9.1 Hidden Markov Models and the Kalman Filter

9.1.1 Introduction

Today we will derive the HMM and Kalman filter equations. We will show that these equations are valid by employing the conditional independence statements made by their graphs.

9.1.2 Hidden Markov Models

Last time we talked about HMMs and how they are clearly and simply described by a graph

\[
\begin{array}{cccccc}
Q_1 & Q_2 & Q_{t-1} & Q_t & Q_{t+1} & Q_{t+2} \\
X_1 & X_2 & X_{t-1} & X_t & X_{t+1} & X_{t+2} \\
\end{array}
\]

Figure 9.1: The HMM graphical model

From this, we can determine the joint probability distribution:

\[
p(Q_{1:T}, X_{1:T}) = \prod_t p(X_t|Q_t)p(Q_t|Q_{t-1}) \quad t \in 1: T
\]

Given this joint distribution, there are three different problems we might want to solve in the context of HMMs:

1. The filtering problem: find \( p(q_t|x_{1:t}) \),
2. The prediction problem: find \( p(q_t|x_{1:s}) \), where \( t > s \),
3. The smoothing problem: find \( p(q_t|x_{1:u}) \), where \( t < u \).

These very problems also arise in the context of the Kalman model (from which the names are derived, e.g., Kalman filtering). We will look at these in the more general context of the LG-HMM model in section 9.2.
9.1.3 The HMM Smoothing Problem

We will derive the algorithm for finding a solution to the HMM smoothing problem. The same procedure can be employed to solve any of these in the general HMM case. To reiterate, our strategy will be to use the HMM graphical model to determine the independence statements that we may exploit to make the computation tractable. First, let's show the HMM graph that we will use throughout the section (this differs from the previous graph only in the naming of the nodes)

![HMM Graph](image)

Figure 9.2: The HMM graphical model

By the definition of conditional probability (setting $u \equiv T$)

$$p(q_t|x_{1:T}) = \frac{p(q_t, x_{1:T})}{p(x_{1:T})}$$

As a basis for comparison, consider brute-force calculation of the numerator. We must marginalize out several variables

$$p(q_t, x_{1:T}) = \sum_{q_1} \sum_{q_2} \cdots \sum_{q_t} \sum_{q_{t+1}} \cdots \sum_{q_T} p(q_1:T, x_{1:T})$$

which requires $O(M^T)$ operations (where $M \equiv |Q|$). This is intractable for any realistic value of $M$. Even if $M = 2$ (so that $Q$ is a binary random variable), for $T = 1000$ we would have to perform $2^{1000}$ operations. Let’s rewrite the smoothing distribution as follows

$$p(q_t|x_{1:T}) = \frac{p(x_{t+1:T}|q_t)p(x_t|q_t)}{p(x_{1:T})}$$

$$= \frac{p(x_{t+1:T}|q_t)p(x_t|q_t)}{p(x_{1:T})}$$

$$= \frac{p(x_{t+1:T}|q_t)p(x_t|q_t)}{\alpha(q_t)\beta(q_t)\gamma(q_t)p(x_{1:T})}$$

Now consider the HMM graph. Applying Bayes ball, we see that the conditional independence statement $x_{t+1:T} \perp x_{1:t} | q_t$ holds, so we can write (9.2) as

$$p(q_t|x_{1:T}) = \frac{p(x_{t+1:T}|q_t)p(x_t|q_t)}{p(x_{1:T})}$$

$$= \frac{p(x_{t+1:T}|q_t)p(x_t|q_t)}{p(x_{1:T})}$$

$$= \frac{\alpha(q_t)\beta(q_t)}{p(x_{1:T})}$$
where $\alpha(q_t) \triangleq p(x_{1:t}, q_t)$ and $\beta(q_t) \triangleq p(x_{t+1:T}|q_t)$. Therefore $\beta(q_t)\alpha(q_t) = p(x_{1:T}, q_t) \forall t$. Since we can write $p(x_{1:T}) = \sum_{q_t} \beta(q_t)\alpha(q_t) \forall t$, the end result (so far) is

$$p(q_t|x_{1:T}) = \frac{\beta(q_t)\alpha(q_t)}{\sum_{q_t} \beta(q_t)\alpha(q_t)} \quad (9.3)$$

It may appear that we haven’t accomplished much, but it is crucial to observe that with some more work on (9.3) we can avoid having to perform (9.1) (an $O(M^T)$ calculation), since we can define $\alpha$ and $\beta$ recursively.

### 9.1.4 $\alpha$-recursion

From above,

$$\alpha(q_{t+1}) \triangleq p(x_{1:t+1}, q_{t+1}) = p(x_{t+1}|q_{t+1}, x_{1:t})p(x_{1:t}, q_{t+1})$$

By applying Bayes ball to the graph, we see that $x_{t+1} \perp\!
\!\!\!\perp x_{1:t}|q_{t+1}$, so we can simplify this expression by removing the formal dependency on $x_{1:t}$:

$$\alpha(q_{t+1}) = p(x_{t+1}|q_{t+1})p(x_{1:t}, q_{t+1}) \quad (9.4)$$

Now let’s re-introduce $q_t$ into this equation by “unmarginalizing in” $q_t$. The RHS of (9.4) becomes

$$p(x_{t+1}|q_{t+1}) \sum_{q_t} p(x_{1:t}, q_t, q_{t+1})$$

$$= p(x_{t+1}|q_{t+1}) \sum_{q_t} p(q_{t+1}|x_{1:t}, q_t)p(x_{1:t}, q_t)$$

The graph further indicates that $q_{t+1} \perp\!
\!\!\!\perp x_{1:t}|q_t$, so

$$\alpha(q_{t+1}) = p(x_{t+1}|q_{t+1}) \sum_{q_t} p(q_{t+1}|q_t) \frac{p(x_{1:t}, q_t)}{\alpha(q_t)}$$

where $(a_{q_t,q_{t+1}})$ is the HMM transition matrix. Why is this valuable? One step in the $\alpha$-recursion requires $O(M^2)$ operations, since there are $M$ different values of $\alpha(q_t)$, and each requires $M$ operations. Since there are $T$ timesteps, the overall complexity is $O(TM^2) \ll O(M^T)$.

### 9.1.5 $\beta$-recursion

Similarly, we can derive a recursion for $\beta$,

$$\beta(q_t) = p(x_{t+1:T}|q_t) = \sum_{q_{t+1}} p(x_{t+1:T}, q_{t+1}|q_t)$$

$$= \sum_{q_{t+1}} p(x_{t+1}|x_{t+2:T}, q_{t+1}, q_t)p(x_{t+2:T}, q_{t+1}|q_t)$$
The graph states that $x_{t+1} \perp \{x_{t+2:T}, q_t\} | q_{t+1}$, so we can delete $x_{t+2:T}$ and $q_t$ from the sum on the RHS, yielding

$$\sum_{q_{t+1}} p(x_{t+1}|q_{t+1}) p(x_{t+2:T}|q_{t+1}, q_t)$$

$$= \sum_{q_{t+1}} p(x_{t+1}|q_{t+1}) p(x_{t+2:T}|q_{t+1}, q_t) p(q_{t+1}|q_t)$$

(9.5)

Furthermore, $x_{t+2:T} \perp q_t | q_{t+1}$, so (9.5) simplifies to

$$\sum_{q_{t+1}} p(x_{t+1}|q_{t+1}) p(x_{t+2:T}|q_{t+1})$$

$$= \sum_{q_{t+1}} p(x_{t+1}|q_{t+1}) p(q_{t+1}|q_t)$$

$$\sum_{q_{t+1}} p(q_{t+1}|q_t)$$

$$\sum_{q_{t+1}} p(q_{t+1}|q_t, x_{1:T})$$

which is a recursive formulation of $\beta$ that is $O(TM^2)$.

$\alpha$ and $\beta$ are called the forward procedure and backward procedure, respectively. $\alpha$ is "forward" in the sense that it begins at $t = 1$ and increases in $t$; likewise $\beta$ is a "backward" procedure in that it begins at $T$ and decreases in $t$.

9.1.6 $\alpha$-\(\gamma\) recursion

The $\alpha$-$\beta$ recursion requires a forward pass and a backward pass through the observed data $x$. This is a limiting factor, since the entire history of observations must be kept in memory. There is a different algorithm called $\alpha$-$\gamma$ recursion that overcomes this limitation. We define

$$\gamma(q_t) \triangleq p(q_t|x_{1:T})$$

We can write $\gamma$ as an equation that does not involve the data $x_{1:T}$ as follows

$$\gamma(q_t) = \sum_{q_{t+1}} p(q_t, q_{t+1}|x_{1:T})$$

$$= \sum_{q_{t+1}} p(q_t|q_{t+1}, x_{1:T}) p(q_{t+1}|x_{1:T})$$

We need to show that we can compute $p(q_t|q_{t+1}, x_{1:T})$ recursively. From the graph, we can see that $q_t \perp x_{t+1:T} | q_{t+1}$, so we can write $\gamma$ as

$$\gamma(q_t) = \sum_{q_{t+1}} p(q_t|q_{t+1}, x_{1:T}) \gamma(q_{t+1})$$

$$= \frac{\sum_{q_{t+1}} p(q_t|q_{t+1}, x_{1:T}) \gamma(q_{t+1})}{\sum_{q_t} p(q_{t+1}|q_t, x_{1:T}) p(q_t, x_{1:T})}$$

$$= \frac{\sum_{q_{t+1}} p(q_t|q_{t+1}, x_{1:T}) p(q_{t+1}, x_{1:T})}{\sum_{q_t} p(q_{t+1}|q_t, x_{1:T}) p(q_t, x_{1:T})} \gamma(q_{t+1})$$

$$\sum_{q_{t+1}} p(q_{t+1}|q_t, x_{1:T}) p(q_t, x_{1:T}) \gamma(q_{t+1})$$
where we used the definition of conditional probability in the second equality. Furthermore, the graph states that $q_{t+1} \perp \perp x_{1:t} | q_t$, and replacing $p(q_t, x_{1:t})$ with $\alpha(q_t)$ we have

$$
\gamma(q_t) = \sum_{q_{t+1}} p(q_{t+1}|q_t) \alpha(q_t) \gamma(q_{t+1})
$$

9.1.7 $\xi$

There is another quantity, $\xi$, needed for computing the transition probabilities in HMMs. It is defined by

$$
\xi(q_t, q_{t+1}) \triangleq p(q_t, q_{t+1}|x_{1:T})
$$

We could determine a recursion on $\xi$, however, it is simpler to use $\alpha$ and $\beta$ to derive $\xi$.

$$
\xi(q_t, q_{t+1}) \equiv p(q_t, q_{t+1}|x)
$$

$$
= p(x|q_t, q_{t+1}) p(q_{t+1}|q_t) p(q_t) \frac{1}{p(x)}
$$

$$
= p(x_0:t, x_{t+1:T}|q_t, q_{t+1}) p(q_{t+1}|q_t) p(q_t) \frac{1}{p(x)}
$$

$$
= p(x_0:t|q_t) p(x_{t+1:T}|q_{t+1}) p(x_{t+2:T}|q_{t+1}) p(q_{t+1}|q_t) \frac{1}{p(x)}
$$

$$
= \alpha(q_t) p(x_{t+1}|q_{t+1}) \beta(q_{t+1}) a_{q_t, q_{t+1}} \frac{1}{p(x)}
$$

It is now easy to calculate $\xi$ given $\alpha$ and $\beta$ or $\alpha$ and $\gamma$.

An important point to bear in mind is that most explanations of this procedure do not clearly state what specific independence statements you are allowed (or not allowed) to make. To truly understand what is occurring within a HMM, you must understand how independence statements allow you to go from $O(M^T)$ operations to $O(TM^2)$ operations.

9.1.8 HMMs and the EM procedure

Recall the maximum likelihood parameter estimation problem:

$$
\lambda^* = \arg\max_\lambda p_\lambda(x) = \arg\max_\lambda \sum_q p(q, x)
$$

where $\lambda = \{A, \pi, \theta\}$ is a set of parameters with $A$ the transition probability matrix, $\pi$ the state initial probabilities, and $\theta$ the parameters of the observation distribution $p_\theta(x_t|q_t)$. To perform this optimization directly is computationally very expensive. The EM algorithm can significantly mitigate this cost.

We can write the auxiliary function as

$$
Q(\lambda, \lambda^p) = \sum_{q_{1:T}} p(x_{1:T}, q_{1:T}|\lambda^p) \log p(x_{1:T}, q_{1:T}|\lambda)
$$
Using the independence statements the graph makes, we can write

\[ p(x_{1:T}, q_{1:T}|\lambda) = p(q_0) \left( \prod_t p(x_t|q_t) \right) \left( \prod_t p(q_t|q_{t-1}) \right) \]

Let’s also write \( p(x_{1:T}, q_{1:T}|\lambda^p) \) as \( p(x_{1:T}|q_{1:T}, \lambda^p)p(x_{1:T}, \lambda^p) \). Note that the term \( p(x_{1:T}, \lambda^p) \) is functionally independent of the sum over \( q_{1:T} \), so we can ignore it during the optimization. Applying the logarithm and putting everything together yields

\[ Q(\lambda, \lambda^p) = \sum_{q_t} p(x_{1:T}, q_{1:T}|\lambda^p) \log p(q_0) + \sum_{q_t} p(x_{1:T}, q_{1:T}|\lambda^p) \sum_t \log p(x_t|q_t) + \sum_{q_t} p(x_{1:T}, q_{1:T}|\lambda^p) \sum_t \log p(q_t|q_{t-1}) \]

We have decoupled the auxiliary function into three separately optimizable components. The first term can be optimized via simple counting. The third term in the equation depends on the form of the observation distribution. If the observation distribution is a mixture of Gaussians, then the form of optimization is very similar to what has been covered in a previous lecture.

### 9.2 The Kalman Filter

Recall the linear Gaussian model.

\[ Y_n \quad \text{X}_n \]

Figure 9.3: The Linear Gaussian Model

When we’re employing factor analysis, all random variables in the model are continuous and jointly Gaussian, and the observations \( y_n \) are related to the hidden variables \( x \) by \( y = \Lambda x + u + \mu \), where \( \Lambda \) is a factor loading matrix, \( x \) is a common factor, and \( u \) is a specific factor. The GM for factor analysis is shown below:

\[ Y_n \quad \text{X}_n \]

Figure 9.4: The Factor Analysis Model

We stated that this was the graphical model for factor analysis, PCA, and other statistical techniques, depending on the assumptions made regarding the matrices, transformations, etc. Factor analysis in the current context is simply one particular set of assumptions.

We now generalize the linear Gaussian model in the same manner. Beginning with a set of linear conditional Gaussian models, we add time edges to the hidden variables. We will refer to this model as LG-HMM (where LG stands for linear Gaussian).
When all random variables in the LG-HMM model are continuous and jointly Gaussian, the LG-HMM is the graphical model for a Kalman filter. We will see that the inference equations in this model are very similar to the \( \alpha \)-recursion.

First, note that all random variables are now vectors. The state update equations for the hidden variables are described by the recursion:

\[
x_{t+1} = Ax_t + Gw_t
\]

Here \( x_t \) is the hidden state, and \( w_t \) is the “noise” term. The model assumes that \( w_t \sim N(0, Q) \), and that \( w_s \perp \perp w_t \), for \( s \neq t \). The joint distribution

\[
p(x_{1:T}, y_{1:T}) \sim N(\mu, \Sigma)
\]

is such that the particular set of independence assumptions made by the LG-HMM hold. Computing the conditional covariance matrix, we see that

\[
x_{t+1} | x_t \sim N(Ax_t, GQG^T)
\]

The dependence between \( y \) and \( x \) is expressed as follows:

\[
y_t = Cx_t + v_t
\]

where \( v_t \sim N(0, R) \), and hence the conditional probability distribution \( p(y_t | x_t) \) is also Gaussian, i.e., \( y_t | x_t \sim N(Cx_t, R) \). Finally, the recursive update equations are initialized as \( x_0 \sim N(0, \Sigma_0) \). Another graphical representation of Kalman filter that makes explicit the dependence on \( w_t \) is illustrated below:

### 9.2.1 Properties of the LG-HMM

1. \( Ex_t = A^T \cdot Ex_0 = 0 \)
2. \( COV(x_t) = A\Sigma_t A^T + GQG^T \) (This is known as the Lyapunov equation.)
Proof.

\[ \text{COV}(x_t) \equiv \Sigma_t = E [(x_t - Ex_t)(x_t - Ex_t)^T] = Ex_t x_t^T \]
\[ = E [(Ax_{t-1} + Gw_{t})(Ax_{t-1} + Gw_{t})^T] \]
\[ = A(E x_{t-1} x_{t-1}^T) A^T + G(E w_{t-1} w_{t-1}^T) G^T, \]

where the last equality follows from the fact that from the graph, \( x_t \perp \perp w_t \), and that several cross terms are identically zero. Since we assume that the noise terms have the same covariance, we have

\[ \text{COV}(x_t) = A \Sigma_t A^T + GQG^T \]

3. \( \text{COV}(x_t, x_{t+1}) = E [x_t (Ax_t + Gw_t)^T] = \Sigma_t A^T \)

There are a number of things we might want to efficiently compute from the joint distribution associated with the LG-HMM:

1. \( p(x_t | y_{0:t}) \) (filtering)
2. \( p(x_t | y_{0:T}) \) (smoothing)
3. \( p(x_t | y_{0:s}), \ s < t \) (prediction)

Kalman filtering is essentially computing (1) in an LG-HMM. We will now investigate the Kalman filtering operation in detail. To find the conditional distribution in \( p(x_t | y_{0:t}) \) where it is known that all random variables are jointly Gaussian, we must find conditional mean and conditional variance, since these are the sufficient statistics for the distribution. Specifically, we will derive a set of operations that produce the conditional mean

\[ \hat{x}_t | t \triangleq E [x_t | y_{0:t}] \]

and the conditional variance

\[ P_t | t \triangleq E [(x_t - \hat{x}_t | t)(x_t - \hat{x}_t | t)^T | y_{0:t}] \]

First we derive a (forward recursive) time update (where intuitively, something happened at time \( t \) but we haven’t received an observation \( y \) of it)

\[ p(x_t | y_{0:t}) \rightarrow p(x_{t+1} | y_{0:t}) \]

and then we derive a measurement update:

\[ p(x_{t+1} | y_{0:t}) \rightarrow p(x_{t+1} | y_{0:t+1}) \]

To find these we must compute two new conditional quantities. We’ll assume we have the sufficient statistics at time \( t \), and find a recursion for the statistics at time \( t + 1 \), i.e., \( (\hat{x}_t | t, P_t | t) \rightarrow (\hat{x}_{t+1} | t, P_{t+1} | t) \). For \( \hat{x}_{t+1} | t \)

\[ \hat{x}_{t+1} | t = E [x_{t+1} | y_{0:t}] \]
\[ = E [Ax_t + Gw_t | y_{0:t}] \]
\[ = AE [x_t | y_{0:t}] + GE [w_t] \quad (\text{since } w_t \perp \perp y_{0:t}) \]
\[ = A \hat{x}_t | t \]
Similarly, for \( P_{t+1|t} \)
\[
P_{t+1|t} = E \left[ (x_{t+1} - \hat{x}_{t+1|t})(x_{t+1} - \hat{x}_{t+1|t})^T | y_{0:t} \right]
\]
\[
= E \left[ (Ax_t + Gw_t - \hat{x}_{t+1|t})(Ax_t + Gw_t - \hat{x}_{t+1|t})^T | y_{0:t} \right]
\]
\[
= E \left[ (Ax_t - \hat{x}_{t|t})(Ax_t - \hat{x}_{t|t})^T | y_{0:t} \right] + E \left[ (Gw_t)(Gw_t)^T \right]
\]
\[
= AP_{t|t}A^T + GQG^T
\]
where the third equality follows from the fact that \( w_t \perp x_{1:t} \), and that several cross terms are identically zero. Thus, the time update is described by
\[
\hat{x}_{t+1|t} \sim N(Ax_{t|t}, AP_{t|t}A^T + GQG^T)
\]

Now we derive the measurement update \( \hat{y}_{t+1|t}, P_{t+1|t} \rightarrow (\hat{y}_{t+1|t+1}, P_{t+1|t+1}) \). We’ll use the following quantities (recall that all random variables are jointly Gaussian, so we require the conditional mean and conditional variance, since these are the sufficient statistics for the distribution):

1. \( E \left[ y_{t+1|t} | y_{0:t} \right] \)
2. \( COV(y_{t+1|t} | y_{0:t}) \)
3. \( COV(x_{t+1|t}, y_{t+1|t} | y_{0:t+1}) \)

For (1), using the notation defined for \( \hat{x} \),
\[
\hat{y}_{t+1|t} \equiv E \left[ y_{t+1|t} | y_{0:t} \right] = E \left[ Cx_{t+1} + v_{t+1} | y_{0:t} \right]
\]

At this point we would like to know if \( v_{t+1} \perp y_{0:t} \). Consider the graph representing the Kalman filter, where we have shown explicitly the dependence on \( v \).

![Figure 9.7: The Extended Kalman filter](image)

By applying Bayes ball to the v-structure involving \( v_{t+1}, y_{t+1}, \) and \( x_{t+1}, v_{t+1} \perp y_{0:t} \) is a valid independence statement made by the graph, so we can simplify our equations. Furthermore, since the random vectors \( v \) have mean zero, we have the result:
\[
\hat{y}_{t+1|t} = C\hat{x}_{t+1|t}
\]

Now let’s work on (2).
\[
COV(y_{t+1|t} | y_{0:t}) = E \left[ (y_{t+1} - \hat{y}_{t+1|t})(y_{t+1} - \hat{y}_{t+1|t})^T | y_{0:t} \right]
\]
\[
= E \left[ (Cx_{t+1} + v_{t+1} - \hat{y}_{t+1|t})(Cx_{t+1} + v_{t+1} - \hat{y}_{t+1|t})^T | y_{0:t} \right]
\]
\[
= CP_{t+1|t}C^T + R
\]
where we used the result derived for (1) and that fact that several cross terms are 0. $R$ is the standard covariance matrix.

We’ll derive the explicit expression for (3) next time.