RECURSIVE ESTIMATION IN HIDDEN MARKOV MODELS

OTTO REED

Abstract. Hidden Markov models are a powerful tool in signal processing and time-series settings. In this paper we consider three fundamental estimation problems of hidden Markov models: filtering, smoothing, and prediction. We first derive recursive expressions for all three distributions in arbitrary state space and subsequently consider the only two settings where these recursions can be implemented exactly: finite state space and linear Gaussian models. In the former we derive the forward-backward algorithm and Viterbi algorithm; in the latter, we derive and explore applications of the famous Kalman filter.

Contents

1. Introduction

Suppose we are climate scientists studying long-term weather patterns who wish to know the weather in Boston in June of 1905. No official weather records survive but we have discovered the personal journal of an enthusiastic Bostonian ice cream salesman who meticulously detailed how much ice cream he sold each day of the summer. From personal experience and rigorous statistical inquiry, we know that the amount of ice cream consumed is highly correlated to the weather. Furthermore, we know that weather can be modeled by a Markov process with state given by "recent weather"; it is a natural assumption that the weather will not be influenced by what it was a century ago.

Thus, we have a "hidden" process, the weather, which can be modeled as Markov, and a set of observations, the ice cream sales, which we can think of as being "generated" by the hidden process. Together, the observable and unobservable components form a *hidden Markov model.* We wish to use our observations to make inferences about the hidden process, i.e., determine the weather from the amount of ice cream sold. However, inference is not the only use of our model. If we decided to retire from climate science in favor of being ice cream salesman, we could instead use our model to more faithfully model ice cream sales so as to prepare our supply more efficiently for a given weather forecast.

Fortunately, hidden Markov models have even more applications than ice cream-based weather inference or weather-based ice cream modeling; in the broadest sense, we can think of the hidden component as some signal and the observed component as a noisy measurement of that signal. This interpretation has a clear connection to life in the digital age, where bits are being streamed constantly to and from devices across the globe. As anyone who has tried to watch live sports on airplane WiFi can attest, the information received is inevitably different from the information transmitted due to radio wave interference (we call such interference the *noise* of the channel). Of course, before we begin to explore estimation problems such as reconstructing the hidden signal from the observed measurement, we must first define the model in which we wish to perform estimation.

2. Foundations of Hidden Markov Models

2.1. **Markov Processes.** As mentioned above, the model we will develop is built on Markov processes. Although Markov processes are one of the simplest types of stochastic (i.e., random) processes, they are undoubtedly one of the most ubiquitous and powerful. The rich theory of Markov processes alone is enough to fill an entire textbook (or several), so we will take only a cursory overview. We begin our exploration with a critical idea in the study of Markov processes: transition kernels. From [1], we have the following definition:

Definition 2.1.1 (Transition Kernel)**.** Let $(\mathrm{X}, \mathcal{X})$ and $(\mathrm{Y}, \mathcal{Y})$ be measurable spaces. A *kernel* from $(\mathrm{X}, \mathcal{X})$ to $(\mathrm{Y}, \mathcal{Y})$ is a function $P : \mathrm{X} \times \mathcal{Y} \to \mathbb{R}^+$ such that for every $x \in \mathrm{X}$ and $A \in \mathcal{Y}$,

(i) the map $A \mapsto P(x, A)$ is a (positive) measure on $(\mathrm{Y}, \mathcal{Y})$;
(ii) the map $x \mapsto P(x, A)$ is $\mathcal{X}$-measurable.

Furthermore, if $P(x, \mathcal{Y}) = 1$ for all $x \in \mathrm{X}$, then P is a *transition kernel.* If additionally $\mathrm{X} = \mathrm{Y}$, then P is a *Markov transition kernel*–or simply *Markov kernel*–on $(\mathrm{X}, \mathcal{X})$.

Intuitively, $P(x, A)$ is the probability that the next state of the process is in the set $A \subset \mathrm{X}$ when the current state of the process is $x \in \mathrm{X}$. Hence, the transition kernel "transitions" the law of the state forward by one time step. An X-valued stochastic process $(X_k)_{k \geq 0}$ (that is, a stochastic process where each X_k takes values in a measurable space $(\mathrm{X}, \mathcal{X})$) on a probability space $(\Omega, \mathcal{F}, \mathbf{P})$ possesses the *Markov property* if

$$\mathbf{P}[X_{k+1} \in A | X_0, \ldots, X_k] = \mathbf{P}[X_{k+1} \in A | X_k] \text{ for all } A \in \mathcal{X},\ k \geq 0$$

Using the language of transition kernels, we can formalize the notion of a *Markov process.*

Definition 2.1.2 (Homogeneous Markov Process)**.** Let $(\Omega, \mathcal{F}, \mathbf{P})$ be a probability space. An X-valued stochastic process $(X_k)_{k \geq 0}$ is said to be a *homogeneous Markov process* under $\mathbf{P}$ if there exists a Markov transition kernel P on the measurable space $(\mathrm{X}, \mathcal{X})$ such that

$$\mathbf{P}[X_{k+1} \in A | X_0, \dots, X_k] = P(X_k, A) \text{ for all } A \in \mathcal{X},\ k \geq 0 \tag{2.1}$$

Remark. It is standard to introduce the notion of a *filtration* to formalize the available information at a given time step. We choose to omit an explicit definition of filtered probability spaces in favor of implicitly using the natural filtration (not only is every process adapted to its natural filtration, but it is also the coarsest such filtration) since we do not utilize the former in our broader discussion of hidden Markov models.

Continuing with our interpretation of transition kernels, we can think of $P(x, A)$ as the probability that state of the process will be in the set $A \subset \mathrm{X}$ in the next time step given that the current state is $x \in \mathrm{X}$. The additional description of "homogeneous" specifies that this probability is the same at each time step k. However, since we will deal almost exclusively with homogeneous Markov processes, we will drop the descriptor "homogeneous" and mention explicitly when the process is not time homogeneous. We will refer to $(\mathrm{X}, \mathcal{X})$ as the *state space* of the process (we will also typically say "on X" to mean "on the state space $(\mathrm{X}, \mathcal{X})$") and the probability measure μ on $(\mathrm{X}, \mathcal{X})$ where $\mu(A) = \mathbf{P}[X_0 \in A]$ as the *initial measure.*

In general, we want to understand how a given Markov process evolves in time. Hence, we seek the *probability distribution* or *law* of the process. As given by [5], the law of a process is the pushforward measure from the probability space $(\Omega, \mathcal{F})$ to the state space $(\mathrm{X}, \mathcal{X})$. Notably, as proved in [7], the law of a Markov process is determined by its finite-dimensional distributions, that is, the joint distribution of $X_0, \dots, X_k$ for all $k \geq 0$. This fact motivates a fundamental result in the study of Markov processes detailed in [3], [1], and Lemma 1.5 of [8]:

Lemma 2.1.3. *Let $(X_k)_{k\geq 0}$ be a Markov process on* X *with transition kernel P and initial measure μ. Then, for any bounded, $\mathcal{X}^{\otimes(k+1)}$-measurable function $f : \mathrm{X}^{k+1} \to \mathbb{R}$,*

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

In other words, the initial measure and transition kernel of a Markov process entirely determine its finite-dimensional distributions, and therefore the law of the process.

Proof. Let $\mathcal{F}(\mathrm{X}^{k+1})$ be the set of all bounded, $\mathcal{X}^{\otimes k+1}$-measurable functions $f : \mathrm{X}^{k+1} \to \mathbb{R}$ and define

$$\mathcal{H} = \{f \in \mathcal{F}(\mathrm{X}^{k+1}) \mid f(x_0, \dots, x_k) = f_0(x_0) \cdots f_k(x_k)\}.$$

Trivially, $\mathcal{X}^{\otimes k+1}$ is a π-system, so, by the monotone class theorem, $\mathcal{F}(\mathrm{X}^{k+1}) \subset \mathcal{H}$. Thus, it is sufficient to consider functions of the form $f(x_0, \ldots, x_k) = f_0(x_0), \ldots, f_k(x_k)$. Note that

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

as desired. □

Before we move on to hidden Markov models, we introduce some helpful notation and definitions. Let $(X_k)_{k\geq 0}$ be a Markov process on X with transition kernel P. For the following, every measure will be on $(\mathrm{X}, \mathcal{X})$, and we will use "measurable" to mean $\mathcal{X}$-measurable.

For any bounded measurable function $f : \mathrm{X} \to \mathbb{R}$, we define the function $Pf : \mathrm{X} \to \mathbb{R}$ by

$$Pf(x) = \int f(y)P(x, dy) \text{ for all } x \in \mathrm{X}.$$

By the Markov property, for any Markov process $(X_k)_{k\geq 0}$, we have

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

For $n \geq 1$, we recursively define the functions $P^n f = PP^{n-1}f$ with the initial condition $P^0 f = f$. Then, by repeated application of the tower property of expectation, it follows that

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

Similarly, for any measure ρ, we define the measure ρP by

$$\rho P(A) = \int P(x, A)\rho(dx) \text{ for all } A \in \mathcal{X}.$$

Remark. Note that $A \in \mathcal{X}$ since P is a Markov transition kernel, although this statement is also true for arbitrary transition kernels from $(\mathrm{X}, \mathcal{X})$ to $(\mathrm{Y}, \mathcal{Y})$. In the latter case, ρP would be a measure on $(\mathrm{Y}, \mathcal{Y})$.

Again, for $n \geq 1$, we recursively define the measures $\rho P^n = \rho P^{n-1} P$ with the initial condition $\rho P^0 = \rho$. If μ is the initial measure of $(X_k)_{k\geq 0}$, by Lemma 2.1.3, we have $\mathbf{P}[X_k \in A] = \mu P^k(A)$ for all $A \in \mathcal{X}$. Hence, μP^k is the law of X_k! Finally, for any measurable function $f : \mathrm{X} \to \mathbb{R}$, we have $(\mu P)(f) = \mu(Pf)$, i.e., $\int f(x) \mu P(dx) = \int Pf(x)\mu(dx)$. Thus, as noted in section 1.1 of [8], this implies that the maps $\mu \mapsto \mu P$ and $f \mapsto Pf$ are dual to each other.

2.2. **Hidden Markov Models.** As discussed in the introduction, a hidden Markov model is a Markov process with two components: an *observable* component and an *unobservable*, i.e., *hidden*, component. More specifically, it is a Markov process $(X_k, Y_k)_{k\geq 0}$ on the state space $\mathrm{X} \times \mathrm{Y}$ where we can "observe" Y_k, but not X_k. Given how natural the signal processing interpretation is, we will refer to $(X_k)_{k\geq 0}$ as the *signal process* on the *signal state space* X and $(Y_k)_{k\geq 0}$ as the *observation process* on the *observation state space* Y. It is important to note that while both the joint process $(X_k, Y_k)_{k\geq 0}$ and the signal process $(X_k)_{k\geq 0}$ are Markov, the observation process $(Y_k)_{k\geq 0}$ generally is not. Thus, Hidden Markov models are capable of modeling non-Markov behavior.

Let us try to motivate a formal definition for a hidden Markov model. As stated in section 1.2 of [8], our definition should encapsulate a Markov process $(X_k, Y_k)_{k\geq 0}$ with two key restrictions:

(i) the signal process $(X_k)_{k\geq 0}$ is Markov;
(ii) the observation Y_k is a "noisy functional" of X_k.

Notation. For the sake of brevity, we will frequently abbreviate expressions such as $X_0, \ldots, X_k$ to $X_{0:k}$ throughout the rest of this paper.

Definition 2.2.1 (Hidden Markov Model)**.** A stochastic process $(X_k, Y_k)_{k\geq 0}$ on a product state space $(\mathrm{X} \times \mathrm{Y}, \mathcal{X} \otimes \mathcal{Y})$ is a *hidden Markov model* if there exist transition kernels $P : \mathrm{X} \times \mathcal{X} \to [0,1]$ and $\Phi : \mathrm{X} \times \mathcal{Y} \to [0,1]$ such that

$$\mathbf{E}[g(X_{k+1}, Y_{k+1})|X_{0:k}, Y_{0:k}] = \iint\limits_{\mathrm{X}\times\mathrm{Y}} g(x,y)\Phi(x,dy)P(X_k,dx)$$

and a probability measure μ on X such that

$$\mathbf{E}[g(X_0, Y_0)|X_{0:k}, Y_{0:k}] = \iint\limits_{\mathrm{X}\times\mathrm{Y}} g(x,y)\Phi(x,dy)\mu(dx)$$

for every bounded $\mathcal{X} \otimes \mathcal{Y}$-measurable function $g : \mathrm{X} \times \mathrm{Y} \to \mathbb{R}$. We call Φ the *observation kernel*, with μ and P again being the *initial measure* and *transition kernel*, respectively.

Note that by Definition 2.1.2, both $(X_k, Y_k)_{k\geq 0}$ and $(X_k)_{k\geq 0}$ are Markov processes. This fact is one of several basic properties of hidden Markov models that form the foundation of our exploration.

Lemma 2.2.2. *Let $(X_k, Y_k)_{k\geq 0}$ be a hidden Markov model on $(\mathrm{X} \times \mathrm{Y}, \mathcal{X} \otimes \mathcal{Y})$ with transition kernel P, observation kernel Φ, and initial measure μ. Then, the following facts hold:*

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

$$\mathbf{P}[Y_0 \in A_0, \ldots, Y_k \in A_k | X_{0:k}] = \Phi(X_0, A_0) \cdots \Phi(X_k, A_k)$$

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

$$\mathbf{E}[f(X_{0:k}, Y_{0:k})] = \int \cdots \int f(x_{0:k}, y_{0:k}) \times \mu(dx_0)\Phi(x_0, dy_0) \prod_{i=1}^{k} \Phi(x_i, dy_i) P(x_{i-1}, dx_i)$$

for every bounded, $(\mathcal{X} \otimes \mathcal{Y})^{\otimes(k+1)}$*-measurable function* $f : (\mathrm{X} \times \mathrm{Y}^{k+1}) \to \mathbb{R}$.

Proof. To prove (i), define the Markov kernel of the joint process on the product state space $(\mathrm{X} \times \mathrm{Y}, \mathcal{X} \otimes \mathcal{Y})$ by

$$Q((x,y),C) = \iint\limits_{\mathrm{X}\times\mathrm{Y}} I_C((x,y))P(x,dx')\Phi(x',dy') \text{ for all } (x,y) \in \mathrm{X} \times \mathrm{Y},\ C \in \mathcal{X} \otimes \mathcal{Y}.$$

Additionally, let $\nu = \Phi \otimes \mu$ be the initial distribution of $(X_k, Y_k)_{k\geq 0}$. Then, comparing with Definition 2.2.1, we see that (i) and (ii) follow directly from Definition 2.1.2. Furthermore, (iii) follows immediately from Definition 2.2.1 and Lemma 2.1.3. □

We will illustrate that the observations $(Y_k)_{k\geq 0}$ only depend on each other through values of the hidden process $(X_k)_{k\geq 0}$ (as given by (iii)) with a simple example.

Example 2.2.3. Let the signal state space $\mathrm{X} = \{0, 1\}$ and suppose the observations $(Y_k)_{k\geq 0}$ depend on the signal $(X_k)_{k\geq 0}$ according to

$$Y_k = \begin{cases} 1 & X_k = 1 \\ -1 & X_k = 0 \end{cases}$$

Further suppose that $\mathbf{P}[X_k] = p$ for $p \in (0, 1),\ k \geq 0$. If $Y_k > 0$ for some $k \geq 0$, then, since $\mathbf{P}[Y_k > 0|X_k = 1] = 1$,

$$\mathbf{P}[Y_{k+1} > 0|Y_k > 0,\ X_k = 1] = \mathbf{P}[Y_{k+1} > 0|X_k = 1] = \mathbf{P}[X_{k+1} = 1] = p.$$

Thus, Y_k and Y_{k+1} are conditionally independent given X_k. Trivially, Y_k and Y_{k+1} are also conditionally independent given X_{k+1}, since Y_{k+1} is given direcly by X_{k+1}.

2.3. **Nondegeneracy.** Although we are still working within the world of theory, we will require a stronger condition on the structure of our observations $(Y_k)_{k\geq 0}$ that is needed if we wish to use our model for practical applications.

Definition 2.3.1 (Nondegeneracy)**.** Let $(X_k, Y_k)_{k\geq 0}$ be a hidden Markov model on $(\mathrm{X} \times \mathrm{Y}, \mathcal{X} \otimes \mathcal{Y})$ with observation kernel Φ. The model has *nondegenerate observations* if there exists a strictly positive $\mathcal{X} \otimes \mathcal{Y}$-measurable density function $\Upsilon : \mathrm{X} \times \mathrm{Y} \to (0, \infty)$ and a probability measure ν on Y such that

$$\Phi(x, B) = \int I_B(y)\Upsilon(x, y)\nu(dy) \text{ for all } x \in \mathrm{X},\ B \in \mathcal{Y}$$

Nondegeneracy guarentees that the model reflects all possible observational values; the positive density condition ensures that every element of the chosen observation state space has a nonzero probability of being observed.

Although this is a simple condition, its importance should not be overlooked. When we assume that a model has nondegenerate observations, we guarentee that we can make inferences about the hidden process from any set of observations $y_0, \ldots, y_k$, even if they do not precisely align with our mathematical definition of the model. If this were not the case, even the slightest amount of noise could lead to data incompatible with our model, making it hopeless to apply in practice. To demonstrate the importance of assuming nondegeneracy, we consider an extreme case of a model that only satisfies our general definition for a hidden Markov model.

Example 2.3.2. Let $\mathrm{X} = \mathrm{Y} = \mathbb{R}$ and let $\delta_k,\ k \geq 0$ be an i.i.d. sequence of random variables whose law is supported on $\mathbb{Z}$ (i.e., $\mathbf{P}[\delta_k \in \mathbb{Z}] = 1$ for all $k \geq 0$). We recursively define $(X_k, Y_k)_{k\geq 0}$ as

$$X_0 = Y_0 = 0, \qquad X_k = X_{k-1} + \delta_k, \qquad Y_k = X_k \qquad (k \geq 1).$$

It is clear that we have a hidden Markov model that satisfies Definition 2.2.1, but not nondegeneracy: for example, $\mathbf{P}[\delta_k = 3.14] = 0$ for all $k \geq 0$. What are the consequences?

Suppose we make a sequence of observations $y_0, \ldots, y_k$ that are generated by our model. Since $\delta_k \in \mathbb{Z}$ for $k \geq 0$, by construction we should expect that each $y_k \in \mathbb{Z}$ as well. However, in practice, it is likely that the signal X_n will be perturbed slightly–no transmission is perfect, after all–causing the corresponding real-world sample y_n to no longer satisfy this property. Any attempt at inference on this observation would certainly fail since according to our model, we have made an impossible measurement. Clearly, a model susceptible to even the smallest amount of noise is insufficient for broader applications.

Although this scenario is highly contrived, it illustrates how Definition 2.2.1 requires further assumptions to yield models where inference can be performed without issue. In fact, the following proposition tells us that if an observation kernel satisfies Definition 2.3.1, then any property of a finite number of observations $Y_0, \ldots, Y_k$ that holds *almost surely* does so for any choice of transition kernel P and intial measure μ.

Proposition 2.3.3. *Let $(X_k, Y_k)_{k\geq 0}$ be a hidden Markov model on $(\mathrm{X} \times \mathrm{Y}, \mathcal{X} \otimes \mathcal{Y})$ with initial measure μ on X, transition kernel P, and observation kernel Φ. Furthermore, let $(\tilde{X}_k, Y_k)_{k\geq 0}$ be a hidden Markov model on $(\mathrm{X} \times \mathrm{Y}, \mathcal{X} \otimes \mathcal{Y})$ with initial measure $\tilde{\mu}$ on X, transition kernel $\tilde{P}$, and observation kernel Φ. Suppose Φ satisfies the nondegeneracy assumption of Definition 2.3.1. Then, for both models, and for $n \geq 0$, the law of $(Y_k)_{k\leq n}$ is absolutely continuous.*

Remark. Note that the Y_k is the same for both hidden Markov models because Y_k only depends on the observation kernel, which remains the same; the initial measure is on X and the transition kernel is on $\mathrm{X} \times \mathcal{X}$.

It follows from this proposition that if $y_0, \ldots, y_k$ is a valid sample path of the observation process for some model of the signal process $(X_k)_{k\geq 0}$ (i.e., has a nonzero probability of being observed in that model), then it is a valid path for *any* model of the signal. While nondegeneracy does not guarentee that inference is robust to error, it ensures that all inference is mathematically well-defined. The following example illustrates a model that *does* satisfy nondegeneracy.

Example 2.3.4. Let $\mathrm{Y} = \mathbb{R}$ and suppose the observations satisfy

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

for some $\mathcal{X}$-measurable function $f : \mathrm{X} \to \mathbb{R}$. Additionally, assume that $\eta_k,\ k \geq 0$ are i.i.d. $\mathcal{N}(0, 1)$. Then, the observation kernel Φ is given by

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

for all $x \in \mathrm{X},\ B \in \mathbb{R}$. Setting

$$\Upsilon(x, z) = \frac{e^{-(z-f(x))^2/2}}{\sqrt{2\pi}},$$

it is clear that the model satisfies Definition 2.3.1.

With Definitions 2.2.1 and 2.3.1, we can now consider a broad range of general hidden Markov models.

2.4. **Two Essential Questions.** Now that we have discussed several examples of hidden Markov models, we can turn to some deeper theory in the field. There are two central questions which will motivate all of our further results: estimation and implementation.

(i) Given a hidden Markov model, (i.e., given the transition kernel P, observation kernel Φ, and initial measure μ) how can we estimate the unobserved signal $(X_k)_{k\geq 0}$ using the observed signal trajectory $y_0, \dots, y_k$?

(ii) How can we implement these methods for estimation precisely on a computer, what assumptions do we need to make, and what would the corresponding algorithms be?

In practice, the signal $(X_k)_{k\geq 0}$ will be completely unobservable. Thus, developing methods to solve the problem of estimation is integral to applying hidden Markov models in the real world. Of course, we must also be able to efficiently compute these methods, so our second question is also fundamental to our exploration.

We will first develop the theory of estimation in a general setting (i.e., arbitrary state space) and then turn to the two scenarios in which our methods are "computable:" finite state space and the linear Gaussian setting.

Since the process $(X_k)_{k\geq 0}$ is hidden, the problem of estimation is to compute

$$\mathbf{E}[f(X_k)|Y_{0:N}]$$

for any choice of k and N and any function $f : \mathrm{X} \to \mathbb{R}$; notice that these objects determine the conditional distribution for the signal X_k given the observations $Y_0, \dots, Y_N$. We refer to the problem as *smoothing* when $k < N$, *filtering* when $k = N$, and *prediction* when $k > N$. We will now develop methods to compute the conditional expectation in each of these three problems; a key feature of filtering, smoothing, and prediction (and an ongoing theme throughout this paper) is that solutions can be found *recursively.*

In general, the signal state space will be infinite, and thus these distributions will be infinite-dimensional, making real-world computation intractable. Of course, if we let the signal state space $\mathrm{X} = \{1, \dots, d\}$ for $d \in \mathbb{N}$, then our transition kernel P, observation kernel Φ can be represented as matrices[1] and any measure or function f on X is entirely determined by a vector (e.g., $f = (f(1), \dots, f(d))^\top$). In this case, all of our computations are matrix-vector operations, which are of course easily implemented precisely on a computer. A more interesting scenario in which the computations reduce to matrix-vector operations is the linear Gaussian setting, where we assume that the signal and observation state spaces are given by a system of linear equations. Every Gaussian variable is completely described by its mean vector and covariance matrix, so as long as our state space is finite-dimensional, these vectors and matrices will be too. Thus, we can successfully use finite matrix computation to solve a class of estimation problems in uncountable state space. As it turns out, the cases where the signal state space is finite or linear Gaussian are the *only* two cases where estimation can be implemented exactly. Fortunately, as we will see, these cases cover a broad range of applications for hidden Markov models.

3. Filtering, Smoothing, and Prediction

As discussed above, filtering, smoothing, and prediction are all *estimation* problems; specifically, problems in which we estimate the random variables $\{X_k\}_{k\geq 0}$ using the observations $Y_0, \dots, Y_N$. However, we have yet to answer a fundamental question: what does it mean to *estimate* a random variable X_k?

[1] We will actually work with the observation density Υ instead of the observation kernel Φ for reasons that will become apparent later.

3.1. **Estimation.** Let X be a real-valued random variable and let Y be a C-valued random variable on a probability space $(\Omega, \mathcal{F}, \mathbf{P})$ and state space $(\mathrm{C}, \mathcal{C})$. Assuming the general idea of a hidden Markov model, suppose we can observe Y but not X and would like to estimate X using Y. We are looking for a function $g(Y)$ that is "close" to X under a specific criterion. Generally, this means that g minimizes some loss function H. For instance, consider the mean square estimation error

$$\mathbf{E}[(X - f(Y))^2]$$

for some function f. As the following lemma tells us, in this case, the function g that minimizes the error is exactly the conditional expectation.

Lemma 3.1.1 (Conditional Expectation Minimizes Mean Squared Error)**.** *Suppose* $\mathbf{E}[X^2] < \infty$ *and let* $g(Y) = \mathbf{E}[X|Y]$*. Then,*

$$g = \arg\min_f \mathbf{E}[(X - f(Y))^2].$$

Proof. Note that $\mathbf{E}[X|Y]$ is a function of Y and

$$\begin{aligned}
&(X\mathbf{E}[X|Y] \geq 0) & \mathbf{E}[(X - \mathbf{E}[X|Y])^2] &\leq \mathbf{E}[X^2 + \mathbf{E}[X|Y]^2] \\
&\text{(linearity of expectation)} & &= \mathbf{E}[X^2] + \mathbf{E}[\mathbf{E}[X|Y]^2] \\
&\text{(Jensen's inequality, monotonicity)} & &\leq \mathbf{E}[X^2] + \mathbf{E}[\mathbf{E}[X^2|Y]] \\
&\text{(total expectation)} & &\leq 2\mathbf{E}[X^2] \\
&\text{(by supposition)} & &< \infty
\end{aligned}$$

Now, let $G = \mathbf{E}[X|Y]$ and consider

$$\begin{aligned}
& & \mathbf{E}[(X - G)^2] &= \mathbf{E}[(X - f(Y) + f(Y) - G)^2] \text{ for some function } f \\
&\text{(linearity)} & &= \mathbf{E}[(X - f(Y))^2] + \mathbf{E}[(f(Y) - G)^2] + 2\mathbf{E}[(X - f(Y))(f(Y) - G)] \\
&\text{(total expectation)} & &= \mathbf{E}[(X - f(Y))^2] + \mathbf{E}[(f(Y) - G)^2] + 2\mathbf{E}[\mathbf{E}[(X - f(Y))(f(Y) - G)|Y]]
\end{aligned}$$

Letting $h(Y) = f(Y) - G$ and pulling out known factors, we have

$$\begin{aligned}
& & &\mathbf{E}[(X - f(Y))^2] + \mathbf{E}[h(Y)^2] + 2\mathbf{E}[h(Y)\mathbf{E}[(X - f(Y))|Y]] \\
&\text{(linearity)} & &= \mathbf{E}[(X - f(Y))^2] + \mathbf{E}[h(Y)^2] + 2\mathbf{E}[h(Y)(\mathbf{E}[X|Y] - \mathbf{E}[f(Y)|Y]] \\
&\text{(stability)} & &= \mathbf{E}[(X - f(Y))^2] + \mathbf{E}[h(Y)^2] + 2\mathbf{E}[h(Y)(\mathbf{E}[X|Y] - f(Y)] \\
&\text{(linearity)} & &= \mathbf{E}[(X - f(Y))^2] + \mathbf{E}[h(Y)^2] - 2\mathbf{E}[h(Y)^2] \\
& & &= \mathbf{E}[(X - f(Y))^2] - \mathbf{E}[h(Y)^2] \\
&\text{(positivity)} & &\leq \mathbf{E}[(X - f(Y))^2].
\end{aligned}$$

This completes the proof. □

Remark. This result is so fundamental to the study of conditional expectation that it is reasonable to define the expectation of $X \in L^2(\Omega, \mathcal{F}, \mathbf{P})$ given $\mathcal{G} \subset \mathcal{F}$ as the orthogonal projection of X onto the L^2-subspace $L^2(\Omega, \mathcal{G}, \mathbf{P})$, since the orthogonal projection (with the L^2 inner product) by definition minimizes the expected squared error $\mathbf{E}[(X - Y)^2]$. As shown in [6], this is a satisfactory approach for proving many elementary properties of conditional expectation, but a more nuanced definition is needed to fully capture all notable results.

Thus, the conditional expectation gives the *least mean square estimate* of the unobserved variable X given the observed variable Y (see [9] for an alternate proof). In general, we would like to consider the estimator $\mathbf{E}[H(X - f(Y))]$ for some arbitrary *loss function* H. Solving the general problem requires the notion of a *conditional distribution.*

Definition 3.1.2 (Regular Conditional Distribution of X given Y)**.** Let X be a $(\mathrm{B}, \mathcal{B})$-valued random variable and let Y be a $(\mathrm{C}, \mathcal{C})$-valued random variable on the probability space $(\Omega, \mathcal{F}, \mathbf{P})$. The *regular conditional distribution* of X given Y is a transition kernel $P_{X|Y} : \mathrm{C} \times \mathcal{B} \to [0, 1]$ which satisfies

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

for every bounded $\mathcal{B}$-measurable function $f : \mathrm{B} \to \mathbb{R}$.

Remark. The function $P_{X|Y}$ is *regular* in the sense that it is a transition kernel. The regularity of $P_{X|Y}$ is precisely why the conditional expectation can be written in terms of integrals for each $y \in \mathrm{C}$.

Intuitively, $P_{X|Y}(y, A) = \mathbf{P}[X \in A|Y = y]$. The following lemma from [8] (Lemma 2.4) tells us that we can use the conditional distribution $P_{X|Y}$ to solve the estimation problem of minimizing $\mathbf{E}[H(X - f(Y))]$ for an arbitrary loss function H.

Lemma 3.1.3. *Let $H : \mathbb{R} \to [0, \infty)$ be a loss function, X be a real-valued random variable such that $\mathbf{E}[H(X)] < \infty$, and Y be a $(\mathrm{C}, \mathcal{C})$-valued random variable. Suppose there exists a $\mathcal{C}$-measurable function $g : \mathrm{C} \to \mathbb{R}$ such that*

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

where $C' \in \mathcal{C}$ such that $\mathbf{P}[Y \in B'] = 1$. Then, g minimizes $f \mapsto \mathbf{E}[H(X - f(Y))]$.

Proof. Note that by construction, we have

$$\int H(x - g(Y)) P_{X|Y}(Y, dx) \leq \int H(x - f(Y)) P_{X|Y}(Y, dx) \text{ almost surely}$$

for any $\mathcal{C}$-measurable function f. It follows from Definition 3.1.2 that

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

To complete the proof, we verify that our expectation is finite: if we let $f(Y) = 0$, then we have $\mathbf{E}[H(X - g(Y))] \leq \mathbf{E}[H(X)] < \infty$. Therefore, g minimizes $f \mapsto \mathbf{E}[H(X - f(Y))]$, as desired. □

We now consider an example that is of great interest in the field of statistical learning.

Example 3.1.4. Let X be a random variable that takes a finite number of values $\{x_1, \ldots, x_n\}$ and define the loss function

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

We wish to choose an estimator g that minimizes $f \mapsto \mathbf{E}[H(X - f(Y))]$, i.e., one that maximizes the probability $\mathbf{P}[X = f(Y)]$. Thus, by Lemma 3.1.3, we should define g by

$$g(y) = \underset{x_1,\dots,x_n}{\arg\max}\, P_{X|Y}(y, \{x_i\}).$$

We call g the *maximum a posteriori (MAP) estimate* of X given Y.

Thus, once we compute the conditional distribution for X given Y, the solution to the optimal estimation problem for an arbitrary loss function H becomes a *deterministic* minimization problem. This fact allows us to narrow our focus to computing the conditional distribution $P_{X|Y}$.

3.2. **Conditional Distributions.** Given two random variables X and Y, how do we compute the conditional distribution $P_{X|Y}$? Fortunately, this problem is straighforward if the law of Y is nondegenerate (one of the many instances in which we see the importance of nondegeneracy). The solution is a cornerstone of modern probability and statistics: Bayes' Theorem. Although we will use the following result extensively, it is so well known that we will omit a proof (see Theorem 2.7 of [8]).

Proposition 3.2.1 (Bayes' Theorem)**.** *Let X be a $(\mathrm{B}, \mathcal{B})$-valued random variable and let Y be a $(\mathrm{C}, \mathcal{C})$-valued random variable on the probability space $(\Omega, \mathcal{F}, \mathbf{P})$. Suppose there exists a $\mathcal{B} \otimes \mathcal{C}$-measurable function $\gamma : \mathrm{B} \times \mathrm{C} \to (0, \infty)$, a probability measure μ_X on B, and a probability measure μ_Y on C such that*

$$\mathbf{E}[f(X,Y)] = \iint\limits_{\mathrm{X}\times\mathrm{Y}} f(x,y)\gamma(x,y)\mu_X(dx)\mu_Y(dy)$$

for every bounded $\mathcal{B} \otimes \mathcal{C}$-measurable function f. Then,

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

is the conditional distribution of X given Y.

Remark. This measure-theoretic form of Bayes' Theorem may look alien compared to the form that is more commonly used, so let us attempt to establish a relation between the two. To start, let us use the notation $P_{X|Y}(A|y)$ and $\gamma(y|x)$ instead of $P_{X|Y}(y, A)$ and $\gamma(x, y)$, since the former is better aligned with the standard way of writing conditional probability, whereas the latter matches the notation we have used for transition kernels. This notation allows us to better see that we can intepret $\gamma(y|x)$ as $\mathbf{P}[Y = y|X = x]$. Now, suppose that B is finite, that is, $\mathrm{B} = x_1, \dots, x_d$. Then, for some $k \in [d]$, the more familiar version of the generalized Bayes' Theorem gives us

$$P_{X|Y}(x_k|y) = \mathbf{P}[X = x_k|Y = y] = \frac{\mathbf{P}[Y = y|X = x_k]\mathbf{P}[X = x_k]}{\sum_{i=1}^{d} \mathbf{P}[Y = y|X = x_i]\mathbf{P}[X = x_i]}.$$

Circling back to an arbitrary set B, we are trying to find the probability that X is in a *set* of points $A \subset \mathrm{B}$ given that $Y = y$. Since B is not necessarily finite, the statement $X = x_k$ becomes $X \in dx$, the sum in the denominator becomes an integral over B, and the numerator becomes an integral over A. Thus, our expression becomes

$$\begin{aligned} P_{X|Y}(A|y) = \mathbf{P}[X \in A|Y = y] &= \frac{\int I_A(x)\mathbf{P}[Y = y|X \in dx]\mathbf{P}[X \in dx]}{\int \mathbf{P}[Y = y|X \in dx]\mathbf{P}[X \in dx]} \\ &= \frac{\int I_A(x)\gamma(y|x)\mu_X(dx)}{\int \gamma(y|x)\mu_X(dx)}, \end{aligned}$$

which is precisely the version presented in the proposition above.

Returning to hidden Markov models, for the remainder of this section, we will consider a given hidden Markov model $(X_k, Y_k)_{k\geq 0}$ with signal state space $(\mathrm{X}, \mathcal{X})$, observation state space $(\mathrm{Y}, \mathcal{Y})$, transition kernel P, observation kernel Φ, and initial measure μ. We will assume that the observations are nondegenerate, that is, Φ has a strictly positive observation density Υ with respect to a reference (Lebesgue) measure ν.

Our goal is to solve the filtering, smoothing, and prediction estimation problems. That is, compute the conditional expectation $\mathbf{E}[f(X_n)|Y_{0:k}]$ for any function f and any n and k, which equivalent to finding the conditional distributions

$$\phi_{k|n} = P_{X_k|Y_{0:n}} \text{ for } k, n \geq 0.$$

Recall that the goal of the *filtering* problem is to compute the *filtering distributions* $\phi_k := \phi_{k|k}$ for $k \geq 0$. Similarly, the goal of the *smoothing* problem is to compute the *smoothing distributions* $\phi_{k|n}$ for $k < n$, and likewise the goal of the *prediction* problem is to compute the *prediction distributions* $\phi_{k|n}$ for $k > n$. The fact that the filtering, smoothing, and prediction problems arise from the choice of times to consider the signal and observation processes is made even more apparent by the conditional distributions. We will now explore how each of the three problems can be computed recursively, as previously promised.

3.3. **Filtering.** We know from Lemma 2.2.2 that the finite-dimensional distributions of $(X_k, Y_k)_{k\geq 0}$ are given by

$$\mathbf{E}[f(X_{0:k}, Y_{0:k})] = \int \cdots \int f(x_{0:k}, y_{0:k}) \prod_{i=0}^{k} \Upsilon(x_i, y_i)\nu(dy_i) \times \mu(dx_0) \prod_{i=1}^{k} P(x_{i-1}, dx_i) \tag{3.1}$$

Using this expression, we can obtain the filtering distribution with Bayes' Theorem.

Definition 3.3.1 (Unnormalized Filtering Distribution)**.** Let $k \geq 0$. The *unnormalized filtering distribution* α_k is the kernel $\alpha_k : \mathrm{Y}^{k+1} \times \mathcal{X} \to \mathbb{R}^+$ defined by

$$\alpha_k(y_{0:k}, A) = \int \cdots \int I_A(x_k)\mu(dx_0)\Upsilon(x_0, y_0) \prod_{i=1}^{k} \Upsilon(x_i, y_i)P(x_{i-1}, dx_i)$$

for all $y_0, \ldots, y_k \in \mathrm{Y}$ and $A \in \mathcal{X}$.

Note that the kernel α_k is not a transition kernel; typically, $\alpha_k(y_{0:k}, \mathrm{X}) \neq 1$ (compare to definition 2.1.1). We will see that the *normalization* of α_k (i.e., constructing a transition kernel from α_k) results in exactly the filtering distribution ϕ_k.

Theorem 3.3.2 (Unnormalized Filtering Recursion)**.** *The filtering distribution ϕ_k can be computed as*

$$\phi_k(y_{0:k}, A) = \frac{\alpha_k(y_{0:k}, A)}{\alpha_k(y_{0:k}, \mathrm{X})} \tag{3.2}$$

for every $A \in \mathcal{X}$ and $y_0, \ldots, y_k \in \mathrm{Y}$. Moreover, the unnormalized filtering distribution α_k can be computed recursively according to

$$\alpha_k(y_{0:k}, A) = \iint I_A(x)\Upsilon(x, y_k)P(x', dx)\alpha_{k-1}(y_{0:k-1}, dx')$$

with the initial condition

$$\alpha_0(y_0, A) = \int I_A(x)\Upsilon(x, y_0)\mu(dx). \tag{3.3}$$

Proof. To apply Bayes, we must first define the probability measures μ_Y and μ_X on Y^{k+1} and X^{k+1}, respectively. Let μ_Y be the product measure

$$\mu_Y(dy_{0:k}) = \nu(dy_0)\cdots\nu(dy_k).$$

Similarly, define μ_X as

$$\mu_X(dx_{0:k}) = P(x_{k-1}, dx_k)\cdots P(x_0, dx_1)\mu(dx_0)$$

and define the function $\gamma : \mathrm{X}^{k+1}\times\mathrm{Y}^{k+1} \to (0,\infty)$ by

$$\gamma(x_{0:k}, y_{0:k}) = \Upsilon(x_0, y_0)\cdots\Upsilon(x_k, y_k).$$

Then, by (3.1), we have

$$\mathbf{E}[f(X_{0:k}, Y_{0:k})] = \int\cdots\int f(x_{0:k}, y_{0:k})\gamma(x_{0:k}, y_{0:k})\mu_X(dx_{0:k})\mu_Y(dy_{0:k}).$$

Hence, we can now apply Bayes' Theorem (Proposition 3.2.1), which gives us

$$\int\cdots\int f(x_{0:k})P_{X_{0:k}|Y_{0:k}}(y_{0:k}, dx_{0:k}) = \frac{\int\cdots\int f(x_{0:k})\gamma(x_{0:k}, y_{0:k})\mu_X(dx_{0:k})}{\int\cdots\int \gamma(x_{0:k}, y_{0:k})\mu_X(dx_{0:k})}.$$

It follows that

$$\begin{aligned}\int f(x)\phi_k(y_{0:k}, dx) &= \int f(x_k)P_{X_k|Y_{0:k}}(y_{0:k}, dx_k)\\ &= \frac{\int\cdots\int f(x_k)\gamma(x_{0:k}, y_{0:k})\mu_X(dx_{0:k})}{\int\cdots\int \gamma(x_{0:k}, y_{0:k})\mu_X(dx_{0:k})}\\ &= \frac{\int f(x_k)\alpha_k(y_{0:k}, dx_k)}{\int \alpha_k(y_{0:k}, dx_k)}\end{aligned}$$

by Definition 3.3.1. Thus, we obtain the filtering distribution:

$$\phi_k(y_{0:k}, A) = \frac{\alpha_k(y_{0:k}, A)}{\alpha_k(y_{0:k}, \mathrm{X})}.$$

Furthermore, by construction, $\alpha(y_0, A) = \displaystyle\int I_A(x)\Upsilon(x, y_0)\mu(dx)$. Therefore, the recursion for α_k follows from the fact that

$$\begin{aligned}\alpha_k(y_{0:k}, A) &= \int\cdots\int I_A(x_k)\gamma(x_{0:k}, y_{0:k})\mu_X(dx_{0:k})\\ &= \iint I_A(x)\Upsilon(x, y_k)P(x', dx)\alpha_{k-1}(y_{0:k-1}, dx'),\end{aligned}$$

completing the proof. □

By combining the expression for the filtering distribution (3.2) with the recursive formula for the unnormalized filtering distribution (3.3), we immediately obtain an expression for computing the former directly–the *normalized* filtering recursion. It is normalized in the sense that each step of the recursion involves a rescaling, as opposed to the unnormalized recursion in which the expression is normalized only after the recursion has terminated.

Corollary 3.3.3 (Filtering Recursion)**.** *The filtering distribution ϕ_k can be computed recursively according to*

$$\phi_k(y_{0:k}, A) = \frac{\iint I_A(x)\Upsilon(x, y_k)P(x', dx)\phi_{k-1}(y_{0:k-1}, dx')}{\iint \Upsilon(x, y_k)P(x', dx)\phi_{k-1}(y_{0:k-1}, dx')}$$

with the initial condition

$$\phi_0(y_0, A) = \frac{\int I_A(x)\Upsilon(x, y_0)\mu(dx)}{\int \Upsilon(x, y_0)\mu(dx)}. \tag{3.4}$$

Proof. Note that by Theorem 3.3.2,

$$\phi_0(y_0, A) = \frac{\alpha_0(y_0, A)}{\alpha_0(y_0, \mathrm{X})} = \frac{\int I_A(x)\Upsilon(x, y_0)\mu(dx)}{\int \Upsilon(x, y_0)\mu(dx)}.$$

The recursion can be read from (3.2) and (3.3). □

Thus, instead of first computing α_k and then normalizing to obtain ϕ_k, we can compute ϕ_k directly. The fact that this computation is recursive makes writing an algorithm to compute the filtering recursion ϕ_k efficiently a straighforward exercise.

3.4. **Smoothing.** The smoothing distributions $\phi_{k|n}$ $(k < n)$, just like the filtering distributions, can be found with Bayes' Theorem. However, the computation will be divided into two parts, since the "past" observations $Y_0, \dots, Y_k$ and the "future" observations $Y_{k+1}, \dots, Y_n$ will be considered differently (that is, "past" and "future" relative to the time k at which we wish to find the conditional distribution of the signal X_k).

Definition 3.4.1 (Unnormalized Smoothing Density)**.** Let $0 \le k < n$. The *unnormalized smoothing density* $\beta_{k|n}$ is the function $\beta_{k|n} : \mathrm{X} \times \mathrm{Y}^{n-k} \to (0, \infty)$ defined as

$$\beta_{k|n}(x_k, y_{k+1:n}) = \int \cdots \int \prod_{i=k+1}^{n} \Upsilon(x_i, y_i)P(x_{i-1}, dx_i)$$

for all $y_{k+1}, \dots, y_n \in \mathrm{Y}$ and $x_k \in \mathrm{X}$.

There are two important things to note about the definition of the unnormalized smoothing density $\beta_{k|n}$. Firstly, $\beta_{k|n}$ is a *density* and not a *distribution* as opposed to the unnormalized smoothing distribution α_k (compare with Definition 3.3.1). Furthermore, the domain of β_n is $\mathrm{X} \times \mathrm{Y}^{n-k}$, since $\beta_{k|n}$ is a function of the observations $y_{k+1}, \dots, y_n$, i.e., all "future" observations.

Remark. The fact that $\beta_{k|n}$ is a *density* means that it is not a kernel, which is clear from the fact that, by definition, $\beta_{k|n}$ is not a function of $\mathcal{X}$ (compare with Definition 2.1.1).

Following in the same vein as the filtering problem, we will again use Bayes to show that the unnormalized smoothing density can be computed recursively.

Theorem 3.4.2 (Unnormalized Smoothing Recursion)**.** *The smoothing distribution $\phi_{k|n}$, $(k < n)$ can be computed as*

$$\phi_{k|n}(y_{0:n}, A) = \frac{\int I_A(x)\beta_{k|n}(x, y_{k+1:n})\alpha_k(y_{0:k}, dx)}{\int \beta_{k|n}(x, y_{k+1:n})\alpha_k(y_{0:k}, dx)} \tag{3.5}$$

for every $A \in \mathcal{X}$ and $y_0, \dots, y_n \in \mathrm{Y}$. Moreover, the unnormalized smoothing densities $\beta_{k|n}$ can be computed with the backward recursion

$$\beta_{k|n}(x, y_{k+1:n}) = \iint \beta_{k+1|n}(x', y_{k+2:n})\Upsilon(x', y_{k+1})P(x, dx') \tag{3.6}$$

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

The proof of this theorem will follow in the spirit of the proof for Theorem 3.3.2.

Proof. Using the same notation as in the proof of Theorem 3.4.2, we have

$$\begin{aligned}
\int f(x)\phi_{k|n}(y_{0:n}, dx) &= \int f(x_k) P_{X_k|Y_{0:n}}(y_{0:n}, dx_k) \\
&= \int \cdots \int f(x_k) P_{X_{0:n}|Y_{0:n}}(y_{0:n}, dx_{0:n}) \\
\text{(Proposition 3.2.1)} \qquad &= \frac{\int \cdots \int f(x_k)\gamma(x_{0:n}, y_{0:n})\mu_X(dx_{0:n})}{\int \cdots \int \gamma(x_{0:n}, y_{0:n})\mu_X(dx_{0:n})}.
\end{aligned}$$

Now, we can split our expression into functions of the observations $y_0, \dots, y_k$ and $y_{k+1}, \dots, y_n$, i.e., $\beta_{k|n}$ and α_k. Note that

$$\begin{aligned}
&\frac{\int \cdots \int f(x_k)\gamma(x_{0:n}, y_{0:n})\mu_X(dx_{0:n})}{\int \cdots \int \gamma(x_{0:n}, y_{0:n})\mu_X(dx_{0:n})} \\
&= \frac{\int \cdots \int \Big(f(x_k)\mu(dx_0)\Upsilon(x_0, y_0)\prod_{i=1}^{k}\Upsilon(x_i, y_i)P(x_{i-1}, dx_i)\Big)\left(\prod_{i=k+1}^{n}\Upsilon(x_i, y_i)P(x_{i-1}, dx_i)\right)}{\int \cdots \int \Big(\prod_{i=1}^{k}\Upsilon(x_i, y_i)P(x_{i-1}, dx_i)\Big)\left(\prod_{i=k+1}^{n}\Upsilon(x_i, y_i)P(x_{i-1}, dx_i)\right)} \\
&= \frac{\int f(x_k)\beta_{k|n}(x_k, y_{k+1:n})\alpha_k(y_{0:k}, dx_k)}{\int \beta_{k|n}(x_k, y_{k+1:n})\alpha_k(y_{0:k}, dx_k)}.
\end{aligned}$$

Thus, we obtain the smoothing distribution

$$\phi_{k|n}(y_{0:n}, A) = \frac{\int I_A(x)\beta_{k|n}(x, y_{k+1:n})\alpha_k(y_{0:k}, dx)}{\int \beta_{k|n}(x, y_{k+1:n})\alpha_k(y_{0:k}, dx)}.$$

Furthermore, by convention, we set the empty product $\prod_\emptyset = 1$, so by construction, $\beta_{n|n} = 1$. Therefore, the backward recursion for $\beta_{k|n}$ follows from the fact that

$$\begin{aligned}
\beta_{k|n}(x, y_{k+1:n}) &= \int \cdots \int \gamma(x_{k+1:n}, y_{k+1:n})\mu_X(dx_{k+1:n}) \\
&= \iint \beta_{k+1|n}(x', y_{k+2:n})\Upsilon(x' y_{k+1})P(x, dx'),
\end{aligned}$$

completing the proof. □

We can again obtain a *normalized* recursion for the smoothing distribution with an argument similar to the normalized filtering recursion. However, note that both the unnormalized smoothing densities $\beta_{k|n}$ and filtering distributions α_k appear in (3.5). Thus, we must first make a *forward* pass in time through the observations to compute the filtering distributions and then a *backward* pass to compute the smoothing densities. This procedure leads to the well-known *forward-backward* algorithm which we explore further in later sections. It is also for this reason that α_k is often called the *forward kernel* and $\beta_{k|n}$ the *backward function.*

Remark. A helpful mneumonic for remembering the direction of the filtering and smoothing recursions is to note that the computation for α_k starts at α_0, with each step of the recursion computing the expression that comes *after* (in time) the expression in the previous step, whereas the computation of $\beta_{k|n}$ starts at $\beta_{n|n}$, with each step of the recursion computing the expression that comes *before* the previous step; that is, "α_k after, $\beta_{k|n}$ before."

Corollary 3.4.3 (Smoothing Recursion)**.** *For $k < n$, define the function $\bar{\beta}_{k|n} : \mathrm{X} \times \mathrm{Y}^{n+1} \to (0, \infty)$ by the backward recursion*

$$\bar{\beta}_{k|n}(x, y_{0:n}) = \frac{\int \bar{\beta}_{k+1|n}(x', y_{0:n})\Upsilon(x', y_{k+1})P(x, dx')}{\iint \Upsilon(x', y_{k+1})P(x, dx')\phi_k(y_{0:k}, dx)} \tag{3.7}$$

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

$$\phi_{k|n}(y_{0:n}, A) = \int I_A(x)\bar{\beta}_{k|n}(x, y_{0:n})\phi_k(y_{0:k}, dx) \tag{3.8}$$

for every $A \in \mathcal{X}$ and $y_0, \ldots, y_n \in \mathrm{Y}$.

Let us compare $\bar{\beta}_{k|n}$ with $\beta_{k|n}$ (Definition 3.4.1). The most significant difference between the two functions is that the domain of $\bar{\beta}_{k|n}$ is now $\mathrm{X} \times \mathrm{Y}^{n+1}$, as opposed to the domain of $\beta_{k|n}$, which is $\mathrm{X} \times \mathrm{Y}^{n-k}$. This difference tells us that $\bar{\beta}_{k|n}$ is a function of *all* observations, whereas $\beta_{k|n}$ is a function of only the "future" observations.

Proof. From the unnormalized smoothing recursion, we immediately obtain

$$\bar{\beta}_{k|n} = \frac{\iint \bar{\beta}_{k+1|n}(x', y_{0:n})\Upsilon(x', y_{k+1})P(x, dx')}{\iint \bar{\beta}_{k+1|n}(x', y_{0:n})\Upsilon(x', y_{k+1})P(x, dx')\phi_k(y_{0:k}, dx)}$$

with $\bar{\beta}_{n|n} = 1$. Thus, we wish to show that for $k < n$

$$\iint \bar{\beta}_{k+1|n}(x', y_{0:n})\Upsilon(x', y_{k+1})P(x, dx')\phi_k(y_{0:k}, dx) = \iint \Upsilon(x', y_{k+1})P(x, dx')\phi_k(y_{0:k}, dx).$$

By the normalized filtering recursion (Corollary 3.3.3), we have

$$\frac{\iint \bar{\beta}_{k+1|n}(x', y_{0:n})\Upsilon(x', y_{k+1})P(x, dx')\phi_k(y_{0:k}, dx)}{\iint \Upsilon(x', y_{k+1})P(x, dx')\phi_k(y_{0:k}, dx)} = \int \bar{\beta}_{k+1|n}(x', y_{0:n})\phi_{k+1}(y_{0:k+1}, dx') = 1$$

by construction, completing the proof. □

Thus, we have both normalized and unnormalized recursions for the filtering and smoothing distributions. We now turn to prediction.

3.5. **Prediction.** Fortunately, the prediction problem, i.e., computing $\phi_{k|n}$ for $k > n$, is the simplest of our three estimation problems. We obtain a single, normalized, form of the prediction recursion.

Theorem 3.5.1 (Prediction Recursion)**.** *The prediction distribution $\phi_{k|n}$ $(k > n)$ can be computed recursively according to*

$$\phi_{k|n}(y_{0:n}, A) = \iint I_A(x)P(x', dx)\phi_{k-1|n}(y_{0:n}, dx') \tag{3.9}$$

for every $A \in \mathcal{X}$ and $y_0, \ldots, y_n \in \mathrm{Y}$, with the initial condition $\phi_{n|n} = \phi_n$.

Remark. The expression $\phi_{n|n} = \phi_n$ is mild abuse of notation; we are setting the starting value of the prediction recursion equal to the filtering distribution ϕ_n, where we have also defined the notation $\phi_n \coloneqq \phi_{n|n}$.

Instead of using Bayes' Theorem (which can certainly be applied), we opt for the instructive proof of Theorem 2.14 in [8] that utilizes some of the fundamental properties of hidden Markov models.

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

$$\mathbf{E}[f(X_k)|Y_{0:n}] = \mathbf{E}[\mathbf{E}[f(X_k)|X_{0:n}, Y_{0:n}]|Y_{0:n}]]$$

for $k > n$. Furthermore, by the Markov property of the signal $(X_k)_{k\geq 0}$,

$$\mathbf{E}[f(X_k)|X_{0:n}, Y_{0:n}] = P^{k-n}f(X_n).$$

Thus, $\mathbf{E}[f(X_k)|Y_{0:n}] = \mathbf{E}[P^{k-n}f(X_n)|Y_{0:n}]$, which is equivalent to

$$\int f(x)\phi_{k|n}(y_{0:n}, dx) = \int P^{k-n}f(x)\phi_n(y_{0:n}, dx),$$

for every bounded $\mathcal{X}$-measurable function f. Note that

$$\int P^{k-n}f(x)\phi_n(y_{0:n}, dx) = \int PP^{k-n-1}f(x)\phi_n(y_{0:n}, dx) = \int Pf(x)\phi_{k-1|n}(y_{0:n}, dx).$$

Therefore,

$$\phi_{k|n}(y_{0:n}, A) = \iint I_A(x)P(x', dx)\phi_{k-1|n}(y_{0:n}, dx'),$$

as desired. □

A simple consequence of this theorem is that

$$\phi_{k+1|k}(y_{0:k}, A) = \iint I_A(x)P(x', dx)\phi_k(y_{0:k}, dx'),$$

so by Corollary 3.3.3, we have

$$\phi_{k+1}(y_{0:k+1}, A) = \frac{\int I_A(x)\Upsilon(x, y_{k+1})\phi_{k+1|k}(y_{0:k}, dx)}{\int \Upsilon(x, y_{k+1})\phi_{k+1|k}(y_{0:k}, dx)}.$$

Hence, the filtering distribution ϕ_{k+1} can be expressed in terms of the one-step predictor $\phi_{k+1|k}$. As a result, the filtering recursion is often interpreted as a two-step procedure:

$$\phi_k \xrightarrow{\text{prediction}} \phi_{k+1|k} \xrightarrow{\text{correction}} \phi_{k+1}.$$

Of course, to fully appreciate the power of this elegant theory we need to consider some specific scenarios of state space. One of the most natural cases to study is finite state space, where, as we will now see, our general theory can be adopted easily to yield several notable algorithms.

4. Finite State Space Models

The setting in which the signal state space X is finite has been integral to the study of hidden Markov models since their inception. Not only do finite nodels emerge naturally in numerous practical applications from telecommunications to DNA sequencing, but finite state space is also a particularly convenient setting for computation. In a finite setting, all of the techniques we developed in the previous section become matrix computations, and can thus be implemented both precisely and efficiently on a computer. Throughout this section, we will consider a hidden Markov model $(X_k, Y_k)_{k\geq 0}$ on the state space $\mathrm{X} \times \mathrm{Y}$ with the restriction that the signal state space is $\mathrm{X} = \{1, \ldots, d\}$ (without loss of generality).[2] We follow the same notation as before for the transition kernel, obervation kernel, and initial measure, and will again adopt the assumption that the observations are nondegenerate. However, since the signal state space X is now finite, it is convenient to think of functions and measures as vectors and kernels as matrices: any function

[2] Technically, we can only label the elements of X without loss of generality. However, for the sake of convenience, we set X equal to the set of integers $1, \ldots, d$, which is truly just a semantic difference.

$f : \mathrm{X} \to \mathbb{R}$ is entirely described by the vector $\boldsymbol{f} = (f(1), \ldots, f(d))^\top \in \mathbb{R}^d$,[3] and any measure μ on X is entirely described by the vector $\boldsymbol{\mu} = (\mu(\{1\}), \ldots, \mu(\{d\}))^\top$. Furthermore, the transition kernel P is naturally represented by a $d \times d$ matrix $\boldsymbol{P}$ where $\boldsymbol{P}_{ij} = P(i, \{j\})$ for $i, j \in \mathrm{X}$. Just as before,

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

and

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

Lastly, we will represent the observation density Υ using matrices, but in a somewhat unique manner: for each observation $y \in \mathrm{Y}$, we define the matrix $\boldsymbol{\Upsilon}(y)$ where

$$(\boldsymbol{\Upsilon}(y))_{ij} = \Upsilon(i, y)\delta_{ij}$$

for $i, j \in \mathrm{X}$. That is, $\boldsymbol{\Upsilon}(y)$ is the $d \times d$ diagonal matrix with nonzero elements $(\boldsymbol{\Upsilon}(y))_{ii} = \Upsilon(i, y)$.

Remark. In much of the stastical literature, finite signal state space hidden Markov models are also assumed to have finite observation space. In this case, the observation kernel Φ is represented by a $d \times e$ matrix where $(\boldsymbol{\Phi})_{ij} = \Phi(i, \{j\})$. Of course, if the observation state space is not finite, then this representation is clunky, since the matrix will not be of finite size. Our definition uses the observation density Υ instead, allowing us to make no assumptions about the structure of Y. The tradeoff is that we must assume nondegeneracy, but, as we established, any "practical" model should be nondegenerate.

In the finite setting, we can interpret the initial measure $\boldsymbol{\mu}$ as the probability distribution of the signal process at time 0, that is, $\boldsymbol{\mu}_i = \mathbf{P}[X_0 = i]$. Furthermore, we can interpret the elements of the transition matrix $\boldsymbol{P}$ as conditional probability, where

$$(\boldsymbol{P})_{ij} = \mathbf{P}[X_{k+1} = j | X_k = i].$$

Thus, the transition matrix gives the probability of "transitioning" to the next state given the current state. Similarly, we can think of the elements of the observation density matrix $\boldsymbol{\Upsilon}(y)$ as the probability of making the observation $y \in \mathrm{Y}$ if the signal is in the state $i \in \mathrm{X}$:

$$(\boldsymbol{\Upsilon}(y))_{ii} = \mathbf{P}[Y_k = y | X_k = i].$$

Following this interpretation, it is intuitive that the non-diagonal elements of $\boldsymbol{\Upsilon}(y)$ are zero, since the signal cannot be in two states at once.

Using this matrix-vector notation, we will now reformulate the results of the estimation problems in the general setting, leading us to several important computational algorithms.

4.1. **Finite State Filtering, Smoothing, and Prediction.**

Remark. For the following, we fix an observation sequence $(y_k)_{k \geq 0}$, allowing us to omit the dependence of e.g., $\phi_{k|n}$ on the observed trajectory $y_0, \ldots, y_k$. Thus, we will write expressions such as $\alpha_k(y_{0:k}, dx)$ as $\alpha_k(dx)$, with the dependence on $y_0, \ldots, y_k$ being implicit. We will further simplify our notation by writing singleton sets as elements. For example, we will write $P(i, j)$ for $P(i, \{j\})$ and $\mu(1)$ for $\mu(\{1\})$.

[3] We will adopt the convention to use column vectors instead of row vectors, as is standard in most linear algebra settings. Notably, in the context of Markov processes, row vectors are typically used–this difference will be noted whenever there is possible ambiguity.

We begin with the unnormalized filtering recursion. Since the unnormalized filter α_k is a measure, we can represent it as a vector $\boldsymbol{\alpha}_k = (\alpha_k(1), \dots, \alpha_k(d))^\top$. Then, from Theorem 3.3.2, we have

$$\boldsymbol{\alpha}_0 = \boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}, \qquad \boldsymbol{\alpha}_k = \boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^\top\boldsymbol{\alpha}_{k-1} \qquad (k \geq 1). \tag{4.1}$$

Note that we have $\boldsymbol{P}^\top\boldsymbol{\alpha}_{k-1}$ in the second expression; this is because the transition matrix is typically applied to a row vector, in which case we would have $\boldsymbol{\alpha}_{k-1}^\top\boldsymbol{P}$. However, since $\boldsymbol{\alpha}_{k-1}$ is a column vector, we have $(\boldsymbol{\alpha}_{k-1}^\top\boldsymbol{P})^\top = \boldsymbol{P}^\top\boldsymbol{\alpha}_{k-1}$.

Letting $\mathbf{1} = (1, \dots, 1)^\top \in \mathbb{R}^d$, we see that the vector form of the normalized filter ϕ_k is $\boldsymbol{\phi}_k = \dfrac{\boldsymbol{\alpha}_k}{\mathbf{1}^\top\boldsymbol{\alpha}_k}$. Although this form of $\boldsymbol{\phi}_k$ is simple, it is still in terms of the unnormalized filter $\boldsymbol{\alpha}$. By Corollary 3.3.3, the normalized filter can be computed directly through the normalized recursion

$$\boldsymbol{\phi}_0 = \frac{\boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}_0}{\mathbf{1}^\top\boldsymbol{\Upsilon}(y_0)\boldsymbol{\mu}}, \qquad \boldsymbol{\phi}_k = \frac{\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^\top\boldsymbol{\phi}_{k-1}}{\mathbf{1}^\top\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^\top\boldsymbol{\phi}_{k-1}} \qquad (k \geq 1). \tag{4.2}$$

We now consider the smoothing problem. Similar to the unnormalized filters, the unnormalized smoothing densities $\beta_{k|n}$ can be represented as vectors $\boldsymbol{\beta}_{k|n} = (\beta_{k|n}(1), \dots, \beta_{k|n}(d))^\top$ (where we have again omitted the dependence on the observations). Then, by Theorem 3.4.2,

$$\boldsymbol{\beta}_{n|n} = \mathbf{1}, \qquad \boldsymbol{\beta}_{k|n} = \boldsymbol{P}\boldsymbol{\Upsilon}(y_{k+1})\boldsymbol{\beta}_{k+1|n} \qquad (k < n). \tag{4.3}$$

The smoothing distributions can be computed from the unnormalized smoothing densities $\boldsymbol{\beta}_{k|n}$ according to Theorem 3.4.2:

$$\boldsymbol{\phi}_{k|n} = \frac{\operatorname{diag}(\boldsymbol{\beta}_{k|n})\boldsymbol{\alpha}_k}{\boldsymbol{\beta}_{k|n}^\top\boldsymbol{\alpha}_k} = \frac{\operatorname{diag}(\boldsymbol{\beta}_{k|n})\boldsymbol{\phi}_k}{\boldsymbol{\beta}_{k|n}^\top\boldsymbol{\phi}_k}, \tag{4.4}$$

where $\operatorname{diag}(\beta)_{k|n}$ is the $d \times d$ diagonal matrix with the entries of $\boldsymbol{\beta}_{k|n}$ along the diagonal. We can also compute the smoothing distributions from the normalized smoothing densities $\bar{\beta}_{k|n}$, represented as vectors $\bar{\boldsymbol{\beta}}_{k|n}$, where

$$\bar{\boldsymbol{\beta}}_{n|n} = \mathbf{1}, \qquad \bar{\boldsymbol{\beta}}_{k|n} = \frac{\boldsymbol{P}\boldsymbol{\Upsilon}(y_{k+1})\bar{\boldsymbol{\beta}}_{k+1|n}}{\mathbf{1}^\top\boldsymbol{\Upsilon}(y_{k+1})\boldsymbol{P}^\top\boldsymbol{\phi}_k} \qquad (k < n). \tag{4.5}$$

Then, by Corollary 3.4.3, the smoothing distributions are given by

$$\boldsymbol{\phi}_{k|n} = \operatorname{diag}(\bar{\boldsymbol{\beta}}_{k|n})\boldsymbol{\phi}_k. \tag{4.6}$$

Lastly, prediction is again our simplest case–the vector form of the prediction recursion follows directly from Theorem 3.5.1:

$$\boldsymbol{\phi}_{n|n} = \boldsymbol{\phi}_n, \qquad \boldsymbol{\phi}_{k+1|n} = \boldsymbol{P}^\top\boldsymbol{\phi}_{k|n} \qquad (k \geq n). \tag{4.7}$$

All of the recursions we have obtained in the finite state space setting involve only matrix computations. Thus, they can be implemented efficiently on a computer. We will now consider one such implementation to compute the filtering and smoothing distributions: the forward-backward algorithm.

4.2. **The Forward-Backward Algorithm.** As previously discussed, both the normalized and unnormalized smoothing recursions involve a *forward* pass (in time) to compute the filtering distributions and subsequently a *backward* pass to compute the smoothing densities (compare with Theorem 3.4.2 and Corollary 3.4.3). Based on our results in the finite state setting, we can sketch an outline for what such a "forward-backward" algorithm might look like. We can read directly from (4.2) to obtain the forward pass, but for the backward pass, we have two options: the normalized and unnormalized smoothing recursions. In general, the normalized recursion is preferable for computational purposes. The unnormalized recursion tends to change in magnitude significantly over time, resulting in catastrophic effects when it approaches or exceeds the upper and lower limits of machine precision. In contrast, the computations of the normalized recursion stay within a relatively stable range, which typically mitigates the risk of floating-point errors and the like. Thus, (4.5) and (4.6) give the backward pass.

Notice that the denominator of $\bar{\boldsymbol{\beta}}_{k|n}$ is just the denominator of $\boldsymbol{\phi}_{k+1}$; thus, the values $\boldsymbol{\Upsilon}(y_k)\boldsymbol{P}^\top\boldsymbol{\phi}_{k-1}$, $(k > 1)$ should be stored to make the algorithm more efficient. Hence, we obtain the *forward-backward algorithm*:

Algorithm 4.1: Forward-Backward Algorithm

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

4.3. **The Viterbi Algorithm.** Let us now consider a new type of problem in the finite state space setting: *decoding* a finite state signal path $x_0, \dots, x_n$ from an observation trajectory $y_0, \dots, y_n$. That is, we wish to do our best to determine the true value of the signal using only our observations. For example, perhaps a friend is shouting something across a crowded cafeteria, or we are making a phone call in an area with poor cellular reception and can only understand one out of every few words through the static. In both cases, we have a *finite alphabet* message (each word is composed from an alphabet of 26 letters) being transmitted through a *noisy channel.* In other words, the medium (air in the cafeteria, the atmosphere, etc.) through which the signal (sound, radio waves, etc.) is sent tends to alter or corrupt the message.

Then, the signal state space X is the signal alphabet, the signal $(X_k)_{0 \le k \le n}$ is the message, and the observation trajectory $(Y_k)_{0 \le k \le n}$ is the likely corrupted received message transmitted through the channel. Our goal is to determine, to the best of our ability, what the transmitted message was from the message we received. Hence, we wish to construct random variables $\hat{X}_0, \dots, \hat{X}_n$ that are functions of the observed sequence, $\hat{X}_k = f_k(Y_{0:n})$, such that the estimate $(\hat{X}_k)_{0 \le k \le n}$ is "as close as possible" to the original signal $(X_k)_{0 \le k \le n}$. Of course, the problem boils down to what "as close as possible" means. Perhaps we wish to construct $\hat{X}_0, \dots, \hat{X}_n$ so that the most expected number of individual symbols (bits, letters, etc.) are decoded correctly. In other words,

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

This approach seems perfectly reasonable, but it has a significant flaw. We will illustrate this with the following simple example.

Example 4.3.1. Let $(X_k, Y_k)_{k\geq 0}$ be a hidden Markov model with signal state space $\mathrm{X} = \{0,1\}$ and transition probabilities $P(0,1) = P(1,0) = 1$, i.e., the signal alternates between 0 and 1. Furthermore, let the initial measure be $\mu(0) = \mu(1) = 1/2$. For simplicity, suppose we have made no observations. Then, $\phi_{k|n}(i) = \mathbf{P}[X_k = i] = 1/2$ for every i, k, n.

We now wish to estimate the signal. Since all of the individual probabilities are $1/2$, any estimate $\hat{X}_{0:n}$ has the same expected number of correctly decoded symbols. So, we may reasonably choose $\hat{X}_k = 0,\ k = 1, \ldots, n$ as the optimal estimator. However, by definition of the transition probabilities, the signal path $X_k = 0$ for all k has probability zero: $P(0,0) = 0$. Thus, our estimation method has resulted in an impossible message!

We can now clearly see the drawback of this method for estimation: an estimate of the signal path which maximizes the number of correctly decoded symbols does not necessarily maximize the probability that the entire signal path is decoded correctly, and, in extreme cases, can even result in an estimate that is not a possible signal path.

Let us now try an alternative estimation approach that directly addresses this issue: construct $\hat{X}_0, \ldots, \hat{X}_n$ so that the probability of the entire signal path being decoded correctly is maximized. That is,

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

Once again, the solution to the maximum probability path estimate can be found recursively. This solution leads to the famous *Viterbi algorithm.* Fortunately, we already know what it means to *estimate* a random variable–this is precisely Lemma 3.1.3. In this scenario, our loss function is constructed from the indicator of whether our estimate matches the original signal: $I_0(x - \hat{x})$. Thus, to find the maximum probability path estimate, we choose the functions f_k such that

$$(f_0(y_{0:n}), \ldots, f_n(y_{0:n})) = \underset{(\hat{x}_{0:n})}{\arg\max} \int \prod_{k=0}^{n} I_0(x_k - \hat{x}_k) P_{X_{0:n}|Y_{0:n}}(y_{0:n}, dx_{0:n}). \tag{4.8}$$

Remark. Note that we have arg max instead of arg min in this expression because we are *maximizing* the probability of our estimate instead of *minimizing* a loss function. Trivially, Lemma 3.1.3 still applies.

We can now apply Bayes Theorem (Proposition 3.2.1), which gives us

$$\int \prod_{k=0}^{n} I_0(x_k - \hat{x}_k) P_{X_{0:n}|Y_{0:n}}(y_{0:n}, dx_{0:n}) = \frac{\mu(\hat{x}_0)\Upsilon(\hat{x}_0, y_0)\prod_{k=1}^{n}\Upsilon(\hat{x}_k, y_k)P(\hat{x}_{k-1}, \hat{x}_k)}{\int\cdots\int \mu(dx_0)\Upsilon(x_0, y_0)\prod_{k=1}^{n}\Upsilon(x_k, y_k)P(x_{k-1}, dx_k)}. \tag{4.9}$$

However, since we are maximizing in terms of $(\hat{x}_0, \ldots, \hat{x}_n)$, the denominator can be ignored. Thus,

$$(f_0(y_{0:n}), \ldots, f_n(y_{0:n})) = \underset{(\hat{x}_{0:n})}{\arg\max}\, \mu(\hat{x}_0)\Upsilon(\hat{x}_0, y_0)\prod_{k=1}^{n}\Upsilon(\hat{x}_k, y_k)P(\hat{x}_{k-1}, \hat{x}_k).$$

Furthermore, since $\log x$ is increasing, $\underset{x}{\arg\max}\, f(x) = \underset{x}{\arg\max}\, \log f(x)$, and thus

$$(f_0(y_{0:n}), \ldots, f_n(y_{0:n})) =$$
$$\underset{(\hat{x}_{0:n})}{\arg\max}\left[\log(\mu(\hat{x}_0)\Upsilon(\hat{x}_0, y_0)) + \sum_{k=1}^{n}(\log P(\hat{x}_{k-1}, \hat{x}_k) + \log\Upsilon(\hat{x}_k, y_k))\right].$$

We have chosen to write our expression in this form in order to introduce the "Viterbi functions"

$$v_\ell(\hat{x}_\ell) = \max_{\hat{x}_{0:\ell-1}} \left[\log(\mu(\hat{x}_0)\Upsilon(\hat{x}_0, y_0)) + \sum_{k=1}^{\ell} (\log P(\hat{x}_{k-1}, \hat{x}_k) + \log \Upsilon(\hat{x}_k, y_k)) \right] \qquad (0 \leq \ell \leq n)$$

which are central to the Viterbi algorithm. The key feature of the Viterbi functions is that they can be compute recursively, as shown by the next theorem. The purpose of these functions–and the main idea of the Viterbi algorithm–is to compute the estimating functions $\{f_k(y_{0:n})\}_{k\geq 0}$ recursively.

Theorem 4.3.2 (Viterbi Recursion)**.** *The functions v_ℓ satisfy the forward 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 $f_\ell(y_{0:n})$, $\ell = 1, \ldots, n$ for the maximum probability path estimate given $Y_0, \ldots, Y_n$ satisfy the backward recursion

$$f_\ell = \arg\max_{\hat{x}_\ell} \{v_\ell(\hat{x}_\ell) + \log P(\hat{x}_\ell, f_{\ell+1})\}$$

with the terminal condition $f_n = \arg\max_{\hat{x}_n} v_n(\hat{x}_n)$.

Proof. The recursions for v_ℓ and f_ℓ are quickly verified by inspection. □

Remark. The backward recursion can be interpreted as repeatedly eliminating potential signal paths until only the most probable path remains. The first step is to find the most probable final state of the true signal and remove all possible paths whose final state differs from this state. Then, of the paths with the corresponding final state, the second-to-last state which maximizes the probability of these paths is found. The third-to-last state is considered next, and the process is repeated until the recursion reaches the first state, leaving only one signal path remaining. If multiple states are equally optimal at any stage of the recursion, then one can be chosen arbitrarily.

To summarize, we wish to find the maximum probability sequence $(\hat{x}_0, \ldots, \hat{x}_n)$, which we know is given by the estimator functions $(f_0(y_{0:n}), \ldots, f_n(y_{0:n}))$. Since these functions give the maximum arguments of (4.8), we can express them more conveniently in terms of functions $v_0, \ldots, v_n$ with arguments of the maxima identical to (4.9). As it turns out, both sets of functions can be computed recursively, which allows us to implement the computation as an algorithm. Furthermore, since we are in the finite state space setting, this algorithm will be entirely composed of matrix computations, and can therefore be implemented efficiently on a computer.

Once again, the algorithm consists of a forward and a backward pass. However, unlike the forward-backward algorithm, the observation path $y_0, \ldots, y_n$ is not directly used in the backward pass of the Viterbi algorithm, and therefore it does not need to be stored in memory. Of course, the values $v_\ell(i)$ must be stored for all ℓ, i. The *Viterbi algorithm* is summarized in Algorithm 4.2.

```
Algorithm 4.2: Viterbi Algorithm
    $v_0(i) \leftarrow \log \boldsymbol{\mu}_i + \log \Upsilon(i, y_0),\ i = 1, \dots, d$;
    for $k = 1, \dots, n$ do
forward | $b_k(i) \leftarrow \underset{j=1,\dots,d}{\arg\max} \{v_{k-1}(j) + \log \boldsymbol{P}_{ji}\},\ i = 1, \dots, d$;
        | $v_k(i) \leftarrow v_{k-1}(b_k(i)) + \log \boldsymbol{P}_{b_k(i) i} + \log \Upsilon(i, y_k),\ i = 1, \dots, d$;
    end
    $f_n \leftarrow \underset{j=1,\dots,d}{\arg\max}\, v_n(j)$;
    for $k = 1, \dots, n$ do
backward | $f_{n-k} \leftarrow b_{n-k+1}(f_{n-k+1})$;
    end
```

5. Linear Gaussian State Space Models

We now consider the second scenario in which the general estimation results we obtained can be computed precisely: the linear Gaussian setting i.e., the setting where the state space of our hidden Markov model is governed by a linear system of equations. However, we have already covered a lot of ground in the study of estimation in hidden Markov models–the general theory gives us the underlying recursion behind computing the filtering, smoothing, and prediction distributions and the case of finite state space yields several of the most important algorithms for computing these distributions. These algorithms are almost identical in the linear Gaussian case (aside from perhaps some different notation) and the general recursions naturally hold as well, since it is just a specific scenario of the general theory. Instead of re-deriving all of the estimation distributions with new notation, we will focus on the most famous result in the linear Gaussian setting: the Stratonovich-Kalman-Bucy, or simply Kalman, filter.[4]

We will consider a hidden Markov model $(X_k, Y_k)_{k \geq 0}$ with signal state space $\mathrm{X} = \mathbb{R}^d$ and observation state space $\mathrm{Y} = \mathbb{R}^{d_Y}$. Thus, the signal state and observation state will be represented by d-dimensional and d_Y-dimensional vectors, respectively. Our state space is governed by two equations:

$$X_k = F_{k-1}(X_{k-1}, \xi_k), \qquad\qquad Y_k = G_k(X_k, \eta_k),$$

where F_{k-1} and G_k are vector-valued functions, ξ_k is random "system noise" and η_k is random "observation noise." To complete our model, we assume that $X_0 = a$ for some $a \in \mathbb{R}^d$. It is immediately clear that this is a hidden Markov model in spirit; we have a signal whose evolution depends only on the previous state and an observation that is a "noisy functional" of our signal. However, as it stands, the signal state space is continuous, so it seems like estimation is still infinite-dimensional.

The key to making this model computationally tractable is to assume that both the state and observation equations are linear (i.e., that the function F_{k-1} and G_k are linear) and that the signal and observation states X_k and Y_k, as well as the noise terms ξ_k and η_k, are jointly Gaussian. As proved in [4], the conditional distribution of a Gaussian variable is a Gaussian (i.e., normal) distribution, so the filtering, smoothing, and prediction distributions will all be Gaussian.

[4] Over time, the filter has become known as the Kalman Filter, after Rudolf E. Kalman, despite critical work on the technique being completed earlier by Richard S. Bucy and the filter being a special case of a more general filtering method developed prior by Ruslan Stratonovich.

Since every Gaussian variable is entirely determined by its mean vector and covariance matrix–which are d-dimensional and $d \times d$ dimensional, respectively, for X_k and ξ_k and d_Y-dimensional and $d_Y \times d$ dimensional, respectively, for Y_k and η_k–our computations will exclusively be with finite-dimensional matrices. Thus, although our state space is infinite, our estimation recursions are once again finite-dimensional!

Formally, our assumptions imply that our state space is given by the following system of equations:

$$X_k = F_{k-1}X_{k-1} + \xi_k, \qquad Y_k = G_k X_k + \eta_k, \tag{5.1}$$

where ξ_k[5] is a random d-dimensional vector, η_k is a random d_Y-dimensional vector, F_{k-1} is a $d \times d$ matrix, and G_k is a $d_Y \times d$ matrix. It is evident that F_{k-1} and G_k are the transition kernel and observation kernel, respectively. We will once again assume that both kernels are time homogeneous and refer to them as F and G.

The linear Gaussian assumption also implies that the signal noise $\xi_k,\ k \geq 0$ are i.i.d. $\mathcal{N}(0, \Sigma_\xi)$ and the observation noise $\eta_k, k \geq 0$ are i.i.d $\mathcal{N}(0, \Sigma_\eta)$, where Σ_ξ and Σ_η are the covariance matrices of the corresponding noise variables. We also assume that $X_0 \sim \mathcal{N}(\mu_0, \Sigma_{X_0})$ for some $\mu_0 \in \mathbb{R}^d$ and that ξ_k are independent of η_k.

We will further prescribe the structure of our model conditional on the available information at time k, i.e., the observations $y_0, \dots, y_{k-1}$. For $k \geq 0$, define the Borel set $\mathcal{F}_k$ by

$$\mathcal{F}_k = \sigma(y_0, \dots, y_k),$$

that is, the sigma algebra generated by the observations $y_{0:k}$. Then, we assume the following:

(i) Conditional on the available information, the observation noise terms have mean zero:

$$\mathbf{E}[\eta_k | \mathcal{F}_{k-1}] = 0.$$

(ii) The covariance matrix for the observation noise is time homogeneous:

$$\mathrm{var}[\eta_k | \mathcal{F}_{k-1}] = \Sigma_\eta.$$

And lastly,

(iii) The observation noise η_k and the signal X_k are uncorrelated:

$$\mathbf{E}[X_k \eta_k^\top | \mathcal{F}_{k-1}] = 0.$$

Together, these assumptions guarentee that the observation noise $(\eta_k)_{k \geq 0}$ is *white noise*, that is, observations generated from an i.i.d. sequence of random variables with mean zero.

Now that we have the foundation for our model, we can derive the Kalman estimation for linear Gaussian state space models. However, we will take a slightly different approach from the finite state space setting, opting for an emphasis on information.

5.1. **One-Step Prediction and Kalman Filtering.** In contrast to the approach we took earlier to derive the general filtering recursion in Theorem 3.3.2, in the linear Gaussian setting we follow the derivation in [2] and compute the Kalman filter from the one-step prediction, i.e., estimating X_{k+1} given $\mathcal{F}_k$. However, since we are in the linear Gaussian setting, we have the additional restriction that our best estimator $\hat{X}$ must be *linear*, i.e.,

$$\hat{X} = \arg\min_f \mathbf{E}[(X - f(Y))^2].$$

[5] It may seem strange to write ξ_k instead of ξ_{k-1}, but the purpose of using the former is to emphasize that ξ_k is independent of X_n for $n < k$, an assumption we will make explicit below.

Fortunately, we know by Lemma 3.1.1 that this is precisely $\mathbf{E}[X|Y]$! Thus, the best estimator for X_{k+1} given $\mathcal{F}_k$ is $\mathbf{E}[X_{k+1}|\mathcal{F}_k]$.

Remark. We will differ slightly from the previous sections in our notation and write

$$\hat{X}_{k|n} = \mathbf{E}[X_k|\mathcal{F}_n],\ k, n \geq 0$$

instead of $\phi_{k|n} = P_{X_k|Y_{0:n}}$, since our estimator is now the conditional expectation instead of the conditional probability. However, we keep the tradition of defining $\hat{X}_k = \hat{X}_{k|k}$.

From L^2-theory, we know that the best linear one-step predictor $\hat{X}_{k+1|k}$ can be interpreted geometrically as the orthogonal projection of X_{k+1} onto the span of $Y_0, \ldots, Y_k$. We will define

$$E_k(X) = \mathbf{E}[X|\mathcal{F}_k],\ k \geq 0$$

to reflect this fact; note that $E_k(X_{k+1}) = \hat{X}_{k+1|k}$. We will also want to keep track of the covariance matrix of our estimate, so we define

$$\Omega_{k+1|k} = \mathbf{E}[(X_{k+1} - \hat{X}_{k+1|k})(X_{k+1} - \hat{X}_{k+1|k})^\top],\ k \geq 0. \tag{5.2}$$

The theme for calculating the prediction $\hat{X}_{k+1|k}$ will again be recursion, this time with an additional sprinkle of information theory. On that note, we define the *innovation* by

$$I_k = Y_k - E_{k-1}(Y_k),\ k > 0, \qquad\qquad I_0 = Y_0$$

that is, I_k is the "new information" contained in the observation Y_k. Note that the innovations are mean 0:

$$\mathbf{E}[I_k] = \mathbf{E}[Y_k] - \mathbf{E}[E_{k-1}(Y_k)] = \mathbf{E}[Y_k] - \mathbf{E}[\mathbf{E}[Y_k|Y_{0:k-1}]] = \mathbf{E}[Y_k] - \mathbf{E}[Y_k] = 0,$$

where the second-to-last equality follows from the law of total expectation.

Since $E_{k-1}(Y_k)$ gives the orthogonal projection of Y_k onto the span of $Y_0, \ldots, Y_{k-1}$ and I_k is equal to precisely Y_k minus this component, I_k is orthogonal to the span of $Y_0, \ldots, Y_{k-1}$. Thus, I_k is orthogonal to I_k for all $n < k$, so $I_k,\ k \geq 0$ are i.i.d., and can also be thought of as white noise.

The critical nondegeneracy assumption manifests here in the form of assuming that the covariance matrix for the innovation, Σ_{I_k}, is invertible. Note that, unlike the covariance matrices for the observation and signal noise, Σ_{I_k} is not time homegeneous.

Remark. The intuition behind introducing the innovation I_k comes from the general theory, which tells us that prediction can be performed recursively–that is, if the prediction $\hat{X}_{k|k-1}$ of X_k is found using the observations $Y_0, \ldots, Y_{k-1}$, then the prediction $\hat{X}_{k+1|k}$ can be found using only the new observation Y_k and the previous prediction $\hat{X}_{k-1|k}$ (and the covariance matrix $\Omega_{k+1|k}$). However, only the *new* information contained in Y_k matters for computing $\hat{X}_{k+1|k}$, since the old information will already be incorporated in $\hat{X}_{k|k-1}$. Thus, we use the innovation I_k (see [1] for further reading on the innovation approach).

We now obtain an expression for the one-step-ahead prediction, that is, a method of computing $\hat{X}_{k+1|k}$ from $\hat{X}_{k|k-1}$.

Theorem 5.1.1 (One-Step Prediction)**.** *The best linear estimate $\hat{X}_{k+1|k}$ is given by*

$$\hat{X}_{k+1|k} = F\hat{X}_{k|k-1} + \Theta_k \Delta_k^{-1}(Y_k - G\hat{X}_{k|k-1}),$$

where $\Delta_k = G\Omega_{k|k-1}G^\top + \Sigma_\eta$ and $\Theta_k = F\Omega_{k|k-1}G^\top$ for $k > 0$.

Proof. By our assumptions of the model, the observation noise η_k is independent from the observations $Y_0, \dots, Y_{k-1}$. Thus,

$$\begin{aligned}
& & E_{k-1}(Y_k) &= E_{k-1}(GX_k + \eta_k) \\
&\text{(linearity)} & &= E_{k-1}(GX_k) + E_{k-1}(\eta_k) \\
&(E_{k-1}(\eta_k) = 0) & &= E_{k-1}(GX_k) \\
&\text{(linearity)} & &= GE_{k-1}(X_k) = G\hat{X}_{k|k-1} \\
& & \implies I_k &= GX_k + \eta_k - G\hat{X}_{k|k-1} = G(X_k - \hat{X}_{k|k-1}) + \eta_k.
\end{aligned}$$

Since we also assume η_k and the signal X_k are independent, it follows that η_k is independent from the estimation $\hat{X}_{k|k-1}$, which is a linear function of X_k and the observations $Y_{0:n-1}$. Thus, η_k is also independent from $G(X_k - \hat{X}_{k|k-1})$, so the covariance matrix Σ_{I_k} of the innovation I_k is the sum of the covariance matrices for the error $X_k - \hat{X}_{k|k-1}$ and η_k:

$$\begin{aligned}
&\Sigma_{I_k} = \Sigma_{X_k - \hat{X}_{k|k-1}} + \Sigma_{\eta_k} \\
&(\Sigma_\eta \text{ is time homogeneous}) \\
&\qquad = \mathbf{E}[(G(X_k - \hat{X}_{k|k-1}) - \mathbf{E}[G(X_k - \hat{X}_{k|k-1})])(G(X_k - \hat{X}_{k|k-1}) - \mathbf{E}[G(X_k - \hat{X}_{k|k-1})])^\top] + \Sigma_\eta \\
&(\mathbf{E}[X_k - \hat{X}_{k|k-1}] = 0) \\
&\qquad = G\mathbf{E}[(X_k - \hat{X}_{k|k-1})(X_k - \hat{X}_{k|k-1})^\top]G^\top + \Sigma_\eta \\
&\qquad = G\Omega_{k|k-1}G^\top + \Sigma_\eta.
\end{aligned}$$

Furthermore, since the span of the observations $Y_0, \dots, Y_k$ is equal to the span of the observations $Y_0, \dots, Y_{k-1}$ and the innovation I_k,

$$\hat{X}_{k+1|k} = E_k(X_{k+1}) = E_{k-1}(X_{k+1}) + \mathbf{E}[X_{k+1}|I_k].$$

Define $\tilde{X}_{k+1} = \mathbf{E}[X_{k+1}|I_k]$. Since $\mathbf{E}[X_{k+1}|I_k]$ is a linear function of I_k, $\tilde{X}_{k+1} = AI_k$ for some matrix A. Recall that $\mathbf{E}[X_{k+1}|I_k]$ can be interpreted geometrically as the orthogonal projection of X_{k+1} onto the span of I_k, so $X_{k+1} - \tilde{X}_{k+1}$ is orthogonal to I_k. Hence,

$$\mathbf{E}[X_{k+1} - \tilde{X}_{k+1}]I_k^\top = \mathbf{E}[\langle (X_{k+1} - \tilde{X}_{k+1}),\ I_k \rangle] = 0.$$

Furthermore, $X_{k+1} - \tilde{X}_{k+1} = X_{k+1} - AI_k$. We wish to solve for A, so we have

$$\begin{aligned}
& & 0 &= \mathbf{E}[X_{k+1} - AI_k]I_k^\top = \mathbf{E}[X_{k+1}I_k^\top] - \mathbf{E}[AI_kI_k^\top] \\
&\text{(linearity)} & &\implies \mathbf{E}[X_{k+1}I_k^\top] = A\mathbf{E}[I_kI_k^\top].
\end{aligned}$$

Note that since the innovation is mean zero,

$$\Sigma_{I_k} = \mathbf{E}[(I_k - \mathbf{E}[I_k])(I_k - \mathbf{E}[I_k])^\top] = \mathbf{E}[I_kI_k^\top] \implies A = \mathbf{E}[X_{k+1}I_k^\top]\Sigma_{I_k}^{-1},$$

where Σ_{I_k} is invertible by the nondegeneracy assumption. Hence,

$$\begin{aligned}
& & \tilde{X}_{k+1} &= \mathbf{E}[X_{k+1}I_k^\top]\Sigma_{I_k}^{-1} \\
& & \implies \hat{X}_{k+1|k} &= E_{k-1}(FX + \xi_{k+1}) + \mathbf{E}[X_{k+1}I_k^\top]\Sigma_{I_k}^{-1}I_k \\
&\text{(linearity)} & &= FE_{k-1}(X_k) + E_{k-1}(\xi_{k+1}) + \mathbf{E}[X_{k+1}I_k^\top]\Sigma_{I_k}I_k \\
&(\xi_{k+1} \text{ is independent from } Y_{1:n-1}) & &= F\hat{X}_{k|k-1} + \mathbf{E}[X_{k+1}I_k^\top]\Sigma_{I_k}^{-1}I_k.
\end{aligned}$$

Furthermore,

$$\mathbf{E}[X_{k+1}I_k^\top] = \mathbf{E}[(FX_k + \xi_{k+1})((X_k - \hat{X}_{k|k-1})^\top G^\top + \xi_k^\top)].$$

Since ξ_{k+1} is independent from X_k, $\hat{X}_{k|k-1}$, and η_k, the expectation of $\xi_{k+1}(X_k - \hat{X}_{k|k-1})$ and $\xi_{k+1}\eta_k^\top$ vanishes. Additionally, note that $\mathbf{E}[\hat{X}_{k|k-1}(X_k - \hat{X}_{k|k-1})^\top] = 0$. Together, these facts give us

$$\begin{aligned}\mathbf{E}[(FX_k + \xi_{k+1})((X_k - \hat{X}_{k|k-1})^\top G^\top + \xi_k^\top)] &= \mathbf{E}[FX_k(X_k - \hat{X}_{k|k-1})G^\top] \\ &= F\mathbf{E}[(X_k - \hat{X}_{k|k-1})(X_k - \hat{X}_{k|k-1})^\top]G^\top \\ &= F\Omega_{k|k-1}G^\top.\end{aligned}$$

Therefore,

$$\hat{X}_{k+1|k} = F\hat{X}_{k|k-1} + F\Omega_{k|k-1}G^\top\Sigma_{I_k}^{-1}I_k = F\hat{X}_{k|k-1} + \Theta_k\Delta_k^{-1}I_k = F\hat{X}_{k|k-1} + \Theta_k\Delta_k^{-1}(Y_k - G\hat{X}_{k|k-1}),$$

as desired. □

Remark. The matrix $\Theta_k\Delta_k^{-1}$ is often called the *Kalman gain matrix*, since it scales the contribution of the innovation I_k to the estimate $\hat{X}_{k+1|k}$. We will see the power of this matrix later when we explore an application of Kalman filtering. Additionally, note that Δ_k is the covariance matrix of I_k.

Since the Kalman gain matrix contains the quadratic error $\Omega_{k|k-1}$, we cannot compute the one-step prediction $\hat{X}_{k+1|k}$ without finding a corresponding recursive update for $\Omega_{k|k-1}$ as well. The following theorem is just that.

Theorem 5.1.2 (One-Step Error Prediction). *The quadratic error $\Omega_{k+1|k}$ can be updated according to*

$$\Omega_{k+1|k} = F\Omega_{k|k-1}F^\top + \Sigma_\xi + \Theta_k\Delta_k^{-1}\Theta_k^\top$$

for $k > 0$.

Proof. By definition,

$$\begin{aligned}\Omega_{k+1|k} &= \mathbf{E}[(X_{k+1} - \hat{X}_{k+1|k})(X_{k+1} - \hat{X}_{k+1|k})^\top] \\ &= \mathbf{E}[X_{k+1}X_{k+1}^\top] - \mathbf{E}[X_{k+1}\hat{X}_{k+1|k}^\top] - \mathbf{E}[\hat{X}_{k+1|k}X_{k+1}] + \mathbf{E}[\hat{X}_{k+1|k}\hat{X}_{k+1|k}^\top],\end{aligned}$$

by linearity. However, note that

$$\mathbf{E}[X_{k+1}\hat{X}_{k+1|k}^\top] = \mathbf{E}[\hat{X}_{k+1|k}X_{k+1}^\top] = \mathbf{E}[\hat{X}_{k+1|k}\hat{X}_{k+1|k}^\top],$$

so

$$\Omega_{k+1|k} = \mathbf{E}[X_{k+1}X_{k+1}^\top] - \mathbf{E}[\hat{X}_{k+1|k}\hat{X}_{k+1|k}^\top]$$

(ξ_{k+1} and X_k are independent)

$$= \mathbf{E}[(FX_k + \xi_{k+1})(FX_k + \xi_{k+1})^\top] - \mathbf{E}[(F\hat{X}_{k+1|k} + \Theta_k\Delta_n^{-1}I_n)(F\hat{X}_{k+1|k} + \Theta_k\Delta_n^{-1}I_n)^\top]$$

($\hat{X}_{k|k-1}$ is orthogonal to I_k)

$$= F\mathbf{E}[X_kX_k^\top]F^\top + \mathbf{E}[\xi_{k+1}\xi_{k+1}^\top] - F\mathbf{E}[\hat{X}_{k|k-1}\hat{X}_{k|k-1}^\top]F^\top + \Theta_k\Delta_k^{-1}\mathbf{E}[I_kI_k^\top]\Delta_k^{-1}\Theta_k^\top$$

(Σ_ξ is time homogenous, Δ_k is symmetric, and $\Sigma_{I_k} = \Delta_k$)

$$\begin{aligned}&= F(\mathbf{E}[X_kX_k^\top] - \mathbf{E}[\hat{X}_{k|k-1}\hat{X}_{k|k-1}])F^\top - \Sigma_\xi + \Theta_k\Delta_k^{-1}\Theta_k^{-1} \\ &= F\Omega_{k|k-1}F^\top + \Sigma_\xi + \Theta_k\Delta_k^{-1}\Theta^\top,\end{aligned}$$

as desired. □

We now have all the tools we need to derive the Kalman filter, i.e., the recursion for $(\hat{X}_k, \Omega_k)_{k\geq 0}$. Recall from Theorem 3.5.1 that the filtering recursion can be interpreted as a two-step procedure: compute the one-step prediction $\hat{X}_{k|k-1}$ and then perform a correction to obtain the filter $\hat{X}_k$. Fortunately, in finding the recursion for the one-step predictor $\hat{X}_{k|k-1}$ and error $\Omega_{k|k-1}$, we have done most of the heavy lifting in finding the filter $\hat{X}_k$. All that remains is to add a correction term to the result of Theorem 5.1.1, which comes in the form of the error covariance matrix $\Omega_{k|k-1}$.

Theorem 5.1.3 (Linear Gaussian Filtering Recursion)**.** *The filter $\hat{X}_k$, $k \geq 0$, is given by*

$$\hat{X}_k = \hat{X}_{k|k-1} + \Omega_{k|k-1} G^\top \Delta_k^{-1}(Y_k - G\hat{X}_{k|k-1}).$$

Proof. Repeating the same innovation argument used to derive the one-step predictor, we have

$$\begin{aligned}
\hat{X}_k &= E_k(X_k) = E_{k-1}(X_k) + \mathbf{E}[X_k | I_k] \\
&= \hat{X}_{k|k-1} + \mathbf{E}[X_k I_k^\top]\mathbf{E}[I_k I_k^\top]^{-1} I_k \\
&= \hat{X}_{k|k-1} + \mathbf{E}[X_k I_k^\top]\Sigma_{I_k}^{-1} I_k \\
&= \hat{X}_{k|k-1} + \mathbf{E}[X_k(G(X_k - \hat{X}_{k|k-1}) + \xi_k)^\top]\Delta_k^{-1} I_k \\
&= \hat{X}_{k|k-1} + \Omega_{k|k-1} G^\top \Delta_k^{-1} I_k \\
&= \hat{X}_{k|k-1} + \Omega_{k|k-1} G^\top \Delta_k^{-1}(Y_k - G\hat{X}_{k|k-1}),
\end{aligned}$$

as desired. □

Just as the derivation for the filtering recursion follows in the footsteps of the the one-step predictor, the derivation of the filtering covariance matrix Ω_k follows in the spirit of the one-step error prediction (Theorem 5.1.2).

Theorem 5.1.4 (Filtering Error Recursion)**.** *The covariance matrix $\Omega_k = \Omega_{k|k}$, $k > 0$ for the filter $\hat{X}_k$ is given by*

$$\Omega_k = \Omega_{k|k-1} - \Omega_{k|k-1} G^\top \Delta_k^{-1} G \Omega_{k|k-1}^\top.$$

Proof. Following the same argument as before, we have

$$\begin{aligned}
\Omega_k &= \mathbf{E}[(X_k - \hat{X}_k)(X_k - \hat{X}_k)^\top] \\
&= \mathbf{E}[X_k X_k^\top]\mathbf{E}[\hat{X}_k \hat{X}_k^\top] \\
\text{(Theorem 5.1.3)} \qquad &= \mathbf{E}[X_k X_k^\top] - \mathbf{E}[(\hat{X}_{k|k-1} + \Omega_{k|k-1} G^\top \Delta_k^{-1} I_k)(\hat{X}_{k|k-1} + \Omega_{k|k-1} G^\top \Delta_k^{-1} I_k)^\top] \\
&= \mathbf{E}[X_k X_k^\top - \hat{X}_{k|k-1}\hat{X}_{k|k-1}^\top] - \Omega_{k|k-1} G^\top \Delta_k^{-1} \mathbf{E}[I_k I_k^\top] \Delta_k^{-1} G \Omega_{k|k-1}^\top \\
&= \Omega_{k|k-1} - \Omega_{k|k-1} G^\top \Delta_k^{-1} G \Omega_{k|k-1}^\top,
\end{aligned}$$

as desired. □

Now that we have derived the recursion for the Kalman filter, we can explore a beautiful and valuable application: generalizing linear models as state space models.

5.2. **Kalman Filtering in Multiple Linear Regression.** To say that linear models are ubiquitous in statistical literature would be a vast understatement. The widespread use of linear models is easily explained by their simpicity to implement and incredible power. In this section we consider a standard linear model

$$y_k = \boldsymbol{Z}_k \boldsymbol{\beta} + \epsilon_k,$$

where $\epsilon_k, k \geq 0$ is *white noise* in the same sense as before, $\boldsymbol{Z}_k = (z_{1,k}, \ldots, z_{d,k})$ is a $d = p+1$-dimensional vector of *explanatory variables* (the expected causes, e.g., smoking, eating grapefruit, being under the age of 20, etc.), and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_d)^\top$ is a d-dimensional time homogeneous vector of unknown parameters (variable features of a family of functions, e.g., slope or y-intercept). The goal of linear models is to perform *regression*, i.e., compute the estimate $\hat{\boldsymbol{\beta}}$ of the vector of parameters $\boldsymbol{\beta}$ that minimizes a given loss function $H(y, \boldsymbol{Z\beta})$. We wish to perform *least squares* regression, in which case our loss function is

$$\mathbf{E}[(y - \boldsymbol{Z\beta})^2],$$

as the name implies. This is precisely the estimation problem that we dealt with in deriving results for estimation in the linear Gaussian setting, so it seems reasonable to attempt to recast this linear model as a state space model and apply the Kalman filter. The key to accomplishing this task is to set the signal $X_k = \boldsymbol{\beta}$ for $k \geq 0$, which gives us the remarkably simple signal state equation $X_k = X_{k-1},\ k > 0$. Hence, the state matrix F_k is time homogeneous and given by $F = I_{d\times d}$ (the $d \times d$ identity matrix) and the signal noise is zero. To complete our rewriting of the linear model, we set the observation matrix $G_k = \boldsymbol{Z}_k$ and the observation noise $\eta_k = \epsilon_k$ for $k \geq 0$, in agreement with (5.1). To summarize, our state space is given by

$$X_k = X_{k-1} \qquad\qquad Y_k = \boldsymbol{Z}_k X_k + \epsilon_k. \tag{5.3}$$

The Kalman gain matrix $K_k = \Theta_k \Delta_k^{-1}$ is now given by

$$K_k = (F\Omega_{k|k-1}G_k^\top)(G_k\Omega_{k|k-1}G_k^\top + \Sigma_\eta)^{\bar{1}} = (\Omega_{k|k-1}\boldsymbol{Z}_k^\top)(\boldsymbol{Z}_k\Omega_{k|k-1}\boldsymbol{Z}_k^\top)^{-1}.$$

We further define $\hat{\boldsymbol{\beta}}_k = \hat{X}_{k|k}$ as the least squares estimate of $\boldsymbol{\beta}_k = \boldsymbol{\beta}$ given the observations $(z_1, y_1), \ldots, (z_k, y_k)$. Applying the Kalman filter (Theorem 5.1.3), we obtain a recursive expression for computing the least squares estimate $\hat{\boldsymbol{\beta}}_{k+1}, k \geq 0$ from the estimate $\hat{\boldsymbol{\beta}}_k$ and the new observation (z_{k+1}, y_{k+1}):

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{k+1} &= \hat{X}_{k+1|k} + \Omega_{k+1|k}G_{k+1}^\top \Delta_{k+1}^{-1}(Y_{k+1} - G_{k+1}\hat{X}_{k+1|k}) \\
&= \hat{\boldsymbol{\beta}}_k + \Omega_{k+1|k}\boldsymbol{Z}_{k+1}^\top(\boldsymbol{Z}_{k+1}\Omega_{k+1|k}\boldsymbol{Z}_{k+1}^\top)^{-1}(Y_{k+1} - \boldsymbol{Z}_{k+1}\hat{\boldsymbol{\beta}}_k) \\
&= \hat{\boldsymbol{\beta}}_k + K_{k+1}(Y_{k+1} - \boldsymbol{Z}_{k+1}\hat{\boldsymbol{\beta}}_k). \qquad (5.4)
\end{aligned}$$

Note that $\hat{\boldsymbol{\beta}}_k = \hat{X}_{k+1|k}$ in the second equality since $\boldsymbol{\beta}$ is time homogeneous. Furthermore, since the observations matrix G now have a time dependence, we have used G_{k+1} in the first line–revisiting Theorem 5.1.3, it is clear that the result still holds. The corresponding recursion for the covariance of the estimate is given by

$$\begin{aligned}
\Omega_{k+1} &= \Omega_{k|k-1} - \Omega_{k|k-1}G_{k+1}^\top \Delta_{k+1}^{-1} G_{k+1}\Omega_{k|k-1}^\top \\
&= \Omega_{k|k-1} - \Omega_{k|k-1}\boldsymbol{Z}_{k+1}^\top(\boldsymbol{Z}_{k+1}\Omega_{k+1|k}\boldsymbol{Z}_{k+1}^\top)^{-1}\boldsymbol{Z}_{k+1}\Omega_{k|k-1}^\top \\
&= (I_{d\times d} - K_{k+1}\boldsymbol{Z}_{k+1})\Omega_{k|k-1}. \qquad (5.5)
\end{aligned}$$

Acknowledgements

Two months before I wrote this paper, I did not so much as know what a stochastic process was; with the help of my mentor Mark Cerenzia, I was able to study the theory of Markov chains and hidden Markov models at a level far beyond what I could have hoped for. I am grateful to Mark for his continuous support and guidance and to Professor Peter May for organizing the REU program which provided me with an invaluable opportunity to continue my math research journey.

I would also like to thank Robert Tunney, with whom I began this journey in my high school years, Professor Beniada Shabani for reigniting my passion for math this year, and lastly Warren Fernandes for encouraging me to always strive for greater challenges.

References

[1] Oliver Cappé, Eric Moulines, and Tobias Rydén. *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer, New York, 2005.

[2] René Carmona. *Statistical Analysis of Financial Data in R*. Springer Texts in Statistics. Springer, New York, 2013.

[3] Charles J. Geyer. Lecture Notes for Stat 8112 - Markov Chains. https://www.stat.umn.edu/geyer/8112/notes/markov.pdf, April 2012.

[4] JoramSoch. The Book of Statistical Proofs, Proof 88: conditional distributions of the multivariate normal distribution. https://statproofbook.github.io/P/mvn-cond, March 2020.

[5] Zakhar Kabluchko. Lecture Notes for Stochastic Processes (Stochastik II). https://www.uni-ulm.de/fileadmin/website_uni_ulm/mawi.inst.110/lehre/ws13/Stochastik_II/Skript_Stochastik_II.pdf, April 2014.

[6] Steve Lalley. Background Reading for Statistics 305 - Brownian Motion and Stochastic Calculus. https://galton.uchicago.edu/~lalley/Courses/385/ConditionalExpectation.pdf, September 2016.

[7] Cosma Shalizi. Lecture Notes for Stat 36-754 - Stochastic Processes (Advanced Probability II) Chapter 2. https://www.stat.cmu.edu/~cshalizi/754/notes/lecture-02.pdf, January 2007.

[8] Ramon van Handel. Lecture Notes for ORF 557 - Hidden Markov Models. https://web.math.princeton.edu/~rvan/orf557/hmm080728.pdf, July 2008.

[9] Daniel Yew Mao Lim. Lecture Notes for API-208 - Program Evaluation: Estimating Program Effectiveness with Empirical Analysis. https://scholar.harvard.edu/files/danielyewmaolim/files/api-208section1.pdf, February 2013.