ell)I_j(x_{\ell+1})\,P_{X_0,\ldots,X_k|Y_0,\ldots,Y_k}(dx_0,\ldots,dx_k) =$$
$$\int A_r(x_{k-1})\sum_{\ell=0}^{k-2} I_i(x_\ell)I_j(x_{\ell+1})\,P_{X_0,\ldots,X_{k-1}|Y_0,\ldots,Y_{k-1}}(dx_0,\ldots,dx_{k-1})$$
$$\times\,\frac{\int \gamma_{k-1}(x_0,\ldots,x_{k-1})\,\mu_{k-1}(dx_0,\ldots,dx_{k-1})}{\int \gamma_k(x_0,\ldots,x_k)\,\mu_k(dx_0,\ldots,dx_k)} = \frac{(\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\tau}_{k-1}^{ij})_r}{\mathbf{1}^*\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\pi}_{k-1}}.$$

Adding the expressions for the two terms completes the proof. □

Using this result, we can now obtain a *forward* algorithm which computes the transition counts and occuptaion times recursively using only a forward pass. This algorithm is summarized as algorithm 3.3.

Remark 3.4. Note that from theorem 3.3, we find immediately that the quantity $\boldsymbol{\omega}_k^i = \sum_{j=1}^d \boldsymbol{\tau}_k^{ij}$ can be computed recursively without computing $\boldsymbol{\tau}_k^{ij}$. As $\omega_k^i = \mathbf{1}^*\boldsymbol{\omega}_k^i$, we obtain a computationally cheaper forward algorithm for computing the occupation times. However, if the transition counts are computed anyway, there is clearly no need to perform this extra recursion.

Algorithm 3.3: Forward Algorithm

$\boldsymbol{\pi}_0 \leftarrow \boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}/\mathbf{1}^*\boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}$;
$\boldsymbol{\tau}_0^{ij} \leftarrow \mathbf{0}$, $i, j = 1, \ldots, d$;
for *k=1,...,n* **do**
  $\tilde{\boldsymbol{\pi}}_k \leftarrow \boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\pi}_{k-1}$;
  $c_k \leftarrow \mathbf{1}^*\tilde{\boldsymbol{\pi}}_k$;
  $\boldsymbol{\pi}_k \leftarrow \tilde{\boldsymbol{\pi}}_k/c_k$;
  $\boldsymbol{\tau}_k^{ij} \leftarrow (\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{\tau}_{k-1}^{ij} + \boldsymbol{I}_j\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^*\boldsymbol{I}_i\boldsymbol{\pi}_{k-1})/c_k$, $i, j = 1, \ldots, d$;
end
$\tau_n^{ij} \leftarrow \mathbf{1}^*\boldsymbol{\tau}_n^{ij}$, $i, j = 1, \ldots, d$;
$\omega_n^i \leftarrow \sum_{j=1}^d \tau_n^{ij}$, $i = 1, \ldots, d$;

We now have two approaches to compute the transition counts and occupation times. Which one is preferable in practice? There is no universal answer to this question. If enough memory is available to store the observation and filter history, and if the time horizon n is fixed, the Baum-Welch algorithm may be computationally cheaper as its cost is of order d^3n operations (each matrix multiplication is of order d^3 and there are n time steps; the fact that there are two passes only contributes a constant factor). In contrast, the forward algorithm has a computational cost of order d^5n (as there are of order d^2 recursions $\boldsymbol{\tau}_k^{ij}$, $i, j = 1, \ldots, d$ being computed simultaneously). Another advantage of the Baum-Welch algorithm is that it allows us to compute arbitrary smoothed estimates, while the forward algorithm is specific to the computation of transition counts; the forward algorithm is therefore only suitable if we are interested exclusively in the latter.

On the other hand, the Baum-Welch algorithm assumes that the time horizon is fixed. If we wanted to compute τ_k^{ij} for all $k = 0, \ldots, n$ using the Baum-Welch algorithm, we would have to repeat the algorithm for every time horizon k separately so that the total computational cost is of order d^3n^2. This may be prohibitive when n is large, while the forward algorithm (with cost d^5n) may do better in this setting. Another advantage of the forward algorithm is that its memory requirements do not depend on the time horizon n, unlike in the Baum-Welch algorithm. Particularly for long time series and for on-line computation, the forward algorithm may then turn out to be preferable.

3.3 The Viterbi Algorithm

Up to this point, we have discussed how to implement the generic filtering, smoothing and prediction problems for finite state signals; these techniques can be applied in a wide variety of applications for various different purposes (see, e.g., section 1.3). We also discussed two special estimation problems—transition counting and occupation time estimation—which we will need later on to solve the important problem of statistical inference (chapter 6).

In this section, we turn to a more specific type of problem: the estimation, or *decoding*, of a a finite state signal path $x_0, \dots, x_n$ from observed data $y_0, \dots, y_n$. Consider, for example, a finite alphabet message that is encoded and transmitted through a noisy channel; the signal state space E then represents the signal alphabet, the signal $(X_k)_{0 \le k \le n}$ is the message, and the observation sequence $(Y_k)_{0 \le k \le n}$ is the encoded and corrupted message as it is received after transmission through the channel. We would like to infer as best we can the transmitted message from the observation sequence: i.e., we are seeking to construct the random variables $\hat{X}_0, \dots, \hat{X}_n$, each of which is a function of the observed sequence $\hat{X}_k = f_k(Y_0, \dots, Y_0)$, such that the estimate $(\hat{X}_k)_{0 \le k \le n}$ is as close as possible to the true signal $(X_k)_{0 \le k \le n}$. The solution of this problem depends, however, on what we mean by 'as close as possible'.

Let us first consider the following problem:

Choose $(\hat{X}_k)_{k \le n}$ such that $\mathbf{E}(\#\{k \le n : X_k = \hat{X}_k\})$ is maximized.

In words, we would like to design the estimate so that as many as possible individual symbols in the message are decoded correctly. First, note that

$$\#\{k \le n : X_k = \hat{X}_k\} = \sum_{k=0}^{n} I_0(X_k - \hat{X}_k).$$

Therefore, by lemma 2.4, we must choose the functions f_k such that

$$(f_0(y_0, \dots, y_n), \dots, f_n(y_0, \dots, y_n)) = \operatorname*{argmax}_{(\hat{x}_0, \dots, \hat{x}_n)} \int \sum_{k=0}^{n} I_0(x_k - \hat{x}_k)\, P_{X_0, \dots, X_n | Y_0, \dots, Y_n}(y_0, \dots, y_n, dx_0, \dots, dx_n).$$

However, due to the elementary fact that the maximum distributes over a sum (i.e., $\max_{z_0, \dots, z_n}(g_0(z_0) + \cdots + g_n(z_n)) = \max_{z_0} g_0(z_0) + \cdots + \max_{z_n} g_n(z_n)$), we may compute each f_k independently:

$$\begin{aligned} f_k(y_0, \dots, y_n) &= \operatorname*{argmax}_{\hat{x}} \int I_0(x - \hat{x})\, P_{X_k | Y_0, \dots, Y_n}(y_0, \dots, y_n, dx) \\ &= \operatorname*{argmax}_i \pi_{k|n}(y_0, \dots, y_n, \{i\}) = \operatorname*{argmax}_i (\boldsymbol{\pi}_{k|n})_i. \end{aligned}$$

Evidently the optimal estimate, in the sense of maximum number of correctly decoded symbols, is obtained by choosing $\hat{X}_k$ to be the MAP estimate of X_k given $Y_0, \dots, Y_n$ (see example 2.6). Computationally, we already know how to obtain this estimate: using either the forward-backward algorithm 3.1 or the Baum-Welch algorithm 3.2 to compute the smoothing distributions $\boldsymbol{\pi}_{k|n}$, the signal estimate is obtained by selecting for each time k the symbol whose smoothing probability $(\boldsymbol{\pi}_{k|n})_i$ is maximal.

The above approach to decoding the signal has an important drawback, however. The problem is most easily illustrated using a trivial example.

Example 3.5. For simplicity, we consider an example where there are no observations. The signal state space is $E = \{0, 1\}$, and the transition probabilities are such that $P(0, \{1\}) = P(1, \{0\}) = 1$. We also choose the initial measure $\mu(\{0\}) = \mu(\{1\}) = 1/2$. As there are no observations (e.g., $Y_k = 0$ for all k), we simply have $\pi_{k|n}(\{i\}) = \mathbf{P}(X_k = i) = 1/2$ for every i, k, n.

We now seek to estimate the signal. As all individual probabilities are $1/2$, the above discussion shows that any choice of estimate $\hat{X}_k$, $k = 0, \ldots, n$ has the same expected number of correctly decoded symbols. We may therefore choose an optimal estimator in this sense by setting $\hat{X}_k = 0$ for all k. However, the signal path $X_k = 0$ for all k has probability zero, as $P(0, \{0\}) = 0$!

Evidently an estimate of the signal path which maximizes the number of correctly decoded individual symbols need not maximize the probability that the entire path is decoded without errors; in particularly bad cases the former technique can even give rise to an estimate which is not actually a valid signal path. The problem is that by maximizing the probability of each symbol individually, we are not necessarily constrained to respect the possible transitions between adjacent symbols. In problems where the latter is important, it may be preferable to solve the following alternative estimation problem:

$$\text{Choose } (\hat{X}_k)_{k \le n} \text{ such that } \mathbf{P}(X_k = \hat{X}_k \text{ for all } k \le n) \text{ is maximized.}$$

In general the two estimation problems will have different solutions.

We now consider how to compute the maximum probability path estimate. The bad news is that as the event $\{X_k = \hat{X}_k \text{ for all } k \le n\}$ can not be written as a disjoint union of events for each time k individually, we can not use the above technique to reduce the problem to the forward-backward or Baum-Welch algorithms. The good news is, however, that we may still compute the maximum probability path estimate using a recursive algorithm, called the *Viterbi algorithm*, which we will develop presently. The Viterbi algorithm is widely used in communications engineering applications—most likely your cell phone incorporates it in some form or another.

To compute the maximum probability path estimate we must choose, by lemma 2.4, the estimate functions f_k such that

$$(f_0(y_0, \ldots, y_n), \ldots, f_n(y_0, \ldots, y_n)) = \operatorname*{argmax}_{(\hat{x}_0, \ldots, \hat{x}_n)} \int \prod_{k=0}^{n} I_0(x_k - \hat{x}_k)\, P_{X_0,\ldots,X_n|Y_0,\ldots,Y_n}(y_0, \ldots, y_n, dx_0, \ldots, dx_n).$$

Using the Bayes formula, we can evaluate explicitly

$$\int \prod_{k=0}^{n} I_0(x_k - \hat{x}_k)\, P_{X_0,\ldots,X_n|Y_0,\ldots,Y_n}(y_0, \ldots, y_n, dx_0, \ldots, dx_n) = \frac{\Upsilon(\hat{x}_0, y_0) \cdots \Upsilon(\hat{x}_n, y_n)\, P(\hat{x}_{n-1}, \{\hat{x}_n\}) \cdots P(\hat{x}_0, \{\hat{x}_1\})\, \mu(\{\hat{x}_0\})}{\int \Upsilon(x_0, y_0) \cdots \Upsilon(x_n, y_n)\, P(x_{n-1}, dx_n) \cdots P(x_0, dx_1)\, \mu(dx_0)}.$$

The denominator does not depend on $\hat{x}_0, \ldots, \hat{x}_n$, however, so evidently

$$(f_0(y_0, \ldots, y_n), \ldots, f_n(y_0, \ldots, y_n)) = \operatorname*{argmax}_{(\hat{x}_0, \ldots, \hat{x}_n)} \Upsilon(\hat{x}_0, y_0) \cdots \Upsilon(\hat{x}_n, y_n)\, P(\hat{x}_{n-1}, \{\hat{x}_n\}) \cdots P(\hat{x}_0, \{\hat{x}_1\})\, \mu(\{\hat{x}_0\}),$$

or, even more conveniently,

$$(f_0(y_0, \ldots, y_n), \ldots, f_n(y_0, \ldots, y_n)) = \operatorname*{argmax}_{(\hat{x}_0, \ldots, \hat{x}_n)} \left[\log(\mu(\{\hat{x}_0\})\Upsilon(\hat{x}_0, y_0)) + \sum_{k=1}^{n} (\log P(\hat{x}_{k-1}, \{\hat{x}_k\}) + \log \Upsilon(\hat{x}_k, y_k)) \right]$$

(we have used that as $\log x$ is increasing, $\operatorname{argmax}_x f(x) = \operatorname{argmax}_x \log f(x)$).

The idea behind the Viterbi algorithm is to introduce the functions

$$v_\ell(\hat{x}_\ell) = \max_{\hat{x}_0, \ldots, \hat{x}_{\ell-1}} \left[\log(\mu(\{\hat{x}_0\})\Upsilon(\hat{x}_0, y_0)) + \sum_{k=1}^{\ell} (\log P(\hat{x}_{k-1}, \{\hat{x}_k\}) + \log \Upsilon(\hat{x}_k, y_k)) \right].$$

The key property of these functions is that they can be computed recursively.

Theorem 3.6 (Viterbi recursion). *The functions v_ℓ satisfy the recursion*

$$v_\ell(\hat{x}_\ell) = \max_{\hat{x}_{\ell-1}} \{v_{\ell-1}(\hat{x}_{\ell-1}) + \log P(\hat{x}_{\ell-1}, \{\hat{x}_\ell\})\} + \log \Upsilon(\hat{x}_\ell, y_\ell)$$

with the initial condition $v_0(\hat{x}_0) = \log(\mu(\{\hat{x}_0\})\Upsilon(\hat{x}_0, y_0))$. Moreover, the estimating functions $f_\ell(y_0, \ldots, y_n)$, $\ell = 1, \ldots, n$ for the maximum probability path estimate given $Y_0, \ldots, Y_n$ satisfy the backward recursion

$$f_\ell = \operatorname*{argmax}_{\hat{x}_\ell} \{v_\ell(\hat{x}_\ell) + \log P(\hat{x}_\ell, \{f_{\ell+1}\})\}$$

with the terminal condition $f_n = \operatorname{argmax}_{\hat{x}_n} v_n(\hat{x}_n)$.

Proof. The result can be read off immediately from the definition of the functions v_ℓ and from the above expression for the estimating functions f_ℓ. □

The Viterbi algorithm can be implemented directly as a computer algorithm; we have summarized it as algorithm 3.4. Note that the algorithm consists of a forward pass and a backward pass, which is reminiscent of the smoothing algorithms earlier in the chapter. However, there is an important difference: the backward pass in the Viterbi algorithm does not explicitly use the observation sequence $y_0, \ldots, y_k$. Therefore the observation history does not need to be stored in memory (but we do need to store at least all $v_\ell(i)$).

Algorithm 3.4: Viterbi Algorithm

$v_0(i) \leftarrow \log \boldsymbol{\mu}_i + \log \Upsilon(i, y_0)$, $i = 1, \ldots, d$;
for *k=1,...,n* **do**
  $b_k(i) \leftarrow \operatorname{argmax}_{j=1,\ldots,d}\{v_{k-1}(j) + \log \boldsymbol{P}_{ji}\}$, $i = 1, \ldots, d$;
  $v_k(i) \leftarrow v_{k-1}(b_k(i)) + \log \boldsymbol{P}_{b_k(i)i} + \log \Upsilon(i, y_k)$, $i = 1, \ldots, d$;
end
$f_n \leftarrow \operatorname{argmax}_{j=1,\ldots,d} v_n(j)$;
for *k=1,...,n* **do**
  $f_{n-k} \leftarrow b_{n-k+1}(f_{n-k+1})$;
end

Remark 3.7. The Viterbi algorithm succeeds in splitting up a global optimization problem so that the optimum can be computed recursively: in each step we maximize over one variable only, rather than maximizing over all n variables simultaneously. The general underlying idea that allows one to solve optimization problems in this manner is Bellman's *dynamic programming principle*; the Viterbi algorithm is an excellent example of this principle in action. We will encounter dynamic programming again repeatedly in chapter 9, where it will be used to solve optimal control problems.

A numerical example

To round off this chapter on a more concrete note, let us briefly work out a simple example. This example is inspired by a problem in biophysics [MJH06], though we will make up some parameters for sake of example. A different example is given as problem 3.5, where you will work through a practical application in communications theory.

The problem in the present example is the following. Recall that the DNA molecule—the carrier of genetic information—consists of two strands that are twisted around one another. In order to regulate the readout of DNA, it is possible for proteins to bind to various parts of the DNA strand; this can either suppress or enhance the expression of a gene. To understand this mechanism more fully, biophysicists are interested in measuring experimentally the dynamics of the binding and dissociation of proteins to the DNA molecule.

One way to do this is to attach to each strand of a DNA molecule a fluorescent dye of a different color: one red and one green, say. We then excite the red dye with a red laser. If the distance between the two dyes is short, then some of the energy can be transfered from the red dye to the green dye, in which case we observe that some green light is emitted. However, the amount of energy transfer depends strongly on the distance between the dyes. The trick is that when a protein binds to the DNA molecule, it wedges itself between the dyes so that their distance is increased. Therefore, when a protein binds, we expect to see a reduction in the amount of emitted green light. If another protein binds, we expect to see a further reduction, etc. By

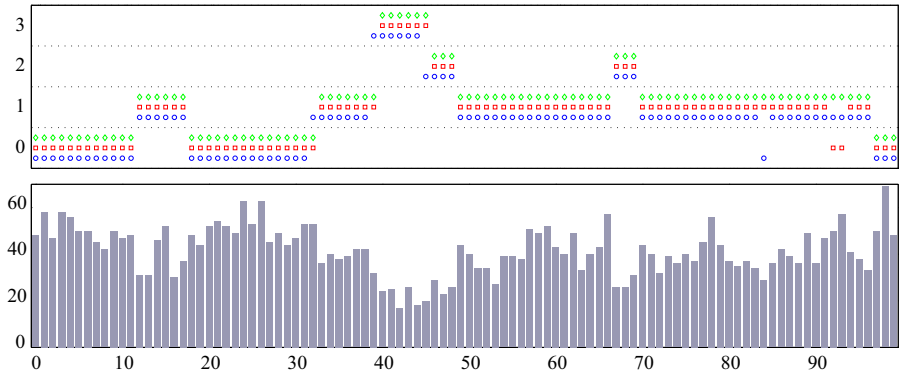

Fig. 3.1. A typical run of the FRET example. The bottom plot shows the photon count in each time bin. The top plot shows the true number of bound proteins (blue circles), the Baum-Welch (red squares) and Viterbi (green diamonds) estimates.

monitoring the green light emitted from the experiment, we can therefore try to estimate when protein binding or dissociation events occur. This is known as a FRET (fluorescence resonance energy transfer) experiment.

Using modern technology, one can easily perform FRET experiments at the single molecule level, so that one can really observe the individual binding and dissociation events. However, there is only so much signal that can be obtained from a single molecule; in particular, in each time interval one only observes a relatively small number of green photons. The observations in such an experiment are therefore subject to Poissonian photon counting statistics. Hidden Markov models provide a tool to decode the individual binding/dissociation events from the noisy photon count data.

For sake of illustration, we make up an example with contrived numerical parameters. Let us assume that at most three proteins reasonably bind to DNA at once. The signal process X_k is the number of proteins bound in the time interval k: it is therefore modeled in the signal state space $E = \{0, 1, 2, 3\}$. For our example, we will presume that

$$\boldsymbol{P} = \begin{bmatrix} .94 & .05 & .01 & .00 \\ .03 & .94 & .02 & .01 \\ .05 & .14 & .80 & .01 \\ .05 & .15 & .30 & .50 \end{bmatrix}, \qquad \boldsymbol{\mu} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

For the observations, we presume that Y_k is Poisson distributed (as is befitting for photon counts) with rate parameter $50 - 10\,X_k$. In particular, $F = \mathbb{Z}_+$ and we can define the reference measure and observation density

$$\varphi(\{j\}) = \frac{e^{-1}}{j!} \quad (j \in \mathbb{Z}_+), \qquad \Upsilon(x, y) = (50 - 10\,x)^y\, e^{-49+10\,x}.$$

This is all we need to use the algorithms in this chapter. A typical numerical trajectory is shown in figure 3.1, together with the Baum-Welch and Viterbi

estimates of the signal (the transition counts and occupation times can of course also be computed, if one is interested).

In a real application one obviously does not wish to impose an arbitrary model as we did here; instead, one would like to infer the various parameters from experimental data. This problem will be treated in detail in chapter 6.

Problems

3.1. Bivariate Smoothing Distributions
(a) Prove theorem 3.2 for general hidden Markov models.
(b) What is the relation between $\bar{\beta}_{k|n}$ (corollary 2.13) and $\tilde{\beta}_{k|n}$?

3.2. Emission Counting
Suppose that $E = \{1, \dots, d\}$ and $F = \{1, \dots, d'\}$ are both finite. Construct a forward-backward as well as a recursive algorithm to compute the *emission counts* $\alpha_n^{ij}(Y_0, \dots, Y_n) = \mathbf{E}(\#\{\ell < n : X_\ell = i \text{ and } Y_\ell = j\}|Y_0, \dots, Y_n)$.

3.3. Smoothing Functionals
We have seen that transition counts and occupation times can be computed in a recursive manner. By the previous problem, this is true also for emission counts. In fact, as is pointed out in [CMR05, section 4.1.2], there is a general class of *smoothing functionals* which can be computed recursively in this manner. Consider a function $t_n(X_0, \dots, X_n)$ which is defined iteratively by

$$t_{n+1}(X_0, \dots, X_{n+1}) = m_n(X_n, X_{n+1})\, t_n(X_0, \dots, X_n) + s_n(X_n, X_{n+1}),$$

where $t_0 : E \to \mathbb{R}$ and $m_n, s_n : E \times E \to \mathbb{R}$ are given functions. Show that $\mathbf{E}(t_n(X_0, \dots, X_n)|Y_0, \dots, Y_n)$ can be computed recursively in a similar manner as transition counts, occupation times and emission counts. Show also that the latter are special cases of this general framework.

3.4. DNA Sequence Alignment II
In problem 1.3 we investigated a technique for DNA sequence recognition and alignment. In that problem, we approached the required computations by brute force. In this particularly simple example this approach is still tractable, but in more realistic settings with longer strings and patterns the computational complexity becomes prohibitive. Fortunately, we now have the tools to perform the computations in a very efficient manner.
(a) Prove that the following holds:

$$\text{score}(y_0, \dots, y_k) = \sigma_k(y_0, \dots, y_k, E) = \mathbf{1}^* \boldsymbol{\sigma}_k,$$

provided that we choose the reference measure φ (see definition 1.9) to be the uniform distribution on $F = \{A, C, G, T\}$.
(b) Reimplement the computations in problem 1.3 using the filtering recursion and the Viterbi algorithm, and verify that everything works as expected.

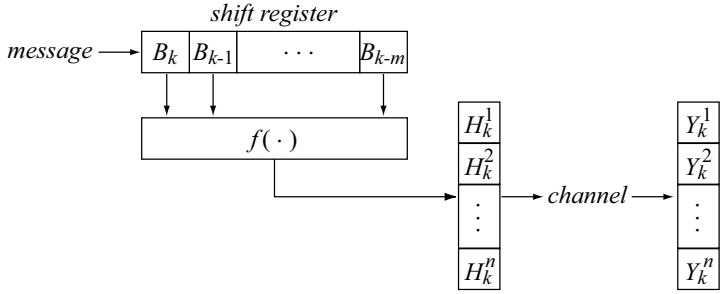

Fig. 3.2. A shift-register encoding model of length m and rate n^{-1}.

3.5. Channel Coding and Decoding

In digital communications one is faced with the basic problem if transmitting a digital message through a noisy channel. The message is modeled as a sequence B_k, $k \geq 0$ of i.i.d. bits ($\mathbf{P}(B_k = 0) = \mathbf{P}(B_k = 1) = 1/2$) and the channel transmits a single bit correctly with probability $p \in]0,1[$ and flips the bit with probability $1-p$ (see example 1.15). Clearly if we were to transmit each message bit directly through the channel, we would lose a fraction $1-p$ of our message. This performance is generally unacceptable.

To circumvent the problem, one must introduce some redundancy into the message to increase the probability of correct decoding after transmission. If n bits are transmitted through the channel for every message bit, the encoder is said to have *rate* n^{-1}. A general encoding architecture is the *shift-register model* of *length* m and rate n^{-1}; this means that when message bit B_k arrives, n bits $H_k = (H_k^1, \ldots, H_k^n) \in \{0,1\}^n$ are transmitted through the channel which are computed as a function of the m previous message bits:

$$H_k = f(B_k, B_{k-1}, \ldots, B_{k-m}), \qquad f : \{0,1\}^{m+1} \to \{0,1\}^n$$

(see figure 3.2). The function f determines the encoding strategy.

(a) Model the shift-register model as a hidden Markov model. The state X_k should contain the $m+1$ bits in the shift register, while the output Y_k consists of the n output bits at time k *after* transmission through the channel.

A specific example of an encoder is a rate $1/2$, length 2 convolutional code. The function f is defined through $H_k^1 = B_k \oplus B_{k-2}$ and $H_k^2 = B_k \oplus B_{k-1} \oplus B_{k-2}$ (a parity check function; here $\oplus$ denotes addition modulo 2).

(b) Implement a computer simulation of the tranmission of a message through a noisy channel, and use the Viterbi algorithm to reconstruct the message from the channel output. Experiment with the parameter p and compare the error frequency obtained with this encoding scheme to the error frequency without encoding. Note: to implement the Viterbi algorithm we must assume a message model (i.e., that the message bits are i.i.d.) However, you might find it fun to experiment with transmitting actual messages (e.g., a text message) through your simulated model (e.g., by converting it to ASCII code).

Remark 3.8. A communications device that uses the Viterbi algorithm to decode an encoded message is known as a *Viterbi decoder*. The convolutional encoding/Viterbi decoding approach is implemented in billions of cell phones. We have discussed a particularly simple example of a rate 1/2 length 2 code, but slightly longer convolutional codes are indeed in very widespread use.

Notes

The finite state setting is the simplest (and oldest) one in which the recursions of the previous chapter can be developed. Nonetheless, a large fraction of the applications of hidden Markov models falls within this setting. On the one hand, the finite state setting is both computationally and theoretically much less demanding than the continuous setting, making it eminently suitable for implementation. On the other hand, surprisingly many applications can be modeled at least approximately as finite state processes, particularly in the area of digital communications. In other applications, regime switching models are common and can successfully capture the statistics of many time series.

There are many variants of the forward-backward algorithm. The original forward-backward algorithm is often attributed to Baum et al. [BPSW70], but a forward-backward type algorithm already appears a decade earlier in the paper of Stratonovich [Str60] (the use of such an algorithm for parameter inference is due to Baum et al., however). The recursive transition and occupation count filters are due to Zeitouni and Dembo [ZD88], see also [EAM95]. The Viterbi algorithm is due to Viterbi [Vit67]. A nice discussion of the Viterbi algorithm together with various applications in communication theory can be found in the tutorial by Forney [For73].

4

Monte Carlo Methods: Interacting Particles

In chapter 2 we completely solved, at least in principle, the filtering problem for any nondegenerate hidden Markov model. However, we also saw that the filtering recursion is infinite dimensional, so that we run into formidable computational difficulties when we wish to apply this technique in practice. With the exception of one special (but important) case—the finite state case of chapter 3—nonlinear filters typically suffer from the curse of dimensionality. Therefore simple approximation methods, such as state space discretization, become rapidly intractable in all but the simplest cases.

There is one approximation method, however, that has turned out to be very successful—the use of *Monte Carlo* or *random sampling* methods to approximate the filtering recursion. Though such algorithms do not manage to entirely avoid the curse of dimensionality, they are flexible, easily implementable and typically lead to good performance even in complicated models. In this chapter we will introduce the Monte Carlo technique in its basic form and prove its convergence to the exact filter in the appropriate limit.

Remark 4.1. We will restrict ourselves to Monte Carlo algorithms for filtering; prediction is trivially incorporated, while Monte Carlo smoothing requires a little more work. Various Monte Carlo algorithms for smoothing can be found in the literature, though these may not be recursive in nature; see [CMR05, ch. 6–8]. A sequential Monte Carlo smoother can be found, e.g., in [DGA00].

4.1 SIS: A Naive Particle Filter

The basic idea behind Monte Carlo approximations is extremely simple, and can be explained in a few lines. We will develop in this section a naive Monte Carlo algorithm based on this idea. However, as we will see, the performance of the naive algorithm is not yet satisfactory, and we will introduce an additional ingredient in the next section in order to obtain a useful algorithm.

Let us begin by noting that by definition 2.8, the unnormalized filtering distribution σ_k evidently satisfies

$$\int f(x)\,\sigma_k(y_0,\ldots,y_k,dx) = \mathbf{E}(f(X_k)\,\Upsilon(X_0,y_0)\cdots\Upsilon(X_k,y_k))$$

for every bounded measurable function f. Now suppose that we have the ability to *simulate* the signal process, i.e., to produce i.i.d. samples from the joint distribution μ_X of the signal values $(X_0,\ldots,X_k)$. Then we can approximate the unnormalized filter σ_k using the law of large numbers:

$$\int f(x)\,\sigma_k(y_0,\ldots,y_k,dx) \approx \frac{1}{N}\sum_{i=1}^N f(x_k^{(i)})\,\Upsilon(x_0^{(i)},y_0)\cdots\Upsilon(x_k^{(i)},y_k),$$

where $x^{(i)} = (x_0^{(i)},\ldots,x_k^{(i)}) \in E^{k+1}$, $i = 1,\ldots,N$ are i.i.d. samples from the signal distribution μ_X. In particular, we can approximate

$$\int f(x)\,\pi_k(y_0,\ldots,y_k,dx) \approx \frac{\sum_{i=1}^N f(x_k^{(i)})\,\Upsilon(x_0^{(i)},y_0)\cdots\Upsilon(x_k^{(i)},y_k)}{\sum_{i=1}^N \Upsilon(x_0^{(i)},y_0)\cdots\Upsilon(x_k^{(i)},y_k)}.$$

The strong law of large numbers immediately guarantees that the right hand side of the expression converges to the left hand side as we increase the number of samples $N \to \infty$ for any bounded (or even just integrable) measurable function f. Thus, for large N, this Monte Carlo approach does indeed give rise to an approximation of the filtering distribution.

Note that the above expression can be written as

$$\int f(x)\,\pi_k(y_0,\ldots,y_k,dx) \approx \sum_{i=1}^N w_k^{(i)}\,f(x_k^{(i)}),$$

where we have defined the *weights* $w_k^{(i)}$ as

$$w_k^{(i)} = \frac{\Upsilon(x_0^{(i)},y_0)\cdots\Upsilon(x_k^{(i)},y_k)}{\sum_{i=1}^N \Upsilon(x_0^{(i)},y_0)\cdots\Upsilon(x_k^{(i)},y_k)}.$$

These weights $w_k^{(i)}$, $i = 1,\ldots,N$ are positive and sum to one. They can therefore be intepreted as probabilities. Note, in particular, that under the signal measure μ_X each sample path i is equally likely by construction (each has probability $1/N$). However, in computing the approximate filter, we reweight each sample path by the corresponding (observation-dependent) weight. The observations therefore enter the picture by modifying the relative importance of each of our simulated sample paths. The Monte Carlo approach can thus be seen as a variant of *importance sampling*.

We now make the key observation that the samples $x_k^{(i)}$ and weights $w_k^{(i)}$ can be generated recursively, just like the exact filter can be computed recursively. This idea allows us to turn the importance sampling technique into

Algorithm 4.1: Sequential Importance Sampling (SIS)

Sample $x_0^{(i)}$, $i = 1, \dots, N$ from the initial distribution μ;
Compute $w_0^{(i)} = \Upsilon(x_0^{(i)}, y_0) / \sum_{i=1}^N \Upsilon(x_0^{(i)}, y_0)$, $i = 1, \dots, N$;
for *k=1,...,n* **do**
  Sample $x_k^{(i)}$ from $P(x_{k-1}^{(i)}, \cdot)$, $i = 1, \dots, N$;
  Compute $w_k^{(i)} = w_{k-1}^{(i)} \Upsilon(x_k^{(i)}, y_k) / \sum_{i=1}^N w_{k-1}^{(i)} \Upsilon(x_k^{(i)}, y_k)$, $i = 1, \dots, N$;
end
Compute approximate filter $\int f(x)\, \pi_n(y_0, \dots, y_n, dx) \approx \sum_{i=1}^N w_n^{(i)} f(x_n^{(i)})$;

algorithm 4.1, called *sequential importance sampling (SIS)* for obvious reasons. It is a simple exercise to verify by induction that the samples and weights generated by this algorithm coincide with the above expressions. Moreover, the SIS algorithm is easily implemented on a computer.

Remark 4.2. Sampling from the conditional distribution $P(x, \cdot)$ is particularly efficient when the signal is modeled as a recursion

$$X_k = F(X_{k-1}, \xi_k) \quad (k \geq 1),$$

where ξ_k, $k \geq 1$ are i.i.d. random variables whose distribution Ξ can be efficiently sampled (e.g., $\Xi = \mathrm{Unif}[0,1]$ or $\Xi = N(0,1)$). Indeed, in this case we may sample $x_k \sim P(x_{k-1}, \cdot)$ simply by sampling $\xi_k \sim \Xi$ and computing $x_k = F(x_{k-1}, \xi_k)$. Similarly, evaluation of the observation density Υ can be done efficiently when the observation process Y_k has a convenient form. For example, consider the common setting where $F = \mathbb{R}^p$ and

$$Y_k = H(X_k) + \eta_k \quad (k \geq 0),$$

where $H : E \to \mathbb{R}^p$ is a given observation function and η_k, $k \geq 0$ are i.i.d. random variables whose distribution has density f_η with respect to the Lebesgue measure on $\mathbb{R}^p$. Then we may choose $\Upsilon(x, y) = f_\eta(y - H(x))$ (problem 4.1).

Unfortunately, the SIS algorithm has some rather severe problems. To see what goes wrong, consider a simple example where X_k is a symmetric random walk on the lattice $\mathbb{Z}^3 \subset \mathbb{R}^3$ and $Y_k = X_k + \varepsilon\, \eta_k$, where η_k are i.i.d. $N(0, \mathrm{Id})$ and $\varepsilon \ll 1$. As the signal to noise ratio is high, we expect the filter distribution π_k to be sharply concentrated around the true location of the signal $X_k = x_k$.

However, in the SIS algorithm, the samples $x_k^{(i)}$ are chosen according to the unconditioned signal distribution μ_X; in particular, if we sample from μ_X at random, only a small fraction of the samples will be close to any fixed location x_k. What will then happen in the SIS algorithm is that after only a few iterations all but one of the Monte Carlo samples will be assigned near-zero weights, so that the effective Monte Carlo approximation consists of only one sample rather than N samples. As a consequence, the approximation error

$$\mathbf{E}\left[\left(\int f(x)\,\pi_k(y_0,\ldots,y_k,dx) - \sum_{i=1}^{N} w_k^{(i)}\, f(x_k^{(i)})\right)^2\right]$$

will typically grow very rapidly as we increase number of iterations k (while keeping the number of samples N fixed), thus rendering the algorithm effectively useless. The problem is, of course, that reweighting a finite number of samples obtained from one distribution to approximate another distribution does not work well if the two distributions are too far apart. To make the SIS algorithm effective, we have to change our sampling strategy so that the distribution of our samples is closer to the filtering distribution π_k.

4.2 SIS-R: Interacting Particles

The idea to resolve the problems of the naive SIS algorithm is surprisingly simple. Recall that the filtering recursion can be seen as a two step procedure:

$$\pi_k \xrightarrow{\text{prediction}} \pi_{k+1|k} \xrightarrow{\text{correction}} \pi_{k+1}.$$

Let us suppose, for the moment, that we have some way of doing the following:

$$\text{Sample } x_k^{(i)},\ i = 1,\ldots,N \text{ from the filtering distribution } \pi_k(y_0,\ldots,y_k,dx).$$

Proceeding as in the SIS algorithm, we can

$$\text{Sample } x_{k+1|k}^{(i)} \text{ from } P(x_k^{(i)},\,\cdot\,) \text{ for every } i = 1,\ldots,N.$$

Then $x_{k+1|k}^{(i)}$, $i = 1,\ldots,N$ are clearly i.i.d. samples from the one step predictive distribution $\pi_{k+1|k}(y_0,\ldots,y_k,dx)$. Let us now compute the weights

$$w_{k+1}^{(i)} = \frac{\Upsilon(x_{k+1|k}^{(i)}, y_{k+1})}{\sum_{i=1}^{N} \Upsilon(x_{k+1|k}^{(i)}, y_{k+1})}.$$

Then, by the filtering recursion and the law of large numbers,

$$\int f(x)\,\pi_{k+1}(y_0,\ldots,y_{k+1},dx) \approx \sum_{i=1}^{N} w_{k+1}^{(i)}\, f(x_{k+1|k}^{(i)})$$

for any bounded measurable function f. In particular, we have approximated the filtering measure by an *empirical measure*:

$$\pi_{k+1}(y_0,\ldots,y_{k+1},dx) \approx \sum_{i=1}^{N} w_{k+1}^{(i)}\, \delta_{x_{k+1|k}^{(i)}}(dx).$$

Algorithm 4.2: Sequential Importance Sampling/Resampling (SIS-R)

Sample $\tilde{x}_0^{(i)}$, $i = 1, \ldots, N$ from the initial distribution μ;
Compute $w_0^{(i)} = \Upsilon(\tilde{x}_0^{(i)}, y_0) / \sum_{i=1}^N \Upsilon(\tilde{x}_0^{(i)}, y_0)$, $i = 1, \ldots, N$;
Sample $x_0^{(i)}$, $i = 1, \ldots, N$ from the distribution $\text{Prob}(\tilde{x}_0^{(j)}) = w_0^{(j)}$;
for *k=1,...,n* **do**
  Sample $\tilde{x}_k^{(i)}$ from $P(x_{k-1}^{(i)}, \cdot)$, $i = 1, \ldots, N$;
  Compute $w_k^{(i)} = \Upsilon(\tilde{x}_k^{(i)}, y_k) / \sum_{i=1}^N \Upsilon(\tilde{x}_k^{(i)}, y_k)$, $i = 1, \ldots, N$;
  Sample $x_k^{(i)}$, $i = 1, \ldots, N$ from the distribution $\text{Prob}(\tilde{x}_k^{(j)}) = w_k^{(j)}$;
end
Compute approximate filter $\int f(x)\, \pi_n(y_0, \ldots, y_n, dx) \approx \frac{1}{N} \sum_{i=1}^N f(x_n^{(i)})$;

In the SIS algorithm, we would now apply the prediction step again to $x_{k+1|k}^{(i)}$ and update the weights. However, recall that we started the present iteration by sampling from the filter π_k. As we have now obtained an approximation of the filtering distribution π_{k+1}, we can begin a new iteration with:

$$\text{Sample } x_{k+1}^{(i)},\ i = 1, \ldots, N \text{ from the approximate filter } \sum_{i=1}^N w_{k+1}^{(i)}\, \delta_{x_{k+1|k}^{(i)}}(dx).$$

Instead of repeatedly updating the weights as in the SIS algorithm, this *resampling step* essentially resets all the weights to $1/N$ at the end of every iteration. The resulting algorithm, which is called *sequential importance sampling with resampling (SIS-R)* or the *bootstrap filter*, is summarized as algorithm 4.2.

What is actually going when we resample? If a sample has a small weight, it will be less likely to be selected in the resampling step. Therefore, some of the samples with small weights will disappear when we resample. On the other hand, as the number of samples N is fixed, some of the samples with large weights will be sampled more than once in the resampling step. Resampling thus has the effect that the samples with low likelihood given the observations 'die' while the samples with high likelihood given the observations 'give birth' to offspring, thus resolving the basic problem of the naive SIS algorithm. This idea is characteristic of a class of algorithms called *evolutionary* or *genetic algorithms*, which propagate a collection of particles by first applying a *mutation* step, where each of the particles moves ('mutates') at random, and a *selection* step, where the less desirable particles die and more desirable particles give birth to offspring ('survival of the fittest').

Beside its obvious advantages, however, the SIS-R algorithm introduces an additional difficulty. Recall that in the SIS algorithm, the paths $(x_0^{(i)}, \ldots, x_k^{(i)})$ were independent for different $i = 1, \ldots, N$. Therefore convergence of the approximate filter to the exact filter as $N \to \infty$ was immediate from the law of large numbers. However, in the SIS-R algorithm, the resampling step kills or duplicates each sample according the observation weights of all the samples. Therefore, the different samples are no longer independent, as they 'interact'

with each other in the resampling step. Such models are known as *interacting particle systems*. The law of large numbers does not apply to dependent samples, however, and proving convergence as $N \to \infty$ now becomes a problem of its own. We will prove convergence of the SIS-R algorithm in the next section.

Remark 4.3. There are many variants of the basis SIS-R algorithm, which can lead to improvements in certain settings. For example, standard Monte Carlo sampling techniques suggest a number of variations on the way sampling or resampling is performed. Another variation is to not resample in every time step, but only when the number of samples with negligible weights becomes too large (this can be computationally advantageous as resampling is expensive). When the signal to noise ratio is very high, the SIS-R algorithm can suffer from the same problem as the SIS algorithm (in this case the weights might become negligible after a single time step, in which case resamping does not help); in this case, some form of regularization might be required to make the algorithm work. A good entry point in the extensive literature on this topic is [DDG01]. In this course, we are more than happy to stick with the basic SIS-R algorithm, which is already surprisingly effective in many cases.

A numerical example

As a simple numerical illustration of the SIS-R method, let us work out a stochastic volatility model for financial time series in the spirit of example 1.12. We consider a single stock whose price we observe in discrete time intervals of length Δ. The price in the kth time step is given by

$$S_k = \exp((r - X_k^2/2)\,\Delta + X_k\,\eta_k\,\sqrt{\Delta})\,S_{k-1} \quad (k \geq 0),$$

where η_k are i.i.d. $N(0,1)$ random variables and r is the interest rate. The volatility X_k satisfies the mean-reverting linear model

$$X_k = X_{k-1} - (X_{k-1} - u)\,\Delta + \sigma\,\xi_k\,\sqrt{\Delta} \quad (k \geq 1),$$

where ξ_k are i.i.d. $N(0,1)$ and u, σ are constants. For sake of example, we have chosen the following parameters: $\Delta = 0.01$, $r = 0.1$, $u = \sigma = 0.5$, $S_{-1} = 20$, $X_0 \sim N(0.5, 0.25)$. This model is a standard hidden Markov model if we choose as our observations the log-returns $Y_k = \log(S_k/S_{k-1})$, $k \geq 0$. The SIS-R algorithm is now easily implemented using the approach outlined in remark 4.2. Indeed, sampling from $P(x, \cdot\,)$ is simply a matter of applying the recursion for X_k, while you may easily verify that we can set

$$\Upsilon(x, y) = |x|^{-1} \exp(-\{y - (r - x^2/2)\,\Delta\}^2/2x^2\Delta).$$

In figure 4.1, we have plotted a typical trajectory of this model. What is shown is the absolute volatility $|X_k|$, as well as its conditional mean and standard deviation as estimated using the SIS-R algorithm with 500 particles.

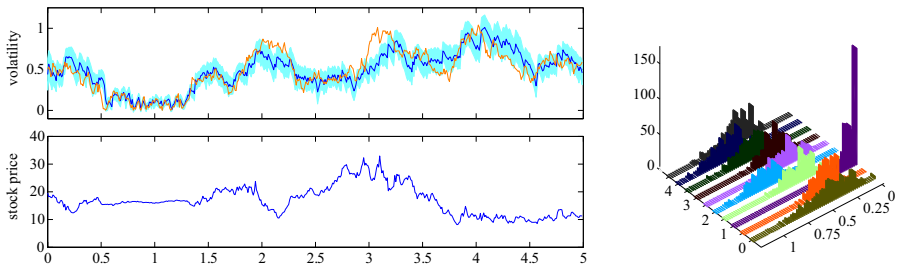

Fig. 4.1. A typical run of the stochastic volatility example. The top plot shows the true volatility (orange) and the filter conditional mean (blue) computed using the SIS-R algorithm from the stock prices, which are shown in the bottom plot. The shaded blue region is the conditional 66% confidence interval computed by the SIS-R algorithm. The plot on the right shows histograms of the SIS-R samples for ten time slices. The SIS-R algorithm was run with 500 particles.

Remark 4.4. Note that in figure 4.1, even though the filter occasionally strays a bit from the true volatility, these little errors correct themselves rather quickly. This would be true even if we had run the simulation for a much longer time interval. It is not, however, entirely obvious why this should be the case—particularly when we make approximations (such as the SIS-R algorithm used here), one might expect that such little errors would accumulate over time and eventually ruin our estimates completely on the long run. We will gain some insight into why this does not happen in the next chapter.

4.3 Convergence of SIS-R

The above discussion strongly suggests that the SIS-R algorithm is a significant improvement over the SIS algorithm. Nonetheless, we have yet to show that the SIS-R algorithm even converges to the exact filter as the number of samples increases $N \to \infty$; unlike in the SIS algorithm, this is not trivial as the SIS-R samples are not independent. The purpose of this section is to fill this gap in our discussion. To be precise, we will prove the following.

Theorem 4.5. *Suppose that the following assumption holds:*

$$\sup_{x \in E} \Upsilon(x, y_k) < \infty, \qquad k = 1, \dots, n.$$

Let $x_n^{(i)}$, $i = 1, \dots, N$ be the random samples generated by the SIS-R algorithm for the observation sequence $y_0, \dots, y_n$. Then

$$\sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \pi_n(y_0, \dots, y_n, dx) - \frac{1}{N} \sum_{i=1}^{N} f(x_n^{(i)}) \right\|_2 \le \frac{C_n}{\sqrt{N}},$$

where $\|X\|_2 = \sqrt{\mathbf{E}(X^2)}$ *and the constant* C_n *does not depend on* N *(but it does typically depend on* n *and* $y_0, \ldots, y_n$*).*

Note in particular that the rate of convergence is of order $N^{-1/2}$, which is characteristic of Monte Carlo algorithms in general. The assumption on the observation density is mild and is satisfied in most cases.

Remark 4.6. Instead of employing a Monte Carlo method, suppose that we approximate the filter by restricting computations to a fixed grid of spacing Δ. Then the approximation error would typically be of order Δ^α for some $\alpha > 0$. In particular, as the number of points in a grid of spacing Δ is of order $N \sim (1/\Delta)^p$ where p is the state space dimension, this non-random algorithm typically has an approximation error of order $N^{-\alpha/p}$. The fact that the error converges very slowly for large p is known as the *curse of dimensionality*. In contrast, in our Monte Carlo algorithm the filter is still approximated by N points, but the approximation error is of order $N^{-1/2}$ where the exponent does not depend on the state space dimension. The Monte Carlo approach is therefore often claimed to *beat* the curse of dimensionality.

However, this claim should be interpreted with a heavy dose of skepticism. Even though the exponent of the error $C_n N^{-1/2}$ does not depend on dimension, the constant C_n may well be very large in high dimensional models. Suppose that $C_n \sim e^{\beta p}$ for some $\beta > 0$; then in order to achieve a fixed approximation error ε, we would have to choose a number of samples of order $N \sim \varepsilon^{-2} e^{2\beta p}$, which rapidly becomes intractable in high dimensional models. Though it is not immediately clear how the constant C_n actually depends on dimension, numerical and some theoretical evidence strongly suggest that also Monte Carlo filter approximations perform poorly in high dimensional state spaces; see [BLB08] and the references therein.

On the other hand, unlike grid methods which can not even be implemented in practice in models whose dimension is higher than 2 or 3, Monte Carlo filtering algorithms are at least computationally tractable. In a sense they can be viewed as 'stochastic grid algorithms' where the locations of the grid points adapt automatically to the problem at hand, even if the number of points required for good approximation may be large. Presently, Monte Carlo filtering appears to be the only approach that can be applied to a general class of higher dimensional problems (in the absence of special structure; if the model is almost linear, some variant of the Kalman filter can be applied). In practice the technique usually works well in concrete problems once the number of particles and the details of the algorithm are fine tuned.

In the following, we presume that the observation sequence $y_0, \ldots, y_n$ is fixed. For notational simplicity, we will not explicitly denote the dependence of π_k on $y_0, \ldots, y_k$ (as we already did for the SIS-R samples and weights).

Let us analyze the steps within one iteration of the SIS-R algorithm. Define the SIS-R empirical measure in step k as (see algorithm 4.2 for notation)

$$\hat{\pi}_k(dx) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x_k^{(i)}}(dx).$$

The SIS-R iteration proceeds as follows:

$$\hat{\pi}_{k-1} \xrightarrow{\text{prediction}} \hat{\pi}_{k|k-1} \xrightarrow{\text{correction}} \hat{\pi}_k^0 \xrightarrow{\text{resampling}} \hat{\pi}_k,$$

where we have defined the empirical measures

$$\hat{\pi}_{k|k-1}(dx) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\tilde{x}_k^{(i)}}(dx), \qquad \hat{\pi}_k^0(dx) = \sum_{i=1}^{N} w_k^{(i)} \delta_{\tilde{x}_k^{(i)}}(dx).$$

To prove theorem 4.5, we will bound each of these steps. We will need the following elementary lemma from Monte Carlo theory.

Lemma 4.7. *Let $x^{(1)}, \ldots, x^{(N)}$ be i.i.d. samples from a (possibly random) probability distribution ν. Then*

$$\sup_{\|f\|_\infty \le 1} \left\| \int f(x)\,\nu(dx) - \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \right\|_2 \le \frac{1}{\sqrt{N}}.$$

Proof. As $x^{(i)}$, $i = 1, \ldots, N$ are independent given ν, we have

$$\begin{aligned}
&\mathbf{E}\left[\left(\int f(x)\,\nu(dx) - \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \right)^2 \,\middle|\, \nu \right] \\
&\quad = \frac{1}{N^2}\sum_{i,j=1}^{N} \mathbf{E}(f(x^{(i)})f(x^{(j)})|\nu) - \left(\int f(x)\,\nu(dx) \right)^2 \\
&\quad = \frac{1}{N}\int f(x)^2\,\nu(dx) + \left(\frac{N^2 - N}{N^2} - 1 \right) \left(\int f(x)\,\nu(dx) \right)^2 \\
&\quad = \frac{1}{N}\left(\int f(x)^2\,\nu(dx) - \left(\int f(x)\,\nu(dx) \right)^2 \right) \le \frac{\|f\|_\infty^2}{N}.
\end{aligned}$$

Taking the expectation of this expression, the claim is easily established. □

We can now proceed to the proof of theorem 4.5.

Proof (Theorem 4.5).
Step 1 (resampling error). From lemma 4.7, we find directly that

$$\sup_{\|f\|_\infty \le 1} \left\| \int f(x)\,\hat{\pi}_k(dx) - \int f(x)\,\hat{\pi}_k^0(dx) \right\|_2 \le \frac{1}{\sqrt{N}}.$$

Therefore, the triangle inequality gives

$$\sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \pi_k(dx) - \int f(x)\, \hat{\pi}_k(dx) \right\|_2 \le \frac{1}{\sqrt{N}} + \sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \pi_k(dx) - \int f(x)\, \hat{\pi}^0_k(dx) \right\|_2 .$$

Step 2 (correction error). By corollary 2.10 and algorithm 4.2

$$\pi_k(dx) = \frac{\Upsilon_k(x)\, \pi_{k|k-1}(dx)}{\int \Upsilon_k(x)\, \pi_{k|k-1}(dx)}, \qquad \hat{\pi}^0_k(dx) = \frac{\Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx)}{\int \Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx)},$$

where we have defined $\Upsilon_k(x) = \Upsilon(x, y_k)$. We now obtain some simple estimates; the following string of inequalities should speak for itself.

$$\begin{aligned}
&\left| \int f(x)\, \pi_k(dx) - \int f(x)\, \hat{\pi}^0_k(dx) \right| \\
&= \left| \frac{\int f(x)\, \Upsilon_k(x)\, \pi_{k|k-1}(dx)}{\int \Upsilon_k(x)\, \pi_{k|k-1}(dx)} - \frac{\int f(x)\, \Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx)}{\int \Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx)} \right| \\
&\le \frac{|\int f(x)\, \Upsilon_k(x)\, \pi_{k|k-1}(dx) - \int f(x)\, \Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx)|}{\int \Upsilon_k(x)\, \pi_{k|k-1}(dx)} \\
&\quad + \left| \frac{\int f(x)\, \Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx)}{\int \Upsilon_k(x)\, \pi_{k|k-1}(dx)} - \frac{\int f(x)\, \Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx)}{\int \Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx)} \right| \\
&= \frac{|\int f(x)\, \Upsilon_k(x)\, \pi_{k|k-1}(dx) - \int f(x)\, \Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx)|}{\int \Upsilon_k(x)\, \pi_{k|k-1}(dx)} \\
&\quad + \frac{|\int f(x)\, \Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx)|}{\int \Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx)} \frac{|\int \Upsilon_k(x)\, \hat{\pi}_{k|k-1}(dx) - \int \Upsilon_k(x)\, \pi_{k|k-1}(dx)|}{\int \Upsilon_k(x)\, \pi_{k|k-1}(dx)} \\
&\le \frac{\|f\|_\infty \|\Upsilon_k\|_\infty}{\int \Upsilon_k(x)\, \pi_{k|k-1}(dx)} \left| \int f_1(x)\, \pi_{k|k-1}(dx) - \int f_1(x)\, \hat{\pi}_{k|k-1}(dx) \right| \\
&\quad + \frac{\|f\|_\infty \|\Upsilon_k\|_\infty}{\int \Upsilon_k(x)\, \pi_{k|k-1}(dx)} \left| \int f_2(x)\, \hat{\pi}_{k|k-1}(dx) - \int f_2(x)\, \pi_{k|k-1}(dx) \right|,
\end{aligned}$$

where $f_1(x) = f(x)\Upsilon_k(x)/\|f\Upsilon_k\|_\infty$ and $f_2(x) = \Upsilon_k(x)/\|\Upsilon_k\|_\infty$. But note that by construction $\|f_1\|_\infty \le 1$ and $\|f_2\|_\infty \le 1$. We therefore evidently have

$$\begin{aligned}
&\sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \pi_k(dx) - \int f(x)\, \hat{\pi}^0_k(dx) \right\|_2 \\
&\le \frac{2\, \|\Upsilon_k\|_\infty}{\int \Upsilon_k(x)\, \pi_{k|k-1}(dx)} \sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \pi_{k|k-1}(dx) - \int f(x)\, \hat{\pi}_{k|k-1}(dx) \right\|_2 .
\end{aligned}$$

Step 3 (prediction error). From lemma 4.7, we find directly that

$$\sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \hat{\pi}_{k|k-1}(dx) - \int f(x)\, \hat{\pi}_{k-1} P(dx) \right\|_2 \le \frac{1}{\sqrt{N}}.$$

Therefore, the triangle inequality gives

$$\begin{aligned}\sup_{\|f\|_\infty \le 1} &\left\| \int f(x)\, \pi_{k|k-1}(dx) - \int f(x)\, \hat{\pi}_{k|k-1}(dx) \right\|_2 \\ &\le \frac{1}{\sqrt{N}} + \sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \pi_{k-1} P(dx) - \int f(x)\, \hat{\pi}_{k-1} P(dx) \right\|_2 \\ &\le \frac{1}{\sqrt{N}} + \sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \pi_{k-1}(dx) - \int f(x)\, \hat{\pi}_{k-1}(dx) \right\|_2,\end{aligned}$$

where the latter inequality holds as $\|Pf\|_\infty \le \|f\|_\infty$ for all functions f.
Step 4 (putting it all together). Collecting our estimates, we have

$$\begin{aligned}\sup_{\|f\|_\infty \le 1} &\left\| \int f(x)\, \pi_k(dx) - \int f(x)\, \hat{\pi}_k(dx) \right\|_2 \\ &\le \frac{1 + D_k}{\sqrt{N}} + D_k \sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \pi_{k-1}(dx) - \int f(x)\, \hat{\pi}_{k-1}(dx) \right\|_2,\end{aligned}$$

where we have defined

$$D_k = \frac{2\, \|\Upsilon_k\|_\infty}{\int \Upsilon_k(x)\, \pi_{k|k-1}(dx)}, \qquad k \ge 1.$$

Iterating this bound, we obtain

$$\sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \pi_n(dx) - \int f(x)\, \hat{\pi}_n(dx) \right\|_2 \le \frac{1}{\sqrt{N}} \sum_{k=0}^{n} (1 + D_k) \prod_{\ell=k+1}^{n} D_\ell,$$

provided that we can obtain a bound on the initial step of the form

$$\sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \pi_0(dx) - \int f(x)\, \hat{\pi}_0(dx) \right\|_2 \le \frac{1 + D_0}{\sqrt{N}}.$$

But it is easily established (problem 4.2), following the same approach as our previous estimates, that this is the case with the constant

$$D_0 = \frac{2\, \|\Upsilon_0\|_\infty}{\int \Upsilon_0(x)\, \mu(dx)},$$

where μ is the initial measure. The proof is complete. □

Problems

4.1. Consider the observation model $Y_k = H(X_k) + \eta_k$ on the observation state space $F = \mathbb{R}^p$, where $H : E \to \mathbb{R}^p$ is measurable and η_k, $k \ge 0$ are i.i.d.

random variables whose law possesses a positive density $f_\eta : \mathbb{R}^p \to]0,\infty[$ with respect to the Lebesgue measure on $\mathbb{R}^p$. Show that this observation model is nondegenerate in the sense of definition 1.9, and argue that we may choose $\Upsilon(x,y) = f_\eta(y - H(x))$ in the filtering/smoothing recursions even though the Lebesgue measure is not a probability measure.

4.2. Prove the following missing estimate in the proof of theorem 4.5:

$$\sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \pi_0(dx) - \int f(x)\, \hat{\pi}_0(dx) \right\|_2 \le \frac{1}{\sqrt{N}} \left(1 + \frac{2\, \|\Upsilon_0\|_\infty}{\int \Upsilon_0(x)\, \mu(dx)} \right),$$

where the notation is the same as in the proof of theorem 4.5.

4.3. A Performance Comparison
In this problem, we will investigate numerically how the SIS and SIS-R algorithms compare to the exact filter. Consider the linear-Gaussian hidden Markov model with real-valued signal and observations

$$X_k = 0.9\, X_{k-1} + \xi_k, \qquad Y_k = X_k + \eta_k,$$

where ξ_k and η_k are $N(0,1)$. Compute the conditional mean and variance
(a) using the exact filtering equation (problem 2.5);
(b) using the SIS algorithm; and
(c) using the SIS-R algorithm.
How do the approximation errors of the SIS and SIS-R algorithms behave as a function of time and of particle number? Experiment with various particle numbers and time horizons, and draw your conclusions.

4.4. Monte Carlo Path Estimation
Our particle filters only approximate the filtering distribution π_k. There are various applications where one must approximate the smoother as well (e.g., to implement the EM algorithm in chapter 6). More generally, one can try to approximate the entire conditional path distribution $P_{X_0,\ldots,X_n|Y_0,\ldots,Y_n}$:

$$\int f(x_0,\ldots,x_n)\, P_{X_0,\ldots,X_n|Y_0,\ldots,Y_n}(dx_0,\ldots,dx_n) \approx \sum_{i=1}^N w_n^{(i)}\, f(x_0^{(n,i)},\ldots,x_n^{(n,i)})$$

for suitable weights $w_n^{(i)}$ and paths $x_k^{(n,i)}$ (note that we have suppressed the dependence of $P_{X_0,\ldots,X_n|Y_0,\ldots,Y_n}$ on $y_0,\ldots,y_n$ for notational simplicity). The smoothing distributions can be obtained as marginals of this distribution.
(a) Modify the SIS algorithm to compute the path distributions.
(b) Prove the following recursion for the exact path distributions:

$$\int f(x_0,\ldots,x_n)\, P_{X_0,\ldots,X_n|Y_0,\ldots,Y_n}(dx_0,\ldots,dx_n) =$$
$$\frac{\int f(x_{0,\ldots,n})\, \Upsilon(x_n,y_n)\, P(x_{n-1},dx_n)\, P_{X_{0,\ldots,n-1}|Y_{0,\ldots,n-1}}(dx_0,\ldots,dx_{n-1})}{\int \Upsilon(x_n,y_n)\, P(x_{n-1},dx_n)\, P_{X_{0,\ldots,n-1}|Y_{0,\ldots,n-1}}(dx_0,\ldots,dx_{n-1})}$$

for all bounded measurable functions f.
(c) Using this identity, propose a variant of the SIS-R algorithm to approximate the conditional path distributions $P_{X_0,\ldots,X_n|Y_0,\ldots,Y_n}$.
(d) Implement your proposed SIS and SIS-R smoothers together with the exact smoother (problem 2.5) for the linear-Gaussian example in the previous problem and investigate their performance numerically.

4.5. Credit Derivative Pricing
Suppose that the value of a firm is modeled by the recursion

$$X_k = (1 + \xi_k)\, X_{k-1}, \qquad \xi_k \sim N(\mu, \sigma^2),\ X_0 > 0$$

(typically $\mu, \sigma^2 \ll 1$). We are also given a threshold $K > 0$ such that the firm goes bankrupt if its value drops below the threshold. The bankruptcy time of the firm is given by $\tau = \min\{k \geq 0 : X_k \leq K\}$.

To finance its operations, the firm issues a zero-coupon bond with maturity N. This means that the firm agrees to pay the bond holder \$1 at time N. However, if the firm goes bankrupt before time N, the bond holder will not get paid. The payoff of the bond is therefore $I_{\tau>N}$. Our question is how to price this bond: if a holder of such a bond wishes to sell the bond on the market at time $k < N$, what price should he ask?

In practice, the value of the firm is not directly observable to investors. Instead, an investor must rely on profit reports and other news issued periodically by the firm in order to form an estimate of the firm's actual value. This news is typically not entirely accurate. In a simple model, we could assume that the information obtained by the investor at time k is of the form

$$Y_k = X_k + \eta_k, \qquad \eta_k \sim N(0, \bar{\sigma}^2),$$

i.e., the investor knows the firm's value up to some 'noise' in the reporting. It can be shown [DL01] that the fair market price S_k of the bond at time $k < N$ is given by (assuming that we are modeling under a risk-neutral measure)

$$S_k = I_{\tau>k}\, r^{k-N}\, \mathbf{P}(\tau > N | \tau > k, Y_0, \ldots, Y_k),$$

where r is the single period risk-free interest rate (i.e., on your bank account).
(a) Develop a SIS-R type algorithm to compute the bond price at time k.
Hint: add a 'coffin' point to the signal and observation state spaces $E = F = \mathbb{R} \cup \{\partial\}$, and construct new signal and observation processes

$$\tilde{X}_k = \begin{cases} X_k & \text{for } k < \tau, \\ \partial & \text{otherwise,} \end{cases} \qquad \tilde{Y}_k = \begin{cases} Y_k & \text{for } k < \tau, \\ \partial & \text{otherwise.} \end{cases}$$

Show that $(\tilde{X}_k, \tilde{Y}_k)_{k\geq 0}$ defines a hidden Markov model and express the price in terms of a prediction problem for this extended hidden Markov model.
(b) Write a computer program that implements your algorithm and plot the bond price as a function of time and maturity. For your simulation, you may choose model parameters that seem reasonable to you.

Remark 4.8. There is a close connection between credit risk models with incomplete information and nonlinear filtering. See [DL01, CC08] and the contribution of R. Frey and W. Runggaldier in [CR09] for further details.

Notes

Variants of the SIS algorithm have been known for a long time [Han70]. As Monte Carlo algorithms are computationally expensive, there appears to have been little interest in such methods until major improvements in computer technology made them practically applicable. The idea of adding a resampling step to the SIS algorithm is due to Gordon, Salmond and Smith [GSS93], who referred to the algorithm as the 'bootstrap algorithm' rather than SIS-R. The first convergence proof of SIS-R is due to Del Moral [Del98a].

Much information on Monte Carlo particle filters can be found in the collection [DDG01]. The convergence proof given here was inspired by the treatment in Crisan and Doucet [CD02]. Some mathematical analysis of the behavior of particle filters in high dimensional state spaces can be found in the recent work of Bickel, Li and Bengtsson [BLB08]. Various approaches to Monte Carlo smoothing can be found in [CMR05] and in [DGA00].

5

Filter Stability and Uniform Convergence

5.1 Orientation

In the previous chapter, we showed that the SIS-R algorithm converges to the exact filter as we let the number of particles N go to infinity:

$$\sup_{\|f\|_\infty \leq 1} \left\| \int f(x)\, \pi_n(y_0, \ldots, y_n, dx) - \frac{1}{N} \sum_{i=1}^{N} f(x_n^{(i)}) \right\|_2 \leq \frac{C_n}{\sqrt{N}}.$$

This means that for a fixed time n, we can obtain an arbitrarily good approximation to the exact filter by choosing N sufficiently large.

However, in many applications one might not necessarily be interested in the filter at any particular fixed time n, but we need to have good estimates available at an arbitrary time. For example, in target tracking problems, the aim is to continually track the location of the target. We can do this, for example, by running the SIS-R algorithm where we compute the approximate filter $\hat{\pi}_k$ in every time step k. Our error bound, however, does not guarantee that this approach will be successful. In particular, if the constants C_k grow rapidly in time, then the SIS-R algorithm may degenerate very rapidly so that we 'lose lock' on the target. A closer look at the proof of theorem 4.5 should make us particularly worried: the constants C_k obtained in the proof are bounded from below by 2^k, which suggests behavior that is entirely unacceptable in tracking applications. Instead, we need to have good performance at an *arbitrary* time for sufficiently large N:

$$\sup_{n \geq 0} \sup_{\|f\|_\infty \leq 1} \left\| \int f(x)\, \pi_n(y_0, \ldots, y_n, dx) - \frac{1}{N} \sum_{i=1}^{N} f(x_n^{(i)}) \right\|_2 \leq \frac{C}{\sqrt{N}}.$$

In other words, would like to show that $\hat{\pi}_k$ converges to π_k *uniformly in time* as $N \to \infty$. This certainly does not follow from theorem 4.5; in fact, one might be led to think that uniform convergence does not hold. It is therefore quite surprising that in many cases (see, e.g., the numerical example in the previous

chapter), numerical simulations strongly suggest that the SIS-R algorithm does converge uniformly in time. That the algorithm works so much better in practice than we would expect is exciting news, but it also means that we are missing something fundamental in our analysis. One of the goals of the present chapter is to gain some understanding of this phenomenon.

The reason that the constants C_k grow exponentially is that the error bound in the proof of theorem 4.5 was obtained by bounding the error incurred in one step of the algorithm. When this single time step bound is iterated, we obtain a bound on the error incurred in k steps of the algorithm. In this approach, however, the error incurred in each step accumulates over time, which causes the constant C_k to grow. In order to prove uniform convergence, we have to show that this accumulation of errors does not actually happen.

Filter stability

The new ingredient that is going to help us is a separate topic in itself: the *stability* property of the filter. Let us forget for the moment about the approximate filter, and consider the exact filtering recursion of corollary 2.10. In order to implement the filter, we need to know the initial measure μ and the transition and observation kernels P and Φ. Unlike the kernels, however, which can be estimated very efficiently using the statistical techniques in chapter 6 (provided that we are given a sufficiently long observation time series), the initial measure is often difficult to estimate. For example, in many cases the noise driving the signal dynamics will cause the signal itself to 'forget' its initial condition (the signal is *ergodic*), so that even an infinite time series of observations can not be used to estimate the initial measure exactly.

One might worry that our inability to estimate the initial measure would mean that filtering becomes useless in real-world problems: using the wrong initial measure in the filtering recursion could have disastrous results. Fortunately, it turns out that this is much less of a problem that one might think. Rather than hurt us, ergodicity of the signal actually helps us here: if the signal itself already forgets its initial condition, then it seems highly likely that this is also true for the filtering distributions. On the other hand, an additional effect helps us in the filtering problem even when the signal is not ergodic: as we are obtaining more and more information from the observations as time increases, it seems likely that this information will eventually supersede the initial measure, which represents our best guess of the location of the signal before any observations were actually made.

For these reasons, it is often the case that as time k increases the filter depends less and less on the choice of the initial measure. Mathematically, what happens is that if one performs the filter recursion (corollary 2.10) with the same observations, transition and observation kernels, but with two different initial measures, then the two resulting filters converge toward one another as $k \to \infty$. In this case, the filter is said to be *stable*.

Stability of the filter means in particular that we may use the 'wrong' initial measure without obtaining unreliable estimates—an important practical issue. However, beside its intrinsic interest, it turns out that the stability property of the filter plays a central role in statistical inference (chapter 7) and in uniform convergence of filter approximations (this chapter). For these reasons, it is of significant interest to characterize the stability properties of filtering models. We will develop one particular approach in the next section; further comments can be found in the notes at the end of the chapter.

Uniform convergence and stability

How does filter stability help us to establish uniform convergence of the SIS-R algorithm? Intuitively, the stability property implies that the filter is insensitive to approximation errors made a long time ago. For this reason, the approximation error can not accumulate: though we make an approximation error in every iteration of the algorithm, the errors made in the previous iterations are progressively 'forgotten' by the filter as time increases.

Let us show how to make this idea precise. We denote by F_k the kth iteration of the exact filtering recursion:

$$\mathsf{F}_k\nu(A) = \frac{\int I_A(x)\,\Upsilon(x,y_k)\,P(x',dx)\,\nu(dx')}{\int \Upsilon(x,y_k)\,P(x',dx)\,\nu(dx')}.$$

In particular, note that $\pi_k = \mathsf{F}_k\pi_{k-1}$ by construction. We can now split the discrepancy between the exact and approximate filter at time k into two parts:

$$\pi_k - \hat{\pi}_k = \overbrace{\mathsf{F}_k\pi_{k-1} - \mathsf{F}_k\hat{\pi}_{k-1}}^{\text{propagation of error}} + \overbrace{\mathsf{F}_k\hat{\pi}_{k-1} - \hat{\pi}_k}^{\text{one step error}}.$$

The first term represents the contribution to the error at time k from the error incurred in the previous iterations, while the second term represents the error incurred in time step k by applying the SIS-R algorithm rather than the exact filter. Splitting up the first term in exactly the same manner, we can write the error at time k as a sum of propagated one step errors:

$$\pi_k - \hat{\pi}_k = \sum_{\ell=0}^{k-1}(\mathsf{F}_k\cdots\mathsf{F}_{\ell+1}\,\mathsf{F}_\ell\hat{\pi}_{\ell-1} - \mathsf{F}_k\cdots\mathsf{F}_{\ell+1}\,\hat{\pi}_\ell) + \mathsf{F}_k\hat{\pi}_{k-1} - \hat{\pi}_k$$

(where $\mathsf{F}_0\hat{\pi}_{-1} = \pi_0$). Now suppose we can establish an estimate of the form

$$\|\mathsf{F}_k\cdots\mathsf{F}_{\ell+1}\nu - \mathsf{F}_k\cdots\mathsf{F}_{\ell+1}\nu'\| \le C_0\,e^{-\gamma(k-\ell)}\,\|\nu-\nu'\|$$

for some $C_0, \gamma > 0$, i.e., we suppose that the filter is *exponentially stable* (we will work below with the norm $\|\nu-\nu'\| = \sup\|\int f\,d\nu - \int f\,d\nu'\|_2$). Then

$$\|\pi_k - \hat{\pi}_k\| \le C_0\sum_{\ell=0}^{k} e^{-\gamma(k-\ell)}\,\|\mathsf{F}_\ell\hat{\pi}_{\ell-1} - \hat{\pi}_\ell\|.$$

Evidently, exponential stability of the filter causes the errors incurred in each time step to be suppressed by a geometric factor. Therefore if the one step errors are uniformly bounded, the total error can now be estimated uniformly in time—which is precisely what we set out to do!

In the remainder of the chapter we will make these ideas precise under certain (strong) technical assumptions. In section 5.2, we first prove exponential stability of the filter. Then, in section 5.3, we develop the above argument in detail and prove uniform convergence of the SIS-R algorithm.

Remark 5.1. The purpose of this chapter is to give a flavor of the stability and uniform approximation properties of filtering problems; an extensive treatment is outside our scope. We therefore develop our results *in the simplest possible setting.* The assumptions that we must impose to make this work are very strong, and there are many applications in which they are not satisfied (some further discussion can be found below). Proving either filter stability or uniform convergence in a general setting is a challenging problem, and to date many open problems remain in this direction.

5.2 Filter Stability: A Contraction Estimate

In this section we are going to prove exponential filter stability under a certain ergodicity assumption on the signal process, called the *mixing condition.* This condition causes the signal itself to forget its initial measure at an geometric rate, and we will show that the filter inherits this property. Note that the second effect described above—that the information gain from the observations can lead to filter stability—does not enter in our analysis. The stability rate which we will prove is an upper bound obtained from the ergodicity of the signal only, and in practice the filter may converge much faster.

Kernels and contraction

Before we can prove filter stability, we need to introduce a simple idea from the ergodic theory of Markov processes.

Lemma 5.2 (Contraction). *Let ν, ν' be (possibly random) probability measures on $(E, \mathcal{E})$ and let $K : E \times \mathcal{E} \to [0,1]$ be a transition kernel. Suppose that the following* minorization condition *holds: there is a fixed probability measure ρ and $0 < \varepsilon < 1$ such that $K(x, A) \geq \varepsilon\, \rho(A)$ for all $x \in E$, $A \in \mathcal{E}$. Then*

$$\sup_{\|f\|_\infty \leq 1} \left\| \int f(x)\, \nu K(dx) - \int f(x)\, \nu' K(dx) \right\|_2 \leq (1-\varepsilon) \sup_{\|f\|_\infty \leq 1} \left\| \int f(x)\, \nu(dx) - \int f(x)\, \nu'(dx) \right\|_2 .$$

Proof. Define $\tilde{K}(x, A) = (1-\varepsilon)^{-1}\{K(x, A) - \varepsilon\,\rho(A)\}$ for all $x \in E$, $A \in \mathcal{E}$. Then the minorization condition guarantees that $\tilde{K}$ is a transition kernel, and we clearly have $\nu K - \nu' K = (1-\varepsilon)\,\{\nu\tilde{K} - \nu'\tilde{K}\}$. Therefore

$$\sup_{\|f\|_\infty \le 1} \left\| \int f(x)\,\nu K(dx) - \int f(x)\,\nu' K(dx) \right\|_2 = (1-\varepsilon) \sup_{\|f\|_\infty \le 1} \left\| \int \tilde{K}f(x)\,\nu(dx) - \int \tilde{K}f(x)\,\nu'(dx) \right\|_2 .$$

The result follows immediately as $\|\tilde{K}f\|_\infty \le \|f\|_\infty$ for any function f. □

Lemma 5.2 shows that under the minorization condition, the map $\nu \mapsto \nu K$ is a strict contraction. This has an immediate application to the ergodicity of Markov processes. Let $(X_k)_{k\ge 0}$ be a Markov process with transition kernel P and initial measure μ, and let $(X'_k)_{k\ge 0}$ be a Markov process with the same transition kernel P but with different initial measure μ'. Note that

$$\mathbf{E}(f(X_k)) = \int f(x)\,\mu P^k(dx), \qquad \mathbf{E}(f(X'_k)) = \int f(x)\,\mu' P^k(dx).$$

Thus if P satisfies the minorization condition, lemma 5.2 shows that

$$\sup_{\|f\|_\infty \le 1} |\mathbf{E}(f(X_k)) - \mathbf{E}(f(X'_k))| \le (1-\varepsilon)^k \sup_{\|f\|_\infty \le 1} |\mathbf{E}(f(X_0)) - \mathbf{E}(f(X'_0))|.$$

In particular, we find that the Markov process is *geometrically ergodic*: the difference between the laws of the Markov process started at two different initial measures decays geometrically in time.

The minorization condition in lemma 5.2 is a special case of the well known *Doeblin condition* for ergodicity. It has an interesting probabilistic interpretation. If the transition kernel P satisfies the condition in lemma 5.2, then we can write $P(x, A) = \varepsilon\,\rho(A) + (1-\varepsilon)\,P'(x, A)$, where P' is another transition kernel. The corresponding Markov process can then be generated as follows:

1. In every time step, flip a coin with Prob(heads) $= \varepsilon$.
2. If the coin comes up tails, choose the next time step according to the transition probability P'.
3. If the coin comes up heads, choose the next time step independently from the present location by sampling from the probability distribution ρ.

Once the coin comes up heads, the Markov process 'resets' to the fixed distribution ρ and the initial condition is forgotten. This idea can be used to provide a probabilistic proof of geometric ergodicity; see problem 5.1.

Exponential stability of the filter

We now consider our usual hidden Markov model $(X_k, Y_k)_{k\ge 0}$. To prove filter stability, we would like to apply the above contraction technique to the filtering

recursion, i.e., we would like to show that $\|\mathsf{F}_k\nu - \mathsf{F}_k\nu'\| \leq (1-\varepsilon)\,\|\nu - \nu'\|$. However, we immediately run into a problem: the filter time step F_k can not be expressed as a kernel, as $\mathsf{F}_k\nu$ is a nonlinear function of ν.

On the other hand, each iteration of the *smoothing* recursion is linear. In particular, define for every $k \leq n$ the transition kernel $K_{k|n}$ as

$$K_{k|n}(x,A) = \frac{\int I_A(x')\,\beta_{k|n}(x', y_{k+1}, \ldots, y_n)\,\Upsilon(x', y_k)\,P(x, dx')}{\int \beta_{k|n}(x', y_{k+1}, \ldots, y_n)\,\Upsilon(x', y_k)\,P(x, dx')}.$$

From theorem 2.12 (see also problem 2.2), we can read off that

$$\pi_{k+1|n}(y_0, \ldots, y_n, A) = \int I_A(x')\,K_{k|n}(x, dx')\,\pi_{k|n}(y_0, \ldots, y_n, dx),$$

i.e., $\pi_{k|n} = \pi_{k-1|n}K_{k|n}$. That the smoothing recursion can be expressed in this form is no coincidence, see problem 2.4. This observation turns out to be the key to our problem: lemma 5.2 can be applied to the kernels $K_{k|n}$.

Lemma 5.3 (Minorization of the smoother). *Suppose that the transition kernel P satisfies the following* mixing condition*: there exists a probability measure ρ and a constant $0 < \varepsilon < 1$ such that*

$$\varepsilon\,\rho(A) \leq P(x,A) \leq \varepsilon^{-1}\,\rho(A) \qquad \text{for all } x \in E,\ A \in \mathcal{E}.$$

Then for every $k \leq n$, the smoothing kernel $K_{k|n}$ satisfies the minorization condition $K_{k|n}(x,A) \geq \varepsilon^2\,\rho_{k|n}(A)$ for some probability measure $\rho_{k|n}$.

Proof. By the mixing condition, we have

$$K_{k|n}(x,A) \geq \varepsilon^2\,\frac{\int I_A(x)\,\beta_{k|n}(x, y_{k+1}, \ldots, y_n)\,\Upsilon(x, y_k)\,\rho(dx)}{\int \beta_{k|n}(x, y_{k+1}, \ldots, y_n)\,\Upsilon(x, y_k)\,\rho(dx)} = \varepsilon^2\,\rho_{k|n}(A).$$

The proof is complete. □

We can now prove stability of the filter.

Theorem 5.4 (Filter stability). *Suppose that the transition kernel P satisfies the mixing condition in lemma 5.3. Then for any two (possibly random) probability measures ν and ν' on E, we have for $k \geq \ell$*

$$\sup_{\|f\|_\infty \leq 1} \left\| \int f(x)\,\mathsf{F}_k \cdots \mathsf{F}_{\ell+1}\nu(dx) - \int f(x)\,\mathsf{F}_k \cdots \mathsf{F}_{\ell+1}\nu'(dx) \right\|_2$$
$$\leq \varepsilon^{-2}\,(1-\varepsilon^2)^{k-\ell} \sup_{\|f\|_\infty \leq 1} \left\| \int f(x)\,\nu(dx) - \int f(x)\,\nu'(dx) \right\|_2.$$

Proof. From theorem 2.12, we can read off that

$$\mathsf{F}_k \cdots \mathsf{F}_{\ell+1}\nu = \nu_{\ell|k} K_{\ell+1|k} \cdots K_{k|k}$$

for any probability measure ν, where we have defined

$$\nu_{\ell|k}(A) = \frac{\int I_A(x)\, \beta_{\ell|k}(x, y_{\ell+1}, \ldots, y_k)\, \nu(dx)}{\int \beta_{\ell|k}(x, y_{\ell+1}, \ldots, y_k)\, \nu(dx)}.$$

Therefore, by lemmas 5.2 and 5.3, we have

$$\begin{aligned}
&\sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \mathsf{F}_k \cdots \mathsf{F}_{\ell+1}\nu(dx) - \int f(x)\, \mathsf{F}_k \cdots \mathsf{F}_{\ell+1}\nu'(dx) \right\|_2 \\
&\quad \le (1-\varepsilon^2)^{k-\ell} \sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \nu_{\ell|k}(dx) - \int f(x)\, \nu'_{\ell|k}(dx) \right\|_2 \\
&\quad \le (1-\varepsilon^2)^{k-\ell}\, \frac{\sup_{x\in E} \beta_{\ell|k}(x)}{\inf_{x\in E} \beta_{\ell|k}(x)} \sup_{\|f\|_\infty \le 1} \left\| \int f(x)\, \nu(dx) - \int f(x)\, \nu'(dx) \right\|_2,
\end{aligned}$$

where the last estimate was obtained in the same manner as step 2 in the proof of theorem 4.5. But by the mixing condition we can bound $\beta_{\ell|k}$ above and below as $\varepsilon\, C_0 \le \beta_{\ell|k}(x, y_{\ell+1}, \ldots, y_k) \le \varepsilon^{-1}\, C_0$, where

$$C_0 = \int \Upsilon(x_{\ell+1}, y_{\ell+1}) \cdots \Upsilon(x_k, y_k)\, P(x_{k-1}, dx_k) \cdots P(x_{\ell+1}, dx_{\ell+2})\, \rho(dx_{\ell+1})$$

(cf. definition 2.11). The proof is easily completed. □

5.3 Uniform Convergence of SIS-R

In this section, we will complete our story by proving that the SIS-R algorithm converges *uniformly* to the exact filter as the number of particles increases $N \to \infty$. We will do this under the following assumption.

Assumption 5.5 (Mixing) *The transition kernel P is mixing, i.e., there exists a probability measure ρ and a constant $0 < \varepsilon < 1$ such that*

$$\varepsilon\, \rho(A) \le P(x, A) \le \varepsilon^{-1}\, \rho(A) \qquad \text{for all } x \in E,\ A \in \mathcal{E}.$$

Moreover, the observation density Υ is bounded from above and below, i.e., there is a constant $0 < \kappa < 1$ such that

$$\kappa \le \Upsilon(x, y) \le \kappa^{-1} \qquad \text{for all } x \in E,\ y \in F.$$

Note that the condition on the observation density is very similar to the mixing condition on the signal transition kernel when it is expressed in terms

of the observation kernel Φ. Some comments about this assumption, which is rather strong, can be found at the end of the section.

As you might expect, we have already done most of the work to complete the proof of uniform convergence: filter stability has been established, and we already know how to bound the one step errors as in the proof of theorem 4.5.

Theorem 5.6. *Suppose assumption 5.5 holds. Let $x_n^{(i)}$ $(i = 1, \dots, N,\ n \geq 0)$ be generated by the SIS-R algorithm for the observations $(y_k)_{k\geq 0}$. Then*

$$\sup_{n\geq 0} \sup_{\|f\|_\infty \leq 1} \left\| \int f(x)\, \pi_n(y_0, \dots, y_n, dx) - \frac{1}{N} \sum_{i=1}^{N} f(x_n^{(i)}) \right\|_2 \leq \frac{C}{\sqrt{N}},$$

where the constant C depends neither on N nor on $(y_k)_{k\geq 0}$.

Proof. As noted in the introduction, we may write

$$\pi_k - \hat{\pi}_k = \sum_{\ell=0}^{k-1} (\mathsf{F}_k \cdots \mathsf{F}_{\ell+1}\, \mathsf{F}_\ell \hat{\pi}_{\ell-1} - \mathsf{F}_k \cdots \mathsf{F}_{\ell+1}\, \hat{\pi}_\ell) + \mathsf{F}_k \hat{\pi}_{k-1} - \hat{\pi}_k$$

(where we use the notation $\mathsf{F}_0 \hat{\pi}_{-1} = \pi_0$). Therefore

$$\begin{aligned} &\sup_{\|f\|_\infty \leq 1} \left\| \int f(x)\, \pi_k(dx) - \int f(x)\, \hat{\pi}_k(dx) \right\|_2 \\ &\quad \leq \sum_{\ell=0}^{k} \varepsilon^{-2} (1-\varepsilon^2)^{k-\ell} \sup_{\|f\|_\infty \leq 1} \left\| \int f(x)\, \mathsf{F}_\ell \hat{\pi}_{\ell-1}(dx) - \int f(x)\, \hat{\pi}_\ell(dx) \right\|_2, \end{aligned}$$

where we have used theorem 5.4. But following exactly the same steps as in the proof of theorem 4.5, we find that

$$\sup_{\|f\|_\infty \leq 1} \left\| \int f(x)\, \mathsf{F}_\ell \hat{\pi}_{\ell-1}(dx) - \int f(x)\, \hat{\pi}_\ell(dx) \right\|_2 \leq \frac{1 + 2\kappa^{-2}}{\sqrt{N}},$$

where we have filled in the bounds on Υ in assumption 5.5. This gives

$$\sup_{\|f\|_\infty \leq 1} \left\| \int f(x)\, \pi_k(dx) - \int f(x)\, \hat{\pi}_k(dx) \right\|_2 \leq \frac{1 + 2\kappa^{-2}}{\sqrt{N}} \frac{1 - (1-\varepsilon^2)^{1+k}}{\varepsilon^4}.$$

We now complete the proof by taking the supremum over k. □

Some comments on Assumption 5.5

Assumption 5.5 is quite restrictive in practice, particularly the lower bounds on P and Υ (the upper bounds are usually not difficult to satisfy). We have already seen that the lower bound in the mixing condition implies that the signal process can be generated by a procedure which, in each time step, resets

the process with probability ε by drawing from a fixed probability distribution ρ. Similarly, it is easy to see that the same interpretation holds for the lower bound on the observation density: in each time step, the observation is drawn with probability κ from the reference measure φ, i.e., independently from the signal, and with probability $1-\kappa$ from the shifted observation kernel $\Phi'(x,dy) = (1-\kappa)^{-1}\{\Phi(x,dy) - \kappa\,\varphi(dy)\}$.

In both cases, the conclusion is evident: assumption 5.5 implies that the signal dynamics and the observations are very noisy. However, even this statement should be interpreted with some care. When the signal state space E is compact, for example, a signal that satisfies the mixing condition can be obtained by discretizing in time a uniformly elliptic diffusion (see [AZ97]). This conforms to the intuition of 'noisy dynamics'. However, when E is noncompact even uniform ellipticity does not suffice. Some intuition can be obtained by considering a simple example (problem 5.3); it appears that in order to satisfy assumption 5.5 in a noncompact setting, one typically needs noise with heavy tails. Many (if not most) reasonable models, both in the compact and in the noncompact setting, do not satisfy the required conditions.

This certainly does not mean that the phenomena introduced in this chapter do not appear in more general models. In fact, both filter stability and uniform approximation is observed numerically in a wide variety of models which are not even close to satisfying assumption 5.5, and various mathematical approaches have been introduced to investigate these problems. There is an important distinction with our results, however. Note that the bounds in theorems 5.4 and 5.6 do not depend on the observation path $(y_k)_{k\geq 0}$: under the assumption 5.5 we obtain stability and approximation results *uniformly in the observations*. With a rare exception, this is no longer true when assumption 5.5 is not satisfied. The lack of uniformity brings with it formidable technical complications, which are beyond the scope of this course.

Problems

5.1. Geometric Ergodicity and Coupling
Let K be a transition kernel on $(E,\mathcal{E})$ such that the minorization condition holds: $K(x,A) \geq \varepsilon\,\rho(A)$ for all $A \in \mathcal{E}$, where $0 < \varepsilon < 1$ and ρ is some probability measure. We are going to give a probabilistic proof of geometric ergodicity (see section 5.2) of the Markov chain with transition kernel K.
(a) Show that $K'(x,A) = (1-\varepsilon)^{-1}\{K(x,A) - \varepsilon\,\rho(A)\}$ is a transition kernel.
(b) Let $(X_k, \tilde{X}_k, \xi_k)_{k\geq 0}$ be a sequence of random variables on some underlying probability space such that the following hold:

1. ξ_k are i.i.d. with $\mathbf{P}(\xi_k = 0) = \varepsilon$ and $\mathbf{P}(\xi_k = 1) = 1-\varepsilon$.
2. X_k is a Markov chain with transition kernel K and initial measure μ.
3. $\tilde{X}_k$ is a Markov chain with transition kernel K' and initial measure $\tilde{\mu}$.
4. $(\xi_k)_{k\geq 0}$, $(X_k)_{k\geq 0}$ and $(\tilde{X}_k)_{k\geq 0}$ are independent of each other.

Now define the following sequence of random variables:

$$Z_k = \begin{cases} \tilde{X}_k & \text{if } \xi_\ell = 1 \text{ for all } \ell \le k; \\ X_k & \text{otherwise.} \end{cases}$$

Show that Z_k is Markov with transition kernel K and initial measure $\tilde{\mu}$.
(c) Show that there exists a random time $\tau < \infty$ a.s. such that $\mathbf{P}(X_k = Z_k \text{ for all } k \ge \tau) = 1$. The random variable τ is called the *coupling time* and the Markov chains X_k and Z_k are said to be *(successfully) coupled*.
(d) Show that the following *coupling inequality* holds:

$$\sup_{\|f\|_\infty \le 1} |\mathbf{E}(f(X_k)) - \mathbf{E}(f(Z_k))| \le 2\,\mathbf{P}(X_k \ne Z_k) \le 2\,\mathbf{P}(k < \tau).$$

Now use this estimate to conclude geometric ergodicity of our Markov chain.

5.2. A Weaker Mixing Condition
(a) Suppose that in assumption 5.5 the mixing condition on the transition kernel is replaced by: there exists an $m \in \mathbb{N}$ such that

$$\varepsilon\,\rho(A) \le P^m(x, A) \le \varepsilon^{-1}\,\rho(A) \qquad \text{for all } x \in E,\ A \in \mathcal{E}.$$

Show that theorem 5.6 still holds under this weaker condition. (Hint: you can no longer show that $K_{\ell+1|n}$ satisfies the minorization condition; however, you can establish minorization for kernels of the form $K_{\ell+1|n} \cdots K_{\ell+m|n}$.)
(b) Suppose that the signal and observation state spaces are both finite. Use the technique in (a) to prove that the filter is exponentially stable whenever the signal is an ergodic Markov chain and the observations satisfy the nondegeneracy condition (definition 1.9). Hint: when a finite state Markov chain is ergodic, there is an integer $k > 0$ such that $(\boldsymbol{P}^k)_{ij} > 0$ for all i, j.

5.3. Mixing Is Hard To Do (in a noncompact space)
Consider a hidden Markov model on $E \times F = \mathbb{R} \times \mathbb{R}$ where

$$X_k = F(X_{k-1}) + \xi_k,$$

where $F : \mathbb{R} \to \mathbb{R}$ is a bounded function. Show that the corresponding transition kernel does not satisfy the mixing condition if ξ_k are i.i.d. $N(0, 1)$, unless $F(x)$ is independent of x. On the other hand, show that the mixing condition is satisfied if ξ_k are i.i.d. exponentially distributed $\xi_k \sim \frac{1}{2}e^{-|x|}dx$. Draw the corresponding conclusions for the existence of upper and lower bounds for the observation density when the observation model is of the form

$$Y_k = H(X_k) + \eta_k,$$

where $H : \mathbb{R} \to \mathbb{R}$ is bounded and η_k are i.i.d. Gaussian or exponential.

5.4. Observability ([CL06])
In the chapter, we have shown that filter stability can be inherited from ergodicity of the signal—in words, if the signal forgets its initial condition, then so does the filter. However, one might expect that the filter can be stable even when the signal is not ergodic. After all, if the observations are 'good enough' one would expect that the information obtained from the observations eventually obsoletes the information contained in the initial measure. In this problem, we will develop a simple result along these lines.
(a) Suppose $\mathbf{P}^\mu$ is the law of a hidden Markov model with transition and observation kernels P and Φ and initial measure μ. Denote by $\mathbf{P}^\nu$ the law with the same kernels but different initial measure ν. Prove that

$$\frac{d\mathbf{P}^\mu}{d\mathbf{P}^\nu} = \frac{d\mu}{d\nu}(X_0) \qquad \text{whenever } \mu \ll \nu.$$

(b) Suppose that $\mu \ll \nu$. Prove that for all bounded measurable f

$$\mathbf{E}^\nu\Big(\frac{d\mu}{d\nu}(X_0)\Big|Y_0,\ldots,Y_k\Big)\,\mathbf{E}^\mu(f(Y_{k+1})|Y_0,\ldots,Y_k)$$
$$= \mathbf{E}^\nu\Big(\mathbf{E}^\nu\Big(\frac{d\mu}{d\nu}(X_0)\Big|Y_0,\ldots,Y_{k+1}\Big)\,f(Y_{k+1})\Big|Y_0,\ldots,Y_k\Big).$$

Hint: review the proof of the Bayes formula (theorem 2.7).
(c) Using part (b) prove the following: whenever $\mu \ll \nu$

$$\mathbf{E}^\mu(|\mathbf{E}^\mu(f(Y_{k+1})|Y_0,\ldots,Y_k) - \mathbf{E}^\nu(f(Y_{k+1})|Y_0,\ldots,Y_k)|) \xrightarrow{k\to\infty} 0$$

for all bounded measurable f. Conclude that

$$\mathbf{E}^\mu(|\mathbf{E}^\mu(\Phi f(X_{k+1})|Y_0,\ldots,Y_k) - \mathbf{E}^\nu(\Phi f(X_{k+1})|Y_0,\ldots,Y_k)|) \xrightarrow{k\to\infty} 0$$

for all bounded measurable f, where $\Phi f(x) = \int f(y)\,\Phi(x,dy)$.
(d) Suppose that $E = \{1,\ldots,d\}$ and $F = \{1,\ldots,d'\}$. Denote by $\boldsymbol{\Phi}$ the matrix with elements $\boldsymbol{\Phi}_{ij} = \Phi(i,\{j\})$. Show that if $\boldsymbol{\Phi}$ is invertible, then

$$\mathbf{E}^\mu(\|\boldsymbol{\pi}^\mu_{k+1|k} - \boldsymbol{\pi}^\nu_{k+1|k}\|_1) \xrightarrow{k\to\infty} 0,$$

where we have denoted the predictor as $(\boldsymbol{\pi}^\mu_{k+1|k})_i = \mathbf{P}^\mu(X_{k+1} = i|Y_0,\ldots,Y_k)$ and $\|\cdot\|_1$ denotes the ℓ_1-norm of a vector.
(e) Suppose that $\boldsymbol{\Phi}$ is invertible and that $\boldsymbol{\Phi}_{ij} > 0$ for all i,j. Using the filtering recursion to express the filter $\boldsymbol{\pi}^\mu_{k+1}$ in terms of the predictor $\boldsymbol{\pi}^\mu_{k+1|k}$, show that in fact $\mathbf{E}^\mu(\|\boldsymbol{\pi}^\mu_k - \boldsymbol{\pi}^\nu_k\|_1) \to 0$ as $k\to\infty$ whenever $\mu \ll \nu$.

Remark 5.7. Note that we have now proved filter stability in this simple setting making only an 'observability' assumption on the observations: we have made no assumptions on the signal! These ideas have some very general ramifications for the stability of nonlinear filters, see [van08b, van08d, van08a].

Notes

In a pioneering paper on filter stability, Ocone and Pardoux [OP96] established that nonlinear filters are stable under very general assumptions, provided only that the signal process is ergodic. However, their approach relies crucially on a result of Kunita [Kun71] whose proof was subsequently discovered to contain a serious gap [BCL04, Bud03]. This gap is largely resolved in [van08c], where additional results can be found on stability in the case of ergodic signals. In such a general setting, however, no rate of convergence can be given.

Atar and Zeitouni [AZ97] were the first to establish exponential stability of nonlinear filters under the strong mixing assumption (early ideas in this direction are in Delyon and Zeitouni [DZ91]). Del Moral and Guionnet [DG01] obtained similar results using a different method, which gives rise to cleaner bounds (which are suitable for application to particle filters). Our treatment of filter stability is loosely based on the approach of Del Moral and Guionnet (see [DG01, lemma 2.3]). Many authors have investigated filter stability under weaker assumptions than the mixing condition. Let us mention, e.g., Chigansky and Liptser [CL04], Le Gland and Oudjane [LO03], and Kleptsyna and Veretennikov [KV08]. An extensive overview of filter stability results can be found in [CR09]. Various questions in filter stability remain open; for example, it appears to be unknown whether geometrically ergodic signals always yield exponential stability of the filter (under mild conditions on the observations).

A standard reference on geometric ergodicity is Meyn and Tweedie [MT93]. For a nice discussion on minorization and coupling, see Rosenthal [Ros95].

The use of filter stability to prove uniform convergence of the SIS-R algorithm is due to Del Moral and Guionnet [DG01]. Our approach is loosely inspired by Le Gland and Oudjane [LO04]. The book by Del Moral [Del04] contains much further information on this topic. An entirely different approach (which still relies on filter stability, however) can be found in Budhiraja and Kushner [BK01]. It should be noted that unlike in the SIS-R algorithm, the approximation error in the SIS algorithm can generally not be controlled uniformly in time [Del98b]. This is a mathematical hint that the SIS-R algorithm should indeed perform better than the SIS algorithm on longer time scales.

6

Statistical Inference: Methods

6.1 Maximum Likelihood and Bayesian Inference

In the previous chapters, we have discussed in detail how the unobserved signal process X_k can be estimated from an observed sample path of the observation process Y_k. In order to obtain such estimates, however, we presumed that the underlying hidden Markov model was completely known. In the present chapter, we start our investigation of the case where one or more of the basic building blocks of the hidden Markov model—the transition kernel P, the observation kernel Φ, and the initial measure μ—are not known precisely. Our goal is to select, or 'learn,' a suitable underlying hidden Markov model from a long 'training' sample path of the observation process. It should be evident that this statistical inference problem is of great practical importance.

To formalize this problem, we will follow the standard statistical practice of introducing a family of candidate models for consideration. To this end, let $(\Theta, \mathcal{H})$ be a measurable space, called the *parameter space*. For each $\theta \in \Theta$, we introduce a separate transition kernel P^θ, observation kernel Φ^θ, and initial measure μ^θ. The law of the hidden Markov model $(X_k, Y_k)_{k\geq 0}$ defined by $P^\theta, \Phi^\theta, \mu^\theta$ will be denoted as $\mathbf{P}^\theta$. Our goal is now to select, on the basis of an observed sequence $y_0, \ldots, y_n$, a *parameter estimate* $\hat{\theta} \in \Theta$ such that the observation statistics are well described by the law $\mathbf{P}^{\hat{\theta}}$. The hidden Markov model defined by $P^{\hat{\theta}}, \Phi^{\hat{\theta}}, \mu^{\hat{\theta}}$ could then be used, for example, to apply the techniques developed in the previous chapters.

What makes for a 'good' parameter estimate $\hat{\theta}$? Note that the estimator depends, by definition, on the observed training data $\hat{\theta} = \hat{\theta}_n(y_0, \ldots, y_n)$. We would like to guarantee that the estimate $\hat{\theta}_n$ is close to the 'true' value of θ for large n, regardless of what the 'true' parameter happens to be. To be precise, we would like to show that $\hat{\theta}_n \to \theta$ in $\mathbf{P}^\theta$-probability (or $\mathbf{P}^\theta$-a.s.) for every $\theta \in \Theta$. When this is the case, the estimator is called *consistent*: this ensures that if the observations are generated by the hidden Markov model with the true parameter value $\theta^\star \in \Theta$, then the parameter estimate is guaranteed to be

close to $\theta^\star$ provided that we are given a sufficient amount of training data. In this chapter we will develop a class of estimators for hidden Markov models and show how they can be implemented in practice; the issue of consistency of these methods will be mostly tackled in the next chapter.

Remark 6.1. The dependence of $P^\theta, \Phi^\theta, \mu^\theta, \mathbf{P}^\theta$ on θ should always be measurable; e.g., we should really think of P^θ as a kernel $P : \Theta \times E \times \mathcal{E} \to [0,1]$ rather than a family of kernels $P^\theta : E \times \mathcal{E} \to [0,1]$. We will take this for granted.

There are two common approaches for constructing parameter estimates: the *Bayesian* approach and the *maximum likelihood* approach. In hidden Markov models, maximum likelihood estimation and has proven to be very successful and can be implemented much more efficiently than Bayesian estimation. Apart from a brief discussion of Bayesian estimation in this section, we will mostly concentrate on the maximum likelihood approach.

Bayesian approach

Let λ be a probability measure on $(\Theta, \mathcal{H})$. Suppose we choose a parameter $\theta^\star$ at random from the distribution λ, and then generate observations $Y_0, \ldots, Y_k$ using the hidden Markov model $\mathbf{P}^{\theta^\star}$. Then $\theta^\star$ and $(X_k, Y_k)_{0 \le k \le n}$ are jointly distributed according to the distribution $\mathbf{P}^\lambda$ on $\Theta \times (E \times F)^n$ defined as

$$\mathbf{E}^\lambda(f(\theta^\star, X_0, \ldots, X_n, Y_0, \ldots, Y_n)) = \int f(\theta, x_0, \ldots, x_n, y_0, \ldots, y_n)\, \mathbf{P}^\theta(dx_0, \ldots, dx_n, dy_0, \ldots, dy_n)\, \lambda(d\theta).$$

We can now estimate the value of $\theta^\star$ using the estimation techniques introduced in section 2.1. For example, to obtain an estimate $\hat{\theta}_n(Y_0, \ldots, Y_n)$ which minimizes the mean square error $\mathbf{E}^\lambda(\|\hat{\theta}_n - \theta^\star\|^2)$ (assume that $\Theta \subset \mathbb{R}^d$ for this to make sense), we would choose $\hat{\theta}_n = \mathbf{E}^\lambda(\theta^\star | Y_0, \ldots, Y_n)$. This is called the *Bayesian* parameter estimator with *prior distribution* λ.

By introducing the prior λ, we have turned the statistical inference problem into a standard estimation problem. However, our estimator will certainly depend on λ. In practice, it is rarely the case that the parameter is actually chosen from a probability distribution; typically we presume that there is a fixed (non-random) but *unknown* parameter value $\theta^\star$ which generates the observations. This does not mean that we can not use a Bayesian estimator, but the choice of prior is rather subjective as it has no inherent significance. Choosing a suitable prior is a bit of an art which we will not go into. Ultimately, the Bayesian estimator should be justified by proving that it is consistent for a suitable choice of prior. We forgo a detailed discussion.

Remark 6.2. One way to eliminate the subjectivity of the prior is to compute the *minimax* estimator $\hat{\theta}_n(Y_0, \ldots, Y_n)$ which minimizes $\sup_\lambda \mathbf{E}^\lambda(\|\hat{\theta}_n - \theta^\star\|^2)$. In other words, the minimax estimator is the Bayesian estimator where the

prior is chosen according to the 'worst case' scenario. The minimax estimator is generally very difficult to compute in practice, however.

Given a prior distribution, how would we compute the Bayesian estimate? At least conceptually, this turns out to be extremely straightforward. Let us define the $\Theta \times E$-valued stochastic process $\tilde{X}_k = (\tilde{X}_k^1, \tilde{X}_k^2)$ by setting $\tilde{X}_k^1 = \theta^\star$, $\tilde{X}_k^2 = X_k$ for all $k \geq 0$. Then it is a simple exercise to show that $(\tilde{X}_k, Y_k)_{k\geq 0}$ is an ordinary hidden Markov model under the Bayesian measure $\mathbf{P}^\lambda$ with the enlarged signal state space $\Theta \times E$ (problem 6.1). The idea of enlarging the state space to include the parameter is called *state augmentation*. It should be clear that the Bayesian estimator $\hat{\theta}_k$ can now be computed using the filter $\tilde{\pi}_k$ for the augmented model. We have already discussed various filtering algorithms in the previous chapters, and these apply also in this setting.

However, this also highlights the practical difficulties of Bayesian estimation. The computational effort needed to compute the filter to reasonable accuracy increases rapidly (typically exponentially) with the dimension of the state space. In many applications, the signal state space has moderate dimension, so that applying the filter for the signal itself is no problem. However, the parameter space may be much larger than the signal state space—a typical example is the case where the signal state space $E = \{1, \ldots, d\}$ is a finite set, but the parameter space Θ consists of all elements of the transition probability matrix $\boldsymbol{P}$. Here filtering of the signal can be done exactly at little cost, but Bayesian estimation requires us to run a filter on a $d(d-1)$-dimensional parameter space: a very expensive problem.

There are, of course, situations where Bayesian estimation can be practical, e.g., when the parameter space happens to be low dimensional or in the special case where the Kalman filter can be applied. If the parameter space is not low dimensional then computing the Bayesian estimator through filtering is typically intractable; however, there are other methods, such as Markov Chain Monte Carlo (MCMC), which are specifically designed to sample from probability distributions in high dimensional spaces. For an entry point to the literature on this topic, see [CMR05, chapter 13].

Maximum likelihood approach

The most common alternative to Bayesian estimation is maximum likelihood estimation. The idea behind this approach is most easily explained in the case where the observation state space is a finite set $F = \{1, \ldots, d'\}$. Let us briefly discuss the idea in this setting; we then return to the general case.

Suppose that we observe a training sequence $y_0, \ldots, y_n$; in this setting there is only a finite number of possible sequences of a fixed length n, as each y_k can only take a finite number of values. Given a fixed parameter $\theta \in \Theta$, the probability that the hidden Markov model defined by θ generated the observed sequence can be evaluated as $\mathbf{P}^\theta(Y_0 = y_0, \ldots, Y_n = y_n)$. The idea behind

maximum likelihood estimation is now simple: we select the parameter θ which gives the highest probability of generating the actually observed training data

$$\hat{\theta}_n(y_0, \ldots, y_n) = \operatorname*{argmax}_{\theta \in \Theta} \mathbf{P}^\theta(Y_0 = y_0, \ldots, Y_n = y_n).$$

The estimate $\hat{\theta}_n$ can therefore be interpreted as the parameter value under which the training data was most likely to be generated.

When F is not a finite set, the probabilities $\mathbf{P}^\theta(Y_0 = y_0, \ldots, Y_n = y_n)$ will typically be zero. However, the idea can still be implemented if we consider the probability *density* of the observations rather than the probability itself. In this general setting, we assume that the probability measure $\mathbf{P}^\theta|_{Y_0,\ldots,Y_n}$ is absolutely continuous with respect to some fixed probability measure $\mathbf{Q}|_{Y_0,\ldots,Y_n}$ for every $\theta \in \Theta$. The maximum likelihood estimate is then defined as

$$\hat{\theta}_n(y_0, \ldots, y_n) = \operatorname*{argmax}_{\theta \in \Theta} \frac{d\mathbf{P}^\theta|_{Y_0,\ldots,Y_n}}{d\mathbf{Q}|_{Y_0,\ldots,Y_n}}(y_0, \ldots, y_n).$$

Note that the estimate does not depend on the choice of $\mathbf{Q}$, as the latter does not depend on θ. The discrete case above follows as a special case.

The maximum likelihood approach seems intuitively plausible. However, it is certainly not entirely obvious (a) that it gives a good estimator; and (b) that it can be computed efficiently in practice. The latter question is the topic of this chapter, while we will tackle the first problem in the next chapter.

Before we can consider any of these problems, however, we need to ask a basic question: what does the likelihood $d\mathbf{P}^\theta|_{Y_0,\ldots,Y_n}/d\mathbf{Q}|_{Y_0,\ldots,Y_n}$ look like in a hidden Markov model? As it turns out, this is a familiar quantity indeed.

Definition 6.3. *From now on, we suppose that $\Phi^\theta(x, dy)$ has a strictly positive density $\Upsilon^\theta(x, y)$ for every $\theta \in \Theta$ with respect to a fixed measure $\varphi(dy)$. We denote by π_k^θ, $\pi_{k|n}^\theta$, σ_k^θ, etc., the conditional measures computed as in chapter 2 for the transition kernel P^θ, observation density Υ^θ, and initial measure μ^θ.*

Proposition 6.4. *Define $\mathbf{Q}|_{Y_0,\ldots,Y_n}(dy_0, \ldots, dy_n) = \varphi(dy_0) \cdots \varphi(dy_n)$. Then*

$$\begin{aligned}\mathbf{L}_n^\theta := \frac{d\mathbf{P}^\theta|_{Y_0,\ldots,Y_n}}{d\mathbf{Q}|_{Y_0,\ldots,Y_n}}(y_0, \ldots, y_n) &= \sigma_n^\theta(y_0, \ldots, y_n, E) \\ &= \sigma_0^\theta(y_0, E) \prod_{k=1}^{n} \int \Upsilon^\theta(x, y_k)\, P^\theta(x', dx)\, \pi_{k-1}^\theta(y_0, \ldots, y_{k-1}, dx').\end{aligned}$$

Proof. It suffices to note that by the definition of σ_n^θ

$$\begin{aligned}\mathbf{E}^\theta(f(Y_0, \ldots, Y_n)) &= \int f(y_0, \ldots, y_n)\, \Upsilon^\theta(x_0, y_0) \cdots \Upsilon^\theta(x_n, y_n) \\ &\quad \times P^\theta(x_{n-1}, dx_n) \cdots P^\theta(x_0, dx_1)\, \mu^\theta(dx_0)\, \varphi(dy_0) \cdots \varphi(dy_n) \\ &= \int f(y_0, \ldots, y_n)\, \sigma_n^\theta(y_0, \ldots, y_n, E)\, \varphi(dy_0) \cdots \varphi(dy_n)\end{aligned}$$

for every bounded measurable function f. □

For a fixed value of θ, the likelihood $\mathbf{L}_n^\theta$ is evidently easily computed using the filtering recursion; for example, in algorithm 3.2 the likelihood is simply $c_0 \cdots c_n$. In order to compute the maximum likelihood estimate, however, we must compute the filter not for a fixed value of θ, but simultaneously for all θ. At first sight, this appears just as difficult as Bayesian estimation.

However, most algorithms for finding the maximum of a function $f(\theta)$ do not require us to evaluate this function a priori at every point θ. Instead, these algorithms typically search for a maximum by starting from an initial guess for θ and iteratively moving this guess in the direction in which the function increases (think, e.g., of steepest descent or Newton-type methods). If we are lucky, such an algorithm converges in a relatively small number of steps, so that we need to run the filter only for a small number of values of θ. In the next section, we will discuss a particular iterative method of this type that is specifically designed for maximum likelihood estimation. The downside of such methods is that they are typically guaranteed to converge only to a *local* maximum of the likelihood, which is not necessarily a global maximum.

To date, this appears to be the state of affairs: the (global) maximum likelihood estimate can be proved to be consistent under suitable assumptions on the model, but to compute the estimate efficiently we can typically only guarantee that a local maximum is found. In practice, this seems to work very well; on the theoretical side, much has been but much also remains to be done.

A particularly simple setting: hypothesis testing

Before we move on to more complicated cases, we discuss a particularly simple setting: the case where the parameter space $\Theta = \{1, \ldots, p\}$ is a finite set. This is known as the *hypothesis testing* problem: we are given p different model possibilities (hypotheses), and our goal is to decide on the basis of observations which of the hypotheses holds true. Though this is not the typical setting of parameter estimation, such problems do appear in applications—for example, the word recognition problem in speech recognition (see example 1.17).

Because there are only finitely many possibilities, the maximum likelihood hypothesis can easily be found. All we need to do is to compute p filters for the observed sequence, one for each parameter value. This can be done using either the exact filtering algorithm if E is finite or using the SIS-R algorithm otherwise (moreover, the computations are easily parallelized as the filters are computed independently). Once the filters are computed, we choose as our estimate the hypothesis with the largest likelihood. However, in this setting Bayesian estimation is also tractable—and gives essentially the same answer!

Proposition 6.5. *Let λ be a Bayesian prior. Then the conditional distribution of the parameter $\theta^\star$ under the Bayesian measure $\mathbf{P}^\lambda$ is given by*

$$\mathbf{E}^\lambda(f(\theta^\star)|Y_0, \ldots, Y_n) = \frac{\int f(\theta)\, \sigma_n^\theta(Y_0, \ldots, Y_n, E)\, \lambda(d\theta)}{\int \sigma_n^\theta(Y_0, \ldots, Y_n, E)\, \lambda(d\theta)}.$$

In particular, provided that $\lambda(\{i\}) > 0$ *for all* $i = 1, \ldots, p$*, the maximum likelihood parameter estimate coincides with the Bayesian MAP estimate as then* $\mathbf{P}^\lambda(\theta^\star = i | Y_0, \ldots, Y_n) \propto \sigma_n^i(Y_0, \ldots, Y_n, E)$ *for all* $i = 1, \ldots, p$*.*

Proof. Problem 6.3. □

This simple setting is exceptional in that consistency can be studied without heroics. So why wait until the next chapter? Though the following result and proof are deceptively simple, and we will need to develop different tools to deal with a more general setting, it should nonetheless give some insightful motivation for the statistical inference methodology.

Theorem 6.6. *When* Θ *is a finite set, the following are equivalent.*

1. *Maximum likelihood estimation is consistent:* $\hat{\theta}_n \to \theta$ $\mathbf{P}^\theta$*-a.s. for all* $\theta \in \Theta$*;*
2. $\mathbf{P}^\theta|_{(Y_k)_{k\geq 0}}$ *and* $\mathbf{P}^{\theta'}|_{(Y_k)_{k\geq 0}}$ *are mutually singular for all* $\theta, \theta' \in \Theta$*,* $\theta \neq \theta'$*.*

Recall that probability measures $\mathbf{P}, \mathbf{Q}$ on a measurable space $(\Omega, \mathcal{G})$ are called mutually singular if there is $S \in \mathcal{G}$ such that $\mathbf{P}(S) = 1$ and $\mathbf{Q}(S) = 0$.

Proof. Suppose that the maximum likelihood estimator is consistent. Then we find that $\mathbf{P}^\theta(\lim_{n\to\infty} \hat{\theta}_n = \theta) = 1$ and $\mathbf{P}^{\theta'}(\lim_{n\to\infty} \hat{\theta}_n = \theta) = 0$ whenever $\theta \neq \theta'$. As by construction $\hat{\theta}_n$ is a function of the observations only, this implies that $\mathbf{P}^\theta|_{(Y_k)_{k\geq 0}}$ and $\mathbf{P}^{\theta'}|_{(Y_k)_{k\geq 0}}$ are mutually singular for $\theta \neq \theta'$.

Now for the converse. The idea is to show that there exists a perfect estimator: i.e., there exists a random variable $\hat{\theta}^0$, which is a function of the observations only, such that $\hat{\theta}^0 = \theta$ $\mathbf{P}^\theta$-a.s. for every $\theta \in \Theta$. We claim that if there exists such a perfect estimator, then the maximum likelihood estimate must be consistent. Let us first show why this is true, and then complete the proof by showing the existence of a perfect estimator.

Let λ be a Bayesian prior as in proposition 6.5. Then

$$\mathbf{P}^\lambda(\theta^\star = \theta | Y_0, \ldots, Y_n) \xrightarrow{n\to\infty} \mathbf{P}^\lambda(\theta^\star = \theta | (Y_k)_{k\geq 0})$$

by the martingale convergence theorem, and as Θ is a finite set we evidently have $\lim_{n\to\infty} \hat{\theta}_n = \operatorname{argmax}_{\theta\in\Theta} \mathbf{P}^\lambda(\theta^\star = \theta | (Y_k)_{k\geq 0})$. By example 2.6, we find that $\hat{\theta} = \lim_{n\to\infty} \hat{\theta}_n$ is the estimator that minimizes the cost $\mathbf{P}^\lambda(\hat{\theta} \neq \theta^\star)$. But if $\hat{\theta}^0$ is a perfect estimator, then clearly $\mathbf{P}^\lambda(\hat{\theta}^0 \neq \theta^\star) = 0$. Therefore, we evidently have $\hat{\theta} = \hat{\theta}^0$ $\mathbf{P}^\lambda$-a.s. You can easily convince yourself that as $\lambda(\{\theta\}) > 0$ for every $\theta \in \Theta$ by assumption, this means that $\hat{\theta} = \theta$ $\mathbf{P}^\theta$-a.s. for every $\theta \in \Theta$, i.e., the maximum likelihood estimator is consistent.

It remains to prove the existence of a perfect estimator. We will construct such an estimator under the assumption that $\mathbf{P}^\theta|_{(Y_k)_{k\geq 0}}$ and $\mathbf{P}^{\theta'}|_{(Y_k)_{k\geq 0}}$ are mutually singular for $\theta \neq \theta'$, thus completing the proof. For every $\theta \neq \theta'$, let $S^{\theta,\theta'} \in \sigma\{Y_k : k \geq 0\}$ be a set such that $\mathbf{P}^\theta(S^{\theta,\theta'}) = 1$ and $\mathbf{P}^{\theta'}(S^{\theta,\theta'}) = 0$. Define $S^\theta = \bigcap_{\theta'\in\Theta} S^{\theta,\theta'}$; then for every θ, we have $\mathbf{P}^\theta(S^\theta) = 1$ and $\mathbf{P}^{\theta'}(S^\theta) = 0$ when $\theta' \neq \theta$. Define the random variable $\hat{\theta}^0(\omega) = \theta$ for $\omega \in S^\theta$; then by construction $\hat{\theta}^0$ is a function of the observations only and $\hat{\theta}^0 = \theta$ $\mathbf{P}^\theta$-a.s. for every $\theta \in \Theta$. Therefore $\hat{\theta}^0$ is a perfect estimator. □

The second condition in this theorem—an *identifiability condition*—is necessary for any estimator to be consistent. Though the rather naive approach and conditions of this theorem do not extend to the case where the parameter space Θ is uncountable, we will see this idea return in the next chapter.

6.2 The EM Algorithm

When the parameter space is not finite, it is very difficult to compute the exact maximum likelihood estimate. We therefore need algorithms to search for the maximum likelihood parameter. In principle one can employ almost any algorithm for finding a maximum of a function (see problem 6.8). The goal of this section is to develop a particular algorithm that is specific to maximum likelihood estimation—the *EM (expectation-maximization) algorithm*—which is widely used in statistical inference problems in hidden Markov models.

EM assumptions

The EM algorithm does not apply to the parameter estimation problem in its most general form; we need to make some assumptions about the nature of the parameter dependence. When these assumptions do not hold, parameter estimation typically becomes a much more difficult problem. Fortunately, it turns out that these assumptions hold in a variety of important examples.

Assumption 6.7 (EM assumptions) *There exists a fixed transition kernel P on E, and probability measures μ on E and φ on F, such that $P^\theta, \Phi^\theta, \mu^\theta$ have densities which have the form of exponential families:*

$$P^\theta(x, dx') = p^\theta(x, x')\, P(x, dx') = \exp\left(\textstyle\sum_{\ell=1}^{d_p} c_\ell(\theta)\, p_\ell(x, x')\right) P(x, dx'),$$
$$\Phi^\theta(x, dy) = \Upsilon^\theta(x, y)\, \varphi(dy) = \exp\left(\textstyle\sum_{\ell=1}^{d_\Upsilon} \gamma_\ell(\theta)\, \Upsilon_\ell(x, y)\right) \varphi(dy),$$
$$\mu^\theta(dx) = u^\theta(x)\, \mu(dx) = \exp\left(\textstyle\sum_{\ell=1}^{d_u} q_\ell(\theta)\, u_\ell(x)\right) \mu(dx).$$

(Necessarily $\int p^\theta(x, x')P(x, dx') = \int \Upsilon^\theta(x, y)\varphi(dy) = \int u^\theta(x)\mu(dx) = 1$.)

Let us give some typical examples.

Example 6.8 (Finite state space). Suppose that $E = \{1, \dots, d\}$, so that we can represent the kernel P^θ as a matrix $\boldsymbol{P}^\theta$. Suppose also that $(\boldsymbol{P}^\theta)_{ij} > 0$ for all $\theta \in \Theta$. Then P^θ satisfies the corresponding EM assumption.

Indeed, let us choose $P(x, dx')$ to be the transition kernel whose transition probability matrix $\boldsymbol{P}$ is given by $(\boldsymbol{P})_{ij} = 1/d$ for all i, j. Then

$$P^\theta(i, \{j\}) = \exp\left(\textstyle\sum_{k,\ell=1}^{d} \log((\boldsymbol{P}^\theta)_{k\ell} d)\, I_k(i)\, I_\ell(j)\right) P(i, \{j\}).$$

Therefore we may set $c_{k\ell}(\theta) = \log((\boldsymbol{P}^\theta)_{k\ell} d)$ and $p_{k\ell}(i,j) = I_k(i)\, I_\ell(j)$.

Note that a minor modification allows us to treat the case where for each i, j either $(\boldsymbol{P}^\theta)_{ij} > 0$ for all $\theta \in \Theta$ or $(\boldsymbol{P}^\theta)_{ij} = 0$ for all $\theta \in \Theta$ (choose a suitable reference kernel P). Also, in a similar fashion, we find that Φ^θ always satisfies the EM assumption if E and F are both finite sets (provided, as always, that Υ^θ is strictly positive for every $\theta \in \Theta$).

Example 6.9 (Gaussian observations). Let us suppose that $E = \{1, \dots, d\}$ is a finite set and that $F = \mathbb{R}$. We assume that $\Phi^\theta(i, dy)$ is a Gaussian distribution for every $i = 1, \dots, d$ and $\theta \in \Theta$. Then Φ^θ satisfies the EM assumption.

Indeed, let $\varphi(dy) = e^{-y^2/2} dy/\sqrt{2\pi}$ and denote by $m_i(\theta)$ and $v_i(\theta)$, respectively, the mean and variance of the Gaussian distribution $\Phi^\theta(i, dy)$. Then

$$\begin{aligned}\Phi^\theta(i, dy) &= \exp\left(\frac{1}{2}\, y^2 - \frac{(y - m_i(\theta))^2}{2 v_i(\theta)} - \log(\sqrt{v_i(\theta)})\right) \varphi(dy) \\ &= \exp\left(\textstyle\sum_{k=1}^3 \sum_{\ell=1}^d \gamma_{k\ell}(\theta)\, \Upsilon_{k\ell}(i, y)\right) \varphi(dy),\end{aligned}$$

where $\Upsilon_{1\ell}(i, y) = I_\ell(i)\, y^2$, $\Upsilon_{2\ell}(i, y) = I_\ell(i)\, y$, $\Upsilon_{3\ell}(i, y) = I_\ell(i)$, and

$$\gamma_{1\ell}(\theta) = \frac{1 - v_\ell(\theta)^{-1}}{2}, \quad \gamma_{2\ell}(\theta) = \frac{m_\ell(\theta)}{v_\ell(\theta)}, \quad \gamma_{3\ell}(\theta) = -\frac{m_\ell(\theta)^2}{2 v_\ell(\theta)} - \log(\sqrt{v_\ell(\theta)}).$$

Along similar lines, we can establish that the EM assumption is satisfied for Φ^θ if $E = \mathbb{R}^p$, $F = \mathbb{R}^q$, and the observations satisfy $Y_k = C(\theta) X_k + D(\theta)\eta_k$ where $\eta_k \sim N(0, \mathrm{Id})$ and $D(\theta)$ is an invertble matrix for every θ.

The EM algorithm

Recall that the maximum likelihood estimate is defined by the expression $\hat{\theta}_n = \operatorname{argmax}_\theta d\mathbf{P}^\theta|_{Y_0,\dots,Y_n} / d\mathbf{Q}|_{Y_0,\dots,Y_n}$ for any reference measure $\mathbf{Q}$. In particular, we may choose $\mathbf{Q} = \mathbf{P}^{\theta'}$ for an arbitrary $\theta' \in \Theta$, so we can write

$$\hat{\theta}_n(y_0, \dots, y_n) = \operatorname*{argmax}_{\theta \in \Theta} \log\left(\frac{d\mathbf{P}^\theta|_{Y_0,\dots,Y_n}}{d\mathbf{P}^{\theta'}|_{Y_0,\dots,Y_n}}(y_0, \dots, y_n)\right).$$

Here we have used the fact that the logarithm is an increasing function.

Now recall that (if this is unfamiliar, do problem 6.2)

$$\log\left(\frac{d\mathbf{P}^\theta|_{Y_0,\dots,Y_n}}{d\mathbf{P}^{\theta'}|_{Y_0,\dots,Y_n}}\right) = \log\left(\mathbf{E}^{\theta'}\left[\frac{d\mathbf{P}^\theta}{d\mathbf{P}^{\theta'}}\,\middle|\, Y_0, \dots, Y_n\right]\right).$$

The maximum of this expression with respect to θ is typically very difficult to compute. However, consider instead the quantity

$$Q_n(\theta, \theta') = \mathbf{E}^{\theta'}\left[\log\left(\frac{d\mathbf{P}^\theta}{d\mathbf{P}^{\theta'}}\right)\,\middle|\, Y_0, \dots, Y_n\right].$$

The key point is that the maximum of this quantity with respect to θ is easy to compute when the EM assumptions hold. Indeed, under assumption 6.7,

$$\log\left(\frac{d\mathbf{P}^{\theta}}{d\mathbf{P}^{\theta'}}(x_0,\ldots,x_n,y_0,\ldots,y_n)\right) = \sum_{k=0}^{n}\sum_{\ell=1}^{d_\Upsilon}\{\gamma_\ell(\theta)-\gamma_\ell(\theta')\}\,\Upsilon_\ell(x_k,y_k) + \sum_{k=1}^{n}\sum_{\ell=1}^{d_p}\{c_\ell(\theta)-c_\ell(\theta')\}\,p_\ell(x_{k-1},x_k) + \sum_{\ell=1}^{d_u}\{q_\ell(\theta)-q_\ell(\theta')\}\,u_\ell(x_0),$$

so that we obtain

$$\begin{aligned} Q_n(\theta,\theta') = &\sum_{k=0}^{n}\sum_{\ell=1}^{d_\Upsilon}\{\gamma_\ell(\theta)-\gamma_\ell(\theta')\}\int \Upsilon_\ell(x,y_k)\,\pi_{k|n}^{\theta'}(dx) \\ &+\sum_{k=1}^{n}\sum_{\ell=1}^{d_p}\{c_\ell(\theta)-c_\ell(\theta')\}\int p_\ell(x,x')\,\pi_{k-1,k|n}^{\theta'}(dx,dx') \\ &+\sum_{\ell=1}^{d_u}\{q_\ell(\theta)-q_\ell(\theta')\}\int u_\ell(x)\,\pi_{0|n}^{\theta'}(dx). \end{aligned}$$

Therefore, the computation of $\operatorname{argmax}_{\theta\in\Theta} Q_n(\theta,\theta')$ can be accomplished in two steps. First, we compute the univariate and bivariate *smoothing* distributions $\pi_{k|n}^{\theta'}$ and $\pi_{k-1,k}^{\theta'}$ (see theorem 3.2). This can be done efficiently using, e.g., the Baum-Welch algorithm 3.2 or a variant of the SIS-R algorithm that computes smoothing distributions. Then, we solve the *deterministic* optimization problem of maximizing the above expression with respect to θ: this is much simpler than the original problem, as the θ-dependence has been separated out from the computation of the conditional expectation. This is the essence of the EM-algorithm: we first perform the E-step (computation of the smoothing distributions), followed by the M-step (maximizing the deterministic expression for Q_n). In many examples, the M-step can in fact be done analytically, so that maximizing $Q_n(\theta,\theta')$ reduces to the smoothing problem only.

At this point calls of protest should be heard. How on earth do we justify exchanging the logarithm and expectation, as we did in order to define $Q_n(\theta,\theta')$? Indeed, the parameter θ that maximizes $Q_n(\theta,\theta')$ is *not* the maximum likelihood estimate. Remarkably, however, the follows does hold: the likelihood of the maximizer θ can be no smaller than the likelihood of θ'!

Lemma 6.10 (EM lemma). *If $\theta = \operatorname{argmax}_{\theta_0\in\Theta} Q(\theta_0,\theta')$, then $\mathbf{L}_n^{\theta} \geq \mathbf{L}_n^{\theta'}$, i.e., the likelihood of θ' can never exceed the likelihood of θ.*

Proof. Note that

$$\log \mathbf{L}_n^{\theta} - \log \mathbf{L}_n^{\theta'} = \log\left(\mathbf{E}^{\theta'}\left[\frac{d\mathbf{P}^{\theta}}{d\mathbf{P}^{\theta'}}\,\middle|\, Y_0,\ldots,Y_n\right]\right) \geq Q_n(\theta,\theta')$$

by Jensen's inequality. But as $Q_n(\theta',\theta')=0$, we must have $Q_n(\theta,\theta')\geq 0$. □

What this simple lemma suggests is that if we start from some candidate parameter estimate θ', then computing a new estimate by maximizing $Q_n(\theta, \theta')$ is guaranteed to improve on our initial estimate. This suggests the following iterative algorithm. We start with an arbitrary candidate parameter $\hat{\theta}_n^{(0)} \in \Theta$. We then construct the sequence of estimates

$$\hat{\theta}_n^{(j)} = \operatorname*{argmax}_{\theta \in \Theta} Q_n(\theta, \hat{\theta}_n^{(j-1)}), \qquad j \geq 1$$

by iterating the E- and M-steps above. This is called the *EM algorithm*. The likelihood of the sequence of EM estimates $\hat{\theta}_n^{(j)}$ will steadily increase as j increases, and we hope that the sequence will converge to the maximum likelihood estimate. In practice, this is difficult to prove; what one can prove is that, under mild conditions, the sequence $\hat{\theta}_n^{(j)}$ converges to a *local* maximum of the likelihood function $\mathbf{L}_n^\theta$. This is briefly sketched in the next chapter.

EM algorithm for a class of hidden Markov models

Let us show the EM algorithm at work in an important class of concrete hidden Markov models. We consider a finite signal state space $E = \{1, \ldots, d\}$ and a real valued observation state space $F = \mathbb{R}$, where the observations take the form $Y_k = m(X_k) + \sqrt{v(X_k)}\,\eta_k$ with $\eta_k \sim N(0,1)$. We wish to estimate all the transition probabilities $\boldsymbol{P}_{ij}$ of the signal, all initial probabilities $\boldsymbol{\mu}_i$ of the signal, and the observation functions m and v (which we interpret as vectors $\boldsymbol{m}_i = m(i)$ and $\boldsymbol{v}_i = v(i)$ as usual). We therefore introduce the parameter space $\Theta = \Sigma_d \times \Delta_d \times \mathbb{R}^d \times \mathbb{R}_+^d$, where Σ_d is the space of $d \times d$ stochastic matrices with strictly positive entries, Δ_d is the space of d-dimensional probability vectors with strictly positive entries, and $\mathbb{R}_+^d$ is the space of d-dimensional vectors with strictly positive entries; we wish to estimate $(\boldsymbol{P}, \boldsymbol{\mu}, \boldsymbol{m}, \boldsymbol{v}) \in \Theta$.

What we are going to do is solve the M-step in the EM algorithm explicitly. To this end, let us first plug in the expressions in examples 6.8 and 6.9 into the general expression for $Q_n(\theta, \theta')$ above. This gives the following:

$$\begin{aligned} Q_n(\theta, \theta') = &- \sum_{k=0}^{n} \sum_{\ell=1}^{d} \left[\frac{(y_k - \boldsymbol{m}_\ell)^2}{2\boldsymbol{v}_\ell} + \log(\sqrt{\boldsymbol{v}_\ell}) \right] (\boldsymbol{\pi}_{k|n}^{\theta'})_\ell \\ &+ \sum_{k=1}^{n} \sum_{\ell,\ell'=1}^{d} \log(\boldsymbol{P}_{\ell\ell'})\, (\boldsymbol{\pi}_{k-1,k|n}^{\theta'})_{\ell\ell'} + \sum_{\ell=1}^{d} \log(\boldsymbol{\mu}_\ell)\, (\boldsymbol{\pi}_{0|n}^{\theta'})_\ell \\ &- \text{a term that is independent of } \theta, \end{aligned}$$

where we have written $\theta = (\boldsymbol{P}, \boldsymbol{\mu}, \boldsymbol{m}, \boldsymbol{v})$. We can now maximize this expression explicitly by taking derivatives with respect to the parameters and setting these to zero (do not forget to take into account the constraints $\sum_{\ell'} \boldsymbol{P}_{\ell\ell'} = 1$ and $\sum_\ell \boldsymbol{\mu}_\ell = 1$, e.g., by substituting $\boldsymbol{P}_{\ell d}$ by $1 - \sum_{\ell' < d} \boldsymbol{P}_{\ell\ell'}$, and similarly for $\boldsymbol{\mu}_d$, before computing the maximum). We leave these routine computations to you (problem 6.4), and jump straight to the result.

Algorithm 6.1: Concrete EM Algorithm/Proposition 6.11

$\ell \leftarrow 0$;
repeat
  $(\boldsymbol{P}, \boldsymbol{\mu}, \boldsymbol{m}, \boldsymbol{v}) \leftarrow \hat{\theta}_n^{(\ell)}$;
  Run the Baum-Welch algorithm 3.2;
  $\hat{\boldsymbol{P}}_{ij} \leftarrow \sum_{k=1}^n (\boldsymbol{\pi}_{k-1,k|n})_{ij} / \sum_{k=1}^n (\boldsymbol{\pi}_{k-1|n})_i$, $i, j = 1, \ldots, d$;
  $\hat{\boldsymbol{\mu}}_i \leftarrow (\boldsymbol{\pi}_{0|n})_i$, $i = 1, \ldots, d$;
  $\hat{\boldsymbol{m}}_i \leftarrow \sum_{k=0}^n y_k \, (\boldsymbol{\pi}_{k|n})_i / \sum_{k=0}^n (\boldsymbol{\pi}_{k|n})_i$, $i = 1, \ldots, d$;
  $\hat{\boldsymbol{v}}_i \leftarrow \sum_{k=0}^n (y_k - \hat{\boldsymbol{m}}_i)^2 \, (\boldsymbol{\pi}_{k|n})_i / \sum_{k=0}^n (\boldsymbol{\pi}_{k|n})_i$, $i = 1, \ldots, d$;
  $\hat{\theta}_n^{(\ell+1)} \leftarrow (\hat{\boldsymbol{P}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{m}}, \hat{\boldsymbol{v}})$;
  $\ell \leftarrow \ell + 1$;
until *parameter estimates converge* ;

Proposition 6.11. *We have* $(\boldsymbol{P}, \boldsymbol{\mu}, \boldsymbol{m}, \boldsymbol{v}) = \operatorname{argmax}_{\theta \in \Theta} Q_n(\theta, \theta')$ *where*

$$\boldsymbol{P}_{ij} = \frac{\sum_{k=1}^n (\boldsymbol{\pi}^{\theta'}_{k-1,k|n})_{ij}}{\sum_{k=1}^n (\boldsymbol{\pi}^{\theta'}_{k-1|n})_i}, \qquad \boldsymbol{\mu}_i = (\boldsymbol{\pi}^{\theta'}_{0|n})_i,$$

$$\boldsymbol{m}_i = \frac{\sum_{k=0}^n y_k \, (\boldsymbol{\pi}^{\theta'}_{k|n})_i}{\sum_{k=0}^n (\boldsymbol{\pi}^{\theta'}_{k|n})_i}, \qquad \boldsymbol{v}_i = \frac{\sum_{k=0}^n (y_k - \boldsymbol{m}_i)^2 \, (\boldsymbol{\pi}^{\theta'}_{k|n})_i}{\sum_{k=0}^n (\boldsymbol{\pi}^{\theta'}_{k|n})_i}.$$

In particular, note that $\boldsymbol{P}_{ij} = \tau_n^{ij;\theta'} / \omega_n^{i;\theta'}$, *where* τ_n *and* ω_n *are the transition counts and occupation times as defined in section 3.2.*

The entire EM algorithm is summarized as algorithm 6.1.

It is interesting to note that the EM iteration has a remarkably intuitive interpretation. For example, the improved estimate of the transition probability from state i to state j is precisely the best estimate—given our present best guess of the parameter values—of the relative frequency of the transitions from i to j. This might be a natural guess for a good estimate, but now we know that this is always guaranteed to improve the likelihood. The estimates for the remaining parameters possess equally intuitive interpretations.

Remark 6.12. We have included estimation of the initial measure $\boldsymbol{\mu}$ in our discussion. However, unlike the remaining parameters which affect the dynamics of the model in every time step, the initial measure is only sampled once in a single realization of the model. Therefore, the maximum likelihood estimate of $\boldsymbol{\mu}$ obtained from a single observation time series is not particularly meaningful—it is an estimate of X_0 rather than of the law of X_0. Estimation of the initial measure can therefore usually be omitted with little loss.

A simple numerical illustration is shown in figure 6.1. The observations were generated from a model on $E = \{1, 2\}$ with true parameters

$$\boldsymbol{P}^\star = \begin{bmatrix} .85 & .15 \\ .05 & .95 \end{bmatrix}, \qquad \boldsymbol{\mu}^\star = \begin{bmatrix} .3 \\ .7 \end{bmatrix}, \qquad \boldsymbol{m}^\star = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \qquad \boldsymbol{v}^\star = \begin{bmatrix} .5 \\ .2 \end{bmatrix}.$$

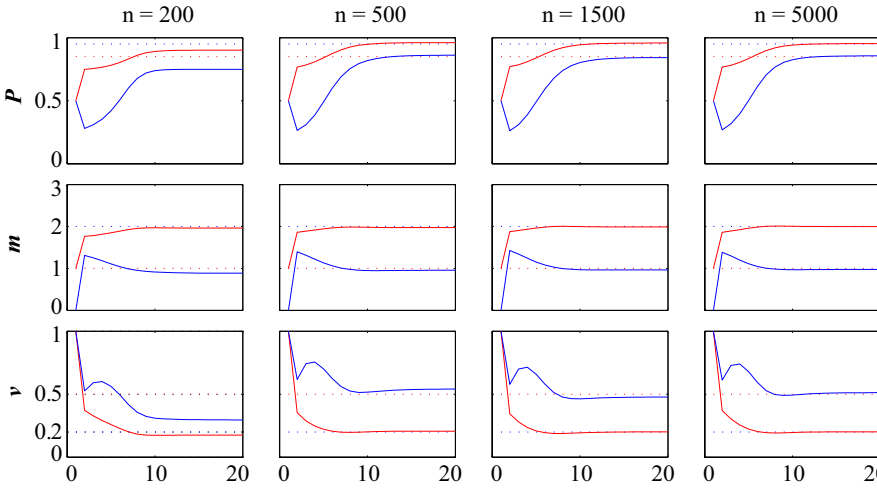

Fig. 6.1. EM algorithm applied to observation time series $y_0, \ldots, y_n$ obtained from the numerical example in section 6.2, for various data lengths n. Shown are the EM estimates of $\boldsymbol{P}_{11}$ (top, red), $\boldsymbol{P}_{22}$ (top, blue), $\boldsymbol{m}_1$ (middle, red), $\boldsymbol{m}_2$ (middle, blue), $\boldsymbol{v}_1$ (bottom, red), and $\boldsymbol{v}_2$ (bottom, blue), as a function of the number of EM iterations. The dotted lines show the true parameter values of each of these quantities.

The EM algorithm was run using the initial guesses

$$\hat{\boldsymbol{P}}^{(0)} = \begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix}, \qquad \hat{\boldsymbol{\mu}}^{(0)} = \begin{bmatrix} .5 \\ .5 \end{bmatrix}, \qquad \hat{\boldsymbol{m}}^{(0)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \hat{\boldsymbol{v}}^{(0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

We see that the EM estimates do indeed converge after a few iterations of the algorithm; moreover, as the length of the observation sequence increases, the EM estimates converge to the true parameter values. The latter suggests that the maximum likelihood estimates are indeed consistent. An interesting thing to note, however, is that the EM estimates have changed the order of the states in E as compared to the model which we used to generate the observations. This is no problem, of course, as changing the order of the points in E just gives a different representation of the same hidden Markov model.

6.3 Model Order Estimation

In the previous sections, we have tacitly assumed that the signal and observation state spaces E and F are fixed at the outset. In order for any form of estimation to make sense, we must indeed fix F—after all, we are trying to estimate on the basis of a given observation sequence $y_0, \ldots, y_n$ which takes values in F. However, it is certainly possible to consider statistical inference problems where different E^θ are chosen for different parameter values $\theta \in \Theta$.

Table 6.1. Order dependence of EM estimates in the numerical example of sec. 6.3

$d = 2$	$d = 3$	$d = 4$
$n^{-1} \log \mathbf{L}_n^{\theta_{\mathrm{EM}}} = -.6371$	$n^{-1} \log \mathbf{L}_n^{\theta_{\mathrm{EM}}} = -.2047$	$n^{-1} \log \mathbf{L}_n^{\theta_{\mathrm{EM}}} = -.2041$
$\boldsymbol{P}_{\mathrm{EM}} = \begin{bmatrix} .73 & .27 \\ .04 & .96 \end{bmatrix}$	$\boldsymbol{P}_{\mathrm{EM}} = \begin{bmatrix} .69 & .17 & .14 \\ .05 & .90 & .05 \\ .06 & .15 & .79 \end{bmatrix}$	$\boldsymbol{P}_{\mathrm{EM}} = \begin{bmatrix} .69 & .06 & .10 & .15 \\ .05 & .36 & .56 & .03 \\ .04 & .29 & .60 & .07 \\ .06 & .06 & .09 & .79 \end{bmatrix}$
$\boldsymbol{m}_{\mathrm{EM}} = \begin{bmatrix} -.031 & 2.5 \end{bmatrix}$	$\boldsymbol{m}_{\mathrm{EM}} = \begin{bmatrix} .0049 & 2.0 & 4.0 \end{bmatrix}$	$\boldsymbol{m}_{\mathrm{EM}} = \begin{bmatrix} .0015 & 1.9 & 2.0 & 4.0 \end{bmatrix}$
$\boldsymbol{v}_{\mathrm{EM}} = \begin{bmatrix} .20 & 1.1 \end{bmatrix}$	$\boldsymbol{v}_{\mathrm{EM}} = \begin{bmatrix} .22 & .20 & .20 \end{bmatrix}$	$\boldsymbol{v}_{\mathrm{EM}} = \begin{bmatrix} .22 & .23 & .18 & .20 \end{bmatrix}$

A setting of particular interest is one where the signal state space E is a finite set of unknown cardinality. This problem appears in various applications. For example, suppose we want to model stock prices using a regime switching model, i.e., where the return and volatility vary according to a finite state Markov process (see example 1.12). It is typically not clear, a priori, how many regimes one should choose in order to faithfully reproduce the observed stock price fluctuations. The number of regimes, called the *model order*, must then be estimated along with the other model parameters.

In principle one would expect that the maximum likelihood approach would work equally well in this case. A promising procedure is the following: for each model order $d = 1, \dots, D$ (recall $E = \{1, \dots, d\}$), we can use the EM algorithm as in the previous section to obtain the (hopefully) maximum likelihood estimate. We therefore obtain a candidate hidden Markov model with parameter θ_d for every $d = 1, \dots, D$. For each of these candidate models, we can compute the observation likelihood $\mathbf{L}_n^{\theta_d}$ from the constants c_k in the Baum-Welch algorithm. The hope is then that if we choose D sufficiently large then $\mathbf{L}_n^\star(d) := \mathbf{L}_n^{\theta_d}$ would attain a maximum for some $d < D$, in which case the maximum likelihood value of d is clearly the model order of choice.

However, this does not quite work out the way one would think. The problem is that a d-state Markov process can always be represented as a $d+1$-state Markov process by duplicating one of the states. You can therefore easily convince yourself that for any hidden Markov model of order d, there is a hidden Markov model of order $d + 1$ whose observation law is precisely the same. Therefore, *the maximum likelihood* $\mathbf{L}_n^\star(d)$ *of order* d *is always nondecreasing in* d*!* In particular, a 'maximum likelihood' model order does not exist. To illustrate this phenomenon, table 6.1 shows the results of a numerical example where an observation time series $y_0, \dots, y_{5000}$ was generated from the following hidden Markov model with three signal states:

$$\boldsymbol{P}^\star = \begin{bmatrix} .70 & .15 & .15 \\ .05 & .90 & .05 \\ .05 & .15 & .80 \end{bmatrix}, \qquad \boldsymbol{\mu}^\star = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}, \qquad \boldsymbol{m}^\star = \begin{bmatrix} 0 \\ 2 \\ 4 \end{bmatrix}, \qquad \boldsymbol{v}^\star = \begin{bmatrix} .2 \\ .2 \\ .2 \end{bmatrix}.$$

As you can see, the likelihood of the order $d = 2$ is significantly smaller than the likelihood of the (true) order $d = 3$, while the likelihood of the order $d = 4$ is essentially equal to the likelihood of the true order. Inspection of the parameter estimates for $d = 4$ shows that the middle state of the true model has been duplicated as the two middle states of the estimated $d = 4$ model, with very little effect on the observation statistics.

Our new-found intuition about the model order estimation problem suggests the following heuristic approach. First, we compute the maximum likelihood function $\mathbf{L}_n^\star(d)$ for every model order $d = 1, \ldots, D$ as described above. When plotted as a function of d, the likelihood should steadily increase for orders below the true model order $d < d^\star$, while the likelihood should be roughly constant for orders greater than the true model order $d \geq d^\star$. The model order estimate is therefore found by looking for the 'corner' in the plot of the likelihood function $\mathbf{L}_n^\star(d)$. This is indeed the essence of a successful model order estimation technique, but this formulation is not very precise mathematically (how is the 'corner' defined?) In particular, we need to be more precise if we wish to prove, e.g., consistency of the estimator.

A way to make this idea precise is to define the model order estimate $\hat{d}_n(y_0, \ldots, y_n)$ as a *penalized maximum likelihood estimator:* we set

$$\hat{d}_n = \operatorname*{argmax}_{d \geq 0} \{\mathbf{L}_n^\star(d) - \varkappa(n, d)\},$$

where $\varkappa(n, d)$ is a given *penalty function* which is strictly increasing in d for every n. The idea is to try to choose $\varkappa(n, d)$ so that it grows less fast with increasing d than does the likelihood $\mathbf{L}_n^\star(d)$ below the true model order $d < d^\star$. As the likelihood levels off after $d > d^\star$, but the penalty $\varkappa(n, d)$ keeps growing, the penalized likelihood $\mathbf{L}_n^\star(d) - \varkappa(n, d)$ will then have a maximum around the true model order $d \approx d^\star$. In essence, the choice of penalty function formalizes how we determine the location of the corner of the likelihood function. The theoretical question is now, of course, how we must choose the penalty function $\varkappa(n, d)$ in order to ensure that the model order estimate is consistent $\hat{d}_n \to d^\star$ as $n \to \infty$. A full development of this idea is quite beyond our scope, but we will sketch some of the necessary ingredients in the next chapter.

Numerical example: General Electric stock prices

We finish this chapter with a brief illustration of the various estimation techniques on real-world data. What we will attempt to do is to fit a regime switching model to historical prices for General Electric Company (NYSE:GE) stock. For our example, we have used the daily closing prices of GE stock in the period of January 1978–December 2007 as a training series. The price data can be obtained free of charge from *Google Finance* (finance.google.com).

Denote by S_k the closing price of GE stock on the kth consecutive trading day since January 3rd, 1978. For the observations of our regime switching model, we choose the sequence of log returns:

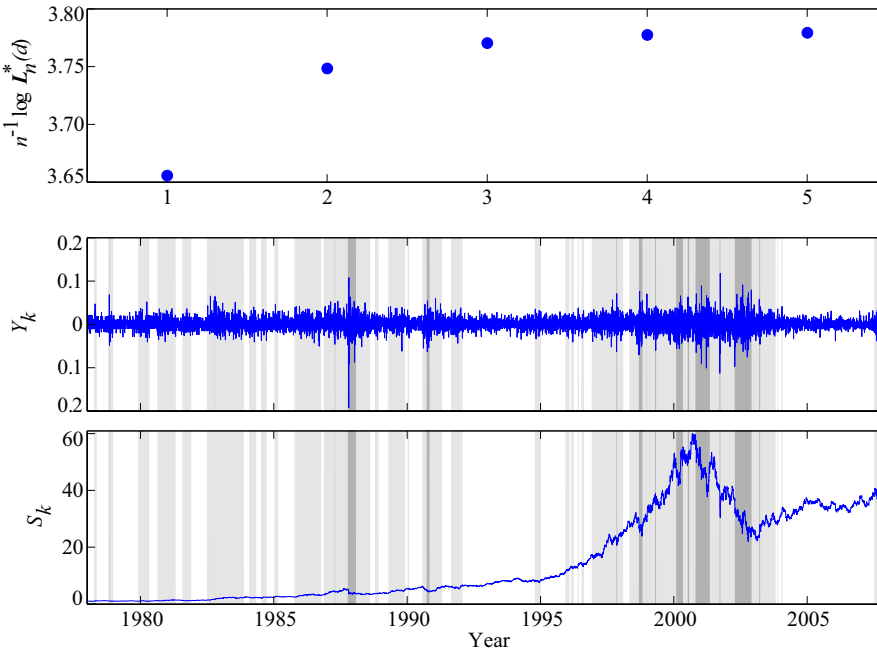

Fig. 6.2. Estimation of regime switching for the daily closing prices of General Electric Company (NYSE:GE) stock in the period January 1978–December 2007. The top plot shows the scaled log-likelihood as a function of model order. The 'corner' of the plot is about $d \approx 3$. The bottom plots show the observation sequence Y_k (i.e., the log returns) and the stock price S_k, respectively. The shading corresponds to the MAP smoothing estimate of the regime for the $d = 3$ model: dark shading is regime 3 (high volatility, negative returns), light shading is regime 2 (medium volatility, high returns), and no shading is regime 1 (low volatility, low returns).

$$Y_k = \log\left[\frac{S_{k+1}}{S_k}\right], \qquad k \geq 0.$$

We model the observation sequence as a regime switching model

$$Y_k = m(X_k) + \sqrt{v(X_k)}\,\eta_k, \qquad k \geq 0,$$

where η_k is an i.i.d. sequence of $N(0,1)$ random variables and X_k is a finite state signal process which represents the regime. Note that in the notation of example 1.12, The volatility is given by $\sigma(X_k) = \sqrt{v(X_k)}$ and the returns are given by $\mu(X_k) = m(X_k) + v(X_k)/2$. To be estimated are the number of regimes, the transition probabilities and the functions m and v.

It is of course not clear, a priori, whether real world stock prices are indeed well represented as a regime switching model. We nonetheless try to estimate the model order as described in this section by computing the likelihood function $\mathbf{L}_n^\star(d)$, and look for the signature of a finite model order: the leveling

off of the likelihood function. To this end, we run the EM algorithm for each model order; as we do not expect the regimes to switch on a daily basis, we choose for every model order an initial guess of $\boldsymbol{P}$ which is close to the identity matrix, and run the EM algorithm for 50 iterations. The likelihood function obtained in this manner is shown as the top plot in figure 6.2. Lo and behold, the plot does indeed level off—it looks like $d \approx 3$ is the corner of the plot. This suggests that a regime switching model of order 3 should form a good description of the statistics of GE prices.

As a by-product of order estimation, we already have an estimate for the order 3 regime switching model. We find the following model parameters:

$$\boldsymbol{P} = \begin{bmatrix} .9901 & .0099 & .0000 \\ .0097 & .9838 & .0065 \\ .0000 & .0368 & .9632 \end{bmatrix}, \qquad \boldsymbol{\mu} = \begin{bmatrix} .9990 \\ .0010 \\ .0000 \end{bmatrix},$$
$$\boldsymbol{m} = \begin{bmatrix} .3833 \\ .8961 \\ -1.392 \end{bmatrix} 10^{-3}, \qquad \boldsymbol{v} = \begin{bmatrix} .0984 \\ .2518 \\ 1.028 \end{bmatrix} 10^{-3}.$$

Note that the three regimes have interesting interpretations. The first is a low return, low volatility regime: a low risk investment. The second is a high return, high volatility regime: a riskier but potentially more rewarding investment. The third regime is one of even higher volatility but negative (!) returns: the signature of a market crash? The bottom plots of figure 6.2 show the smoothed MAP estimates of the regime as a function of time. It is interesting to note that the stock market crash of 1987, as well as two periods of sharp decline after 2001, are estimated as being in the third regime.

Problems

6.1. Bayesian State Augmentation
What are the transition and observation kernels and the initial measure of the augmented hidden Markov model $(\tilde{X}_k, Y_k)_{k\geq 0}$ under the Bayesian measure $\mathbf{P}^\lambda$? (See section 6.1 for the relevant definitions).

6.2. Let $\mathbf{P}, \mathbf{Q}$ be probability measures on $(\Omega, \mathcal{G})$ such that $\mathbf{P}$ is absolutely continuous with respect to $\mathbf{Q}$, and let $\mathcal{G}' \subset \mathcal{G}$ be a sub-σ-field. Show that

$$\frac{d\mathbf{P}|_{\mathcal{G}'}}{d\mathbf{Q}|_{\mathcal{G}'}} = \mathbf{E}_{\mathbf{Q}}\left[\frac{d\mathbf{P}}{d\mathbf{Q}}\,\middle|\,\mathcal{G}'\right].$$

In particular, $\mathbf{P}|_{\mathcal{G}'}$ is also absolutely continuous with respect to $\mathbf{Q}|_{\mathcal{G}'}$.

6.3. Prove proposition 6.5.

6.4. Complete the proof of proposition 6.11.

6.5. Multiple Training Sequences
In our discussion of maximum likelihood estimation, we have presumed that statistical inference is to be performed on the basis of a single observation sequence. However, in many applications one might have multiple independent observation sequences available from the same hidden Markov model. For example, in speech recognition, the training set consists of multiple independent speech samples of the same word or phrase. The training sequences are independent but may have different lengths.
(a) Explain how maximum likelihood estimation works in this setting.
(b) Adapt the EM algorithm to cover this setting.

6.6. The EM Algorithm: Finite Signal and Observation State Spaces
Suppose that $E = \{1, \ldots, d\}$ and $F = \{1, \ldots, d'\}$ are both finite sets.
(a) Work out the details of the EM algorithm for estimating all the transition, observation and initial probabilities $P(i, \{j\})$, $\Phi(i, \{j\})$, $\mu(\{i\})$.
(b) Give a probabilistic interpretation of the EM estimates in terms of the quantities discussed in chapter 3 (recall in particular problem 3.2).

6.7. The EM Algorithm: Linear-Gaussian Models
Develop the EM algorithm in the linear-Gaussian setting of problem 2.5. To be estimated are the matrices A, B, C, D, P_0 and the vectors a, c, μ_0,

6.8. Gradient Based Optimization
We have discussed how to find a (local) maximum of the likelihood $l_n(\theta) = \log \mathbf{L}_n^\theta$ using the EM algorithm. However, one can in principle apply any numerical algorithm for finding the maximum of a function. Typically such algorithms require one to evaluate the derivatives of the objective function. For example, the method of *steepest descent* has us compute iteratively

$$\hat{\theta}_n^{(j+1)} = \hat{\theta}_n^{(j)} + \gamma_j \, \nabla l_n(\hat{\theta}_n^{(j)}),$$

where γ_i are nonnegative constants. For a suitable choice of γ_i, the estimates $\hat{\theta}_n^{(j)}$ are known to converge to a stationary point of $l_n(\theta)$. An alternative that does not require us to find suitable constants γ_i (a nontrivial task) is to apply the Newton-Raphson root finding algorithm to ∇l_n:

$$\hat{\theta}_n^{(j+1)} = \hat{\theta}_n^{(j)} - \nabla^2 l_n(\hat{\theta}_n^{(j)})^{-1} \nabla l_n(\hat{\theta}_n^{(j)}).$$

Here $\hat{\theta}_n^{(j)}$ will converge to a zero of ∇l_n, i.e., to a stationary point of the likelihood (typically a local maximum, hopefully the global maximum).
(a) For the model of proposition 6.11, compute the first derivatives of $l_n(\theta)$ with respect to the model parameters θ. Do the expressions look familiar?
(b) Choose a simple example of a hidden Markov model. Simulate an observation sequence and implement the EM and the Newton-Raphson algorithms to re-estimate the parameters. Compare the performance of the two algorithms.

Remark 6.13. Gradient-based algorithms often converge much faster than the EM algorithm. However, they are more complicated to implement than the EM algorithm and may also be less stable numerically. Moreover, unlike in the EM algorithm, the likelihood is not guaranteed to be nondecreasing with successive iterations of a gradient-based algorithm. Both algorithms are of interest and there is no universal answer to which one is better: this depends on the setting and on the available computational resources.

6.9. Model Order Estimation
(a) Choose a one-state, two-state and three-state hidden Markov model and simulate an observation time series from each. Now run the model order estimation procedure on each of these time series, and show that you are led to select the correct model order in every case.
(b) Estimate the number of regimes in the stock prices of your favorite company. Financial time series can be obtained, e.g., from *Google Finance*.

Notes

The method of maximum likelihood estimation was pioneered by Fisher in the early 20th century. A modern introduction, chiefly in the most common setting with i.i.d. observations, can be found in [van98, van00, IH81]. The case where the observations are generated by a hidden Markov model is made much more difficult by the fact that the observations are not independent. Maximum likelihood estimation in hidden Markov models was first investigated by Baum and Petrie [BP66] for finite signal and observation states spaces.

The EM algorithm for hidden Markov models dates back to Baum, Petrie, Soules and Weiss [BPSW70]. The method uses no special features of hidden Markov models; indeed, it turns out to be a special instance of the general algorithm for maximum likelihood estimation introduced independently by Dempster, Laird and Rubin [DLR77], who coined the term EM algorithm. Other approaches for computing maximum likelihood estimates in hidden Markov models, including the use of Monte Carlo filters when the signal state space is not finite, are reviewed in [CMR05].

Model order estimation in hidden Markov models dates back to Finesso [Fin90]. It is related to the model selection problem in statistics, see [CH08] for an introduction. In practice it is often both mathematically and computationally easier to estimate the model order through a 'quasi-likelihood' approach, as initially suggested by Rydén [Ryd95], rather than computing the full maximum likelihood estimate for every model order. See the notes at the end of the next chapter for further references on this topic. Other methods beside penalized (quasi-)likelihood methods have also been suggested for estimating the model order. For example, Celeux and Durand [CD08] utilize a cross-validation technique, while Cvitanić, Rozovskii and Zaliapin [CRZ06] employ a method that is specifically designed for continuous time observations.

7

Statistical Inference: Consistency

In the previous chapter, we introduced various maximum likelihood based methods for statistical inference. The purpose of this chapter is to give a flavor of the theoretical underpinnings of these methods. This is a challenging topic and, as in our discussion of filter stability in chapter 5, an extensive treatment is beyond our scope. We therefore mainly focus on proving consistency of the maximum likelihood estimator, and we will not hesitate to impose very strong conditions in order to reduce the proofs to the *simplest possible setting*. Many of our assumptions can be weakened, for which we refer to the references given in the notes at the end of the chapter. In addition, some more advanced topics beyond consistency are briefly sketched (without proofs) in section 7.3.

7.1 Consistency of the Maximum Likelihood Estimate

Recall that the maximum likelihood estimate is defined as $\hat{\theta}_n = \text{argmax}_\theta \mathbf{L}_n^\theta$, and our goal is to prove consistency: $\hat{\theta}_n \to \theta^\star$ as $n \to \infty$ $\mathbf{P}^{\theta^\star}$-a.s.

We have already completed a successful trial run in the hypothesis testing setting (theorem 6.6). Though the proof of this theorem can not be adapted to more general models, the basic approach will provide the necessary inspiration. The main idea of the proof of theorem 6.6 can be abstracted as follows:

1. Show that $L_n(\theta) := \mathbf{L}_n^\theta / C_n$ converges $\mathbf{P}^{\theta^\star}$-a.s. as $n \to \infty$ to some limiting random variable $L(\theta)$ for every $\theta \in \Theta$, where C_n is a suitable normalizing process which does not depend on θ.
2. Show that $L(\theta)$ $\mathbf{P}^{\theta^\star}$-a.s. has a unique maximum at $\theta = \theta^\star$.
3. Conclude that $\hat{\theta}_n = \text{argmax}_{\theta\in\Theta} L_n(\theta) \to \text{argmax}_{\theta\in\Theta} L(\theta) = \theta^\star$ $\mathbf{P}^{\theta^\star}$-a.s.

In theorem 6.6 the process C_n was chosen such that $L_n(\theta)$ is the Bayesian conditional probability of the parameter θ at time n for a suitable prior, and the identifiability requirement established that $L(\theta)$ has a unique maximum at $\theta = \theta^\star$. The third step is trivial in the hypothesis testing problem, as in this setting the parameter space Θ is a finite set.

When Θ is not finite, however, the third step is far from obvious: the fact that $L_n(\theta) \to L(\theta)$ for every $\theta \in \Theta$ does not in general guarantee that the maximum of L_n (the maximum likelihood estimate) converges to the maximum of L (the true parameter), which defeats the purpose of proving that $L_n \to L$ in the first place. This is illustrated in the following example.

Example 7.1. Consider the functions

$$f(x) = e^{-x^2}, \qquad f_n(x) = e^{-x^2} + 2\,e^{-(nx-n+\sqrt{n})^2}, \qquad x \in [-1,1].$$

Then $f_n(x) \to f(x)$ as $n \to \infty$ for all $x \in [-1,1]$. However, $\operatorname{argmax}_x f_n(x) \to 1$ as $n \to \infty$, while $\operatorname{argmax}_x f(x) = 0$. Thus $f_n \to f$ pointwise, but the maximum of f_n does not converge to the maximum of f.

Evidently, $L_n(\theta) \to L(\theta)$ for every θ is not enough. However, as the following elementary calculus lemma shows, our problems are resolved if we replace pointwise convergence by *uniform* convergence $\sup_\theta |L_n(\theta) - L(\theta)| \to 0$.

Lemma 7.2. *Suppose Θ is compact. Let $L_n : \Theta \to \mathbb{R}$ be a sequence of continuous functions that converges uniformly to a function $L : \Theta \to \mathbb{R}$. Then*

$$\operatorname*{argmax}_{\theta\in\Theta} L_n(\theta) \to \operatorname*{argmax}_{\theta\in\Theta} L(\theta) \qquad \text{as } n \to \infty.$$

Proof. As a continuous function on a compact space attains its maximum, we can find a (not necessarily unique) $\theta_n \in \operatorname{argmax}_{\theta\in\Theta} L_n(\theta)$ for all n. Then

$$\begin{aligned}
0 &\le \sup_{\theta\in\Theta} L(\theta) - L(\theta_n) = \sup_{\theta\in\Theta}\{L(\theta) - L_n(\theta) + L_n(\theta)\} - L(\theta_n) \\
&\le \sup_{\theta\in\Theta}\{L(\theta) - L_n(\theta)\} + \sup_{\theta\in\Theta} L_n(\theta) - L(\theta_n) \\
&= \sup_{\theta\in\Theta}\{L(\theta) - L_n(\theta)\} + L_n(\theta_n) - L(\theta_n) \le 2 \sup_{\theta\in\Theta} |L(\theta) - L_n(\theta)| \xrightarrow{n\to\infty} 0.
\end{aligned}$$

Suppose that θ_n does not converge to the set of maxima of $L(\theta)$. Then there exists by compactness a subsequence $\{\theta'_m\} \subset \{\theta_n\}$ which converges to $\theta' \notin \operatorname{argmax}_{\theta\in\Theta} L(\theta)$. But $L(\theta)$ is continuous (as $L_n \to L$ uniformly and each L_n is continuous), so $L(\theta'_m) \to L(\theta') < \sup_{\theta\in\Theta} L(\theta)$. This is a contradiction. □

With our new insight, the outline of a consistency proof now looks as follows: (i) define $L_n(\theta)$ so that it converges *uniformly* to a limit $L(\theta)$; and (ii) prove that $L(\theta)$ $\mathbf{P}^{\theta^\star}$-a.s. has a unique maximum at $\theta = \theta^\star$. It is here, however, that the real difficulties of the general setting enter the picture: uniform convergence is not so easy to achieve. For example, when Θ is a continuous space, the Bayesian normalization used in the hypothesis testing problem can not lead to uniform convergence. Indeed, in this case $L_n(\theta)$ is the Bayesian density of the parameter θ with respect to the prior distribution λ (proposition 6.5). If the estimator is consistent, then the Bayesian conditional

distribution of the parameter should converge to a point mass at $\theta^\star$; therefore the density $L_n(\theta)$ should converge to a nonzero value only if $\theta = \theta^\star$, so that the convergence $L_n(\theta) \to L(\theta)$ can certainly not be uniform in θ.

We have thus arrived at the key difficulty of proving consistency in the general setting: we must find a replacement for the quantity $L_n(\theta)$ which converges *uniformly* to a limit. Remarkably, the appropriate notion comes from an unexpected source: the classical Shannon-McMillan-Breiman (SMB) theorem in *information theory*. For the version of this theorem that is suitable for our purposes, we require the following (see definition 6.3 for notation).

Assumption 7.3 *The following hold.*

1. *Θ is a compact subset of $\mathbb{R}^p$.*
2. *There is a $0 < \varepsilon < 1$ and a family of probability measures ρ^θ such that*
$$\varepsilon\, \rho^\theta(A) \le P^\theta(x, A) \le \varepsilon^{-1}\, \rho^\theta(A) \qquad \text{for all } x \in E,\ A \in \mathcal{E},\ \theta \in \Theta.$$
3. *There is a constant $0 < \kappa < 1$ such that*
$$\kappa \le \Upsilon^\theta(x, y) \le \kappa^{-1} \qquad \text{for all } x \in E,\ y \in F,\ \theta \in \Theta.$$
4. *P^θ and Υ^θ are Lipschitz: for some $c_1, c_2 > 0$*
$$\sup_{x \in E} \sup_{A \in \mathcal{E}} |P^\theta(x, A) - P^{\theta'}(x, A)| \le c_1\, \|\theta - \theta'\|,$$
$$\sup_{x \in E} \sup_{y \in F} |\Upsilon^\theta(x, y) - \Upsilon^{\theta'}(x, y)| \le c_2\, \|\theta - \theta'\|.$$
5. *The initial measures μ^θ are stationary:*
$$\mu^\theta(A) = \int \mu^\theta(dx)\, P^\theta(x, A) \qquad \text{for all } A \in \mathcal{E},\ \theta \in \Theta.$$

Remark 7.4. The stationarity assumption on μ^θ is natural, but all our results hold also without this assumption. See problem 7.2.

Proposition 7.5 (Uniform SMB). *Define $\ell_n(\theta) = n^{-1} \log \mathbf{L}_n^\theta$, and suppose assumption 7.3 holds. Then $\ell_n(\theta)$ is continuous and $\ell(\theta) = \lim_{n\to\infty} \ell_n(\theta)$ exists $\mathbf{P}^{\theta^\star}$-a.s. for every $\theta \in \Theta$. Moreover, $\ell_n \to \ell$ uniformly $\mathbf{P}^{\theta^\star}$-a.s.*

Before we prove this result, let us complete the proof of consistency.

Theorem 7.6 (Consistency). *Suppose that assumption 7.3 holds and that the following* identifiability condition *holds true:*

$$\ell(\theta) \text{ has a unique maximum at } \theta = \theta^\star \quad \mathbf{P}^{\theta^\star}\text{-a.s.}$$

Then the maximum likelihood estimate is consistent.

Proof. Note that the maximum likelihood estimate can be written as $\hat\theta_n = \operatorname{argmax}_{\theta\in\Theta} \ell_n(\theta)$. The result follows from proposition 7.5 and lemma 7.2. □

In the next section we will investigate further the identifiability condition in theorem 7.6 and discuss how one might go about verifying it. The remainder of this section is devoted to the proof of proposition 7.5.

A law of large numbers

The basis for our proof is the following representation:

$$\ell_n(\theta) = \frac{1}{n}\sum_{k=0}^{n} \log\left[\int \Upsilon^\theta(x, Y_k)\, \pi^\theta_{k|k-1}(Y_0, \ldots, Y_{k-1}, dx)\right] := \frac{1}{n}\sum_{k=0}^{n} D^\theta_k,$$

where we have used the convention $\pi^\theta_{0|-1}(dx) = \mu^\theta(dx)$. This expression can be read off directly from proposition 6.4.

Apparently the quantity $\ell_n(\theta)$ can be written as a time average of the random variables D^θ_k. Limit theorems for time averages of random variables are called *laws of large numbers (LLN)*. For independent random variables, for example, we encountered a type of LLN as lemma 4.7. The random variables D^θ_k are not independent, so we will use the following LLN instead.

Lemma 7.7 (LLN). *Let $(Z_k)_{k\geq 0}$ be a sequence of random variables such that $|\mathbf{E}(Z_k|Z_0, \ldots, Z_\ell)| \leq C\,\rho^{k-\ell}$ a.s. for all $0 \leq \ell \leq k$ and some constants $C > 0$, $0 < \rho < 1$. Then $S_n := n^{-1}\sum_{k=0}^n Z_k \to 0$ a.s. as $n \to \infty$.*

Proof. We first prove mean square convergence. To this end, note that

$$\mathbf{E}(S_n^2) = \frac{1}{n^2}\sum_{k=0}^{n} \mathbf{E}(Z_k^2) + \frac{2}{n^2}\sum_{k=0}^{n}\sum_{\ell=0}^{k-1} \mathbf{E}(Z_k Z_\ell).$$

But $\mathbf{E}(Z_k^2) \leq C^2$ and $|\mathbf{E}(Z_k Z_\ell)| = |\mathbf{E}(\mathbf{E}(Z_k|Z_0, \ldots, Z_\ell)\, Z_\ell)| \leq C^2\,\rho^{k-\ell}$, so it is easily established that $\mathbf{E}(S_n^2) \leq K/n$ for some $K < \infty$. In particular, $\mathbf{E}(S_n^2) \to 0$. We now strengthen to a.s. convergence. For any $\alpha > 1$ and $\varepsilon > 0$,

$$\sum_{k=1}^{\infty} \mathbf{P}(|S_{\alpha^k}| > \varepsilon) \leq \sum_{k=1}^{\infty} \frac{\mathbf{E}(S_{\alpha^k}^2)}{\varepsilon^2} \leq \frac{K}{\varepsilon^2}\sum_{k=1}^{\infty} \alpha^{-k} < \infty.$$

By the Borel-Cantelli lemma, we find that $S_{\alpha^k} \to 0$ a.s. as $k \to \infty$ for any $\alpha > 1$. For any integer n, denote by $k^\alpha_+(n)$ the smallest integer such that $n \leq \alpha^{k^\alpha_+(n)}$ and by $k^\alpha_-(n)$ the largest integer such that $\alpha^{k^\alpha_-(n)} < n$. Then

$$\frac{\alpha^{k^\alpha_-(n)}}{\alpha^{k^\alpha_+(n)}}\frac{1}{\alpha^{k^\alpha_-(n)}}\sum_{\ell=0}^{\alpha^{k^\alpha_-(n)}} (Z_\ell + C) \leq \frac{1}{n}\sum_{\ell=0}^{n}(Z_\ell + C) \leq \frac{\alpha^{k^\alpha_+(n)}}{\alpha^{k^\alpha_-(n)}}\frac{1}{\alpha^{k^\alpha_+(n)}}\sum_{\ell=0}^{\alpha^{k^\alpha_+(n)}} (Z_\ell + C),$$

where we have used that $Z_\ell + C \geq 0$ a.s. But for n large enough we must evidently have $k^\alpha_+(n) = k^\alpha_-(n) + 1$, so that we obtain

$$C\alpha^{-1} \leq \liminf_{n\to\infty} \frac{1}{n}\sum_{\ell=0}^{n}(Z_\ell + C) \leq \limsup_{n\to\infty} \frac{1}{n}\sum_{\ell=0}^{n}(Z_\ell + C) \leq C\alpha \quad \text{a.s.}$$

As $\alpha > 1$ was arbitrary, we find that $S_n \to 0$ a.s. □

The proof of proposition 7.5 proceeds in three steps. First, we show that

$$\ell(\theta) := \lim_{k\to\infty} \mathbf{E}^{\theta^*}(D_k^\theta) \quad \text{exists for every } \theta \in \Theta.$$

Second, we will show that there exist $C > 0$ and $0 < \rho < 1$ such that

$$|\mathbf{E}^{\theta^\star}(D_k^\theta - \mathbf{E}^{\theta^\star}(D_k^\theta)\,|\,X_0,\ldots,X_\ell,Y_0,\ldots,Y_\ell)| \le C\,\rho^{k-\ell}$$

for all $0 \le \ell \le k$. The law of large numbers then guarantees that

$$\ell_n(\theta) = \frac{1}{n}\sum_{k=0}^{n}\{D_k^\theta - \mathbf{E}^{\theta^\star}(D_k^\theta)\} + \frac{1}{n}\sum_{k=0}^{n}\mathbf{E}^{\theta^\star}(D_k^\theta) \xrightarrow{n\to\infty} \ell(\theta) \quad \mathbf{P}^{\theta^\star}\text{-a.s.}$$

for every $\theta \in \Theta$. Finally, in the third step we will show that this convergence is in fact uniform in θ, thus completing the proof.

Two key consequences of filter stability

From the definition of D_k^θ, it is evident that the long time properties of $\ell_n(\theta)$ are intimately related with the long time properties of the prediction filter $\pi_{k|k-1}^\theta$. It should therefore come as no surprise that the filter stability theory from chapter 5 makes an appearance; indeed, assumption 7.3 was chiefly designed to make this possible (compare with assumption 5.5).

The techniques from chapter 5 will be used in the form of two key lemmas, which we will prove first. The first lemma shows that the quantity D_k^θ, which depends on the observations $Y_0,\ldots,Y_k$, can be approximated uniformly by a function of a fixed number of observations $Y_{k-\ell},\ldots,Y_k$ only.

Lemma 7.8 (Finite memory approximation). *Define for* $0 < \ell < k$

$$D_{k,\ell}^\theta := \log\left[\int \Upsilon^\theta(x,Y_k)\,\pi_{\ell|\ell-1}^\theta(Y_{k-\ell},\ldots,Y_{k-1},dx)\right].$$

If assumption 7.3 holds, then $|D_{k,\ell}^\theta - D_k^\theta| \le 2\,\kappa^{-2}\,\varepsilon^{-2}(1-\varepsilon^2)^\ell$.

Proof. By assumption $\Upsilon^\theta(x,y) \in [\kappa,\kappa^{-1}]$ for some $0 < \kappa < 1$. Using the inequality $|\log(x) - \log(x')| \le \kappa^{-1}|x - x'|$ for all $x, x' \in [\kappa,\kappa^{-1}]$, we estimate

$$\begin{aligned}|D_{k,\ell}^\theta - D_k^\theta| \le \kappa^{-1}\Bigg|&\int \tilde\Upsilon^\theta(x,Y_k)\,\pi_{\ell-1}^\theta(Y_{k-\ell},\ldots,Y_{k-1},dx)\\ &- \int \tilde\Upsilon^\theta(x,Y_k)\,\pi_{k-1}^\theta(Y_0,\ldots,Y_{k-1},dx)\Bigg|,\end{aligned}$$

where we have defined $\tilde\Upsilon^\theta(x,y) = \int \Upsilon^\theta(x',y)\,P^\theta(x,dx')$. But note that

$$\begin{aligned}\pi_{k-1}^\theta(Y_0,\ldots,Y_{k-1},dx) &= \mathsf{F}_{k-1}^\theta\cdots\mathsf{F}_{k-\ell}^\theta\pi_{k-\ell-1}^\theta(dx),\\ \pi_{\ell-1}^\theta(Y_{k-\ell},\ldots,Y_{k-1},dx) &= \mathsf{F}_{k-1}^\theta\cdots\mathsf{F}_{k-\ell}^\theta\mu^\theta(dx),\end{aligned}$$

where F_n^θ are the filter recursion map as defined in chapter 5. Taking into account $|\tilde\Upsilon(x,y)| \le \kappa^{-1}$, the proof is completed by invoking theorem 5.4. □

The second lemma shows that $\theta \mapsto D_k^\theta$ is Lipschitz continuous *uniformly* in k. This is, of course, similar to the uniform approximation theorem 5.6.

Lemma 7.9 (Equicontinuity). *Suppose that assumption 7.3 holds. Then there is a $K < \infty$ such that $|D_k^\theta - D_k^{\theta'}| \le K\|\theta - \theta'\|$ for all $k \ge 1$.*

Proof. It is easily established as in the proof of lemma 7.8 that

$$|D_k^\theta - D_k^{\theta'}| \le \kappa^{-2} \sup_{\|f\|_\infty \le 1} \left| \int f(x)\, \pi_{k-1}^\theta(dx) - \int f(x)\, \pi_{k-1}^{\theta'}(dx) \right|.$$

But note that for every $\ell \ge 0$, probabilty measure μ and $\theta, \theta' \in \Theta$

$$\begin{aligned}
&\sup_{\|f\|_\infty \le 1} \left| \int f(x)\, \mathsf{F}_\ell^\theta \mu(dx) - \int f(x)\, \mathsf{F}_\ell^{\theta'} \mu(dx) \right| \\
&\quad \le \left| \frac{\tilde\Upsilon^\theta(x,y)}{\int \tilde\Upsilon^\theta(x,y)\, \mu(dx)} - \frac{\tilde\Upsilon^{\theta'}(x,y)}{\int \tilde\Upsilon^{\theta'}(x,y)\, \mu(dx)} \right| \\
&\quad \le \frac{|\tilde\Upsilon^\theta(x,y) - \tilde\Upsilon^{\theta'}(x,y)|}{\int \tilde\Upsilon^\theta(x,y)\, \mu(dx)} + \frac{\tilde\Upsilon^{\theta'}(x,y) \int |\tilde\Upsilon^{\theta'}(x,y) - \int \tilde\Upsilon^\theta(x,y)|\, \mu(dx)}{\int \tilde\Upsilon^\theta(x,y)\, \mu(dx) \int \tilde\Upsilon^{\theta'}(x,y)\, \mu(dx)} \\
&\quad \le (\kappa^{-1} + \kappa^{-3}) \sup_{x,y} |\tilde\Upsilon^{\theta'}(x,y) - \tilde\Upsilon^\theta(x,y)| \\
&\quad \le (\kappa^{-1} + \kappa^{-3})\{c_2 + \kappa^{-1} c_1\} \|\theta - \theta'\|,
\end{aligned}$$

where c_1 and c_2 are defined in assumption 7.3. Moreover, using lemma 5.2

$$\begin{aligned}
&\sup_{\|f\|_\infty \le 1} \left| \int f(x)\, \mu^\theta(dx) - \int f(x)\, \mu^{\theta'}(dx) \right| \\
&\quad = \sup_{\|f\|_\infty \le 1} \left| \int f(x)\, P^\theta(x', dx)\, \mu^\theta(dx') - \int f(x)\, P^{\theta'}(x', dx)\, \mu^{\theta'}(dx') \right| \\
&\quad \le c_1 \|\theta - \theta'\| + (1 - \varepsilon) \sup_{\|f\|_\infty \le 1} \left| \int f(x)\, \mu^\theta(dx) - \int f(x)\, \mu^{\theta'}(dx) \right|,
\end{aligned}$$

so $\sup_{\|f\|_\infty \le 1} |\int f d\mu^\theta - \int f d\mu^{\theta'}| \le \varepsilon^{-1} c_1 \|\theta - \theta'\|$. The proof is now completed by following the same argument as in the proof of theorem 5.6. □

Proof of proposition 7.5

Step 1 (convergence of $\mathbf{E}^{\theta^\star}(\ell_n(\theta))$). Define $\Delta_\ell := \mathbf{E}^{\theta^\star}(D_\ell^\theta)$. As we assume that $\mu^{\theta^\star}$ is stationary, $(Y_k)_{k \ge 0}$ is a stationary stochastic process under $\mathbf{P}^{\theta^\star}$. Therefore $\Delta_\ell = \mathbf{E}^{\theta^\star}(D_{k,\ell}^\theta)$ for any $0 < \ell < k$, and we can estimate

$$|\Delta_{m+n} - \Delta_m| = |\mathbf{E}^{\theta^\star}(D_{m+n}^\theta) - \mathbf{E}^{\theta^\star}(D_{m+n,m}^\theta)| \le 2\, \kappa^{-2}\, \varepsilon^{-2} (1 - \varepsilon^2)^m$$

by lemma 7.8. Thus evidently $\sup_{n \ge 0} |\Delta_{m+n} - \Delta_m| \to 0$ as $m \to \infty$, i.e., Δ_k is a Cauchy sequence and is therefore convergent. By Cesàro's theorem

(problem 7.1), $\mathbf{E}^{\theta^\star}(\ell_n(\theta)) = n^{-1}(\Delta_0 + \cdots + \Delta_n)$ converges also.

Step 2 (convergence of $\ell_n(\theta)$). We have shown that $\ell(\theta) := \lim_{n\to\infty} \mathbf{E}^{\theta^\star}(\ell_n(\theta))$ exists for every $\theta \in \Theta$. We aim to show that in fact $\ell_n(\theta) \to \ell(\theta)$ $\mathbf{P}^{\theta^\star}$-a.s. for every $\theta \in \Theta$. By the LLN, this follows if we can show that

$$|\mathbf{E}^{\theta^\star}(D_k^\theta | X_0, \ldots, X_\ell, Y_0, \ldots, Y_\ell) - \mathbf{E}^{\theta^\star}(D_k^\theta)| \le C\,\rho^{k-\ell}$$

for all $0 \le \ell \le k$ and some constants $C > 0$, $0 < \rho < 1$.

To this end, note that $D^\theta_{k+n,n-1} = f_n(Y_{k+1}, \cdots, Y_{k+n})$ for fixed $n > 1$ and all $k \ge 0$. By the hidden Markov property (definition 1.6), there is a function g_n such that $\mathbf{E}^{\theta^\star}(D^\theta_{\ell+n,n-1} | X_0, \ldots, X_\ell, Y_0, \ldots, Y_\ell) = g_n(X_\ell)$ for all $\ell \ge 0$. We claim that for any $n > 1$ and $\ell, m \ge 0$ the following estimate holds:

$$|\mathbf{E}^{\theta^\star}(D^\theta_{\ell+m+n,n-1} | X_0, \ldots, X_\ell, Y_0, \ldots, Y_\ell) - \mathbf{E}^{\theta^\star}(D^\theta_{\ell+m+n,n-1})| \le 2\kappa^{-1}(1-\varepsilon)^m.$$

Indeed, this follows from lemma 5.2, the Markov property of X_k, and the tower property of the conditional expectation. Therefore by lemma 7.8

$$|\mathbf{E}^{\theta^\star}(D^\theta_{\ell+m+n} | X_0, \ldots, X_\ell, Y_0, \ldots, Y_\ell) - \mathbf{E}^{\theta^\star}(D^\theta_{\ell+m+n})| \le 2\,\kappa^{-1}\,(1-\varepsilon)^m + 4\,\kappa^{-2}\,\varepsilon^{-2}(1-\varepsilon^2)^{n-1}.$$

Substituting $m = n-2$ and $m = n-1$, respectively, we can estimate

$$|\mathbf{E}^{\theta^\star}(D^\theta_{\ell+k} | X_0, \ldots, X_\ell, Y_0, \ldots, Y_\ell) - \mathbf{E}^{\theta^\star}(D^\theta_{\ell+k})| \le C_0\,(1-\varepsilon^2)^{k/2-1}$$

for all $k \ge 2$, where $C_0 = 2\,\kappa^{-1} + 4\,\kappa^{-2}\,\varepsilon^{-2}$. The condition of the LLN is now easily verified by setting $\rho = \sqrt{1-\varepsilon^2}$ and choosing C sufficiently large.

Step 3 (uniform convergence of $\ell_n(\theta)$). By lemma 7.9, $\ell_n(\theta)$ is Lipschitz continuous for every n. As $\ell_n(\theta) \to \ell(\theta)$ $\mathbf{P}^{\theta^\star}$-a.s. for every $\theta \in \Theta$, evidently $\ell(\theta)$ is Lipschitz continuous also with the same Lipschitz constant.

As Θ is compact, it can be covered by a finite number of balls of radius δ for any given $\delta > 0$. Thus there exists for every $\delta > 0$ a finite collection of points $\Theta_\delta \subset \Theta$, $\#\Theta_\delta < \infty$ such that every $\theta \in \Theta$ is within distance δ from one of the points in Θ_δ. By lemma 7.9 we can estimate

$$\sup_{\theta\in\Theta} |\ell_n(\theta) - \ell(\theta)| \le 2\delta + \max_{\theta\in\Theta_\delta} |\ell_n(\theta) - \ell(\theta)|.$$

As $\ell_n \to \ell$ pointwise and Θ_δ is a finite set,

$$\limsup_{n\to\infty} \sup_{\theta\in\Theta} |\ell_n(\theta) - \ell(\theta)| \le 2\delta \quad \mathbf{P}^{\theta^\star}\text{-a.s.}$$

But $\delta > 0$ was arbitrary, so $\ell_n \to \ell$ uniformly. □

7.2 Identifiability

Our main consistency theorem 7.6 states that the maximum likelihood estimate is consistent provided that the the model is identifiable in the sense that $\ell(\theta)$ has a unique maximum at $\theta = \theta^\star$ $\mathbf{P}^{\theta^\star}$-a.s. This requirement should seem rather mysterious: why would one expect this to be the case? And even so, how does one verify this in practice? The purpose of this section is to reduce the abstract identifiability condition to a much more intuitive statement, and to show how the condition might be verified.

Our treatment of identifiability is based on the following observation. Note that $\ell(\theta) = \lim_{n\to\infty} \mathbf{E}^{\theta^\star}(\ell_n(\theta))$ $\mathbf{P}^{\theta^\star}$-a.s., as was established in the proof of proposition 7.5. We therefore have $\mathbf{P}^{\theta^\star}$-a.s.

$$\ell(\theta^\star) - \ell(\theta) = \lim_{n\to\infty} \frac{1}{n} \mathbf{E}^{\theta^\star}\left(\log\left[\frac{\mathbf{L}_n^{\theta^\star}}{\mathbf{L}_n^{\theta}}\right]\right) = \lim_{n\to\infty} \frac{1}{n} \mathbf{E}^{\theta^\star}\left(\log\left[\frac{d\mathbf{P}^{\theta^\star}|_{Y_0,\ldots,Y_n}}{d\mathbf{P}^{\theta}|_{Y_0,\ldots,Y_n}}\right]\right).$$

The quantity on the right is a familiar quantity in information theory.

Definition 7.10. *For any two probability measures* $\mathbf{P}$ *and* $\mathbf{Q}$*, the quantity*

$$D(\mathbf{P}||\mathbf{Q}) = \begin{cases} \mathbf{E}_{\mathbf{P}}\left(\log\left[\frac{d\mathbf{P}}{d\mathbf{Q}}\right]\right) & \text{if } \mathbf{P} \ll \mathbf{Q}, \\ \infty & \text{otherwise}, \end{cases}$$

is called the relative entropy *(or* Kullback-Leibler divergence*) between* $\mathbf{P}, \mathbf{Q}$*.*

As we will shortly see, the relative entropy can be seen as a measure of distance between probability measures. Evidently the quantity

$$\ell(\theta^\star) - \ell(\theta) = \lim_{n\to\infty} \frac{1}{n} D(\mathbf{P}^{\theta^\star}|_{Y_0,\ldots,Y_n}||\mathbf{P}^{\theta}|_{Y_0,\ldots,Y_n}) \quad \mathbf{P}^{\theta^\star}\text{-a.s.}$$

represents the rate of growth of the relative entropy distance between the laws of the observation process over an increasing time horizon. This quantity is therefore known as the *relative entropy rate* between the laws of the observations $(Y_k)_{k\geq 0}$ under $\mathbf{P}^{\theta^\star}$ and $\mathbf{P}^{\theta}$. To establish identifiability, our aim is to show that $\ell(\theta^\star) - \ell(\theta) > 0$ for $\theta \neq \theta^\star$: this is equivalent to the statement that $\ell(\theta)$ has a unique maximum at $\theta = \theta^\star$. To this end, we will need some elementary properties of the relative entropy.

Lemma 7.11. *For any probability measures* $\mathbf{P}$ *and* $\mathbf{Q}$*, the following hold.*

1. $D(\mathbf{P}||\mathbf{Q}) \geq 0$ *and* $D(\mathbf{P}||\mathbf{P}) = 0$.
2. *If* $\sup_{n\geq 0} D(\mathbf{P}|_{Y_0,\ldots,Y_n}||\mathbf{Q}|_{Y_0,\ldots,Y_n}) < \infty$*, then* $\mathbf{P}|_{(Y_k)_{k\geq 0}} \ll \mathbf{Q}|_{(Y_k)_{k\geq 0}}$.

Remark 7.12. $D(\mathbf{P}||\mathbf{Q})$ can be seen as a measure of distance between probability measures in the sense that it is nonnegative and vanishes only if $\mathbf{P} = \mathbf{Q}$. Note, however, that it is not a true distance in the mathematical sense, as it is not symmetric in $\mathbf{P}$ and $\mathbf{Q}$ and does not satisfy the triangle inequality.

Proof. That $D(\mathbf{P}||\mathbf{P}) = 0$ is trivial. To prove that $D(\mathbf{P}||\mathbf{Q}) \geq 0$, it suffices to assume $\mathbf{P} \ll \mathbf{Q}$. As $f(x) = x \log x$ is convex, Jensen's inequality gives

$$D(\mathbf{P}||\mathbf{Q}) = \mathbf{E}_{\mathbf{Q}}\left(f\left(\frac{d\mathbf{P}}{d\mathbf{Q}}\right)\right) \geq f\left(\mathbf{E}_{\mathbf{Q}}\left(\frac{d\mathbf{P}}{d\mathbf{Q}}\right)\right) = f(1) = 0.$$

Now define the function $f^+(x) = x \log^+ x$ ($\log^+ x = \max(\log x, 0)$), and note that $|f^+(x) - f(x)| \leq \exp(-1)$ for all x. Therefore

$$\sup_{n\geq 0} \mathbf{E}_{\mathbf{Q}}\left(f^+\left(\frac{d\mathbf{P}|_{Y_0,\ldots,Y_n}}{d\mathbf{Q}|_{Y_0,\ldots,Y_n}}\right)\right) \leq \exp(-1) + \sup_{n\geq 0} D(\mathbf{P}|_{Y_0,\ldots,Y_n}||\mathbf{Q}|_{Y_0,\ldots,Y_n}).$$

It is a well known fact in measure-theoretic probability that the finiteness of the left hand side implies $\mathbf{P}|_{(Y_k)_{k\geq 0}} \ll \mathbf{Q}|_{(Y_k)_{k\geq 0}}$; e.g., [Shi96, page 527]. □

We are now armed to prove our key identifiability theorem.

Theorem 7.13 (Identifiability). *If assumption 7.3 holds, then* $\mathbf{P}^{\theta^\star}$*-a.s.*

1. $\ell(\theta) \leq \ell(\theta^\star)$ *for all* $\theta \in \Theta$*; and*
2. $\ell(\theta) = \ell(\theta^\star)$ *if and only if* $\mathbf{P}^{\theta}|_{(Y_k)_{k\geq 0}} = \mathbf{P}^{\theta^\star}|_{(Y_k)_{k\geq 0}}$.

In particular, if every $\theta \in \Theta$ *gives rise to a distinct law of the observations* $\mathbf{P}^{\theta}|_{(Y_k)_{k\geq 0}}$*, then* $\theta^\star$ *is the unique maximum of* $\ell(\theta)$ $\mathbf{P}^{\theta^\star}$*-a.s.*

Proof. As relative entropy is nonnegative, it is immediate that the relative entropy rate $\ell(\theta^\star) - \ell(\theta)$ is nonnegative. This establishes the first claim.

We now turn to the second claim. Note that by the definition of the relative entropy rate and the property $D(\mathbf{P}||\mathbf{P}) = 0$ of the relative entropy, $\mathbf{P}^{\theta^\star}|_{(Y_k)_{k\geq 0}} = \mathbf{P}^{\theta}|_{(Y_k)_{k\geq 0}}$ clearly implies that $\ell(\theta^\star) = \ell(\theta)$. The converse statement is much less trivial, and we will prove it in two steps. In the first step, we will show that $\ell(\theta^\star) = \ell(\theta)$ implies that $\mathbf{P}^{\theta^\star}|_{(Y_k)_{k\geq 0}} \ll \mathbf{P}^{\theta}|_{(Y_k)_{k\geq 0}}$. In the second step, we will prove that the latter implies $\mathbf{P}^{\theta^\star}|_{(Y_k)_{k\geq 0}} = \mathbf{P}^{\theta}|_{(Y_k)_{k\geq 0}}$.

Step 1. Suppose that $\ell(\theta^\star) = \ell(\theta)$. Then

$$\begin{aligned}|\mathbf{E}^{\theta^\star}(D_n^{\theta^\star} - D_n^{\theta})| &= |\mathbf{E}^{\theta^\star}(D_n^{\theta^\star} - D_n^{\theta}) - \ell(\theta^\star) + \ell(\theta)| \\ &\leq |\mathbf{E}^{\theta^\star}(D_n^{\theta^\star}) - \ell(\theta^\star)| + |\mathbf{E}^{\theta^\star}(D_n^{\theta}) - \ell(\theta)| \leq 4\kappa^{-2}\varepsilon^{-2}(1-\varepsilon^2)^n,\end{aligned}$$

where the latter estimate was established in the first step of the proof of proposition 7.5. Defining $K = 4\kappa^{-2}\varepsilon^{-2}\sum_{k=0}^{\infty}(1-\varepsilon^2)^k < \infty$, we can write

$$D(\mathbf{P}^{\theta^\star}|_{Y_0,\ldots,Y_n}||\mathbf{P}^{\theta}|_{Y_0,\ldots,Y_n}) = \sum_{k=0}^{n}\mathbf{E}^{\theta^\star}(D_k^{\theta^\star} - D_k^{\theta}) \leq 4\kappa^{-2}\varepsilon^{-2}\sum_{k=0}^{n}(1-\varepsilon^2)^k < K$$

for all $n \geq 0$. That $\mathbf{P}^{\theta^\star}|_{(Y_k)_{k\geq 0}} \ll \mathbf{P}^{\theta}|_{(Y_k)_{k\geq 0}}$ follows from lemma 7.11.

Step 2. We now suppose that $\mathbf{P}^{\theta^\star}|_{(Y_k)_{k\geq 0}} \neq \mathbf{P}^{\theta}|_{(Y_k)_{k\geq 0}}$. We will show that under this assumption the laws of $(Y_k)_{k\geq 0}$ under $\mathbf{P}^{\theta^\star}$ and $\mathbf{P}^{\theta}$ are mutually

singular. This implies, conversely, that if the laws of the observations are absolutely continuous, then they must in fact be equal.

When the laws of the observations under $\mathbf{P}^{\theta^\star}$ and $\mathbf{P}^\theta$ are not equal, there exists an $n < \infty$ and a bounded function f such that $\mathbf{E}^{\theta^\star}(f(Y_1, \ldots, Y_n)) \neq \mathbf{E}^\theta(f(Y_1, \ldots, Y_n))$. Define $Z_k = f(Y_{k+1}, \ldots, Y_{k+n})$, and note that by stationarity $\mathbf{E}^{\theta'}(Z_k) = \mathbf{E}^{\theta'}(f(Y_1, \ldots, Y_n))$ for all k and θ'. Moreover, we can establish as in the second step of the proof of proposition 7.5 that for every $\theta' \in \Theta$

$$|\mathbf{E}^{\theta'}(Z_k|Z_0, \ldots, Z_\ell) - \mathbf{E}^{\theta'}(Z_k)| \leq C\,\rho^{k-\ell}$$

for some $C > 0$, $0 < \rho < 1$. Therefore, by the LLN,

$$\frac{1}{r}\sum_{k=0}^{r} Z_k \xrightarrow{n\to\infty} \mathbf{E}^{\theta'}(f(Y_1, \ldots, Y_n)) \quad \mathbf{P}^{\theta'}\text{-a.s.}$$

for every $\theta' \in \Theta$. In particular, the event $\{\frac{1}{r}\sum_{k=0}^{r} Z_k \to \mathbf{E}^{\theta^\star}(f(Y_1, \ldots, Y_n))\}$ has unit probability under $\mathbf{P}^{\theta^\star}$ and zero probability under $\mathbf{P}^\theta$. Thus evidently the laws of $(Y_k)_{k\geq 0}$ under $\mathbf{P}^{\theta^\star}$ and $\mathbf{P}^\theta$ are mutually singular. □

Evidently the identifiability condition of theorem 7.6 is much more natural than would initially seem: indeed, it is the weakest possible type of assumption, as obviously no inference procedure can distinguish between two models which give rise to the same observations (of course, assumption 7.3 can be weakened significantly). Note that identifiability in our setting is in fact the same condition as in the hypothesis testing problem of theorem 6.6, once we note that under assumption 7.3 two distinct observation laws are automatically mutually singular—this is precisely the second step in the above proof.

In fact, the above theorem allows us to stengthen theorem 7.6 somewhat. The proof of the following is an immediate extension of theorem 7.6.

Corollary 7.14 (Consistency). *Suppose that assumption 7.3 holds, and let $\hat{\theta}_n$ be a sequence of maximum likelihood estimates. Then $\hat{\theta}_n$ converges $\mathbf{P}^{\theta^\star}$-a.s. to the set of parameters which give rise to the same observation law as $\theta^\star$.*

To wrap up our discussion, let us give an example.

Example 7.15. Let the signal state space $E = \{1, \ldots, d\}$ be finite, and suppose that the observations are real-valued and take the form $Y_k = h(X_k) + \sigma\xi_k$ where ξ_k are i.i.d. $N(0, 1)$ and $\sigma > 0$. The parameter space Θ consists of all transition probabilities $\boldsymbol{P}_{ij}$ of the signal, the noise variance σ, and all observation values $\boldsymbol{h}_i = h(i)$ which we presume to be distinct $\boldsymbol{h}_i \neq \boldsymbol{h}_j$ for $i \neq j$. We have seen various examples of this type of model in chapter 6. Note, however, that this example does not satisfy the strong assumption 7.3 that we have made throughout this chapter. An example of a model that does satisfy our assumptions is given as problem 7.3 below.

We would like to investigate when two distinct parameters $\theta, \theta' \in \Theta$ give rise to the same observation law. First, note that under any $\mathbf{P}^\theta$, the characteristic function of Y_k can be written as

$$\mathbf{E}^\theta(e^{i\lambda Y_k}) = \mathbf{E}^\theta(e^{i\lambda\sigma\xi_k})\,\mathbf{E}^\theta(e^{i\lambda h(X_k)}) = e^{-\sigma^2\lambda^2/2}\sum_{i=1}^{d}\mathbf{P}^\theta(X_k = i)\,e^{i\lambda \boldsymbol{h}_i},$$

i.e., a Gaussian envelope times a purely oscillatory term. Note that σ can be uniquely determined from the law of Y_k—it is the unique $\sigma > 0$ such that $e^{\sigma^2\lambda^2/2}\,\mathbf{E}^\theta(e^{i\lambda Y_k})$ neither converges to zero nor diverges as $\lambda \to \infty$. Therefore, if the laws of the observations under $\mathbf{P}^\theta$ and $\mathbf{P}^{\theta'}$ are the same, then $\sigma = \sigma'$.

Using the same technique, we can investigate the multivariate laws:

$$\mathbf{E}^\theta(e^{i\{\lambda_0 Y_0 + \cdots + \lambda_n Y_n\}}) = \mathbf{E}^\theta(e^{i\{\lambda_0 h(X_0) + \cdots + \lambda_n h(X_n)\}})\,e^{-\sigma^2\{\lambda_0^2 + \cdots + \lambda_n^2\}/2}.$$

As σ can be determined uniquely from the observation law, we find that if the laws of the observations under $\mathbf{P}^\theta$ and $\mathbf{P}^{\theta'}$ are the same, then the law of the process $h(X_k)$ under $\mathbf{P}^\theta$ and of the process $h'(X_k)$ under $\mathbf{P}^{\theta'}$ are the same. But as we have assumed that the observation values are distinct, it must be the case that $(\boldsymbol{P}, \boldsymbol{h})$ and $(\boldsymbol{P}', \boldsymbol{h}')$ coincide *up to a permutation of the points in* E. Indeed, if we exchange the transition probabilities of two points in the signal state space, then the law of the observations does not change provided that we also exchange the corresponding observation values.

We therefore conclude that the model in this example is identifiable up to a permutation of the points in the signal state space. A result along the lines of corollary 7.14 (provided that the assumption 7.3 is weakened) would then imply that the maximum likelihood estimate converges to some permutation of the true model. We have indeed already seen precisely this in practice—see figure 6.1 in the previous chapter. Alternatively, one could force the model to be completely identifiable, for example, by restricting Θ to the subset where the observation values are ordered $\cdots < \boldsymbol{h}_i < \boldsymbol{h}_{i+1} < \cdots$.

7.3 Advanced Topics

In the previous sections, we have developed consistency of the maximum likelihood estimate in the simplest possible setting. Even under our strong assumptions, the necessary theory is quite involved. More advanced topics beyond consistency complicate matters even further, and a full treatment is definitely beyond our scope. Nonetheless it is useful to give a flavor of some advanced topics—asymptotic normality, consistency of model order estimation, and local convergence of the EM algorithm—without going into the full details. In this section, we will briefly sketch how one could go about developing these topics. We will mostly outline or skip the proofs, and we refer to the references given in the notes at the end of the chapter for a full development.

Asymptotic Normality

We have shown that, under suitable assumptions, the maximum likelihood estimate is consistent. This means that for large times n, the parameter estimate $\hat{\theta}_n$ is close to the true parameter value $\theta^\star$. However, consistency does not tell us *how* close the estimate is to the true parameter value at a given time n, so that in practice (where n is always finite) it is not entirely clear how reliable the estimate actually is. In many applications it is important to obtain not only a parameter estimate, but also a corresponding *confidence interval* which gives an indication as to how well we can trust the estimate.

Let us briefly recall how confidence intervals are obtained in the simplest statistical setting. Let μ^θ be a family of probability distributions on $\mathbb{R}$ with finite mean m^θ and variance V^θ, and suppose that we observe a sequence $X_1, X_2, \ldots$ of i.i.d. random variables with distribution μ^θ. Then

$$\hat{m}_n = \frac{1}{n}\sum_{k=1}^{n} X_k \xrightarrow{n\to\infty} m^\theta \quad \mathbf{P}^\theta\text{-a.s.}$$

for every θ by the law of large numbers. In particular, $\hat{m}_n$ is a consistent estimator of the mean m^θ. We would now like to estimate how close $\hat{m}_n$ actually is to m^θ. Note that by the central limit theorem

$$\sqrt{n}\{\hat{m}_n - m^\theta\} = \frac{1}{\sqrt{n}}\sum_{k=1}^{n}\{X_k - m^\theta\} \xrightarrow[\mathbf{P}^\theta\text{-weakly}]{n\to\infty} N(0, V^\theta).$$

Therefore, for large n, the estimate $\hat{m}_n$ is approximately distributed as a Gaussian random variable with mean m^θ and variance V^θ/n. The quantiles of this Gaussian distribution then define the corresponding asymptotic confidence intervals; for example, the standard 95% confidence interval is given by $\hat{m}_n \approx m^\theta \pm 1.96\sqrt{V^\theta/n}$. In practice V^θ is not known (as it requires us to know the unknown parameter θ), so that V^θ is replaced by any consistent estimator of V^θ such as the empirical variance.

In order to extend this idea to maximum likelihood estimation, we would have to prove that for some (co)variance matrix Σ (which may depend on $\theta^\star$)

$$\sqrt{n}\{\hat{\theta}_n - \theta^\star\} \xrightarrow[\mathbf{P}^{\theta^\star}\text{-weakly}]{n\to\infty} N(0, \Sigma).$$

When this is the case, the maximum likelihood estimate is said to be *asymptotically normal*, and confidence intervals can be obtained along the same lines as in the i.i.d. case as described above.

There is a standard trick that is used to prove asymptotic normality of maximum likelihood estimates. The idea is that the first derivatives of a smooth function must vanish at its maximum. Let us presume that regularity conditions have been imposed so that $\ell_n(\theta)$ is sufficiently smooth. Then

$$0 = \nabla\ell_n(\hat{\theta}_n) = \nabla\ell_n(\theta^\star) + \nabla^2\ell_n(\theta^\star)\{\hat{\theta}_n - \theta^\star\} + R_n(\hat{\theta}_n, \theta^\star),$$

where we have Taylor expanded the likelihood gradient $\nabla \ell_n$ to first order around $\theta^\star$. In particular, we find that

$$\sqrt{n}\{\hat{\theta}_n - \theta^\star\} = -(\nabla^2 \ell_n(\theta^\star))^{-1}\{\nabla \ell_n(\theta^\star) + R_n(\hat{\theta}_n, \theta^\star)\}\sqrt{n}.$$

To establish asymptotic normality, it then suffices to prove the following:

$$-\nabla^2 \ell_n(\theta^\star) \xrightarrow[\mathbf{P}^{\theta^\star}-\text{a.s.}]{n\to\infty} J(\theta^\star), \qquad \sqrt{n}\,\nabla \ell_n(\theta^\star) \xrightarrow[\mathbf{P}^{\theta^\star}-\text{weakly}]{n\to\infty} N(0, J(\theta^\star)),$$

and $R_n(\hat{\theta}_n, \theta^\star)\sqrt{n} \to 0$, in order to establish asymptotic normality with covariance matrix $\Sigma = J(\theta^\star)^{-1}$ (in order to compute confidence intervals in practice one may now replace the unknown quantity $J(\theta^\star)$ by the computable quantity $-\nabla^2 \ell_n(\hat{\theta}_n)$). This procedure is reminiscent of the proof of the Cramér-Rao bound, and it turns out that the matrix $J(\theta^\star)$ can indeed be interpreted as the Fisher information matrix in this setting.

Proving convergence of the derivatives of the likelihood has much in common with our proof of consistency. Indeed, the basic approach is mostly the same, except that we must supplement our law of large numbers for dependent random variables (lemma 7.7) with a suitable central limit theorem for dependent random variables. As is to be expected the details of the proof are messy, and we will not go into the matter any further here.

Remark 7.16. An alternative technique for obtaining confidence intervals, which does not require asymptotic normality and may in fact give more precise results, is the parametric bootstrap method (see problem 7.4). The bootstrap can be computationally intensive to compute, however.

Model Order Estimation

We now turn to the model order estimation discussed in the previous chapter. Recall that in this case the signal state space $E_d = \{1, \ldots, d\}$ is a finite set, but the *model order* d is not known in advance. In this case the parameter space is $\Theta = \bigcup_{d \geq 0} \Theta_d$, where Θ_d is the parameter set for the models with fixed order d (i.e., a point in Θ_d consists of all possible transition probabilities and observation parameters for a hidden Markov model of order d).

Recall that if $\theta^\star \in \Theta_{d^\star}$ is the true model parameter, then there exists for every $d > d^\star$ a parameter $\theta \in \Theta_d$ which gives rise to the same observation law. The model order estimation problem is therefore inherently *non-identifiable*. A consistent estimator would be guaranteed to converge to a model parameter with the correct observation law, but this parameter might well be of a much larger model order than is necessary to describe the observed training data. Our goal is therefore to find an estimator $\hat{\theta}_n$ which is not only consistent, but also gives rise (as $n \to \infty$) to a parameter estimate with the smallest possible order. In other words, we would like to estimate the smallest integer d such that the observation law can be described by a hidden Markov model of order d—we will refer to this quantity as the *true* model order $d^\star$.

Let us define $\ell_n^\star(d) = \max_{\theta\in\Theta_d} \ell_n(\theta)$. The maximizer $\hat{\theta}_n(d)$ in this expression is the maximum likelihood estimate of order d. Because there exists for every $d' > d$ a $\theta' \in \Theta_{d'}$ with the same observation law as $\hat{\theta}_n(d)$, and hence with the same likelihood, the likelihood function $\ell_n^\star(d)$ is nondecreasing with increasing model order d. Moreover, assuming that the maximum likelihood estimates are consistent, it will be the case that $\lim_{n\to\infty} \ell_n^\star(d) := \ell^\star(d)$ satisfies $\ell^\star(d) = \ell^\star(d^\star)$ for all $d > d^\star$. In other words, for large n, the likelihood function $\ell_n^\star(d)$ is increasing for $d < d^\star$ and is flat for $d \geq d^\star$.

How to estimate $d^\star$? As discussed in the previous chapter, a promising idea is to define the order estimate $\hat{d}_n$ as a *penalized* maximum likelihood estimate

$$\hat{d}_n = \operatorname*{argmax}_{d\geq 0} \{\ell_n^\star(d) - \zeta(n)\,\iota(d)\},$$

where the penalty functions ζ and ι are to be chosen such that $\hat{d}_n \to d^\star$ $\mathbf{P}^{\theta^\star}$-a.s. We are now going to argue how this can be done, albeit—with apologies to the reader—with a lot of handwaving and imprecision.

The essential idea is to require the following three conditions:

1. $\iota(d)$ is a strictly increasing function.
2. $\zeta(n) \to 0$ as $n \to \infty$.
3. $\{\ell_n^\star(d^\star) - \ell_n^\star(d)\}/\zeta(n) \to 0$ as $n \to \infty$ $\mathbf{P}^{\theta^\star}$-a.s. for $d \geq d^\star$.

Let us show the relevance of these conditions. First, note that

$$\ell_n^\star(d) - \zeta(n)\,\iota(d) \xrightarrow{n\to\infty} \ell^\star(d).$$

As the latter is flat for $d \geq d^\star$ but is increasing for $d < d^\star$, the order estimate will satisfy $\hat{d}_n \geq d^\star$ for large enough n. In other words, as $n \to \infty$, the order estimate will not *under*estimate the true model order. On the other hand,

$$\frac{\{\ell_n^\star(d^\star) - \zeta(n)\,\iota(d^\star)\} - \{\ell_n^\star(d) - \zeta(n)\,\iota(d)\}}{\zeta(n)} \xrightarrow{n\to\infty} \iota(d) - \iota(d^\star)$$

for $d > d^\star$. As ι is strictly increasing, the right hand side is strictly positive, so that evidently for large n we have $\hat{d}_n \leq d^\star$. In other words, as $n \to \infty$, the order estimate will not *over*estimate the true model order. This can only imply that $\hat{d}_n \to d^\star$ as $n \to \infty$, which is precisely what we want to show.

Remark 7.17. Note that without further assumptions these claims only work if we impose and upper bound on d: otherwise we have to prove that the convergence statements hold uniformly in d, as was of essence in lemma 7.2.

The main difficulty is now to choose $\zeta(n)$ such that the third condition above holds. Note that as $\ell^\star(d) = \ell^\star(d^\star)$ for $d > d^\star$, we have $\ell_n^\star(d^\star) - \ell_n^\star(d) \to 0$ as $n \to \infty$ for $d > d^\star$. In essence, we would like to show that $\zeta(n)$ converges to zero at a slower rate than $\ell_n^\star(d^\star) - \ell_n^\star(d)$. We must therefore try to estimate the latter rate. Heuristic arguments based on the law of iterated logarithm

lead one to expect that this rate is of order $O(\log\log n/n)$, in which case one could choose something like $\zeta(n) = \log n/n$. In some particular cases this argument can be made rigorous, though a detailed development of the necessary technicalities is quite intricate and is most certainly beyond our scope. It also appears that a completely satisfactory general result has yet to be obtained. The reader is referred to the notes at the end of the chapter.

Convergence of the EM Algorithm

To compute maximum likelihood estimates in practice, we have introduced the EM algorithm in the previous chapter. We have seen that each EM iteration increases the likelihood, but this does not guarantee that repeated iteration of the EM algorithm will cause the parameter estimate to converge to the global maximum of the likelihood. Indeed, this is generally not guaranteed; the EM algorithm may even converge to different limits depending on which initial guess was used for the parameter value.

In this section, we will sketch a simple argument that shows that the EM algorithm converges to a *critical point* of the likelihood (i.e., a point where all the first derivatives of the likelihood vanish) under certain conditions. As the likelihood typically has several local maxima, this implies that the algorithm generally converges to a local maximum. There does not appear to be a simple way to guarantee that this local maximum is actually a global maximum. In practice, one might try to run the algorithm several times started at different initial guesses, and choose the run which leads to the largest likelihood.

Define the map $T : \Theta \to \Theta$ as $T(\theta) = \operatorname{argmax}_{\theta_0 \in \Theta} Q_n(\theta_0, \theta)$, where Q_n is the defining quantity of the EM algorithm. Then the EM algorithm consists of computing iteratively $\hat\theta^j = T(\hat\theta^{j-1})$ from some initial guess $\hat\theta^0 \in \Theta$.

Proposition 7.18 (Local EM convergence). *Assume the following:*

1. *Θ is an open subset of $\mathbb{R}^p$.*
2. *$Q_n(\theta', \theta)$ and $\mathbf{L}_n^{\theta'}$ are continuously differentiable w.r.t. θ' for every θ.*
3. *$Q_n(\theta', \theta)$ is strictly concave in θ' for every θ.*
4. *$Q_n(\theta_0, \theta)$ attains its maximum at a unique point $\theta_0 = T(\theta)$ for every θ.*
5. *The map T is continuous.*

Define $\hat\theta^j$ recursively as $\hat\theta^j = T(\hat\theta^{j-1})$ given an arbitrary initial guess $\hat\theta^0 \in \Theta$. Then every convergent subsequence of $\{\hat\theta^j\}$ converges to a critical point of $\mathbf{L}_n^\theta$.

Proof. Let $j_k \nearrow \infty$ be a sequence such that $\hat\theta^{j_k} \to \hat\theta^\infty$ as $k \to \infty$. As T is continuous, we find that $\hat\theta^{j_k+1} = T(\hat\theta^{j_k}) \to T(\hat\theta^\infty)$. In particular, as the likelihood is continuous and is nondecreasing with respect to T (lemma 6.10),

$$\mathbf{L}_n^{\hat\theta^{j_k}} \le \mathbf{L}_n^{\hat\theta^{j_k+1}} \le \mathbf{L}_n^{\hat\theta^{j_k+1}} \quad \xrightarrow{j\to\infty} \quad \mathbf{L}_n^{\hat\theta^\infty} \le \mathbf{L}_n^{T(\hat\theta^\infty)} \le \mathbf{L}_n^{\hat\theta^\infty}.$$

Therefore $\mathbf{L}_n^{\hat\theta^\infty} = \mathbf{L}_n^{T(\hat\theta^\infty)}$. We claim that this implies that $\hat\theta^\infty = T(\hat\theta^\infty)$.

Indeed, suppose that $T(\theta) \neq \theta$. As $Q_n(\theta', \theta)$ is strictly concave, it has a unique global maximum at $\theta' = T(\theta)$. Therefore $Q_n(T(\theta), \theta) > Q_n(\theta, \theta) = 0$, and $\mathbf{L}_n^{T(\theta)} > \mathbf{L}_n^\theta$ by lemma 6.10. Conversely, $\mathbf{L}_n^{T(\theta)} = \mathbf{L}_n^\theta$ must imply $T(\theta) = \theta$.

It remains to show that every fixed point of the map T is a critical point of the likelihood. Note that as $Q_n(\theta', \theta)$ is continuously differentiable, its derivatives with respect to θ' must vanish at the maximum $\theta' = T(\theta)$. In particular, if $T(\theta) = \theta$, then $\nabla_{\theta'} Q_n(\theta', \theta)|_{\theta'=\theta} = 0$. We claim that $\nabla_{\theta'} Q_n(\theta', \theta)|_{\theta'=\theta} = \nabla \log \mathbf{L}_n^\theta$. Indeed, for fixed $\theta \in \Theta$, the function $f(\theta') := \log \mathbf{L}_n^{\theta'} - \log \mathbf{L}_n^\theta - Q_n(\theta', \theta)$ is continuously differentiable, $f(\theta') \geq 0$ for all θ' by lemma 6.10, and $f(\theta) = 0$. Therefore $\theta' = \theta$ is a minimum of $f(\theta')$, and as Θ is an open set and f is continuously differentiable this implies that $0 = \nabla f(\theta) = \nabla \log \mathbf{L}_n^\theta - \nabla_{\theta'} Q_n(\theta', \theta)|_{\theta'=\theta}$. This establishes the claim. □

The assumptions of this simple result are far from the weakest possible, but the statement is fairly representative of the type of convergence that can be established for the EM algorithm. The assumptions are not difficult to verify for a slightly simplified form of the the model of proposition 6.11 where the observation variance $\boldsymbol{v}$ is presumed to be known and fixed (problem 7.5).

Problems

7.1. Cesàro's theorem
Prove *Cesàro's theorem*: if x_n is a sequence of real numbers that converges $x_n \to x$ as $n \to \infty$, then $n^{-1} \sum_{k=0}^n x_k \to x$ as $n \to \infty$ also.

7.2. Relaxing Stationarity
The mixing assumption on P^θ (the second item in assumption 7.3) guarantees that there exists for every $\theta \in \Theta$ a unique stationary measure $\tilde{\mu}^\theta$. The assumption that $\mu^\theta = \tilde{\mu}^\theta$ for all $\theta \in \Theta$ is therefore natural: one might well expect the hidden Markov model to start off in steady state. On the other hand, the proof of proposition 7.5 requires only minor modifications in order to eliminate the stationarity assumption entirely.
(a) Prove that the mixing assumption on P^θ (the second item in assumption 7.3) implies that there is a unique stationary measure $\tilde{\mu}^\theta$ for every $\theta \in \Theta$. (Hint: use lemma 5.2 and the Banach fixed point theorem.)
(b) Show that $\mu(P^\theta)^k \to \tilde{\mu}^\theta$ as $k \to \infty$ for every initial measure μ and $\theta \in \Theta$.
(c) Modify the proof of proposition 7.5 to show that the result already holds when the last item in assumption 7.3 is replaced by the following assumption: $\sup_{A \in \mathcal{E}} |\mu^\theta(A) - \mu^{\theta'}(A)| \leq c_3 \|\theta - \theta'\|$ for all $\theta, \theta' \in \Theta$.

7.3. Identifiability: Finite Signal and Observation State Spaces
Suppose that the signal and observation state spaces $E = \{1, \ldots, d\}$ and $F = \{1, \ldots, d'\}$ are both finite. Give a simple sufficient condition in this setting for the hidden Markov model to be identifiable.

Remark 7.19. In the setting of problem 7.3 identifiability has been characterized completely: see Ito, Amari and Kobayashi [IAK92]. This necessary and sufficient condition is algebraic in nature and quite complicated. A simple sufficient condition is easily obtained, however.

7.4. Confidence Intervals

In this problem, you are going to investigate numerically two methods for obtaining confidence intervals for maximum likelihood estimates. To keep things simple, let us consider a model with a one-dimensional parameter space $\Theta = [0, 1]$. The signal state space is $E = \{0, 1\}$ with initial measure $\mu(\{0\}) = \mu(\{1\}) = 1/2$ and transition probabilities

$$P(0, \{0\}) = P(1, \{1\}) = \theta, \qquad P(0, \{1\}) = P(1, \{0\}) = 1 - \theta \qquad (\theta \in \Theta).$$

The observation state space is $F = \mathbb{R}$ with $Y_k = X_k + \eta_k$, $\eta_k \sim N(0, 1)$.

Throughout this problem, let us fix a true parameter value $\theta^\star$ and a reasonably large time horizon n. Before we compute confidence intervals on the basis of observed data, let us simulate the exact distribution of the maximum likelihood estimate as a benchmark. Note that in reality this distribution is not computable: after all, $\theta^\star$ is really an unknown parameter.

(a) Simulate a large number of observation sample paths $Y_0, \ldots, Y_n$ under the true model parameter $\theta^\star$, and compute the maximum likelihood estimate $\hat{\theta}_n$ for each path. Plot a histogram of the distribution of $\sqrt{n}\{\hat{\theta}_n - \theta^\star\}$.

Now simulate a single sample path $Y_0, \ldots, Y_n$ of the observations under the true model parameter $\theta^\star$. In the following parts we will obtain approximate confidence intervals on the basis of this observed path only.

(b) *Asymptotic normality* suggests that for large n, the quantity $\sqrt{n}\{\hat{\theta}_n - \theta^\star\}$ is approximately distributed as a Gaussian with zero mean and variance $-(d^2\ell_n(\theta)/d\theta^2)^{-1}|_{\theta=\hat{\theta}_n}$. Obtain an expression for this quantity and compute it for the observed path. Plot the resulting Gaussian distribution and compare with the histogram obtained in part (a).

A different method to obtain approximate confidence intervals is the *parametric bootstrap*. This works as follows. First, compute the maximum likelihood estimate $\hat{\theta}_n$ on the basis of the observations. Next, repeat part (a) of this problem under the assumption that $\hat{\theta}_n$ is the true model parameter value. Note that this procedure does *not* depend on the *actual* parameter value $\theta^\star$. As $\hat{\theta}_n$ is close to $\theta^\star$ for large n, the parametric bootstrap distribution should be close to the actual distribution of $\sqrt{n}\{\hat{\theta}_n - \theta^\star\}$ for large n.

(c) Compute the parametric bootstrap distribution given our observed path. Compare with the exact and approximate distributions in parts (a) and (b).

7.5. Convergence of the EM Algorithm

Verify that the assumptions in proposition 7.18 for convergence of the EM algorithm are satisfied for a simplified form of the the model of proposition 6.11 where the observation variance $\boldsymbol{v}$ is known and fixed.

Notes

The first proof of consistency and asymptotic normality of the maximum likelihood estimator for hidden Markov models was given by Baum and Petrie [BP66] for the case where the signal and observation state spaces are both finite. Remarkably, it took almost three decades for this result to be extended to more general models. Leroux [Ler92] was the first to prove consistency for a finite signal state space but general observation state space. Bickel, Ritov and Rydén [BRR98] subsequently proved asymptotic normality in this setting. Meanwhile, Mevel [Mev97] developed a different approach based on the ergodic properties of the filter. For the case where the signal state space is not finite, Jensen and Petersen [JP99], Douc and Mathias [DM01], and Douc, Moulines and Rydén [DMR04] prove consistency and asymptotic normality under slightly weaker conditions than we have imposed in this chapter.

Our proof of consistency is close in spirit, but does not follow directly any of the above references. The LLN used in our proof is extremely primitive; we could have easily used the standard ergodic theorems of Birkhoff or Kingman (see, e.g., [Kal02]), but the proofs of these results are much more complicated. The proof of our LLN utilizes a simple device due to Etemadi [Ete81] to strengthen the trivial mean square convergence to almost sure convergence. Our proof of identifiability appears to be new.

The basic approach to proving consistency and asymptotic normality outlined in this chapter, and used in the above references, is essentially the 'classical' approach (see [van98]) for the analysis of maximum likelihood estimates. A modern approach uses empirical process theory to establish uniform laws of large numbers and uniform central limit theorems for the likelihood. Such methods do not require a compact parameter space, but instead impose entropic bounds on the complexity of the model class [van00]. It remains an open problem to adapt this much more general approach to hidden Markov models: the fact that hidden Markov model observations are not i.i.d. complicates the application of empirical process theory. Another open problem is to weaken the strong mixing condition on the signal in the case of a general signal state space. The general setting is not yet entirely understood; in particular, at present the known results for a finite state space are more general than can be obtained by applying the general theory.

The analysis of the model order estimation problem requires us to study the rate of convergence of the likelihood function. Some results in this direction can be found, e.g., in Mevel and Finesso [MF04] and in Gerencsér and Molnár-Sáksa [GMS03]. These results do not appear to be sufficiently strong to prove consistency of penalized likelihood methods. In the setting where the signal and observation state spaces are finite, Gassiat and Boucheron [GB03] prove consistency of a penalized likelihood method for model order estimation using a particular penalty. This result and previous results following initial work of Finesso [Fin90] are reviewed in [CMR05, chapter 15]. Recent results in a setting where the observation state space is not finite can be found in [CGG08].

Using the full likelihood functions for model order selection is often overkill, however. It is often sufficient to consider 'quasi-likelihood' type functions, as suggested by Rydén [Ryd95]. The idea is that when the signal state space is finite, the marginal law of a single observation Y_k is a finite mixture of observation densities. One can then use estimators of the form

$$\hat{d}_n = \operatorname*{argmax}_{d \geq 0} \left\{ \sup_{\theta \in \bar{\Theta}_d} \frac{1}{n} \sum_{k=0}^{n} \log g^{\theta}(Y_k) - \varkappa(n, d) \right\},$$

where g_θ is a suitable class of functions and $\varkappa(n, d)$ is a suitable penalty function, to estimate the number of elements in the mixture, without relying on the full joint likelihood of all the observations. This approach is both mathematically and computationally simpler than a full-blown penalized likelihood method. See, e.g., Gassiat [Gas02] and Poskitt and Zhang [PZ05].

Our proof of convergence of the EM algorithm is from [BPSW70].

The parametric bootstrap is a classical technique in statistics to obtain confidence intervals for estimators by simulation (see, e.g., [van98]). In the hidden Markov model setting, see [MZ97].

Some results about consistency of Bayes estimates, under similar conditions as we have imposed in this chapter, can be found in Papavasiliou [Pap06].

References

[AZ97] R. Atar and O. Zeitouni. Exponential stability for nonlinear filtering. *Ann. Inst. H. Poincaré Probab. Statist.*, 33:697–725, 1997.

[BBS88] H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Trans. Automat. Control*, 33:780–783, 1988.

[BCL04] P. Baxendale, P. Chigansky, and R. Liptser. Asymptotic stability of the Wonham filter: Ergodic and nonergodic signals. *SIAM J. Control Optim.*, 43:643–669, 2004.

[BH85] B. Bru and H. Heinich. Meilleures approximations et médianes conditionnelles. *Ann. Inst. H. Poincaré Probab. Statist.*, 21:197–224, 1985.

[BH04] R. Bhar and S. Hamori. *Hidden Markov Models. Applications to Financial Economics*. Kluwer Academic Publishers, Dordrecht, 2004.

[BHL99] D. Brigo, B. Hanzon, and F. Le Gland. Approximate nonlinear filtering by projection on exponential manifolds of densities. *Bernoulli*, 5:495–534, 1999.

[BJ87] R. S. Bucy and P. D. Joseph. *Filtering for stochastic processes with applications to guidance*. Chelsea Publishing Co., New York, second edition, 1987.

[BK01] A. Budhiraja and H. J. Kushner. Monte Carlo algorithms and asymptotic problems in nonlinear filtering. In *Stochastics in finite and infinite dimensions*, Trends Math., pages 59–87. Birkhäuser Boston, Boston, MA, 2001.

[BLB08] P. Bickel, B. Li, and T. Bengtsson. Sharp failure rates for the bootstrap particle filter in high dimensions. In B. Clarke and S. Ghosal, editors, *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*, volume 3 of *IMS Collections*, pages 318–329. IMS, Beachwood, OH, 2008.

[BLK01] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, 2001.

[BN93] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes. Theory and Application*. Prentice-Hall, Englewood Cliffs, NJ, 1993.

[BP66] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Statist.*, 37:1554–1563, 1966.

[BP03] J. Bröcker and U. Parlitz. Analyzing communication schemes using methods from nonlinear filtering. *Chaos*, 13:195–208, 2003.

[BPSW70] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, 41:164–171, 1970.

[BRR98] P. J. Bickel, Y. Ritov, and T. Rydén. Asymptotic normality of the maximum-likelihood estimator for general hidden Markov models. *Ann. Statist.*, 26:1614–1635, 1998.

[Bud03] A. Budhiraja. Asymptotic stability, ergodicity and other asymptotic properties of the nonlinear filter. *Ann. Inst. H. Poincaré Probab. Statist.*, 39:919–941, 2003.

[CC08] A. Capponi and J. Cvitanić. Credit risk modeling with misreporting and incomplete information, 2008. Preprint.

[CD02] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Signal Process.*, 50:736–746, 2002.

[CD08] G. Celeux and J.-B. Durand. Selecting hidden Markov model state number with cross-validated likelihood. *Comp. Stat.*, 2008. To appear.

[CDMS97] H. Carvalho, P. Del Moral, A. Monin, and G. Salut. Optimal nonlinear filtering in GPS/INS integration. *IEEE Trans. Aerospace Electr. Syst.*, 33:835–850, 1997.

[CGG08] A. Chambaz, A. Garivier, and E. Gassiat. A MDL approach to HMM with Poisson and Gaussian emissions. application to order indentification. *J. Statist. Plan. Inf.*, 2008. To appear.

[CH08] G. Claeskens and N. L. Hjort. *Model Selection and Model Averaging*, volume 27 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2008.

[CL04] P. Chigansky and R. Liptser. Stability of nonlinear filters in nonmixing case. *Ann. Appl. Probab.*, 14:2038–2056, 2004.

[CL06] P. Chigansky and R. Liptser. On a role of predictor in the filtering stability. *Electron. Comm. Probab.*, 11:129–140, 2006.

[CMR05] O. Cappé, E. Moulines, and T. Rydén. *Inference in hidden Markov models*. Springer Series in Statistics. Springer, New York, 2005.

[CR09] D. Crisan and B. Rozovsky, editors. *The Oxford University Handbook of Nonlinear Filtering*. Oxford University Press, 2009. To appear.

[CRZ06] J. Cvitanić, B. Rozovskii, and I. Zaliapin. Numerical estimation of volatility values from discretely observed diffusion data. *J. Comp. Finance*, 9:1–36, 2006.

[DDG01] A. Doucet, N. De Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. Statistics for Engineering and Information Science. Springer-Verlag, New York, 2001.

[Del98a] P. Del Moral. Measure-valued processes and interacting particle systems. Application to nonlinear filtering problems. *Ann. Appl. Probab.*, 8:438–495, 1998.

[Del98b] P. Del Moral. A uniform convergence theorem for the numerical solving of the nonlinear filtering problem. *J. Appl. Probab.*, 35:873–884, 1998.

[Del04] P. Del Moral. *Feynman-Kac formulae*. Probability and its Applications (New York). Springer-Verlag, New York, 2004. Genealogical and interacting particle systems with applications.

[DG01] P. Del Moral and A. Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. *Ann. Inst. H. Poincaré Probab. Statist.*, 37:155–194, 2001.

[DGA00] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comp.*, 10:197–208, 2000.

[DL01] D. Duffie and D. Lando. Term structures of credit spreads with incomplete accounting information. *Econometrica*, 69:633–664, 2001.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 39:1–38, 1977. With discussion.

[DM01] R. Douc and C. Matias. Asymptotics of the maximum likelihood estimator for general hidden Markov models. *Bernoulli*, 7:381–420, 2001.

[DMR04] R. Douc, É. Moulines, and T. Rydén. Asymptotic properties of the maximum likelihood estimator in autoregressive models with Markov regime. *Ann. Statist.*, 32:2254–2304, 2004.

[DZ91] B. Delyon and O. Zeitouni. Lyapunov exponents for filtering problems. In *Applied stochastic analysis (London, 1989)*, volume 5 of *Stochastics Monogr.*, pages 511–521. Gordon and Breach, New York, 1991.

[EAM95] R. J. Elliott, L. Aggoun, and J. B. Moore. *Hidden Markov models*, volume 29 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 1995.

[EM02] Y. Ephraim and N. Merhav. Hidden Markov processes. *IEEE Trans. Inf. Th.*, 48:1518–1569, 2002.

[Ete81] N. Etemadi. An elementary proof of the strong law of large numbers. *Z. Wahrsch. Verw. Gebiete*, 55:119–122, 1981.

[Fin90] L. Finesso. *Consistent Estimation of the Order for Markov and Hidden Markov Chains*. PhD thesis, Univ. Maryland, College Park, 1990.

[For73] G. D. Forney. The Viterbi algorithm. *Proc. IEEE*, 61:268–278, 1973.

[Gas02] E. Gassiat. Likelihood ratio inequalities with applications to various mixtures. *Ann. Inst. H. Poincaré Probab. Statist.*, 38:897–906, 2002.

[GB03] E. Gassiat and S. Boucheron. Optimal error exponents in hidden Markov models order estimation. *IEEE Trans. Inform. Theory*, 49:964–980, 2003.

[GC03] V. Genon-Catalot. A non-linear explicit filter. *Statist. Probab. Lett.*, 61:145–154, 2003.

[GMS03] L. Gerencsér and G. Molnár-Sáksa. Adaptive encoding and prediction of hidden Markov processes. In *Proc. European Control Conf. 2003*, 2003.

[GSS93] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140:107–113, 1993.

[Han70] J. E. Handschin. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica–J. IFAC*, 6:555–563, 1970.

[IAK92] H. Ito, S.-I. Amari, and K. Kobayashi. Identifiability of hidden markov information sources and their minimum degrees of freedom. *IEEE Trans. Inf. Th.*, 38:324–333, 1992.

[IH81] I. A. Ibragimov and R. Z. Has'minskiĭ. *Statistical estimation*, volume 16 of *Applications of Mathematics*. Springer-Verlag, New York, 1981.

[Jaz70] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

[JP99] J. L. Jensen and N. V. Petersen. Asymptotic normality of the maximum likelihood estimator in state space models. *Ann. Statist.*, 27:514–535, 1999.

[Kal02] O. Kallenberg. *Foundations of modern probability*. Probability and its Applications. Springer-Verlag, New York, second edition, 2002.

[KD01] H. J. Kushner and P. Dupuis. *Numerical methods for stochastic control problems in continuous time*. Springer, second edition, 2001.

[Kos01] T. Koski. *Hidden Markov models for bioinformatics*, volume 2 of *Computational Biology Series*. Kluwer Academic Publishers, Dordrecht, 2001.

[KP98] T. Kailath and H. V. Poor. Detection of stochastic processes. *IEEE Trans. Inform. Theory*, 44:2230–2259, 1998. Information theory: 1948–1998.

[Kro98] A. Krogh. An introduction to hidden Markov models for biological sequences. In S. L. Salzberg and D. B. Searls, editors, *Computational Methods in Molecular Biology*, pages 45–63. Elsevier, Amsterdam, 1998.

[KSH00] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, New York, 2000.

[Kun71] H. Kunita. Asymptotic behavior of the nonlinear filtering errors of Markov processes. *J. Multivar. Anal.*, 1:365–393, 1971.

[KV08] M. L. Kleptsyna and A. Yu. Veretennikov. On discrete time ergodic filters with wrong initial data. *Probab. Theory Related Fields*, 141:411–444, 2008.

[Ler92] B. G. Leroux. Maximum-likelihood estimation for hidden Markov models. *Stochastic Process. Appl.*, 40:127–143, 1992.

[LMR97] S. Lototsky, R. Mikulevicius, and B. L. Rozovskii. Nonlinear filtering revisited: a spectral approach. *SIAM J. Control Optim.*, 35:435–461, 1997.

[LO03] F. Le Gland and N. Oudjane. A robustification approach to stability and to uniform particle approximation of nonlinear filters: The example of pseudo-mixing signals. *Stochastic Process. Appl.*, 106:279–316, 2003.

[LO04] F. Le Gland and N. Oudjane. Stability and uniform approximation of nonlinear filters using the Hilbert metric and application to particle filters. *Ann. Appl. Probab.*, 14:144–187, 2004.

[ME07] R. S. Mamon and R. J. Elliott, editors. *Hidden Markov Models in Finance*. International Series in Operations Research & Management Science. Springer-Verlag, New York, 2007.

[Mev97] L. Mevel. *Statistique asymptotique pour les modèles de Markov cachés*. PhD thesis, Univ. Rennes 1, 1997.

[MF04] L. Mevel and L. Finesso. Asymptotical statistics of misspecified hidden Markov models. *IEEE Trans. Automat. Control*, 49:1123–1132, 2004.

[MJH06] S. A. McKinney, C. Joo, and T. Ha. Analysis of single-molecule FRET trajectories using hidden Markov modeling. *Biophys. J.*, 91:1941–1951, 2006.

[MT93] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Communications and Control Engineering Series. Springer-Verlag London Ltd., London, 1993.

[MZ97] I. L. MacDonald and W. Zucchini. *Hidden Markov and other models for discrete-valued time series*, volume 70 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 1997.

[Nor98] J. R. Norris. *Markov chains*, volume 2 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 1998. Reprint of 1997 original.

[OP96] D. Ocone and E. Pardoux. Asymptotic stability of the optimal filter with respect to its initial condition. *SIAM J. Control Optim.*, 34:226–243, 1996.

[Pap06] A. Papavasiliou. Parameter estimation and asymptotic stability in stochastic filtering. *Stochastic Process. Appl.*, 116:1048–1065, 2006.

[Pic86] J. Picard. Nonlinear filtering of one-dimensional diffusions in the case of a high signal-to-noise ratio. *SIAM J. Appl. Math.*, 46:1098–1125, 1986.

[Pic91] J. Picard. Efficiency of the extended Kalman filter for nonlinear systems with small noise. *SIAM J. Appl. Math.*, 51:843–885, 1991.

[PP05] G. Pagès and H. Pham. Optimal quantization methods for nonlinear filtering with discrete-time observations. *Bernoulli*, 11:893–932, 2005.

[PZ05] D. S. Poskitt and J. Zhang. Estimating components in finite mixtures and hidden Markov models. *Aust. N. Z. J. Stat.*, 47:269–286, 2005.

[Rab89] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286, 1989.

[Rev75] D. Revuz. *Markov chains*. North-Holland Publishing Co., Amsterdam, 1975. North-Holland Mathematical Library, Vol. 11.

[Ros95] J. S. Rosenthal. Convergence rates for Markov chains. *SIAM Rev.*, 37:387–405, 1995.

[Ryd95] T. Rydén. Estimating the order of hidden Markov models. *Statistics*, 26:345–354, 1995.

[Saw81] G. Sawitzki. Finite-dimensional filter systems in discrete time. *Stochastics*, 5:107–114, 1981.

[SH04] J. Sass and U. G. Haussmann. Optimizing the terminal wealth under partial information: The drift process as a continuous time Markov chain. *Finance Stoch.*, 8:553–577, 2004.

[She02] L. Shepp. A model for stock price fluctuations based on information. *IEEE Trans. Inform. Theory*, 48:1372–1378, 2002. Special issue on Shannon theory: perspective, trends, and applications.

[Shi73] A. N. Shiryaev. *Statistical sequential analysis: Optimal stopping rules*. American Mathematical Society, Providence, R.I., 1973. Translations of Mathematical Monographs, vol. 38.

[Shi96] A. N. Shiryaev. *Probability*, volume 95 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1996.

[Str60] R. L. Stratonovich. Conditional Markov processes. *Teor. Verojatnost. i Primenen.*, 5:172–195, 1960.

[van98] A. W. van der Vaart. *Asymptotic statistics*, volume 3 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 1998.

[van00] S. A. van de Geer. *Applications of empirical process theory*, volume 6 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2000.

[van08a] R. van Handel. Discrete time nonlinear filters with regular observations are always stable, 2008. Preprint, arXiv:0807.1072.

[van08b] R. van Handel. Observability and nonlinear filtering. *Probab. Th. Rel. Fields*, 2008. To appear.

[van08c] R. van Handel. The stability of conditional Markov processes and Markov chains in random environments, 2008. Preprint, arXiv:0801.4366.

[van08d] R. van Handel. Uniform observability of hidden Markov models and filter stability for unstable signals, 2008. Preprint, arXiv:0804.2885.

[Vit67] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Th.*, IT-13:260–269, 1967.

[ZD88] O. Zeitouni and A. Dembo. Exact filters for the estimation of the number of transitions of finite-state continuous-time Markov processes. *IEEE Trans. Inform. Theory*, 34:890–893, 1988.