19 Filtering and State Estimation

Our study of estimating parameters from observations has presumed that there are unchanging parameters to be estimated. For many (if not most) applications this is not so: not only are the parameters varying, but finding their variation in time may be the goal of the data analysis. This chapter and the next bring time back into the picture. Here we will look at the problem of estimating a time-dependent set of parameters that describe the state of a system, given measurements of observable quantities along with some kind of model for the relationship between the observations and the underlying state. For example, in order to navigate, an airplane must know where it is. Many relevant signals arrive at the airplane, such as radar echoes, GPS messages, and gyroscopic measurements. The first task is to reduce these raw signals to position estimates, and then these estimates must be combined along with any relevant past information to provide the best overall estimate of the plane's position. Closely related tasks are *smoothing*, *noise reduction*, and *signal separation*, using the collected data set to provide the best estimate of previous states (given new measurements, where do we think the airplane was?), and *prediction*, using the data to forecast a future state (where is the airplane going?). These tasks are often described as filtering problems, even though they really are general algorithm questions, because they evolved from early implementations in analog filters.

We will start with the simple example of *matched filters* to detect a known signal, extend that to *Wiener filters* to separate a signal from noise, and then turn to the much more general, useful, and important *Kalman filters*. Much of estimation theory is based on linear techniques; since the world is not always obligingly linear, we will next look at how nonlinearity makes estimation more difficult, and simpler. The chapter closes with the use of *Hidden Markov Models* to help find models as well as states.

19.1 MATCHED FILTERS

Consider a signal $x(t)$ passed through a linear filter with impulse response $f(t)$ (go back to Chapter 3 if you need a review of linear systems theory). The frequency domain response of the output $Y(\omega)$ will be the product of the Fourier transforms of the input and the filter

$$Y(\omega) = X(\omega)F(\omega) \quad , \tag{19.1}$$

and the time domain response will be the convolution

$$y(t) = x(t) * f(t) = \int_0^T x(t-u)f(u)\,du \quad , \tag{19.2}$$

where the limits of the integral are the interval during which the signal has been applied to the filter. The magnitude of the output can be bounded by *Schwarz's inequality*:

$$\begin{aligned} y^2(t) &= \left| \int_0^T x(t-u)f(u)\,du \right|^2 \\ &\le \int_0^T |x(t-u)|^2\,du \int_0^T |f(u)|^2\,du \quad . \end{aligned} \tag{19.3}$$

By inspection, this bound will be saturated (reach its maximum value) if

$$f(u) = A\,x^*(t-u) \tag{19.4}$$

for any constant A. The filter will produce the maximum output for a given input signal if the impulse response of the filter is proportional to the complex conjugate of the signal reversed in time. This is called a *matched filter*, and is used routinely to detect and time known signals. For example, to measure the arrival time of radar echoes, the output from a filter matched to the transmitted pulses goes to a comparator, and the time when the output exceeds a preset threshold is used to determine when a pulse has arrived.

19.2 WIENER FILTERS

Next, consider a time-invariant filter with impulse response $f(t)$ that receives an input $x(t) + \eta(t)$ and produces an output $y(t)$, with $x(t)$ a desired signal and $\eta(t)$ noise added to the signal (such as from the front end of an amplifier). In the time domain the output is the convolution

$$y(t) = \int_{-\infty}^{\infty} f(u)[x(t-u) + \eta(t-u)]\,du \quad , \tag{19.5}$$

for now assuming that the signals are defined for all time. How should the filter be designed to make $y(t)$ as close as possible to $x(t)$? One way to do this is by minimizing the mean square error between them (in Chapter 12 we saw that this implicitly assumes Gaussian statistics, but is an assumption that is commonly and relatively reliably used more broadly). This problem was solved for a linear filter by Norbert Wiener at MIT's Radiation Laboratory in the 1940s, therefore the solution is called a *Wiener filter*.

The expected value of the error at time t is

$$\langle E^2 \rangle = \langle [x(t+\alpha) - y(t)]^2 \rangle \quad , \tag{19.6}$$

where the average is over an ensemble of realizations of the noise process. An offset α has been added to cover the three cases of:

- $\alpha < 0$: *smoothing* the past
- $\alpha = 0$: *filtering* the present
- $\alpha > 0$: *predicting* the future

Substituting in equation (19.5),

$$\langle E^2\rangle = \langle x^2(t+\alpha)\rangle - 2\int_{-\infty}^{\infty} f(u)\underbrace{\langle x(t+\alpha)[x(t-u)+\eta(t-u)]\rangle}_{\equiv C_{x,x+\eta}(\alpha+u)}\,du \qquad (19.7)$$

$$+\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u)f(v)\underbrace{\langle [x(t-u)+\eta(t-u)][x(t-v)+\eta(t-v)]\rangle}_{\equiv C_{x+\eta,x+\eta}(u-v)}\,du\,dv \quad .$$

We must find the $f(t)$ that minimizes the sum of these integrals over the correlation functions. Since the first term does not depend on the filter function $f(t)$ it can't contribute to the minimization and we will drop it. Because of the double integral we can't use the Euler–Lagrange equation derived in Chapter 5, but we can use a similar argument. Assume that $f(t)$ is the optimal filter that we are looking for, and let $g(t)$ be any arbitrary filter added to it, giving a new filter $f(t)+\epsilon g(t)$. In terms of this the new error is

$$\begin{aligned}\langle E^2\rangle &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[f(u)+\epsilon g(u)][f(v)+\epsilon g(v)]C_{x+\eta,x+\eta}(u-v)\,du\,dv\\ &\quad -2\int_{-\infty}^{\infty}[f(u)+\epsilon g(u)]C_{x,x+\eta}(\alpha+u)\,du \quad . \end{aligned}\qquad (19.8)$$

We can now differentiate with respect to ϵ and look for the minimum at $\epsilon = 0$:

$$\begin{aligned}\left.\frac{\partial\langle E^2\rangle}{\partial\epsilon}\right|_{\epsilon=0} &= 0\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(u)f(v)C_{x+\eta,x+\eta}(u-v)\,du\,dv\\ &\quad +\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u)g(v)C_{x+\eta,x+\eta}(u-v)\,du\,dv\\ &\quad +2\epsilon\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(u)g(v)C_{x+\eta,x+\eta}(u-v)\,du\,dv\\ &\quad -2\int_{-\infty}^{\infty} g(u)C_{x,x+\eta}(\alpha+u)\,du \quad . \end{aligned}\qquad (19.9)$$

The first two terms are the same (interchanging dummy integration variables and using the symmetry of the correlation functions), and the third one vanishes at $\epsilon = 0$, so we're left with

$$\int_{-\infty}^{\infty} g(\tau)\left[-C_{x,x+\eta}(\alpha+\tau)+\int_{-\infty}^{\infty} f(u)C_{x+\eta,x+\eta}(u-\tau)\right]\,du\,d\tau = 0 \qquad (19.10)$$

Since $g(\tau)$ is arbitrary, the only way this can be equal to zero for all choices of g is if the term in brackets vanishes

$$\int_{-\infty}^{\infty} f(u)C_{x+\eta,x+\eta}(u-\tau)\,du = C_{x,x+\eta}(\alpha+\tau) \quad . \qquad (19.11)$$

This is now a simpler integral equation to be solved for f, but there is a crucial subtlety in equation (19.11). If we solve it for all τ, positive and negative, then it will require that the filter function f be defined for both positive and negative times. This is a *noncausal* filter. The only way that the filter can have access to the signal at all times (other than

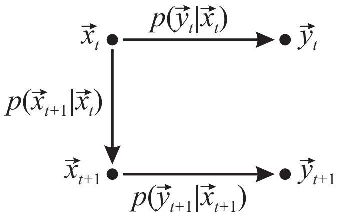

Figure 19.1. Update of an internal state and an external observable.

by being psychic) is if it is applied after the full time record has been recorded, which is fine for off-line applications. If we don't mind a noncausal filter, the convolution in equation (19.11) is easily solved by taking (two-sided) Laplace transforms

$$F(s)C_{x+\eta,x+\eta}(s) = C_{x,x+\eta}(s)e^{\alpha s} \tag{19.12}$$

and so

$$F(s) = \frac{C_{x,x+\eta}(s)e^{\alpha s}}{C_{x+\eta,x+\eta}(s)} \quad . \tag{19.13}$$

This has a simple interpretation: the Wiener filter rolls the response off when the signal is much smaller than the noise, and sets the gain to unity if the signal is much larger than the noise. Prediction or smoothing is done simply by the complex phase shift of a linear system.

If a causal filter is needed so that the Wiener filter can be used in real-time then $f(\tau)$ must vanish for negative τ, and equation (19.11) must be solved only for $\tau \geq 0$. In this case it is called the *Wiener–Hopf* equation, and is much more difficult to solve, although there are many special techniques available for doing so because it is so important [Brown & Hwang, 1997]. Beyond Wiener filters, signal separation for nonlinear systems requires the more general time series techniques to be introduced in Chapter 20.

19.3 KALMAN FILTERS

Wiener filters are impressively optimal, but practically not very useful. It is important to remember that everything is optimal with respect to something. In the case of Wiener filters we found the "best" linear time-invariant filter, but by design it is therefore linear time-invariant. The result is an almost trivial kind of signal separation, simply cutting off the response where the signal is small compared to the noise. Furthermore, it does not easily generalize to more complex problems with multiple degrees of freedom.

Now consider the general system shown in Figure 19.1. There is an internal state $\vec{x}$, for example, the position, velocity, and acceleration of an airplane as well as the orientations of the control surfaces. Its state is updated in discrete time according to a distribution function $p(\vec{x}_{t+1}|\vec{x}_t)$, which includes both the deterministic and random influences. For the airplane, the deterministic part is the aerodynamics, and the random part includes factors such as turbulence and control errors. The internal state is not directly accessible, but rather must be inferred from measurements of observables $\vec{y}$ (such as the airplane's pitot tube, GPS receiver, and radar returns), which are related to $\vec{x}$ by a relation $p(\vec{y}|\vec{x})$ that can include a random component due to errors in the measurement process. How should the measurements be combined to estimate the system's state? Further, is it possible to iteratively update the state estimate given new measurements without recalculating it from scratch? Kalman filters provide a general solution to this important problem.

To start, let's assume that there are just two random variables, x and y, that have a joint probability distribution $p(x,y)$. Given a measurement of y, what function $\hat{x}(y)$ should we use to estimate x? Once again, we will do this by picking the estimate that minimizes the mean square error over the distribution. This means that we want to minimize

$$\begin{aligned}
\langle[x-\hat{x}(y)]^2\rangle &= \int\int [x-\hat{x}(y)]^2\ p(x,y)\ dx\ dy \\
&= \int\int [x^2 - 2x\hat{x}(y) + \hat{x}^2(y)]\ \underbrace{p(x,y)}_{p(x|y)p(y)}\ dx\ dy \\
&= \int x^2 \underbrace{\int p(x,y)\ dy}_{p(x)}\ dx - 2\int \hat{x}(y) \underbrace{\int x\ p(x|y)dx}_{\equiv\ \langle x|y\rangle}\ p(y)\ dy \\
&\quad + \int \hat{x}^2(y) \underbrace{\int p(x,y)\ dx}_{p(y)}\ dy \\
&= \int x^2\ p(x)\ dx + \int [\hat{x}^2(y) - 2\hat{x}(y)\langle x|y\rangle]\ p(y)\ dy \\
&= \int x^2\ p(x)\ dx \\
&\quad + \int [\hat{x}^2(y) - 2\hat{x}(y)\langle x|y\rangle + \langle x|y\rangle^2 - \langle x|y\rangle^2]\ p(y)\ dy \\
&= \int x^2\ p(x)\ dx \\
&\quad - \int \langle x|y\rangle^2\ p(y)\ dy + \int [\hat{x}(y) - \langle x|y\rangle]^2\ p(y)\ dy\ .
\end{aligned} \tag{19.14}$$

All integrals are over the limits of the distribution, and in the last line we completed the square. The first two terms don't depend on the unknown estimator $\hat{x}(y)$, and so are irrelevant to the minimization. The last term is the product of two non-negative functions, which will be minimized if the left hand one vanishes:

$$\hat{x}(y) = \langle x|y\rangle = \int x\ p(x|y)\ dx \quad . \tag{19.15}$$

In retrospect, this is perhaps an obvious result: the minimum mean square estimator simply is the expected value. This result easily generalizes to multi-dimensional distributions.

Now let's assume that the system's update rule is linear, with additive noise $\vec{\eta}$

$$\vec{x}_t = \mathbf{A}_{t-1}\cdot\vec{x}_{t-1} + \vec{\eta}_t \tag{19.16}$$

(we will later relax the assumption of linear updates), and assume a linear relationship

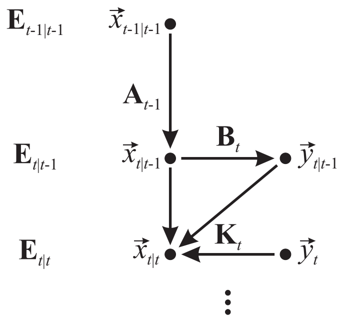

Figure 19.2. Steps in Kalman filtering.

between the state and the observable with additive noise $\vec{\epsilon}$

$$\vec{y}_t = \mathbf{B}_t \cdot \vec{x}_t + \vec{\epsilon}_t \quad . \tag{19.17}$$

The noise sources are assumed to be uncorrelated in time, but can have correlations among the components, as measured by the noise covariance matrices $\mathbf{N}^x$ and $\mathbf{N}^y$

$$\mathbf{N}^x = \langle \vec{\eta}\vec{\eta}^T \rangle \qquad \langle \eta_i(t)\eta_j(t') \rangle = N^x_{ij}\delta_{tt'}$$

$$\mathbf{N}^y = \langle \vec{\epsilon}\vec{\epsilon}^T \rangle \qquad \langle \epsilon_i(t)\epsilon_j(t') \rangle = N^y_{ij}\delta_{tt'} \tag{19.18}$$

(where as usual $\vec{\epsilon}^T$ is the transpose of $\vec{\epsilon}$). The two noise sources are taken to be uncorrelated with each other

$$\langle \vec{\eta}\vec{\epsilon}^T \rangle = \vec{0} \tag{19.19}$$

and to have zero mean

$$\langle \vec{\eta} \rangle = \langle \vec{\epsilon} \rangle = \vec{0} \quad . \tag{19.20}$$

The elements of Kalman filtering are shown in Figure 19.2. $\vec{x}_t$ is the true (but inaccessible) state of the system at time t, $\vec{y}_t$ the observable, and $\mathbf{E}_t$ is the covariance matrix of the error in the estimate of $\vec{x}$. The notation $\vec{x}_{n|m}$ represents the best estimate for $\vec{x}_n$ given the record of measurements up to time m

$$\vec{x}_{n|m} = \langle \vec{x}_n | \vec{x}_m, \vec{x}_{m-1}, \ldots \rangle = \int \vec{x}_n \; p(\vec{x}_n | \vec{x}_m, \vec{x}_{m-1}, \ldots) \, d\vec{x}_n \quad . \tag{19.21}$$

The first step in Kalman filtering is to use the best estimate of the previous system state, $\vec{x}_{t-1|t-1}$, to predict the new state $\vec{x}_{t|t-1}$. This is then used to predict the observable $\vec{y}_{t|t-1}$. Then, when the true new observable $\vec{y}_t$ is measured, it and the estimate $\vec{y}_{t|t-1}$ are combined to estimate the new internal state $\vec{x}_{t|t}$. There are two very important and perhaps nonobvious elements of this figure. First, the state estimate updates are done on just the previous state, without needing the full record, but (given the assumptions of the model) this provides just as good an estimate. Second, this estimate will be much better than if the new observable alone was used to estimate the internal state. Kalman filtering is an example of *recursive estimation*: to determine the present estimate it is necessary to know the previous one, which in turn depends on the one before that, and so forth back to the initial conditions.

To do the first prediction step, recall that if two variables a and b with probabilities $p_a(a)$ and $p_b(b)$ are added, then the distribution for their sum $c = a + b$ is the convolution

$$p(c) = \int p_b(b) p_a(c - b) \; db \quad . \tag{19.22}$$

Since $\vec{x}_t$ depends only on the previous value $\vec{x}_{t-1}$ plus the noise term, the expected value will depend only on the previous expected value $\vec{x}_{t-1|t-1}$:

$$\vec{x}_{t|t-1} = \int \vec{x}_t \; p(\vec{x}_t|\vec{x}_{t-1}) \; d\vec{x}_t \quad . \tag{19.23}$$

The conditional distribution $p(\vec{x}_t|\vec{x}_{t-1})$ consists of the deterministic distribution $\delta(\vec{x}_t - \mathbf{A}_t \cdot \vec{x}_{t-1})$ convolved by the (zero mean) noise distribution p_η, so

$$\vec{x}_{t|t-1} = \int \vec{x}_t \; p_\eta(\vec{x}_t - \mathbf{A}_{t-1} \cdot \vec{x}_{t-1}) \; d\vec{x}_t \quad , \tag{19.24}$$

and since the noise distribution is zero mean

$$\vec{x}_{t|t-1} = \mathbf{A}_{t-1} \cdot \vec{x}_{t-1|t-1} \quad . \tag{19.25}$$

Similarly,

$$\vec{y}_{t|t-1} = \mathbf{B}_t \cdot \vec{x}_{t|t-1} \quad . \tag{19.26}$$

This gives us the estimates for the new internal state and observable. To update the internal state estimate, these can be linearly combined with the new observation $\vec{y}_t$

$$\vec{x}_{t|t} = \vec{x}_{t|t-1} + \mathbf{K}_t \cdot (\vec{y}_t - \vec{y}_{t|t-1}) \quad . \tag{19.27}$$

The matrix $\mathbf{K}_t$ is called the *Kalman gain matrix*; we will derive the optimal form for it. Given this estimate we can define the error covariance matrix in terms of the (inaccessible) true state $\vec{x}_t$ by

$$\mathbf{E}_{t|t} = \langle (\vec{x}_t - \vec{x}_{t|t})(\vec{x}_t - \vec{x}_{t|t})^T \rangle \quad . \tag{19.28}$$

The difference between the true state and the estimate is

$$\vec{x}_t - \vec{x}_{t|t} = \vec{x}_t - \vec{x}_{t|t-1} - \mathbf{K}_t \cdot (\vec{y}_t - \vec{y}_{t|t-1}) \quad , \tag{19.29}$$

and the difference between the predicted and the true observation is

$$y_t - y_{t|t-1} = \mathbf{B}_t \cdot \vec{x}_t + \vec{\epsilon}_t - \mathbf{B}_t \cdot \vec{x}_{t|t-1} \quad . \tag{19.30}$$

Combining these,

$$\begin{aligned} \vec{x}_t - \vec{x}_{t|t} &= \vec{x}_t - \vec{x}_{t|t-1} - \mathbf{K}_t \mathbf{B}_t \cdot (\vec{x}_t - \vec{x}_{t|t-1}) - \mathbf{K}_t \cdot \vec{\epsilon}_t \\ &= (\mathbf{I} - \mathbf{K}_t \mathbf{B}_t) \cdot (\vec{x}_t - \vec{x}_{t|t-1}) - \mathbf{K}_t \cdot \vec{\epsilon}_t \quad . \end{aligned} \tag{19.31}$$

Therefore the error matrix is updated by

$$\begin{aligned} \mathbf{E}_{t|t} &= (\mathbf{I} - \mathbf{K}_t \mathbf{B}_t) \; \langle (\vec{x}_t - \vec{x}_{t|t-1})(\vec{x}_t - \vec{x}_{t|t-1})^T \rangle \; (\mathbf{I} - \mathbf{K}_t \mathbf{B}_t)^T \\ &\quad + \mathbf{K}_t \langle \vec{\epsilon}_t \vec{\epsilon}_t^T \rangle \mathbf{K}_t^T \\ &= (\mathbf{I} - \mathbf{K}_t \mathbf{B}_t) \; \mathbf{E}_{t|t-1} \; (\mathbf{I} - \mathbf{K}_t \mathbf{B}_t)^T + \mathbf{K}_t \mathbf{N}_t^y \mathbf{K}_t^T \end{aligned} \tag{19.32}$$

(there are no cross terms because the measurement noise $\vec{\epsilon}_t$ is independent of the state estimation error $\vec{x}_t - \vec{x}_{t|t-1}$). The diagonal terms of the error covariance matrix are the state errors; we want to choose the Kalman gain matrix $\mathbf{K}$ to minimize the sum of the diagonal terms of the matrix, i.e., minimize the trace

$$\mathrm{Tr}(\mathbf{E}_{t|t}) = \left\langle |\vec{x}_t - \vec{x}_{t|t}|^2 \right\rangle \quad . \tag{19.33}$$

To do this minimization, we will use two matrix identities

$$\frac{d\ \mathrm{Tr}(\mathbf{AB})}{d\mathbf{A}} = \mathbf{B}^T \quad \text{(if } \mathbf{AB} \text{ is square)} \tag{19.34}$$

and

$$\frac{d\ \mathrm{Tr}(\mathbf{ACA}^T)}{d\mathbf{A}} = 2\mathbf{AC} \quad \text{(if } \mathbf{C} \text{ is symmetric)} \quad , \tag{19.35}$$

where

$$\left(\frac{df}{d\mathbf{A}}\right)_{ij} \equiv \frac{df}{dA_{ij}} \tag{19.36}$$

(these can be proved by writing out the components). Equation (19.32) can be expanded out as

$$\begin{aligned}\mathbf{E}_{t|t} = &\ \mathbf{E}_{t|t-1} - \mathbf{K}_t\mathbf{B}_t\mathbf{E}_{t|t-1} - \mathbf{E}_{t|t-1}\mathbf{B}_t^T\mathbf{K}_t^T \\ &+\mathbf{K}_t(\mathbf{B}_t\mathbf{E}_{t|t-1}\mathbf{B}_t^T + \mathbf{N}_t^y)\mathbf{K}_t^T\end{aligned} \tag{19.37}$$

(recalling that $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$). Applying the two matrix identities to take the derivative of the trace of this equation with respect to $\mathbf{K}$, and using the fact that the trace is unchanged by taking the transpose $\mathrm{Tr}(\mathbf{E}_{t|t-1}\mathbf{B}_t^T\mathbf{K}_t^T) = \mathrm{Tr}([\mathbf{E}_{t|t-1}\mathbf{B}_t^T\mathbf{K}_t^T]^T) = \mathrm{Tr}(\mathbf{K}_t\mathbf{B}_t\mathbf{E}_{t|t-1})$, gives

$$\frac{d\ \mathrm{Tr}(\mathbf{E}_{t|t})}{d\ \mathbf{K}_t} = -2(\mathbf{B}_t\mathbf{P}_{t|t-1})^T + 2\mathbf{K}_t(\mathbf{B}_t\mathbf{E}_{t|t-1}\mathbf{B}_t^T + \mathbf{N}_t^y) = 0 \quad . \tag{19.38}$$

This equation defines the Kalman gain matrix that makes the error extremal; checking the second derivative shows that this is a minimum. Solving for the optimal gain matrix,

$$\mathbf{K}_t = \mathbf{E}_{t|t-1}\mathbf{B}_t^T\left(\mathbf{B}_t\mathbf{E}_{t|t-1}\mathbf{B}_t^T + \mathbf{N}_t^y\right)^{-1} \quad . \tag{19.39}$$

Substituting the gain matrix back into equation (19.37), the third and fourth terms cancel, leaving

$$\mathbf{E}_{t|t} = \mathbf{E}_{t|t-1} - \mathbf{E}_{t|t-1}\mathbf{B}_t^T(\mathbf{B}_t\mathbf{E}_{t|t-1}\mathbf{B}_t^T + \mathbf{N}_t^y)^{-1}\mathbf{B}_t\mathbf{E}_{t|t-1} \tag{19.40}$$

or

$$\mathbf{E}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{B}_t)\mathbf{E}_{t|t-1} \quad . \tag{19.41}$$

This gives the update rule for the error matrix given a new measurement of the observable.

The last piece that we need is the predicted error after the state prediction step $\vec{x}_{t+1|t} = \mathbf{A}_t \cdot \vec{x}_{t|t}$, which will be

$$\begin{aligned}\mathbf{E}_{t+1|t} &= \left\langle (\vec{x}_{t+1} - \vec{x}_{t+1|t})(\vec{x}_{t+1} - \vec{x}_{t+1|t})^T \right\rangle \\ &= \left\langle (\mathbf{A}_t \cdot \vec{x}_t + \vec{\eta}_t - \mathbf{A}_t \cdot \vec{x}_{t|t})(\mathbf{A}_t \cdot \vec{x}_t + \vec{\eta}_t - \mathbf{A}_t \cdot \vec{x}_{t|t})^T \right\rangle \\ &= \left\langle (\mathbf{A}_t \cdot (\vec{x}_t - \vec{x}_{t|t}) + \vec{\eta}_t)(\mathbf{A}_t \cdot (\vec{x}_t - \vec{x}_{t|t}) + \vec{\eta}_t)^T \right\rangle\end{aligned}$$

$$
\begin{aligned}
&= \langle \mathbf{A}_t \cdot (\vec{x}_t - \vec{x}_{t|t})(\vec{x}_t - \vec{x}_{t|t})^T \cdot \mathbf{A}^T \rangle + \langle \vec{\eta}_t \vec{\eta}_t^T \rangle \\
&= \mathbf{A}_t \mathbf{E}_{t|t} \mathbf{A}_t^T + \mathbf{N}_t^x \quad . \qquad (19.42)
\end{aligned}
$$

This completes the derivation of the Kalman filter, the linear estimator with the minimum square error. Recapping, the procedure starts with an initial estimate for the state $\vec{x}_{t|t-1}$ and error $\mathbf{E}_{t|t-1}$, and then the sequence is:

- Estimate the new observable

$$\vec{y}_{t|t-1} = \mathbf{B}_t \cdot \vec{x}_{t|t-1} \quad .$$

- Measure a new value for the observable

$$\vec{y}_t \quad .$$

- Compute the Kalman gain matrix

$$\mathbf{K}_t = \mathbf{E}_{t|t-1} \mathbf{B}_t^T \left(\mathbf{B}_t \mathbf{E}_{t|t-1} \mathbf{B}_t^T + \mathbf{N}_t^y \right)^{-1} \quad .$$

- Estimate the new state

$$\vec{x}_{t|t} = \vec{x}_{t|t-1} + \mathbf{K}_t \cdot (\vec{y}_t - \vec{y}_{t|t-1}) \quad .$$

- Update the error matrix

$$\mathbf{E}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{B}_t)\, \mathbf{E}_{t|t-1} \quad .$$

- Predict the new state

$$\vec{x}_{t+1|t} = \mathbf{A}_t \cdot \vec{x}_{t|t} \quad .$$

- Predict the new error

$$\mathbf{E}_{t+1|t} = \mathbf{A}_t \mathbf{E}_{t|t} \mathbf{A}_t^T + \mathbf{N}_t^x \quad .$$

Remarkably, the iterative application of this recursive algorithm gives the best estimate of $\vec{x}(t)$ from the history of $\vec{y}(t)$ that can be made by a linear estimator; it cannot be improved by analyzing the entire data set off-line [Catlin, 1989].

Stepping back from the details of the derivation, these equations have very natural limits. If $\mathbf{B} \rightarrow 0$ (the observable $\vec{y}$ does not depend on the internal state $\vec{x}$) or $\mathbf{N}^y \rightarrow \infty$ (the observable is dominated by measurement noise) then $\mathbf{K} \rightarrow 0$ and the measurements are not used in the state estimate. Conversely, if $\mathbf{N}^y \rightarrow 0$ and $\mathbf{B} \rightarrow \mathbf{I}$ (there is no noise in the observable, and the transformation from the internal state reduces to the identity matrix) then the update replaces the internal state with the new measurement. Problem 18.1 looks at the suggestive form of these equations for the case of small measurement noise.

19.4 NONLINEARITY AND ENTRAINMENT

The derivation of the Kalman filter has assumed linearity in two places: linear observables and dynamics, and linear updates of the state following a new measurement. The former can be relaxed by local linearization; we'll return to the latter in the next chapter.

The nonlinear governing equations now are

$$\vec{x}_t = \vec{f}(\vec{x}_{t-1}) + \vec{\eta}_t \qquad \vec{y}_t = \vec{g}(\vec{x}_t) + \vec{\epsilon}_t \quad . \tag{19.43}$$

The system governing equation is needed to predict the new state $\vec{x}_t = \vec{f}(\vec{x}_{t-1})$, and to predict the new error

$$\begin{aligned}\mathbf{E}_{t+1|t} &= \langle (\vec{x}_{t+1} - \vec{x}_{t+1|t})(\vec{x}_{t+1} - \vec{x}_{t+1|t})^T \rangle \\ &= \langle [\vec{f}(\vec{x}_t) + \vec{\eta}_t - \vec{f}(\vec{x}_{t|t})][\vec{f}(\vec{x}_t) + \vec{\eta}_t - \vec{f}(x_{t|t})]^T \rangle \quad .\end{aligned} \tag{19.44}$$

If the prediction error is not large (i.e., the noise $\vec{\eta}$ is small), then $\vec{f}$ can be replaced by its local linearization

$$\begin{aligned}\vec{f}(\vec{x}_t) - \vec{f}(\vec{x}_{t|t}) &\approx \left.\frac{\partial \vec{f}}{\partial \vec{x}}\right|_{\vec{x}_{t|t}} \cdot (\vec{x}_t - \vec{x}_{t|t}) \\ &\equiv \mathbf{A}_t \cdot (\vec{x}_t - \vec{x}_{t|t}) \quad .\end{aligned} \tag{19.45}$$

With this revised definition for **A** then equation (19.42) can be used as before. Similarly, the observable equation appears in the derivation of the Kalman gain matrix as

$$\begin{aligned}\vec{y}_t - \vec{y}_{t|t-1} &= \vec{g}(\vec{x}_t) + \vec{\epsilon}_t - \vec{g}(\vec{x}_{t|t-1}) \\ &\approx \left.\frac{\partial \vec{g}}{\partial \vec{x}}\right|_{\vec{x}_{t|t-1}} \cdot (\vec{x}_t - \vec{x}_{t|t-1}) + \vec{\epsilon}_t \\ &\equiv \mathbf{B}_t \cdot (\vec{x}_t - \vec{x}_{t|t-1}) + \vec{\epsilon}_t \quad .\end{aligned} \tag{19.46}$$

Once again, by taking **B** to be the local linearization this is the same as equation (19.30). Redefining the Kalman filter to use local linearizations of nonlinear observables and dynamics in the gain and error calculations gives the *extended Kalman filter* (the nonlinear functions can be retained in the the state and observable predictions). As with most things nonlinear it is no longer possible to prove the same kind of optimality results about an extended Kalman filter, a liability that is more than made up for by its broader applicability.

The magic of Kalman filtering happens in the step

$$\vec{x}_{t|t} = \vec{x}_{t|t-1} + \mathbf{K}_t \cdot (\vec{y}_t - \vec{y}_{t|t-1}) \quad . \tag{19.47}$$

A correction is added to the internal state based on the difference between what you predicted and what you observed, scaled by how much you trust your predictions versus the observations. Officially, to be able to apply this you must know enough about the system to be able to calculate the noise covariances in both the dynamics and the measurements. In practice this is often not the case, particularly since the "noise" represents all aspects of the system not covered by the model. Then the noise terms become adjustable parameters that are selected to give satisfactory performance (Problem 18.2 provides an example of this trade-off).

The success of Kalman filtering even when it is not formally justified hints at the power of equation (19.47). Many nonlinear systems share the property that a small interaction with an independent copy of the system can cause their states to become synchronized. This process is called *entrainment*. For example, let $d\vec{x}/dt = \vec{f}(\vec{x})$, and take $d\vec{x}'/dt = \vec{f}(\vec{x}')$ to obey the same governing equation but have different initial conditions. Then if we couple one of the degrees of the freedom of the two systems with a linear correction that seeks to drive those variables to the same value,

$$\frac{dx_i}{dt} = f_i(x_i) + \epsilon(x_i' - x_i) \quad , \tag{19.48}$$

then for most choices of f, ϵ, and i, $\vec{x}$ will approach $\vec{x}'$ as long as x_i interacts with the other components of $\vec{x}$ and there is dissipation to damp out errors. Because dissipation reduces the dimension of the subspace of a system's configuration space that it actually uses [Temam, 1988], it's needed to separate the tugs from the coupling between the systems from the internal evolution of the system. ϵ is a small parameter that controls the trade-off between responding quickly and ignoring noise.

Entrainment requires that the largest Lyapunov exponent associated with the coupling between the systems is negative [Pecora *et al.*, 1997]; this does not even require the systems to be identical [Parlitz *et al.*, 1997]. A formerly-familiar example is provided by mechanical clocks on the wall of a clock shop; the vibrations coupled through the wall could entrain the clock mechanisms so that they would tick in synchrony.

Entrainment can be used to design systems whose simple dynamics replaces complicated algorithms for the job of state estimation. An example is *spread spectrum acquisition*, a very important task in engineering practice. A transmitter that seeks to make optimal use of a communications channel uses a linear feedback shift register (LFSR, Chapter 6) to generate ideal pseudo-random noise as a modulation source. To decode the message, the receiver must maintain a copy of the transmitter's shift register that remains faithful even if there is noise in the transmission or the two system's clocks drift apart. This is conventionally done by a coding search for the best setting of the receiver's shift register [Simon *et al.*, 1994].

An LFSR uses a binary recursion

$$x_n = \sum_{i=1}^{N} a_i x_{n-i} \pmod 2 \quad , \tag{19.49}$$

with the a_i's chosen to make the z-transform irreducible. It's possible to add small perturbations to this discrete function if the LFSR is replaced by an *analog feedback shift register* (*AFSR*) [Gershenfeld & Grinstein, 1995],

$$x_n = \frac{1}{2}\left[1 - \cos\left(\pi \sum_{i=1}^{N} a_i x_{n-i}\right)\right] \quad . \tag{19.50}$$

This analog function matches the value of the LFSR for binary arguments. Because the magnitude of the slope of the map is less than 1 at the digital values, these are stable fixed points [Guckenheimer & Holmes, 1983] that attract an arbitrary initial condition of the register onto the LFSR sequence. If we now add to this a small correction

$$x_n = \frac{1}{2}\left[1 - \cos\left(\pi \sum_{i=1}^{N} a_i x_{n-i}\right)\right] + \epsilon(x_n' - x_n) \quad , \tag{19.51}$$

where x_n' is a signal received from another shift register, then the two systems can entrain. This is shown in Figure 19.3. If ϵ is large the receiver locks quickly but it will also try to follow any modulation and noise in the signal; if ϵ is small it will take longer to lock but will result in a cleaner estimate of the transmitter's state.

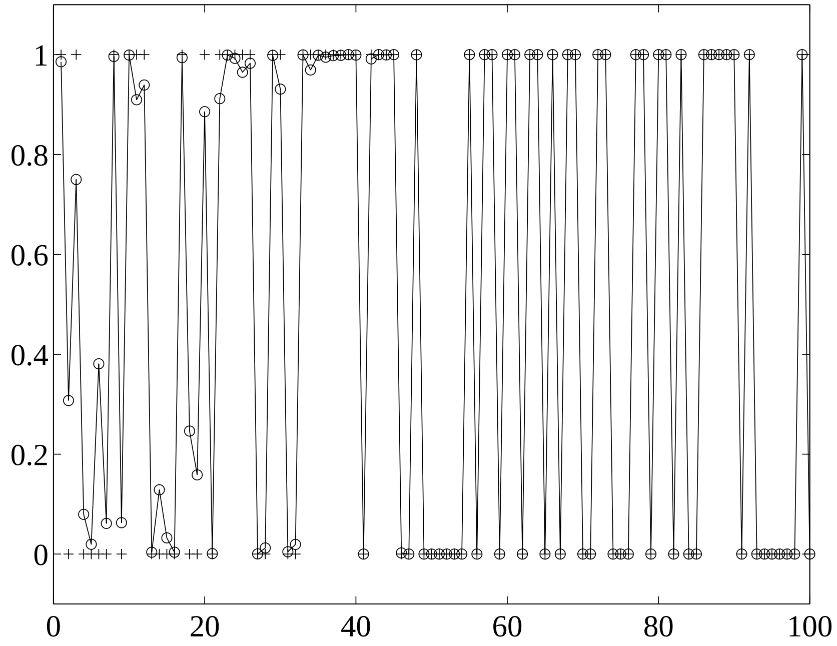

Figure 19.3. Entrainment of an analog feedback shift register (○) with a linear feedback shift register (+).

19.5 HIDDEN MARKOV MODELS

The job of a Kalman filter is to provide an estimate of the internal state given a history of measurements of an external observable. It presumes, however, that you already know how to calculate the transition probabilities, and further that you're not interested in the probability distribution for the internal state. A *Hidden Markov Model* (*HMM*) addresses these limitations.

For example, consider coin flipping done by a corrupt referee who has two coins, one biased and the other fair, with the biased coin occasionally switched in surreptitiously. The observable is whether the outcome is heads or tails; the hidden variable is which coin is being used. Figure 19.4 shows this situation. There are transition probabilities between the hidden states A and B, and emission probabilities associated with the observables 0 and 1. Given this architecture, and a set of measurements of the observable, our task is to deduce both the fixed transition probabilities and the changing probabilities to see the internal states. This is a discrete HMM; it's also possible to use HMMs with continuous time models [Rabiner, 1989].

Just as with the relationship between AR and MA models (Section 20.1), an HMM can be approximated by an ordinary Markov model, but the latter might require an enormous order to capture the behavior of the former because the rules for dynamics in the present can depend on a change of models that occurred a long time ago.

We'll assume that the internal state x can take on N discrete values, for convenience taken to be $x = \{1, \ldots, N\}$. Call x_t the internal state at time t, and let $\{y_1, \ldots, y_T\}$

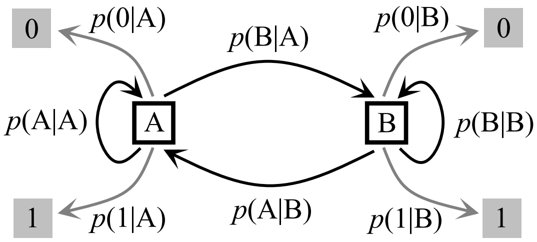

Figure 19.4. A Hidden Markov Model.

be a set of measurements of the observable. An HMM is specified by three sets of probabilities: $p(x_{t+1}|x_t)$, the internal transitions, $p(y_t|x_t)$, the emission of an observable given the internal state, and $p(x_1)$, the initial distribution of the internal states.

The key quantity to estimate is $p(x_t, x_{t+1}, y_1, \ldots, y_T)$, the probability to see a pair of internal states along with the observations. From this we can find the probability to see a transition given the observations,

$$\begin{aligned} p(x_{t+1}|x_t, y_1, \ldots, y_T) &= \frac{p(x_t, x_{t+1}, y_1, \ldots, y_T)}{p(x_t, y_1, \ldots, y_T)} \\ &= \frac{p(x_t, x_{t+1}, y_1, \ldots, y_T)}{\sum_{x_{t+1}=1}^{N} p(x_t, x_{t+1}, y_1, \ldots, y_T)} \quad , \end{aligned} \tag{19.52}$$

and then the absolute transition probability can be estimated by averaging over the record

$$p(x_{t+1} = j|x_t = i) \approx \frac{1}{T} \sum_{t'=1}^{T} p(x_{t'+1} = j|x_{t'} = i, y_1, \ldots, y_T) \quad . \tag{19.53}$$

Similarly, the probability of seeing an internal state is

$$\begin{aligned} p(x_t|y_1, \ldots, y_T) &= \frac{p(x_t, y_1, \ldots, y_T)}{p(y_1, \ldots, y_T)} \\ &= \frac{\sum_{x_{t+1}=1}^{N} p(x_t, x_{t+1}, y_1, \ldots, y_T)}{\sum_{x_t=1}^{N} \sum_{x_{t+1}=1}^{N} p(x_t, x_{t+1}, y_1, \ldots, y_T)} \quad , \end{aligned} \tag{19.54}$$

which can be used to estimate the observable probability by another sum over the data

$$p(y_t = j|x_t = i) \approx \frac{\sum_{t'|y_{t'}=j} p(x_{t'} = i|y_1, \ldots, y_T)}{\sum_{t'=1}^{T} p(x_{t'} = i|y_1, \ldots, y_T)} \quad , \tag{19.55}$$

as well as the absolute probability of an internal state

$$p(x = i) \approx \frac{1}{T} \sum_{t=1}^{T} p(x_t = i|y_1, \ldots, y_T) \quad . \tag{19.56}$$

There is a problem lurking in the estimation of these quantities. Consider the probability of the model to produce the observations $p(y_1, \ldots, y_T)$. Since we don't know the sequence of internal states we have to sum over all of the possibilities

$$p(y_1, \ldots, y_T) = \sum_{x_1=1}^{N} \cdots \sum_{x_T=1}^{N} p(x_1, \ldots, x_T, y_1, \ldots, y_T) \quad . \tag{19.57}$$

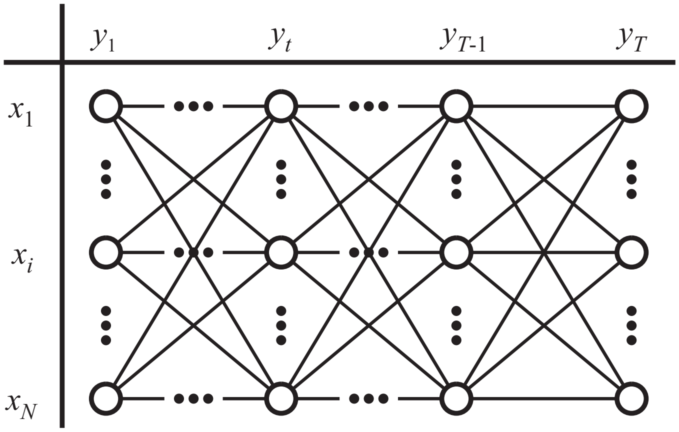

Figure 19.5. Hidden Markov Model trellis.

This is a set of T sums over N terms, requiring N^T operations. That's a a big number! A model with 10 internal states and an observed sequence of 100 points requires adding 10^{100} terms, which is larger than the number of atoms in the universe ($\sim 10^{70}$). The problem may be seen in Figure 19.5. The observed outputs are written across the top, with the possible internal states under them. The exponential explosion comes in the number of different paths through this *trellis*.

The trellis also points to a solution: each column depends only on the previous column, and so we are doing far too much work by recalculating each column over and over for every path that passes through it. Let's start with the last step. Notice that it can be written as a sum over the internal states,

$$\begin{aligned} p(y_1,\ldots,y_T) &= \sum_{x_T=1}^{N} p(x_T,y_1,\ldots,y_T) \\ &= \sum_{x_T=1}^{N} p(y_T|x_T,y_1,\ldots,y_{T-1})\, p(x_T,y_1,\ldots,y_{T-1}) \quad . \end{aligned} \tag{19.58}$$

Because the output probability depends only on the internal state this can be simplified to

$$p(y_1,\ldots,y_T) = \sum_{x_T=1}^{N} p(y_T|x_T)\, p(x_T,y_1,\ldots,y_{T-1}) \quad . \tag{19.59}$$

Factored again over the previous step,

$$\begin{aligned} p(y_1,\ldots,y_T) &= \sum_{x_T=1}^{N} p(y_T|x_T) \sum_{x_{T-1}=1}^{N} p(x_T,x_{T-1},y_1,\ldots,y_{T-1}) \\ &= \sum_{x_T=1}^{N} p(y_T|x_T) \sum_{x_{T-1}=1}^{N} p(x_T|x_{T-1},y_1,\ldots,y_{T-1})\, p(x_{T-1},y_1,\ldots,y_{T-1}) \\ &= \sum_{x_T=1}^{N} p(y_T|x_T) \sum_{x_{T-1}=1}^{N} p(x_T|x_{T-1})\, p(x_{T-1},y_1,\ldots,y_{T-1}) \quad , \end{aligned} \tag{19.60}$$

dropping the dependence of the internal transition probability on anything but the previous state. Continuing in this fashion back to the beginning we find that

$$
\begin{aligned}
p(y_1,\ldots,y_T) \;=\; & \sum_{x_T=1}^{N} p(y_T|x_T) \sum_{x_{T-1}=1}^{N} p(x_T|x_{T-1})\; p(y_{T-1}|x_{T-1}) \\
& \cdots \sum_{x_2=1}^{N} p(x_3|x_2)\; p(y_2|x_2) \sum_{x_1=1}^{N} p(x_2|x_1)\; p(y_1|x_1)\; p(x_1) \quad .
\end{aligned}
\tag{19.61}
$$

The x_1 sum has N terms and must be done for all values of x_2, a total of N^2 operations. Since there are T of these, the cost of the calculation drops to $\mathcal{O}(N^2T)$ – quite a saving over N^T. As in so many other areas, a hard problem becomes easy if it is written recursively. For an HMM this is called the *forward algorithm.*

The same idea works in reverse. Start with the probability to see a sequence of observables given a starting initial state, and factor it over the first step:

$$
\begin{aligned}
p(y_t,\ldots,y_T|x_t) &= \sum_{x_{t+1}=1}^{N} p(x_{t+1},y_t,\ldots,y_T|x_t) \\
&= \sum_{x_{t+1}=1}^{N} p(y_t|x_t,x_{t+1},y_{t+1},\ldots,y_T)\; p(x_{t+1},y_{t+1},\ldots,y_T|x_t) \\
&= p(y_t|x_t) \sum_{x_{t+1}=1}^{N} p(y_{t+1},\ldots,y_T|x_t,x_{t+1})\; p(x_{t+1}|x_t) \\
&= p(y_t|x_t) \sum_{x_{t+1}=1}^{N} p(x_{t+1}|x_t)\; p(y_{t+1},\ldots,y_T|x_{t+1}) \quad .
\end{aligned}
\tag{19.62}
$$

Continuing on to the end,

$$
\begin{aligned}
p(y_t,\ldots,y_T|x_t) \;=\; & p(y_t|x_t) \sum_{x_{t+1}=1}^{N} p(x_{t+1}|x_t)\; p(y_{t+1}|x_{t+1}) \\
\times & \sum_{x_{t+2}=1}^{N} p(x_{t+2}|x_{t+1})\; p(y_{t+2}|x_{t+2}) \;\cdots \sum_{x_{T-1}=1}^{N} p(x_{T-1}|x_{T-2})\; p(y_{T-1}|x_{T-1}) \\
\times & \sum_{x_T=1}^{N} p(x_T|x_{T-1})\; p(y_T|x_T) \quad .
\end{aligned}
\tag{19.63}
$$

This is called (can you guess?) the *backwards algorithm.*

Now return to the probability to see a pair of internal states and the observations. This can be factored as

$$
\begin{aligned}
p(x_t,x_{t+1},y_1,\ldots,y_T) = {} & p(x_t,y_1,\ldots,y_t)\; p(x_{t+1}|x_t,y_1,\ldots,y_t) \\
& p(y_{t+1},\ldots,y_T|x_t,x_{t+1},y_t,\ldots,y_T) \ ,
\end{aligned}
\tag{19.64}
$$

or dropping irrelevant variables,

$$
p(x_t,x_{t+1},y_1,\ldots,y_T) = p(x_t,y_1,\ldots,y_t)\; p(x_{t+1}|x_t)\; p(y_{t+1},\ldots,y_T|x_{t+1}) \quad .
$$

There are three factors on the right. The first is what we find from the forward algorithm, the middle one is the transition probability specified by the HMM, and the last is the result of the backward algorithm. Therefore this quantity can be calculated for all points by a linear-time pass through the data.

Once that's been done the resulting distributions can be used to update the transition probabilities according to equations (19.53) and (19.55). This procedure can then be iterated, first using the transition probabilities and the observables to update the estimate of the internal probabilities, then using the internal probabilities to find new values of the transition probabilities. Going back and forth between finding probabilities given parameters and finding the most likely parameters given probabilities is just the *Expectation-Maximization* (*EM*) algorithm that we saw in Section 16.3, which in the context of HMMs is called the *Baum–Welch* algorithm. It finds the maximum likelihood parameters starting from initial guesses for them. For a model with continuous parameters the M step becomes a maximization with respect to the parameterized distribution of internal states.

The combination of the forward-backward algorithm and EM finds the parameters of an HMM but it provides no guidance into choosing the architecture. The need to specify the architecture is the weakness, and strength, of using HMMs. In applications where there is some *a priori* insight into the internal states it is straightforward to build that in. A classic example, which helped drive the development of HMMs, is in speech recognition. Here the outputs can be parameters for a sound synthesis model, say ARMA coefficients (Section 20.1), and the internal states are phonemes and then words. It's hard to recognize these primitives from just a short stretch of sound, but the possible utterances are a strong function of what has preceeded them. The same thing applies to many other recognition tasks, such as reading handwriting, where a scrawled letter can be interpreted based on its context. An HMM provides the means to express these ideas.

The most important application of an HMM comes not on the training data but in applying the resulting model to deduce the hidden states given new data. To do this we want to find the most likely sequence of states given the data,

$$\begin{aligned}\underset{x_1\ldots x_T}{\operatorname{argmax}}\, p(x_1,\ldots,x_T|y_1,\ldots,y_T) =& \underset{x_1\ldots x_T}{\operatorname{argmax}} \frac{p(x_1,\ldots,x_T,y_1,\ldots,y_T)}{p(y_1,\ldots,y_T)} \\ =& \underset{x_1\ldots x_T}{\operatorname{argmax}}\, p(x_1,\ldots,x_T,y_1,\ldots,y_T) \qquad (19.65)\end{aligned}$$

(the denominator can be dropped because it doesn't affect the maximization). $\operatorname{argmax}_x f(x)$ is defined to be the argument x that gives f the maximum value, as compared to $\max_x f(x)$ which is the value of f at the maximum.

Naively this requires checking the likelihood of every path through the trellis, an $\mathcal{O}(N^T)$ calculation. Not surprisingly, the same recursive trick that we used before also works here. Start by factoring out the final step and dropping terms that are irrelevant to the distribution,

$$\begin{aligned}&\max_{x_1\ldots x_T} p(x_1,\ldots,x_T,y_1,\ldots,y_T) \\ =& \max_{x_1\ldots x_T} p(x_T,y_T|x_1,\ldots,x_{T-1},y_1,\ldots,y_{T-1})\; p(x_1,\ldots,x_{T-1},y_1,\ldots,y_{T-1}) \\ =& \max_{x_1\ldots x_T} p(x_T,y_T|x_{T-1})\; p(x_1,\ldots,x_{T-1},y_1,\ldots,y_{T-1}) \qquad (19.66)\end{aligned}$$

$$
\begin{aligned}
&= \max_{x_1 \ldots x_T} p(y_T|x_T, x_{T-1})\, p(x_T|x_{T-1})\, p(x_1, \ldots, x_{T-1}, y_1, \ldots, y_{T-1}) \\
&= \max_{x_T} p(y_T|x_T) \max_{x_1 \ldots x_{T-1}} p(x_T|x_{T-1})\, p(x_1, \ldots, x_{T-1}, y_1, \ldots, y_{T-1}) \quad .
\end{aligned}
$$

Continuing in this fashion back to the beginning,

$$
\begin{aligned}
&\max_{x_1 \ldots x_T} p(x_1, \ldots, x_T, y_1, \ldots, y_T) \\
&= \max_{x_T} p(y_T|x_T) \max_{x_{T-1}} p(x_T|x_{T-1})\, p(y_{T-1}|x_{T-1}) \\
&\quad \cdots \max_{x_2} p(x_3|x_2)\, p(y_2|x_2) \max_{x_1} p(x_2|x_1)\, p(y_1|x_1)\, p(x_1) \quad .
\end{aligned}
\tag{19.67}
$$

This is now once again an $\mathcal{O}(N^2T)$ calculation. It is called the *Viterbi* algorithm, and is very important beyond HMMs in decoding signals sent through noisy channels that have had correlations introduced by a convolutional coder [Sklar, 1988]. There is one subtlety in implementing it: each maximization has a set of outcomes based on the unknown value of the following step. This is handled by using the maximum value for each outcome and keeping track of which one was used at each step, then backtracking from the end of the calculation once the final maximization is known.

Figures 19.1 and 19.5 were used to help explain Kalman filtering and HMMs by drawing the connections among the variables. It's possible to go much further with such diagrams, using them to write down probabilistic models with more complex dependencies than what we've covered, and applying graphical techniques to arrive at the kinds of simplifications we found to make the estimation problems tractable [Jordan, 1999]. Such architectural complexity is useful when there is advance knowledge to guide it; the next chapter turns to the opposite limit.

19.6 GRAPHICAL NETWORKS

$$
\begin{aligned}
m_{x_i \to f_i}(x_i) &= \prod_{f_j\ (\sim f_i)} m_{f_j \to x_i}(x_i) \\
m_{f_i \to x_i}(x_i) &= \sum_{\sim x_i} f_i(\{\vec{x}\}_i) \prod_{x_j\ (\sim x_i)} m_{x_j \to f_i}(x_j)
\end{aligned}
\tag{19.68}
$$

$m_{\bullet \to x_i}(x_i) = 1$
likelihood, log-likelihood

19.7 SELECTED REFERENCES

[Brown & Hwang, 1997] Brown, Robert Grover, & Hwang, Patrick Y.C.(1997). *Introduction to Random Signals and Applied Kalman Filtering*. 3rd edn. New York, NY: Wiley.

A good practical introduction to estimation theory.

[Catlin, 1989] Catlin, Donald E. (1989). *Estimation, Control, and the Discrete Kalman Filter*. Applied Mathematical Sciences, vol. 71. New York, NY: Springer Verlag.

The rigorous mathematical background of Kalman filtering.

[Honerkamp, 1994] Honerkamp, Josef (1994). *Stochastic Dynamical Systems: Concepts, Numerical Methods, Data Analysis.* New York, NY: VCH. Translated by Katja Lindenberg.

The connection between estimation theory and the theory of stochastic dynamical systems.

19.8 PROBLEMS

(18.1) What is the approximate Kalman gain matrix in the limit of small measurement noise? How is the error matrix updated in this case?

(18.2) Take as a test signal a periodically modulated sinusoid with noise added,

$$y_n = \sin[0.1t_n + 4\sin(0.01t_n)] + \eta \equiv \sin(\theta_n) + \eta \quad , \tag{19.69}$$

where η is a Gaussian noise process with $\sigma = 0.1$. Design an extended Kalman filter to estimate the noise-free signal. Use a two-component state vector $\vec{x}_n = (\theta_n, \theta_{n-1})$, and assume for the internal model a linear extrapolation $\theta_{n+1} = \theta_n + (\theta_n - \theta_{n-1})$. Take the system noise matrix $\mathbf{N}^x$ to be diagonal, and plot the predicted value of y versus the measured value of y if the standard deviation of the system noise is chosen to be 10^{-1}, 10^{-3}, and 10^{-5}. Use the identity matrix for the initial error estimate.

(18.3) HMM coin-flipping ...