Chapter 1

4F7 Statistical Signal Analysis

Please note that the margins of these notes can be used to check factual recall simply by covering up the right hand text.

1.1 State Space Models

1.1.1 Preliminaries

White Noise Sequences

The notation $W_n \sim \mathrm{WN}(0, Q_n)$ denotes a sequence of random vectors, $W_1, W_2, \ldots$ with zero mean and second moment

$$\mathbb{E}[W_n W_m^T] = \begin{cases} 0 & m \neq n \\ Q_n & m = n \end{cases} \tag{1.1}$$

State Space Model

A state space model for a time series, $\{Y_n\}$ consists of two equations:

$$Y_n = G_n X_n + V_n \qquad n = 1, 2, \ldots \qquad \text{(observation)} \tag{1.2}$$

$$X_{n+1} = F_n X_n + W_n \qquad n = 1, 2, \ldots \qquad \text{(evolution)} \tag{1.3}$$

where $V_n \sim \mathrm{WN}(0, R_n)$ and $\{G_n\}$ is some *deterministic* sequence of matrices. Similarly, $W_n \sim \mathrm{WN}(0, Q_n)$ and $\{F_n\}$ is some deterministic sequence of matrices. In certain models, observations may be noiseless.

Additional assumptions

Additionally, the following assumptions are made:

$$\mathrm{cov}[X_1, W_n] = 0 \quad n \geq 1 \tag{1.4}$$

$$\mathrm{cov}[X_1, V_n] = 0 \quad n \geq 1 \tag{1.5}$$

$$\mathrm{cov}[V_n, W_n] = 0 \quad n \geq 1, \; m \geq 1 \tag{1.6}$$

Note that these assumptions are made on **correlation, not independence**. Note that $\mathrm{cov}(X, Y) = \mathbb{E}[XY^T] - \mathbb{E}[X]\mathbb{E}[Y]^T$. For example, $X \sim \mathcal{N}(0, 1)$ and $Y = |X|$ are not independent, but they are uncorrelated.

It is possible to express X_n as a linear function of X_1 and $\{W_i\}_{i=1}^{n-1}$:

$$X_n = \Big(\prod_{i=1}^{n-1} F_i\Big) X_1 + \sum_{i=1}^{n-1} W_i \prod_{j=i}^{n-1} F_i \tag{1.7}$$

The state at time n is a linear function of the initial state and driving noise sequence. Using this form of X_n, it is possible to prove the following facts:

Try proving these yourself!

$$\mathbb{E}[X_m W_n^T] = 0 \qquad m \leq n \tag{1.8}$$

$$\mathbb{E}[Y_m W_n^T] = 0 \qquad m \leq n \tag{1.9}$$

$$\mathbb{E}[X_m V_n^T] = 0 \qquad \forall m, n \tag{1.10}$$

$$\mathbb{E}[Y_m V_n^T] = 0 \qquad m < n \tag{1.11}$$

In situations where there is no mention of a hidden state process, if a time series Y_n can be expressed through some hidden state process, then this time series has a state-space representation.

1.1.2 Linear Prediction

Linear Prediction

Note: for this section, a scalar valued state-space model is used.

Given random variables $Y_1, \dots, Y_n$ and some random variable X which we seek to estimate, linear predictors of X using $Y_1, \dots, Y_n$ take the form:

$$\hat{X} = h_0 + \sum_{i=1} h_i Y_i \tag{1.12}$$

for some constants $h_0, \dots, h_n$. The best linear predictor is the predictor with coefficients $h_0^*, \dots, h_n^*$ which satisfy:

$$h_0^*, \dots, h_n^* = \arg \min_{(h_0,\dots,h_n)} \mathbb{E}\left[\left(h_0 + \sum_{i=1} h_i Y_i - X\right)^2\right] \tag{1.13}$$

The best linear predictor (i.e. the predictor with the above coefficients) is denoted as $K[\cdot|\cdot]$ i.e. a function of two arguments.

Try proving this yourself!

By differentiating the cost function with respect to the filter parameter, it is possible to show that:

$$h_0 = \mathbb{E}[X] - \sum_i h_i \mathbb{E}[Y_i] \tag{1.14}$$

$$0 = \mathbb{E}[\Big\{\Big(\sum_i h_i(Y_i - \mathbb{E}[Y_i])\Big) - (X - \mathbb{E}[X])\Big\}(Y_k - \mathbb{E}[Y_k])] \quad \forall k \tag{1.15}$$

Note that in the proof, the additional term of $\mathbb{E}[Y_i]$ can be added as the expectation of terms corresponding to this term are zero. Writing $\mathbf{p} = \begin{bmatrix}\text{cov}[X, Y_i] & \cdots & \text{cov}[X, Y_n]\end{bmatrix}^T$, Eq. 1.15 can be written as:

$$\mathbf{\Sigma}_Y \mathbf{h} = \mathbf{p}. \tag{1.16}$$

The Kalman Filter

Therefore, the best linear predictor is of the form:

$$K[X|Y_1, \dots, Y_n] = \mathbb{E}[X] + \sum_{i=1}^{N} h_i(Y_i - \mathbb{E}[Y_i]) \tag{1.17}$$

Properties of the Kalman Filter

where $\mathbf{\Sigma}_Y \mathbf{h} = \mathbf{p}$. This has the following properties:

- The error of the estimate has zero mean.

$$\mathbb{E}[X - K[X|Y_{1:n}]] = 0 \tag{1.18}$$

- The error is orthogonal to all Y_i.

$$\mathbb{E}[(X - K[X|Y_{1:n}])Y_i] = 0 \tag{1.19}$$

  This follows directly from Eq. 1.15, noting that the term corresponding to $\mathbb{E}[Y_i]$ is zero.

Prove this property, and the below properties.

- If $\text{cov}[Y_i, Y_n] = 0$ for $i < n$, then:

$$K[X|Y_{1:n}] = K[X|Y_{1:n-1}] + K[X|Y_n] - \mathbb{E}[X] \tag{1.20}$$

  Note: proof hinges on the condition ensuring that the coefficients do not change.

- Linearity:

$$K[aX + bU + c|Y_{1:n}] = aK[X|Y_{1:n}] + bK[U|Y_{1:n}] + c \tag{1.21}$$

- If $\tilde{\mathbf{Y}} = \mathbf{CY} + \mathbf{b}$ where $\tilde{C}$ is an invertible matrix, then:

$$K[X|Y_{1:n}] = K[X|\tilde{Y}_{1:n}] \tag{1.22}$$

  This can be shown by showing that the filter coefficients obtained are effectively the same in both cases.

- If $\text{cov}[U, Y_i] = 0$, then:

$$K[U|Y_{1:n}] = \mathbb{E}[U] \tag{1.23}$$

- $K[Y_i|Y_{1:n}] = Y_i$.

1.1.3 Kalman Filtering

Objective

The objective is to compute the filter, $K[X_n|Y_{1:n}]$, recursively in time for a state-space model:

$$Y_n = g_n X_n + V_n \tag{1.24}$$
$$X_{n+1} = f_n X_n + W_n \tag{1.25}$$

for $n = 1, 2, \dots$ with $V_n \sim \text{WN}(0, r_n)$ and $W_n \sim \text{WN}(0, q_n)$. Additionally,

$$\text{cov}(X_1, W_n) = \text{cov}(X_1, V_n) = \text{cov}(V_m, W_n) = 0$$

for all $n \geq 1$ and $m \geq 1$. To derive the filter, it is assumed that $K[X_n|Y_{1:n}]$ has already been calculated.

Prediction
Complete Proof

For the prediction step, we seek to calculate $K[X_{n+1}|Y_{1:n}]$. It is straightforward to show that:

$$K[X_{n+1}|Y_{1:n}] = f_n K[X_n|Y_{1:N}] \tag{1.26}$$

Similarly, it is easy to show that:

$$K[Y_{n+1}|Y_{1:N}] = g_{n+1} f_n K[X_n|Y_{1:N}] \tag{1.27}$$

Prediction MSE Update

The mean square error is also updated sequentially. Denote the mean square error as $\sigma_n = \mathbb{E}[(X_n - K[X_n|Y_{1:n}])^2]$. Then

$$\begin{aligned}
\underbrace{\mathbb{E}[(X_{n+1} - K[X_{n+1}|Y_{1:n}])^2]}_{\tilde{\sigma}_{n+1}} &= \mathbb{E}[(f_n X_n + W_n - f_n K[X_n|Y_{1:n}])^2] \\
&= f_n^2 \mathbb{E}[(X_n - K[X_n|Y_{1:n}])^2] + \mathbb{E}[W_n^2] + 2 f_n \underbrace{\mathbb{E}[(X_n - K[X_n|Y_{1:n}]) W_n]}_{0} \\
&= f_n^2 \sigma_n + q_n \qquad (1.28)
\end{aligned}$$

Why is the final term zero?

i.e. the mean square error is inflated by q_n due to the additional noise and scaled by f_n. The final term is zero because $X_n - K[X_n|Y_{1:n}]$ is linear in $(X_1, W_1, \dots W_{n-1}, V_1, \dots, V_n)$. The expected value of this term multiplied by W_n is zero. Together, Equations 1.26 and 1.28 make up the Kalman prediction step for a state-space model.

Derivation of the Kalman Update Step

The *innovations* are defined as:

$$I_{n+1} = Y_{n+1} - K[Y_{n+1}|Y_{1:n}] = Y_{n+1} - g_{n+1} f_n K[X_n|Y_{1:n}] \tag{1.29}$$

these can be thought of as the unpredictable part of Y. Note that $\text{cov}(I_{n+1}, Y_i) = 0, i < n+1$ as the error is orthogonal to the inputs. Additionally, note that $Y_{n+1} = I_{n+1} + K[Y_{n+1}|Y_{1:n}]$, meaning that there is an invertible matrix and bias, $\mathbf{C}$ and $\mathbf{b}$ such that:

Find these matrices

$$\begin{bmatrix} Y_{1:n} & I_{n+1} \end{bmatrix}^T = \mathbf{C} \begin{bmatrix} Y_{1:n} & Y_{n+1} \end{bmatrix}^T + \mathbf{b}$$

Therefore,

$$K[\cdot|Y_{1,n+1}] = K[\cdot|Y_{1:n}, I_{n+1}] = K[\cdot|Y_{1:n}] + K[\cdot|I_{n+1}] - \mathbb{E}[\cdot] \tag{1.30}$$

Thus

$$K[X_{n+1}|Y_{1:n+1}] = \underbrace{K[X_{n+1}|Y_{1:n}]}_{\text{prediction}} + K[X_{n+1}|I_{n+1}] - \mathbb{E}[X_{n+1}] \tag{1.31}$$

Now, noting that $\mathbb{E}[I_n] = 0$,

$$K[X_{n+1}|I_{n+1}] = \mathbb{E}[X_{n+1}] + \frac{\mathbb{E}[X_{n+1}I_{n+1}]}{\mathbb{E}[I_{n+1}^2]} I_{n+1} \tag{1.32}$$

Recall and expand the definition of I_{n+1}:

$$\begin{aligned} I_{n+1} &= g_{n+1}X_{n+1} + V_{n+1} - K[g_{n+1}X_{n+1} + V_{n+1}|Y_{1:n}] \\ &= g_{n+1}(X_{n+1} - K[X_{n+1}|Y_{1:n}]) + V_{n+1} \end{aligned} \tag{1.33}$$

Therefore, the denominator is

$$\begin{aligned} \mathbb{E}[I_{n+1}^2] &= g_{n+1}^2 \underbrace{\mathbb{E}[(X_{n+1} - K[X_{n+1}|Y_{1:n}])^2]}_{\bar{\sigma}_{n+1}} + \mathbb{E}[V_{n+1}^2] + \underbrace{2\mathbb{E}[(X_{n+1} - K[X_{n+1}|Y_{1:n}])V_{n+1}]}_{0} \\ &= g_{n+1}^2(f_n^2\sigma_n + q_n) + r_{n+1} \end{aligned} \tag{1.34}$$

Cross-Terms - why zero?

The cross term is 0 because X_{n+1} is a linear function of $(X_1, W_1, \ldots, W_n)$ and $K[X_{n+1}|Y_{1:n}]$ is a linear function of $(X_1, W_1, \ldots, W_n, V_1, \ldots, V_n)$. The numerator is:

$$\mathbb{E}[X_{n+1}I_{n+1}] = g_{n+1}\mathbb{E}[X_{n+1}(X_{n+1-K[X_{n+1}|Y_{1:n}]})] + \underbrace{\mathbb{E}[X_{n+1}V_{n+1}]}_{0} \tag{1.35}$$

$$= g_{n+1}\underbrace{\mathbb{E}[(X_{n+1} - K[X_{n+1}|Y_{1:n}])^2]}_{\bar{\sigma}_{n+1}} - g_{n+1}\overbrace{\mathbb{E}[K[X_{n+1}|Y_{1:n}]\underbrace{(X_{n+1} - K[X_{n+1}|Y_{1:n}])}_{\text{prediction error}}]}^{0} \tag{1.36}$$

Why is the final term zero?

Kalman Update Step

Since the prediction error is zero mean and orthogonal to all variables used to predict it. Therefore, the new prediction is:

$$\underbrace{K[X_{n+1}|Y_{1:n+1}]}_{\hat{X}_{n+1}} = \underbrace{K[X_{n+1}|Y_{1:n}]}_{\bar{X}_{n+1}} + \frac{g_{n+1}\bar{\sigma}_{n+1}}{g_{n+1}^2\bar{\sigma}_{n+1} + r_{n+1}} I_{n+1} \tag{1.37}$$

The filter's mean squared error must now be calculated. $\sigma_n = \mathbb{E}[(X_{n+1} - \hat{X}_{n+1})^2]$. Subtracting the prediction from Eq. 1.31 and rearranging yield:

$$\begin{aligned} X_{n+1} - \bar{X}_{n+1} &= (X_{n+1} - \hat{X}_{n+1}) + \underbrace{(\mathbb{E}[X_{n+1}] - K[X_{n+1}|I_{n+1}])}_{\alpha} \\ \bar{\sigma}_{n+1} &= \sigma_{n+1} + \mathbb{E}[\alpha^2] + \underbrace{2\mathbb{E}[\alpha(X_{n+1} - \hat{X}_{n+1})]}_{0} \\ &= \sigma_{n+1} + \frac{\mathbb{E}[X_{n+1}I_{n+1}]^2}{\mathbb{E}[I_{n+1}^2]} \end{aligned} \tag{1.38}$$

the cross term is zero because the prediction error is orthogonal to the terms used to define it, and the final equality is by Eq. 1.32. Using the previous expressions for these terms yields the update for the mean square error:

Kalman MSE Update

$$\sigma_{n+1} = \frac{\bar{\sigma}_{n+1}r_{n+1}}{g_{n+1}^2\bar{\sigma}_{n+1} + r_{n+1}} \tag{1.39}$$

1.2 Hidden Markov Models

Define HMM

A hidden Markov model is comprised of two stochastic processes, a hidden state, X, and observation process, Y, which are both Markov i.e.

$$p(x_k|x_0,\dots,x_{k-1}) = p(x_k|x_{k-1}) = f(x_{k-1},x_k) \tag{1.1}$$

$$p(x_k|x_0,\dots,x_{k-1},y_0,\dots,y_{k-1}) = p(y_k|x_k) = g(x_k,y_k) \tag{1.2}$$

$$\begin{tikzcd}
 & Y_{t-1} & Y_t & \\
X_{t-2} \arrow[r, "f(X_{t-2},x_{t-1})"'] & X_{t-1} \arrow[u, "g(X_{t-1},y_{t-1})"'] \arrow[r, "f(X_{t-1},x_t)"'] & X_t \arrow[u, "g(X_t,y_t)"'] \arrow[r, "f(X_t,x_{t+1})"'] & X_{t+1}
\end{tikzcd}$$

Figure 1.1: Evolution of RVs in a HMM

Note: HMMs are significantly more 'powerful' than state space models; there is no underlying assumption that evolution is linear.

Joint Density

The joint density of $(X_0,Y_0,\dots,X_k,Y_k)$ can be written straightforwardly:

$$p(x_0,y_0,\dots,x_k,y_k) = \prod_{i=0}^{k} p(y_i|x_i,x_0,\dots x_{i-1},y_0,\dots y_{i-1})p(x_i|x_0,\dots x_{i-1},y_0,\dots y_{i-1})$$

$$= p(x_0)g(x_k,y_k)\prod_{i=0}^{k} g(x_i,y_i)f(x_i,x_{i+1}) \tag{1.3}$$

Examples of HMM models include:

1. In the Gaussian state-space model, the driving noises are Gaussian: *Gaussian State Space Model*

$$X_{k+1} = aX_k + bW_{k+1} \qquad Y_k = cX_k + dV_k \tag{1.4}$$

for $k = 0,1,\dots$ where $V_k \sim \mathcal{N}(0,1)$, $W_k \sim \mathcal{N}(0,1)$ and $X_0 \sim \mathcal{N}(\bar{\mu}_0,\bar{\sigma}_0)$

2. The stochastic volatility model is: *Stochastic Volatility Model*

$$X_{k+1} = aX_{k-1} + bW_k \qquad Y_k = c\exp\frac{X_k}{2}V_k \tag{1.5}$$

for $k = 0,1,\dots$ where $V_k \sim \mathcal{N}(0,1)$, $W_k \sim \mathcal{N}(0,1)$ and $X_0 \sim \mathcal{N}(\bar{\mu}_0,\bar{\sigma}_0)$

3. A discrete valued Markov change, observed in Gaussian noise. $X_k \in S = \{1,\dots,n\}$ with:

$$P(X_k = i_k|X_{k-1} = i_{k-1}) = P_{i_{k-1},i_k} \tag{1.6}$$

P is the transition probability matrix with the sum of each row being 1 (for correct normalisation). Given $X_k = i_k$, the observed process is:

$$Y_k = c_{i_k} + d_{i_k}V_k \tag{1.7}$$

for $k = 0,1,\dots$ where $V_k \sim \mathcal{N}(0,1)$. $c_1,d_1,\dots,c_n,d_n$ are real valued constants.

Inference Objectives for HMMs

There are multiple possible inference objectives for the HMM model:

1. **Filtering:** compute the density of the current hidden state.

$$p(x_k|y_0,\dots,y_k) \tag{1.8}$$

2. **Prediction:** compute the future hidden state.

$$p(x_{k+m}|y_0,\dots,y_k) \tag{1.9}$$

3. **Smoothing:** improve estimates of previous hidden states.

$$p(x_{k-m}|y_0,\dots,y_k) \tag{1.10}$$

HMMs have exact solutions for discrete values of X_k and for continuous valued X_k if the Gaussian state-space model is used.

Exact Computations for the Discrete HMM

For the discrete valued HMM, the exact sequential computations are:

$$p(i_{k+1}|y_{0:k}) = [\pi_k^T P]_{i_{k+1}} \tag{1.11}$$

$$\pi_{k+1}^T = \frac{\pi_k^T P B_{k+1}}{\pi_k^T P B_{k+1}\mathbf{1}} \tag{1.12}$$

where pi_k is the column vector of $p(x_k = i_k|y_{0:k})$, $\mathbf{1}$ is a column vector of ones and B_k is defined as

$$B_k = \begin{bmatrix} g(1,y_k) & & \\ & \ddots & \\ & & g(n,y_k) \end{bmatrix} \tag{1.13}$$

Bayesian Kalman Filter Update Equations

Derive this

In general, assuming that $p(x_k|y_{0:k})$ is known, the Kalman filter equations are:

$$p(x_{k+1}|y_{0:k}) = \int p(x_k|y_{0:k}) f(x_k, x_{k+1})\, dx_k \tag{1.14}$$

$$p(x_{k+1}|y_{0:k+1}) = \frac{p(x_{k+1}|y_{0:k}) g(x_{k+1}, y_{k+1})}{\int p(x_{k+1}|y_{0:k}) g(x_{k+1}, y_{k+1}) dx_{k+1}} \tag{1.15}$$

Bayesian Kalman Filter - Prove the Update Equations

For the special case of the Gaussian state space model, the Kalman filter prediction equation, assuming that $p(x_k|y_{0:k}) = \mathcal{N}(\mu_k, \sigma_k)$, is

$$\bar{\mu}_{k+1} = a\mu_k \qquad \bar{\sigma}_{k+1} = a^2\sigma_k + b^2 \tag{1.16}$$

The update equations are then:

$$\mu_{k+1} = \bar{\mu}_{k+1} + \frac{c\bar{\mu}_{k+1}}{c^2\bar{\mu}_{k+1} + d}(y_{k+1} - c\bar{\mu}_{k+1}) \tag{1.17}$$

$$\sigma_{k+1} = \bar{\sigma}_{k+1} - \frac{c^2\bar{\sigma}_{k+1}^2}{c^2\bar{\sigma}_{k+1}^2 + d^2} \tag{1.18}$$

Thus with an initial density for the first hidden state, the *Bayesian Kalman Filter* equations are very straightforward to implement.

1.3 Importance Sampling

Let $\pi(x)$ be a probability density function with $x \in \mathbb{R}^n$. Often, we want to compute integrals with respect to π:

$$\mathbb{E}_\pi[h(X)] = \int h(x)\pi(x)\, dx \tag{1.1}$$

When do we use importance sampling?

for some function $h : \mathbb{R}^n \to \mathbb{R}$. In general, **it may not be possible to sample from π**, meaning that a direct Monte Carlo estimate cannot be computed.

Importance Sampling

Important Sampling estimates $\mathbb{E}_\pi[h(X)]$ using *Monte Carlo* using some probability density function, $q(x)$, from which we can sample.

$$\mathbb{E}_\pi[h(X)] \simeq \frac{1}{N}\sum_{i=1}^{N} h(X^{(i)}) \underbrace{\frac{\pi(X^{(i)})}{q(X^{(i)})}}_{w^{(i)}} \quad \text{where } X^{(i)} \overset{iid}{\sim} q \tag{1.2}$$

Note that this is an unbiased estimate. $w^{(i)}$ is the importance weight.

Show Unbiased

Self-Normalised Importance Sampling

In general, it cannot be assumed that the target density normalises, as often it is difficult to calculate the correct normalising constant. Let $\pi(x) \geq 0\ \forall x$. Then, the *Self Normalised Importance Sampling* estimate is constructed as follows:

$$\mathbb{E}_{\pi^*}[h(X)] \simeq \frac{\sum_{i=1}^{N} h(X^{(i)}) w^{(i)}}{\sum_{i=1}^{N} w^{(i)}} \quad \text{where } X^{(i)} \overset{iid}{\sim} q \tag{1.3}$$

Note that the denominator uses importance sampling (with the same set of samples) to estimate the normalising constant. Note that this estimate is biased.

Show Estimate & Bias

Central Limit

Let $X_1, \dots$ be a sequence of iid RVs with $\mathbb{E}[X_i] = 0$ and $\mathbb{E}[X_i^2] = \sigma^2$. Define:

$$Z_N = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} X^{(i)} \tag{1.4}$$

As $N \to \infty$, the distribution of Z_n approaches $\mathcal{N}(0, \sigma^2)$.

Efficiency Loss - Proof

Other that q, note that superscript $*$ denotes a normalised density, or quantities produced using normalised densities. Define

$$S_1 = \frac{1}{N} \sum_{i=1}^{N} h(X^{(i)}) w^*(X^{(i)}) \tag{1.5}$$

$$S_2 = \frac{1}{N} \sum_{i=1}^{N} w^*(X^{(i)}) \tag{1.6}$$

and note that the self-normalised importance sampling estimate can be expressed as S_1/S_2. Additionally:

$$s_1 = \mathbb{E}[S_1] = \mathbb{E}_{\pi*}[h(X)] \qquad s_2 = \mathbb{E}[S_2] = 1 \tag{1.7}$$

Now consider the variation of the sample about the target value:

$$\begin{aligned} \frac{S_1}{S_2} - s_1 &= \frac{1}{S_2}(S_1 - S_2 s_1) \\ &= \frac{1}{S_2}\frac{1}{N} \sum_{i=1}^{N} (h(X^{(i)}) - s_1) w^*(X^{(i)}) \end{aligned} \tag{1.8}$$

Note that each term in the summation has zero mean and variance:

Show Zero Mean

$$\begin{aligned} \sigma^2 &= \mathbb{E}_q[(h(X) - s_1)^2 w^*(X)^2] \\ &= \mathbb{E}_{\pi^*}[(h(X) - s_1)^2 w^*(X)] \end{aligned} \tag{1.9}$$

Applying the central limit theorem, for large N,

$$\frac{S_1}{S_2} - s_1 \to \frac{1}{S_2}\frac{1}{\sqrt{N}} \mathcal{N}(0, \sigma^2) \tag{1.10}$$

Also note that for large N, $S_2 \to 1$, meaning the the importance sampling estimate has the following asymptotic distribution:

$$\frac{S_1}{S_2} \overset{\text{large } N}{\sim} \mathcal{N}(\mathbb{E}_{\pi*}[h(X)], \frac{\sigma^2}{N}) \tag{1.11}$$

so the self-normalised importance sampling estimate is asymptotically unbiased.

Typical Monte Carlo Variance

The variance of a Monte Carlo estimate, assuming sampling from the correct distribution could be performed, is:

$$\frac{1}{N}\underbrace{\mathbb{E}_{\pi^*}[(h(X)-\mathbb{E}_{\pi*}[h(X)])^2]}_{\sigma_0^2} \tag{1.12}$$

Equating the variance of a normal Monte Carlo estimate and the self-normalised importance sampling estimate gives:

$$N_{\text{IS}} = N\frac{\sigma^2}{\sigma_0^2} \tag{1.13}$$

Often $\sigma^2/\sigma_0^2 > 1$, meaning that more samples from q are required to match the quality of the estimate using π^*. There is a loss of efficiency due to the trial probability density function q and the target. It can be difficult to find a good candidate for the trial distribution, especially as the dimension of the problem increases (the ratio between the variances can increase with the problem size).

1.4 Sampling Techniques for HMMs

Target Density

The target density is the conditional density of all the states given all the observations.

$$\pi_n^*(x_0,\dots,x_n) = p(x_0,\dots,x_n|y_0,\dots,y_n) \propto \underbrace{p(x_0,y_0,\dots,x_n,y_n)}_{\pi_n(x_0,\dots,x_n)} \tag{1.1}$$

Trial Density

The trial distribution has the following Markov structure:

$$q_n(x_0,\dots,x_n) = q_0(x_0)q_1(x_0,x_1)\dots q_n(x_{n-1},x_n) \tag{1.2}$$

Importance Weights

where each $q_i(x_{i-1},x_i)$ is normalised in the second argument. Typically we may sample from the state transition density, $f(\cdot,\cdot)$. Given this structure, the importance weights can be written:

$$w(X_{0:n}^{(i)}) = \frac{p(X_0^{(i)})g(X_0^{(i)},y_0)}{q(X_0^{(i)})}\frac{f(X_0^{(i)},X_1^{(i)})g(X_1^{(i)},y_1)}{q(X_0^{(i)},X_1^{(i)})}\cdots\frac{f(X_{n-1}^{(i)},X_n^{(i)})g(X_n^{(i)},y_n)}{q(X_{n-1}^{(i)},X_n^{(i)})} \tag{1.3}$$

1.4.1 Sequential Importance Sampling

The idea behind sequential importance sampling is to exploit the Markov structure of q_n when sampling from it. Given samples, $X_{0:n}^{(i)} \sim q_n$, and weights, $w(X_{0:n}^{(i)})$, sequential importance sampling can be performed as follows:

Extension

1. Sample $X_{n+1}^{(i)}$ from $q_{n+1}(X_n^{(i)},x_{n+1})$ for each sample, i. Then, simply append the new sample onto the existing samples, $X_{0:n+1}^{(i)} = \left[X_{0:n}^{(i)},X_{n+1}^{(i)}\right]$.

Weight Update

2. Re-weight each sample:

$$w_{n+1}^{(i)} = w_{n+1}^{(i)}\underbrace{\frac{f(X_n^{(i)},X_{n+1}^{(i)})g(X_{n+1}^{(i)},y_n)}{q(X_n^{(i)},X_{n+1}^{(i)})}}_{u_{n+1}^{(i)}} \tag{1.4}$$

$u_{n+1}^{(i)}$ is known as the incremental weight. This expression for the incremental weight follows directly from Eq. 1.3.

Problem with Sequential Importance Sampling

The problem is sequential importance sampling is that eventually the weights will collapse, with most of the weights becoming small and one larger than the rest. Thus, one of the N normalised weights ($w_n^{(i)} / \sum_j w_n^{(j)}$) will be close to 1, meaning that the self-normalised estimate will collapse to a single sample estimate.

Developing a Solution

We note that samples with small normalised weights should not be carried forward, whilst those with larger normalised weights should. Additionally, the population size should be kept at the original level, N, or only a single sample will remain. Resampling is one technique of resolving this problem.

1.4.2 Resampling

Algorithm 1 Resampling

Input: N weighted samples, $(X_{0:n}^{(i)}, w_n^{(i)})$ which unbiasedly approximate π_n i.e.

$$\mathbb{E}[h_n(X_{0:n}^{(i)})w_n^{(i)}] = \int h_n(x_{0:n})\pi_n(x_{0:n})\, dx_{0:n} \tag{1.5}$$

```
Let $W_n = \sum_j w_n^{(j)}$
for $i = 1, \dots, N$ do
    Sample $J$ from the set $\{1, \dots, N\}$ with $Pr(J = j) = w_n^{(j)} / W_n$
    $X_{0:n}^{(i)} \leftarrow X_{0:n}^{(J)}$
    $w_n^{(i)} \leftarrow W_n / N$
end
```

Proof: Unbiasedness of Resampling

The unbiased of resampling is easy to show, given that the input samples are also biased. To complete this proof, simply write down and rearrange the expected value of the output, considering first the randomness in J.

1.4.3 Sequential Importance Sampling with Resampling

The resampling technique is developed as follows: after resampling and reweighting occurs, the particle is then extended and weighted according to the sample generated i.e.

$$X_{n+1}^{(i)} \sim q_{n+1}(X_{0:n}^{(i)}, x_{n+1}) \tag{1.6}$$

$$w_{n+1}^{(i)} = \underbrace{\frac{W_n}{N}}_{w_n^{(i)}} \underbrace{\frac{f(X_n^{(i)}, X_{n+1}^{(i)})g(X_{n+1}^{(i)}, y_n)}{q(X_n^{(i)}, X_{n+1}^{(i)})}}_{u_{n+1}^{(i)}} \tag{1.7}$$

Verify Unbiasedness

This gives an unbiased estimate of π_{n+1} that can be verified by considering:

$$\mathbb{E}[h_{n+1}(X_{0:n}^{(J)}, X_{n+1}^{(i)})w_{n+1}^{(i)}] \tag{1.8}$$

When evaluating this, the randomness in each original sample, the value of J, and the value of $X_{n+1}^{(i)}$ must be considered in reverse order.

The Particle Filter is this technique, applied specifically to Hidden Markov Models. Note that by setting $h_T(x_{0:T}) = 1$, the particle filter calculates the probability of the observed data.

1.4.4 Analysis of the Particle Filter

Pertinent Questions

When using the particle filter, we want to know:

1. Do the number of particles, N, need to increase with the length of the time-series to get a stable estimate of $p(y_{0:T})$?

2. How does the particle filter compare to normal sequential importance sampling without the re-sampling steps?

Prove this Expansion

Consider the average sum of weights at time T:

$$\begin{aligned} p(y_{0:T}) &\simeq \frac{W_T}{N} \\ &= \frac{W_{T-1}}{N}\sum_{i=1}^{N}\frac{u_T^{(i)}}{N} \\ &= \Big(\frac{1}{N}\sum_{i=1}^{N}\frac{p(X_0^{(i)})g(X_0^{(i)})}{q_0(X_0^{(i)})}\Big)\prod_{j=1}^{T}\Big(\sum_{i=1}^{N}\frac{u_j^{(i)}}{N}\Big) \end{aligned} \tag{1.9}$$

Simplifying Example

This is not easy to analyse in general. Thus, let the hidden state process be IID according to $p(\cdot)$. Then:

$$p(y_{0:T}) = \prod_{i=0}^{T} p(y_i) \tag{1.10}$$

with $p(y_i) = \int p(x_n)g(x_n, y_n)dx_n$. Additionally, consider $q_n(x_{n-1}, x_n) = p(x_n)$. Then:

$$p(y_{0:T}) \simeq \prod_{j=0}^{T}\Big(\sum_{i=1}^{N}\frac{g(X_j^{(i)}, y_j)}{N}\Big) = \prod_{j=0}^{T}\frac{U_j}{N} \tag{1.11}$$

Consider the variance of this estimate, noting independence:

$$\frac{\text{var}\,[\frac{W_T}{N}]}{p(y_0)^2\cdots p(y_T)^2} = \mathbb{E}\left\{\left(\frac{U_0}{Np(y_0)}\right)^2\right\}\cdots\mathbb{E}\left\{\left(\frac{U_T}{Np(y_T)}\right)^2\right\} - 1 \tag{1.12}$$

Show This

For any value of n:

$$\mathbb{E}\left\{\left(\frac{U_n}{Np(y_n)}\right)^2\right\} = 1 + \frac{1}{N}\underbrace{\text{var}\left[\frac{g(X_n, y_n)}{p(y_n)}\right]}_{c_n} \tag{1.13}$$

Therefore, noting that $1 + \frac{c_n}{N} < \exp(\frac{c_0}{N})$ by Taylor expansion, we note that:

$$\frac{\text{var}\,[\frac{W_T}{N}]}{p(y_0)^2\cdots p(y_T)^2} < \exp\Big(\frac{c_0 + \cdots + c_T}{N}\Big) - 1 \tag{1.14}$$

Giving a variance which grows with T, unless N is also increased with T.

1.5 Model Calibration

Parametrising a HMM

A hidden Markov model can be parametrised by a vector by modifying the state transition density and observation density to depend on a parameter, $\theta \in \mathbb{R}^d$ i.e.

$$f_\theta(x_{k-1}, x_k) \qquad g_\theta(x_k, y_k)$$

Choosing θ_k^*

A typical approach used to choose these parameters is to use the *maximum likelihood principle*:

$$\theta_k^* = \arg\max_\theta \underbrace{\log p_\theta(y_{0:k})}_{\mathcal{L}_k(\theta)} \tag{1.1}$$

Why is this hard?

which, under some assumptions, provides a consistent estimator i.e. $\theta_k^* \to \theta^*$ as the data length increases and θ^* is the 'optimal' value. However, we do not observe the hidden states and thus it is difficult to find the maximiser. We resort to iterative methods.

1.5.1 Gradient Methods

One simple approach is to apply gradient ascent:

$$\theta^{(i+1)} = \theta^{(i)} + \eta_i \nabla_\theta \mathcal{L}(\theta^{(i)}) \tag{1.2}$$

where ∇_θ denotes the vector gradient in the usual way and $\{\eta_i\}$ is a positive step size sequence tending to zero. The parameter will converge to a local maximum of the likelihood, which may not necessarily be the global maximiser. Typically, we may use *Stochastic Approximation* where a Monte Carlo estimate of the gradient is used, in which case the step size sequence should be chosen such that the step sizes are positive, tend to zero but with $\sum_i \eta_i = \infty$. A typical choice is:

Stochastic Approximation

$$\eta_i = \frac{1}{i^\alpha} \tag{1.3}$$

with $\alpha \in [0.5, 1]$. Fixed step sizes give oscillatory behaviour around a maximum.

It can be shown that:

Stochastic Optimisation

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta_k} = \int \left(\frac{\partial}{\partial \theta_k} \log p_\theta(x_{0:n}, y_{0:n}) \right) p_\theta(x_{0:n}|y_{0:n}) \, dx_{0:n} \tag{1.4}$$

which can be calculated using the particle filter. Additionally, it is easy to show that:

Proof

$$\frac{\partial}{\partial \theta_k} \log p_\theta(x_{0:n}, y_{0:n}) = \sum_{j=1}^{n} \left(\frac{\partial}{\partial \theta_k} \log f_\theta(x_{j-1}, x_j) + \frac{\partial}{\partial \theta_k} \log g_\theta(x_j, y_j) \right) \tag{1.5}$$

1.5.2 Expectation Maximisation

The EM algorithm is an alternative iterative procedure for maximising the log-likelihood. Define:

Auxiliary Function

$$\mathcal{Q}(\theta, \theta') = \int \log\{p_{\theta'}(x_{0:k}, y_{0:k})\} p_\theta(x_{0:k}|y_{0:k}) dx_{0:k} \tag{1.6}$$

θ is the current best guest of the global maximiser and we optimise over θ'. Considering the strictly concave function $\log(x)$ and the variable $\frac{p_{\theta'}}{p_\theta}$, Jensen's Inequality directly implies that for any $\theta' \neq \theta$:

Prove using Jensen

$$\int \log p_{\theta'}(x_{0:k}|y_{0:k}) p_\theta(x_{0:k}|y_{0:k}) \, dx < \int \log p_\theta(x_{0:k}|y_{0:k}) p_\theta(x_{0:k}|y_{0:k}) \, dx \tag{1.7}$$

which effectively states that the average log likelihood under the true density is larger than under any other density. Rearranging gives:

Rearrange as required

$$\log p_{\theta'}(y_{0:k}) - \log p_\theta(y_{0:k}) > \mathcal{Q}(\theta, \theta') - \mathcal{Q}(\theta, \theta) \tag{1.8}$$

Thus by maximising over θ' are each step means that the log likelihood will increase. The EM algorithm repeatedly maximises the auxiliary function. The E-step computes $\mathcal{Q}(\theta, \theta')$ and the M-step maximises this over θ'.

For the general HMM we cannot compute the integral defining $\mathcal{Q}(\theta, \theta')$ though we can for the Gaussian state-space model.

EM for Exponential Family Models

A HMM belongs to the exponential family if:

Exponential Family

$$f_\theta(x_{n-1}, x_n) g_\theta(x_n, y_n) = h(x_{n-1}, x_n, y_n) \exp\left[\phi(\theta)^T S(x_{n-1}, x_n, y_n) - m(\theta) \right] \tag{1.9}$$

where $\phi(\theta)$ and $S(x_{n-1}, x_n, y_n)$ are vector valued functions of the same dimension. Then:

$$p_{\theta'}(x_{0:k}, y_{0:k}) = \exp\left[\sum_{n=0}^{k} \phi(\theta')^T S(x_{n-1}, x_n, y_n) - (k+1)m(\theta')\right] \times \prod_{n=1}^{k} h(x_{n-1}, x_n, y_n) \quad (1.10)$$

Derive This

The maximisation step then reduces to maximising:

$$\phi(\theta')^T \underbrace{\int \left(\sum_{n=0}^{k} S(x_{n-1}, x_n, y_n)\right) p_\theta(x_{0:k}|y_{0:k})\, dx_{0:k}}_{\bar{S}} - (k+1)m(\theta') \quad (1.11)$$

$\bar{S}$ can be computed easily using the particle filter, and note that it is independent of θ'. Then, we require:

$$\max_\theta \phi(\theta)^T \bar{S} + (k+1)m(\theta') \quad (1.12)$$

which can be solved easily, but note that the type of stationary point found needs to be checked.

Linking Gradient Ascent and EM

Note that:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta_k} = \left.\frac{\partial \mathcal{Q}(\theta, \theta')}{\partial \theta'}\right|_{\theta'=\theta} \quad (1.13)$$

which can be seen easily simply via substitution. Gradient ascent replaces the maximisation step of the EM algorithm; the new value of θ' is the current value plus a small change in the direction of increasing the auxiliary function.