From Hidden Markov Models to Linear Dynamical Systems

Thomas P. Minka

(revised July 18, 1999)

Abstract

Hidden Markov Models (HMMs) and Linear Dynamical Systems (LDSs) are based on the same assumption: a hidden state variable, of which we can make noisy measurements, evolves with Markovian dynamics. Both have the same independence diagram and consequently the learning and inference algorithms for both have the same structure. The only difference is that the HMM uses a discrete state variable with arbitrary dynamics and arbitrary measurements while the LDS uses a continuous state variable with linear-Gaussian dynamics and measurements. We show how the forward-backward equations for the HMM, specialized to linear-Gaussian assumptions, lead directly to Kalman filtering and Rauch-Tung-Streibel smoothing. We also investigate the most general possible modeling assumptions which lead to efficient recursions in the case of continuous state variables.

1 Introduction

Both HMMs and LDSs specify a joint probability distribution over states $\mathbf{s}_1..\mathbf{s}_T$ and measurements $\mathbf{x}_1..\mathbf{x}_T$. In both cases, the distribution factors in the same way:

$$p(\mathbf{s}_1..\mathbf{s}_T, \mathbf{x}_1..\mathbf{x}_T) = p(\mathbf{s}_1)p(\mathbf{x}_1|\mathbf{s}_1)\prod_{t=2}^{T} p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{x}_t|\mathbf{s}_t) \quad (1)$$

where all terms are implicitly conditioned on the parameters of the model. This factorization is equivalent to the following independence diagram:

$$\begin{tikzcd}
\mathbf{s}_1 \arrow[r] \arrow[d] & \mathbf{s}_2 \arrow[r] \arrow[d] & \mathbf{s}_3 \arrow[r] \arrow[d] & \circ \quad \circ \quad \circ \arrow[r] & \mathbf{s}_T \arrow[d] \\
\mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & & \mathbf{x}_T
\end{tikzcd}$$

which is a graphical way to say that the hidden states form a Markov chain that emits a time series of outputs. To convert a graph into a factorization, make one term per node, conditioned on the nodes pointing into it. In this way, mathematical manipulations can be replaced with graphical ones.

Besides a Markov chain (horizontal arrows) with noisy measurements (vertical arrows), this graph can also be considered a mixture model (vertical arrows) with coupling between the assignment variables (horizontal arrows). It is just a matter of whether you consider the horizontal or vertical links to be fundamental. More complex independence diagrams can be interpreted in even more different ways.

The main point of this paper is that the independence structure reflected in the graph determines the overall structure of the algorithms. The choice of conditional distributions only affects computational details, which are the only difference between the HMM and the LDS.

Given one or more measurement sequences from the model, there are three basic tasks we want to perform:

Classification Compute the probability that a measurement sequence $\mathbf{x}_1..\mathbf{x}_T$ came from this model.

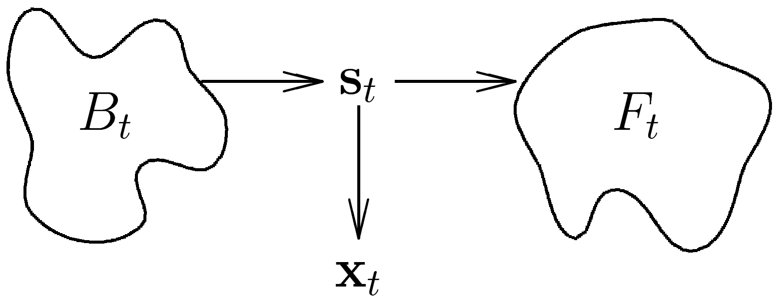
Inference Compute the probability that the system was in state $\mathbf{z}$ at time t, i.e. $p(\mathbf{s}_t = \mathbf{z}|\mathbf{x}_1..\mathbf{x}_T)$.

Learning Determine the parameter settings which maximize the probability of the measurement sequences.

The first two tasks are accomplished via forward-backward recursions which propagate information across the graph. The third task is performed iteratively by EM, where the E-step is the inference task and the M-step is a linear maximum-likelihood problem. Only the first two tasks are discussed here; the M-step for a LDS is straightforward and can be found in Ghahramani & Hinton (1996).

2 Generic forward-backward propagation

This section develops the complete solution to the classification and inference tasks. In exchange for making the restrictive assumption of a Markov chain, we get an exact, efficient inference algorithm. The reason is that each state variable $\mathbf{s}_t$ separates the graph into three independent parts:



where

$$B_t = \{\mathbf{x}_1..\mathbf{x}_{t-1}, \mathbf{s}_1..\mathbf{s}_{t-1}\} \tag{2}$$
$$F_t = \{\mathbf{x}_{t+1}..\mathbf{x}_T, \mathbf{s}_{t+1}..\mathbf{s}_T\} \tag{3}$$

Therefore,

$$p(B_t, \mathbf{s}_t, \mathbf{x}_t, F_t) = p(B_t, \mathbf{s}_t)p(\mathbf{x}_t|\mathbf{s}_t)p(F_t|\mathbf{s}_t) \tag{4}$$

If we want $p(\mathbf{s}_t, \mathbf{x}_1..\mathbf{x}_T)$, we can integrate out the other state variables:

$$\begin{aligned} p(\mathbf{s}_t, \mathbf{x}_1..\mathbf{x}_T) &= \int_{\mathbf{s}_1..\mathbf{s}_{t-1}} \int_{\mathbf{s}_{t+1}..\mathbf{s}_T} p(B_t, \mathbf{s}_t, \mathbf{x}_t, F_t) && (5) \\ &= \left(\int_{\mathbf{s}_1..\mathbf{s}_{t-1}} p(B_t, \mathbf{s}_t)\right) p(\mathbf{x}_t|\mathbf{s}_t) \left(\int_{\mathbf{s}_{t+1}..\mathbf{s}_T} p(F_t|\mathbf{s}_t)\right) && (6) \\ &= p(B_t^x, \mathbf{s}_t)p(\mathbf{x}_t|\mathbf{s}_t)p(F_t^x|\mathbf{s}_t) && (7) \end{aligned}$$

where

$$B_t^x = \{\mathbf{x}_1..\mathbf{x}_{t-1}\} \tag{8}$$
$$F_t^x = \{\mathbf{x}_{t+1}..\mathbf{x}_T\} \tag{9}$$

The probability of the entire measurement sequence is $p(\mathbf{x}_1..\mathbf{x}_T) = \sum_{\mathbf{s}_t} p(\mathbf{s}_t, \mathbf{x}_1..\mathbf{x}_T)$ and the probability of being in state $\mathbf{z}$ at time t is $p(\mathbf{s}_t = \mathbf{z}, \mathbf{x}_1..\mathbf{x}_T)/p(\mathbf{x}_1..\mathbf{x}_T)$, so the joint distribution (7) is all we need to solve the classification and inference tasks.

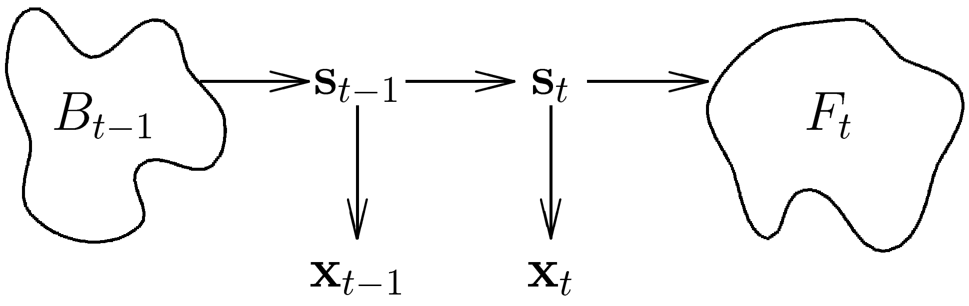
The idea is to compute the left and right terms in (7) recursively on the left and right subgraphs. To this end, define

$$\alpha_t(\mathbf{s}_t) = p(B_t^x, \mathbf{s}_t)p(\mathbf{x}_t|\mathbf{s}_t) = p(B_t^x, \mathbf{x}_t, \mathbf{s}_t) \tag{10}$$
$$\beta_t(\mathbf{s}_t) = p(F_t^x|\mathbf{s}_t) \tag{11}$$

corresponding to the notation in Rabiner (1989). Therefore

$$p(\mathbf{s}_t, \mathbf{x}_1..\mathbf{x}_T) = \alpha_t(\mathbf{s}_t)\beta_t(\mathbf{s}_t) \tag{12}$$

To derive the recursions, consider two consecutive time-steps:



In this diagram, $B_t = B_{t-1} \cup \{\mathbf{s}_{t-1}, \mathbf{x}_{t-1}\}$ and $F_{t-1} = \{\mathbf{s}_t, \mathbf{x}_t\} \cup F_t$.

The independence diagram tells us

$$\begin{aligned}
\alpha_t(\mathbf{s}_t) = p(\mathbf{x}_t|\mathbf{s}_t)p(B_t^x, \mathbf{s}_t) &= p(\mathbf{x}_t|\mathbf{s}_t) \int_{\mathbf{z}} p(B_{t-1}^x, \mathbf{s}_{t-1} = \mathbf{z}, \mathbf{x}_{t-1}, \mathbf{s}_t) &(13)\\
&= p(\mathbf{x}_t|\mathbf{s}_t) \int_{\mathbf{z}} p(B_{t-1}^x, \mathbf{s}_{t-1} = \mathbf{z})p(\mathbf{x}_{t-1}|\mathbf{s}_{t-1} = \mathbf{z})p(\mathbf{s}_t|\mathbf{s}_{t-1} = \mathbf{z}) &(14)\\
&= p(\mathbf{x}_t|\mathbf{s}_t) \int_{\mathbf{z}} p(\mathbf{s}_t|\mathbf{s}_{t-1} = \mathbf{z})\alpha_{t-1}(\mathbf{z}) &(15)\\
\beta_{t-1}(\mathbf{s}_{t-1}) = p(F_{t-1}^x|\mathbf{s}_{t-1}) &= \int_{\mathbf{z}} p(\mathbf{s}_t = \mathbf{z}, \mathbf{x}_t, F_t^x|\mathbf{s}_{t-1}) &(16)\\
&= \int_{\mathbf{z}} p(\mathbf{s}_t = \mathbf{z}|\mathbf{s}_{t-1})p(\mathbf{x}_t|\mathbf{s}_t = \mathbf{z})p(F_t^x|\mathbf{s}_t = \mathbf{z}) &(17)\\
&= \int_{\mathbf{z}} p(\mathbf{s}_t = \mathbf{z}|\mathbf{s}_{t-1})p(\mathbf{x}_t|\mathbf{s}_t = \mathbf{z})\beta_t(\mathbf{z}) &(18)
\end{aligned}$$

To start off the recursions:

$$\begin{aligned}
\alpha_1(\mathbf{s}_1) &= p(\mathbf{s}_1)p(\mathbf{x}_1|\mathbf{s}_1) &(19)\\
\beta_T(\mathbf{s}_T) &= 1 &(20)
\end{aligned}$$

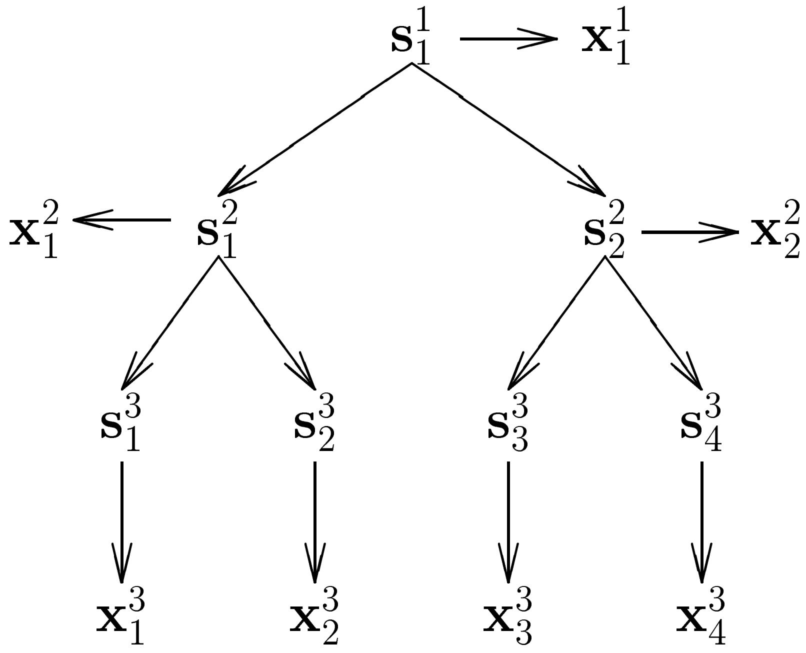
Putting these equations together lets us compute all $\alpha_t(\cdot)$ (going forward in time) and all $\beta_t(\cdot)$ (going backward in time) and therefore all marginals for $\mathbf{s}_t$. A backward step is needed because we want the best estimate of $\mathbf{s}_t$ using all the data at hand, including data received after time t. It may be helpful to consider the graph to be a parallel machine, where each node is a processor and the links are communication paths. The forward-backward recursions make sure that every $\mathbf{s}_t$ processor gets information about the entire measurement sequence.

If we instead want the marginal pairwise density of $\mathbf{s}_{t-1}$ and $\mathbf{s}_t$, the diagram tells us, going left to right:

$$\begin{aligned}
p(\mathbf{s}_{t-1}, \mathbf{s}_t|B_{t-1}^x, F_t^x, \mathbf{x}_{t-1}, \mathbf{x}_t) &\propto p(B_{t-1}^x, \mathbf{s}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{s}_{t-1})p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{x}_t|\mathbf{s}_t)p(F_t^x|\mathbf{s}_t) &(21)\\
&= \alpha_{t-1}(\mathbf{s}_{t-1})p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{x}_t|\mathbf{s}_t)\beta_t(\mathbf{s}_t) &(22)
\end{aligned}$$

This obviously generalizes to triples of state variables and so on. Therefore any inference task on this model can be solved by computing the α's and β's.

Since this algorithm is derived from independence relationships alone, it is clear that any transition density $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ and any emission density $p(\mathbf{x}_t|\mathbf{s}_t)$ can be used, in principle. These densities may depend on time, i.e. the process need not be stationary. Furthermore, if the independence diagram were not a chain but rather a tree, exact inference could still be performed efficiently. This case is illustrated below:



Since each hidden state node still partitions the graph into independent pieces, a similar set of recursions can be derived, resulting in an upward-downward algorithm (Pearl, 1988). This is left as an exercise.

2.1 Scaling factors

In preparation for the Linear Dynamical System equations, it is helpful to reformulate the forward-backward recursions in terms of scaled α's and β's. The rescaling is also useful for avoiding numerical underflow (Rabiner, 1989).

Define

$$c_t = p(\mathbf{x}_t|\mathbf{x}_1..\mathbf{x}_{t-1}) \tag{23}$$

which is the inverse of Rabiner's (1989) scaling factor, also called c_t. Now factor c_t out of the original definition of $\alpha_t(\mathbf{s}_t)$:

$$\begin{aligned}\alpha_t(\mathbf{s}_t) &= p(\mathbf{s}_t, B_t^x, \mathbf{x}_t) = p(\mathbf{s}_t, \mathbf{x}_1..\mathbf{x}_t) && (24)\\ &= p(\mathbf{x}_1..\mathbf{x}_t)p(\mathbf{s}_t|\mathbf{x}_1..\mathbf{x}_t) && (25)\\ &= (\prod_{\tau=1}^{t} c_\tau)\hat{\alpha}_t(\mathbf{s}_t) && (26)\end{aligned}$$

which defines $\hat{\alpha}_t(\mathbf{s}_t) = p(\mathbf{s}_t|\mathbf{x}_1..\mathbf{x}_t)$. Equation 15 can be turned into a recursion for $\hat{\alpha}$:

$$(\prod_{\tau=1}^{t} c_\tau)\hat{\alpha}_t(\mathbf{s}_t) = p(\mathbf{x}_t|\mathbf{s}_t)\int_{\mathbf{z}} p(\mathbf{s}_t|\mathbf{s}_{t-1}=\mathbf{z})(\prod_{\tau=1}^{t-1} c_\tau)\hat{\alpha}_{t-1}(\mathbf{z}) \tag{27}$$

$$c_t\hat{\alpha}_t(\mathbf{s}_t) = p(\mathbf{x}_t|\mathbf{s}_t)\int_{\mathbf{z}} p(\mathbf{s}_t|\mathbf{s}_{t-1}=\mathbf{z})\hat{\alpha}_{t-1}(\mathbf{z}) \tag{28}$$

where c_t is computed as the factor that normalizes $\hat{\alpha}_t(\mathbf{s}_t)$. This way $\hat{\alpha}_t(\mathbf{s}_t)$ and all of the c_t stay within machine precision during the forward propagation.

Similarly, define

$$\beta_t(\mathbf{s}_t) = (\prod_{\tau=t+1}^{T} c_\tau)\hat{\beta}_t(\mathbf{s}_t) \tag{29}$$

where c_t is the same scaling factor used for $\hat{\alpha}_t$. Equation 18 turns into

$$\hat{\beta}_{t-1}(\mathbf{s}_{t-1}) = \frac{1}{c_t}\int_{\mathbf{z}} p(\mathbf{s}_t=\mathbf{z}|\mathbf{s}_{t-1})p(\mathbf{x}_t|\mathbf{s}_t=\mathbf{z})\hat{\beta}_t(\mathbf{z}) \tag{30}$$

which keeps $\hat{\beta}$ within machine precision.

The marginal distributions now become exact in terms of the scaled α's and β's (the distributions do not require normalization):

$$p(\mathbf{s}_t|\mathbf{x}_1..\mathbf{x}_T) = \hat{\alpha}_t(\mathbf{s}_t)\hat{\beta}_t(\mathbf{s}_t) \tag{31}$$

$$p(\mathbf{s}_{t-1},\mathbf{s}_t|\mathbf{x}_1..\mathbf{x}_T) = \frac{1}{c_t}\hat{\alpha}_{t-1}(\mathbf{s}_{t-1})p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{x}_t|\mathbf{s}_t)\hat{\beta}_t(\mathbf{s}_t) \tag{32}$$

Note that the forward and backward recursions could be derived from these equations alone, by integrating the latter equation over $\mathbf{s}_{t-1}$ or $\mathbf{s}_t$, respectively.

The probability of the measurement sequence is now

$$p(\mathbf{x}_1..\mathbf{x}_T) = \prod_{\tau=1}^{T} c_\tau \tag{33}$$

2.2 Sequential vs. parallel recursions

The recursions just given are *parallel* in the sense that $\hat{\alpha}$ and $\hat{\beta}$ are computed independently of each other. It is also possible to derive a *sequential* recursion, where the $\hat{\beta}$'s are computed from the $\hat{\alpha}$'s instead of from the data. This is the type of recursion that is used for Linear Dynamical Systems. The idea is to make a backward recursion for the product $\hat{\alpha}_t(\mathbf{s}_t)\hat{\beta}_t(\mathbf{s}_t) = p(\mathbf{s}_t|\mathbf{x}_1..\mathbf{x}_T)$ as follows (Murphy, 1998):

$$\hat{\alpha}_{t-1}(\mathbf{s}_{t-1})\hat{\beta}_{t-1}(\mathbf{s}_{t-1}) = \int_{\mathbf{z}} p(\mathbf{s}_{t-1}|\mathbf{s}_t=\mathbf{z},\mathbf{x}_1..\mathbf{x}_T)\hat{\alpha}_t(\mathbf{z})\hat{\beta}_t(\mathbf{z}) \tag{34}$$

$$p(\mathbf{s}_{t-1}|\mathbf{s}_t,\mathbf{x}_1..\mathbf{x}_T) = \frac{p(\mathbf{s}_t|\mathbf{s}_{t-1})\hat{\alpha}_{t-1}(\mathbf{s}_{t-1})}{\int_{\mathbf{z}} p(\mathbf{s}_t|\mathbf{s}_{t-1}=\mathbf{z})\hat{\alpha}_{t-1}(\mathbf{z})} \tag{35}$$

This comes from multiplying (30) by $\hat{\alpha}_{t-1}(\mathbf{s}_{t-1})$ and applying (28) for c_t.

3 Linear dynamical systems

When the state variables $\mathbf{s}_t$ are discrete, the integrals above become sums and the $\hat{\alpha}$ and $\hat{\beta}$ functions can be stored and computed explicitly. This leads to the HMM forward-backward algorithm, which is completely general in terms of what the transition and emission probabilities can be. When the state variables are continuous, the $\hat{\alpha}$ and $\hat{\beta}$ functions must instead be stored implicitly and the integrals solved analytically. This leads to restrictions on the kinds of transition and emission densities we can handle efficiently, i.e. in time linear in the length of the measurement sequence.

To get a linear-time algorithm, each $\hat{\alpha}$ and $\hat{\beta}$ function must be represented with a constant number of parameters (constant with respect to T). Otherwise the integral at each step would take time proportional to T. For example, if we use a mixture of Gaussians, the number of components must remain constant. This is a very stringent requirement, since it means multiplication by the transition and emission densities cannot make the function more complex but only change its parameters. The only family of probability distributions with this property—closure under multiplication—is the exponential family. Fortunately, the exponential family is quite large, including the Gaussian, Gamma, Poisson, and Beta distributions. Any distribution which can be written as $p(\mathbf{z}, \theta) = \exp(\theta^{\mathrm{T}} f(\mathbf{z}) + g(\theta))$ is in the family.

Let Q denote the abstract choice of density for our model, e.g. $Q =$ Gaussian. Conditioned on s_t, the variables x_t and s_{t+1} must have density Q, though with arbitrary parameters. They must also have marginal density Q if s_t has density Q (because of (15)). If $Q =$ Gaussian, then this means all conditional densities must be linear:

$$\begin{aligned}
p(\mathbf{s}_t|\mathbf{s}_{t-1}) &\sim \mathcal{N}(\mathbf{A}\mathbf{s}_{t-1}, \Gamma) && (36)\\
p(\mathbf{x}_t|\mathbf{s}_t) &\sim \mathcal{N}(\mathbf{C}\mathbf{s}_t, \Sigma) && (37)\\
\text{where } \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V}) &= \frac{1}{|2\pi\mathbf{V}|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^{\mathrm{T}}\mathbf{V}^{-1}(\mathbf{x}-\mathbf{m}))
\end{aligned}$$

This model can also be written, more conventionally, as a set of linear equations driven by noise (e.g. as in Ghahramani & Hinton (1996)):

$$\begin{aligned}
\mathbf{s}_t &= \mathbf{A}\mathbf{s}_{t-1} + \mathbf{w}_t && (38)\\
\mathbf{w}_t &\sim \mathcal{N}(0, \Gamma) && (39)\\
\mathbf{x}_t &= \mathbf{C}\mathbf{s}_t + \mathbf{v}_t && (40)\\
\mathbf{v}_t &\sim \mathcal{N}(0, \Sigma) && (41)
\end{aligned}$$

Hence choosing $Q =$ Gaussian leads to exactly the class of Linear Dynamical Systems (LDSs). In their fullest generality, the parameters $\mathbf{A}, \mathbf{C}, \Gamma$ and Σ may depend on t, i.e. the model may be nonstationary. An additional parameter of the model is the initial state distribution $\mathcal{N}(\mathbf{m}_0, \mathbf{V}_0)$ which starts off the recursions.

3.1 Forward recursion: Kalman filter

Once we've decided on a linear-Gaussian restriction, the next step is to perform the integrals in the forward-backward equations. The representation for $\hat{\alpha}_t$ can be any fixed-length mixture of Gaussians. For simplicity, let it be one Gaussian: $\hat{\alpha}_t(\mathbf{s}_t) \sim \mathcal{N}(\mathbf{m}_t, \mathbf{V}_t)$. The forward equation (28) becomes

$$c_t\hat{\alpha}_t(\mathbf{s}_t) \quad = \quad \mathcal{N}(\mathbf{x}_t; \mathbf{C}\mathbf{s}_t, \Sigma) \int_{\mathbf{z}} \mathcal{N}(\mathbf{s}_t; \mathbf{A}\mathbf{z}, \Gamma)\, \mathcal{N}(\mathbf{z}; \mathbf{m}_{t-1}, \mathbf{V}_{t-1}) \tag{42}$$

The following fact comes in handy. Suppose we have a Gaussian random vector partitioned into $\mathbf{x}$ and $\mathbf{y}$ with the following mean and covariance:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mathbf{m}_x \\ \mathbf{A}\mathbf{m}_x \end{bmatrix}, \begin{bmatrix} \mathbf{V}_x & \mathbf{V}_x\mathbf{A}^{\mathrm{T}} \\ \mathbf{A}\mathbf{V}_x & \mathbf{A}\mathbf{V}_x\mathbf{A}^{\mathrm{T}} + \mathbf{V}_y \end{bmatrix}) \tag{43}$$

Since $p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) = p(\mathbf{y})p(\mathbf{x}|\mathbf{y})$, we get

$$\begin{aligned} \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{V}_y)\, \mathcal{N}(\mathbf{x}; \mathbf{m}_x, \mathbf{V}_x) &= \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{m}_x, \mathbf{A}\mathbf{V}_x\mathbf{A}^{\mathrm{T}} + \mathbf{V}_y) \\ &\quad\ \mathcal{N}(\mathbf{x}; \mathbf{m}_x + \mathbf{K}(\mathbf{y} - \mathbf{A}\mathbf{m}_x), (\mathbf{I} - \mathbf{K}\mathbf{A})\mathbf{V}_x) &(44) \\ \text{where } \mathbf{K} &= \mathbf{V}_x\mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{V}_x\mathbf{A}^{\mathrm{T}} + \mathbf{V}_y)^{-1} &(45) \end{aligned}$$

using the rule for conditioning a Gaussian (Minka, 1997).

Applying this rule twice to (42):

$$\begin{aligned} c_t\hat{\alpha}_t(\mathbf{s}_t) &= \mathcal{N}(\mathbf{x}_t; \mathbf{C}\mathbf{s}_t, \Sigma)\, \mathcal{N}(\mathbf{s}_t; \mathbf{A}\mathbf{m}_{t-1}, \mathbf{A}\mathbf{V}_{t-1}\mathbf{A}^{\mathrm{T}} + \Gamma) &(46) \\ &= \mathcal{N}(\mathbf{x}_t; \mathbf{C}\mathbf{A}\mathbf{m}_{t-1}, \mathbf{C}\mathbf{P}_{t-1}\mathbf{C}^{\mathrm{T}} + \Sigma) \\ &\quad\ \mathcal{N}(\mathbf{s}_t; \mathbf{A}\mathbf{m}_{t-1} + \mathbf{K}_t(\mathbf{x}_t - \mathbf{C}\mathbf{A}\mathbf{m}_{t-1}), (\mathbf{I} - \mathbf{K}_t\mathbf{C})\mathbf{P}_{t-1}) &(47) \\ \text{where } \mathbf{K}_t &= \mathbf{P}_{t-1}\mathbf{C}^{\mathrm{T}}(\mathbf{C}\mathbf{P}_{t-1}\mathbf{C}^{\mathrm{T}} + \Sigma)^{-1} &(48) \\ \text{and } \mathbf{P}_{t-1} &= \mathbf{A}\mathbf{V}_{t-1}\mathbf{A}^{\mathrm{T}} + \Gamma &(49) \end{aligned}$$

Therefore:

$$\begin{aligned} \mathbf{m}_t &= \mathbf{A}\mathbf{m}_{t-1} + \mathbf{K}_t(\mathbf{x}_t - \mathbf{C}\mathbf{A}\mathbf{m}_{t-1}) &(50) \\ \mathbf{V}_t &= (\mathbf{I} - \mathbf{K}_t\mathbf{C})\mathbf{P}_{t-1} &(51) \\ c_t &= \mathcal{N}(\mathbf{x}_t; \mathbf{C}\mathbf{A}\mathbf{m}_{t-1}, \mathbf{C}\mathbf{P}_{t-1}\mathbf{C}^{\mathrm{T}} + \Sigma) &(52) \end{aligned}$$

which are exactly the Kalman filter equations reported by Ghahramani & Hinton (1996). By design, the computational cost is constant per time step.

Since $\hat{\alpha}_1(\mathbf{s}_1) = p(\mathbf{s}_1|\mathbf{x}_1)$, the recursions start off with

$$\begin{aligned} \mathbf{m}_1 &= \mathbf{m}_0 + \mathbf{K}_1(\mathbf{x}_1 - \mathbf{C}\mathbf{m}_0) &(53) \\ \mathbf{V}_1 &= (\mathbf{I} - \mathbf{K}_1\mathbf{C})\mathbf{V}_0 &(54) \\ c_1 &= \mathcal{N}(\mathbf{x}_1; \mathbf{C}\mathbf{m}_0, \mathbf{C}\mathbf{V}_0\mathbf{C}^{\mathrm{T}} + \Sigma) &(55) \\ \mathbf{K}_1 &= \mathbf{V}_0\mathbf{C}^{\mathrm{T}}(\mathbf{C}\mathbf{V}_0\mathbf{C}^{\mathrm{T}} + \Sigma)^{-1} &(56) \end{aligned}$$

Note that the transition density could be such that $\mathbf{s}_t$ is a constant $\mathbf{s}$, in which case we have a recursive solution for the posterior of $\mathbf{s}$ given a series of independent, noisy observations. This again emphasizes that the observation density must come from the exponential family: it is the only family for which parameter estimation can be performed recursively, via sufficient statistics (this is the Pitman-Koopman theorem).

3.2 Backward recursion: Kalman smoothing

Now we can derive a sequential backward recursion for $\hat{\alpha}_t(\mathbf{s}_t)\hat{\beta}_t(\mathbf{s}_t) = \mathcal{N}(\mathbf{s}_t; \hat{\mathbf{m}}_t, \hat{\mathbf{V}}_t)$. This is the marginal distribution for $\mathbf{s}_t$, given the entire measurement sequence. It is also possible to derive a parallel backward recursion for $\hat{\beta}$; see Murphy (1998).

Multiplying the backward equation (30) by $\hat{\alpha}_{t-1}(\mathbf{s}_{t-1})$ gives us

$$\hat{\alpha}_{t-1}(\mathbf{s}_{t-1})\hat{\beta}_{t-1}(\mathbf{s}_{t-1}) = \mathcal{N}(\mathbf{s}_{t-1}; \mathbf{m}_{t-1}, \mathbf{V}_{t-1}) \int_{\mathbf{z}} \mathcal{N}(\mathbf{z}; \mathbf{A}\mathbf{s}_{t-1}, \Gamma)\mathcal{N}(\mathbf{x}_t; \mathbf{C}\mathbf{z}, \Sigma)\frac{\mathcal{N}(\mathbf{z}; \hat{\mathbf{m}}_t, \hat{\mathbf{V}}_t)}{c_t\hat{\alpha}_t(\mathbf{z})} \quad (57)$$

Combining the first two terms via (44) and substituting (46) for $c_t\hat{\alpha}_t(\mathbf{z})$ causes massive cancellation, leaving

$$\begin{aligned}
\hat{\alpha}_{t-1}(\mathbf{s}_{t-1})\hat{\beta}_{t-1}(\mathbf{s}_{t-1}) &= \int_{\mathbf{z}} \mathcal{N}(\mathbf{s}_{t-1}; \mathbf{m}_{t-1} + \mathbf{J}_{t-1}(\mathbf{z} - \mathbf{A}\mathbf{m}_{t-1}), (\mathbf{I} - \mathbf{J}_{t-1}\mathbf{A})\mathbf{V}_{t-1})\mathcal{N}(\mathbf{z}; \hat{\mathbf{m}}_t, \hat{\mathbf{V}}_t) \\
&= \int_{\mathbf{z}} \mathcal{N}(\mathbf{s}_{t-1} - \mathbf{m}_{t-1} + \mathbf{J}_{t-1}\mathbf{A}\mathbf{m}_{t-1}; \mathbf{J}_{t-1}\mathbf{z}, (\mathbf{I} - \mathbf{J}_{t-1}\mathbf{A})\mathbf{V}_{t-1})\mathcal{N}(\mathbf{z}; \hat{\mathbf{m}}_t, \hat{\mathbf{V}}_t) \\
&= \mathcal{N}(\mathbf{s}_{t-1} - \mathbf{m}_{t-1} + \mathbf{J}_{t-1}\mathbf{A}\mathbf{m}_{t-1}; \mathbf{J}_{t-1}\hat{\mathbf{m}}_t, \mathbf{J}_{t-1}\hat{\mathbf{V}}_t\mathbf{J}_{t-1}^{\mathrm{T}} + (\mathbf{I} - \mathbf{J}_{t-1}\mathbf{A})\mathbf{V}_{t-1}) \\
&= \mathcal{N}(\mathbf{s}_{t-1}; \mathbf{m}_{t-1} + \mathbf{J}_{t-1}(\hat{\mathbf{m}}_t - \mathbf{A}\mathbf{m}_{t-1}), \mathbf{V}_{t-1} + \mathbf{J}_{t-1}\hat{\mathbf{V}}_t\mathbf{J}_{t-1}^{\mathrm{T}} - \mathbf{J}_{t-1}\mathbf{A}\mathbf{V}_{t-1}) \\
\text{where } \mathbf{J}_{t-1} &= \mathbf{V}_{t-1}\mathbf{A}^{\mathrm{T}}(\mathbf{P}_{t-1})^{-1} \quad (58)
\end{aligned}$$

Therefore:

$$\begin{aligned}
\hat{\mathbf{m}}_{t-1} &= \mathbf{m}_{t-1} + \mathbf{J}_{t-1}(\hat{\mathbf{m}}_t - \mathbf{A}\mathbf{m}_{t-1}) \quad (59) \\
\hat{\mathbf{V}}_{t-1} &= \mathbf{V}_{t-1} + \mathbf{J}_{t-1}(\hat{\mathbf{V}}_t - \mathbf{P}_{t-1})\mathbf{J}_{t-1}^{\mathrm{T}} \quad (60)
\end{aligned}$$

since $\mathbf{A}\mathbf{V}_{t-1} = \mathbf{P}_{t-1}\mathbf{J}_{t-1}^{\mathrm{T}}$. These match the equations given by Ghahramani & Hinton (1996).

Since $\hat{\beta}_T(\mathbf{s}_T) = 1$, the backward recursion starts with

$$\begin{aligned}
\hat{\mathbf{m}}_T &= \mathbf{m}_T \quad (61) \\
\hat{\mathbf{V}}_T &= \mathbf{V}_T \quad (62)
\end{aligned}$$

Equation 32 for the marginal pairwise density becomes

$$p(\mathbf{s}_{t-1}, \mathbf{s}_t|\mathbf{x}_1..\mathbf{x}_T) = \mathcal{N}(\mathbf{s}_{t-1}; \mathbf{m}_{t-1}, \mathbf{V}_{t-1})\mathcal{N}(\mathbf{s}_t; \mathbf{A}\mathbf{s}_{t-1}, \Gamma)\mathcal{N}(\mathbf{x}_t; \mathbf{C}\mathbf{s}_t, \Sigma)\frac{\mathcal{N}(\mathbf{s}_t; \hat{\mathbf{m}}_t, \hat{\mathbf{V}}_t)}{c_t\hat{\alpha}_t(\mathbf{s}_t)} \quad (63)$$

which is virtually the same as the backward equation above. Proceeding as before:

$$p(\mathbf{s}_{t-1}, \mathbf{s}_t | \mathbf{x}_1..\mathbf{x}_T) = \mathcal{N}(\mathbf{s}_{t-1}; \mathbf{m}_{t-1} + \mathbf{J}_{t-1}(\mathbf{s}_t - \mathbf{A}\mathbf{m}_{t-1}), \mathbf{V}_{t-1} - \mathbf{J}_{t-1}\mathbf{A}\mathbf{V}_{t-1})\mathcal{N}(\mathbf{s}_t; \hat{\mathbf{m}}_t, \hat{\mathbf{V}}_t)$$

Therefore $\mathbf{s}_{t-1}$ and $\mathbf{s}_t$ are jointly normal:

$$\begin{bmatrix} \mathbf{s}_{t-1} \\ \mathbf{s}_t \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \hat{\mathbf{m}}_{t-1} \\ \hat{\mathbf{m}}_t \end{bmatrix}, \begin{bmatrix} \hat{\mathbf{V}}_{t-1} & \mathbf{J}_{t-1}\hat{\mathbf{V}}_t \\ \hat{\mathbf{V}}_t\mathbf{J}_{t-1}^{\mathrm{T}} & \hat{\mathbf{V}}_t \end{bmatrix}) \tag{64}$$

The recursion reported by Ghahramani & Hinton (1996) to compute the cross-covariance between $\mathbf{s}_{t-1}$ and $\mathbf{s}_t$ is not required; the answer is simply $\mathbf{J}_{t-1}\hat{\mathbf{V}}_t$.

Every Linear Dynamical System has the property that the entire set of state variables and measurement variables $\{\mathbf{s}_1..\mathbf{s}_T, \mathbf{x}_1..\mathbf{x}_T\}$ is jointly Gaussian. In other words, the variables all form a long Gaussian random vector whose mean and covariance matrix can be computed in terms of $\mathbf{A}, \mathbf{C}, \Gamma$ and Σ. In principle, one could use this fact to compute any probability of interest; the forward-backward recursions just provide a particularly efficient way to do so.

Acknowledgements

I am indebted to Rosalind Picard and Aaron Bobick for improving the presentation and to Kenneth Russell for helpful discussions.

References

[1] Zoubin Ghahramani and Geoffrey E. Hinton. Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, University of Toronto, 1996. `http://www.cs.utoronto.ca/~zoubin/`.

[2] Thomas P. Minka. Old and new matrix algebra useful for statistics. `http://vismod.www.media.mit.edu/~tpminka/papers/tutorial.html`, 1997.

[3] Kevin P. Murphy. Filtering, smoothing and the junction tree algorithm. `http://www.cs.berkeley.edu/~murphyk/publ.html`, 1998.

[4] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Francisco, CA, 1988.

[5] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, Feb. 1989.