STAT 248 - Analysis of Time Series
Full Lecture Notes

Spring 2022, UC Berkeley

Aditya Guntuboyina

October 7, 2022

Contents

1 Lecture One **4**
1.1 State Space Models 4
1.2 Examples of State Space Models 5
1.2.1 Direct Examples: Tracking 5
1.2.2 Trend Estimation 6
1.3 Recommended Reading for Today 7

2 Lecture Two **7**
2.1 Local Level and Local Linear Models 8
2.2 Stochastic Volatility Models 9
2.3 Dynamic Regression Model 9
2.4 Recommended Reading for Today 10

3 Lecture Three **10**
3.1 Connection to the Periodogram 13
3.2 Recommended Reading for Today 16

4 Lecture Four **17**
4.1 The Autoregressive Model 19
4.2 Recommended Reading for Today 21

5 Lecture Five **21**
5.1 Outline of Approach to Calculate Smoothing Distributions 22
5.2 Linear Gaussian State Space Models 23
5.3 Recommended Reading for Today 23

6 Lecture Six **24**
6.1 General Approach for calculating Filtering Distributions 24
6.2 The Kalman Filter 25
6.3 Recommended Reading for Today 27

7 Lecture Seven **28**
7.1 The Kalman Filter 28
7.2 Some Examples 28
7.2.1 Tracking One: Velocity Model 29

7.2.2 Tracking Two: Acceleration Model . . . 29
7.2.3 Tracking Three: Local Linear Model . . . 31
7.3 Use of the Kalman Filter for Parameter Estimation by Maximum Likelihood 31
7.4 Recommended Reading for Today . . . 32

8 Lecture Eight **32**
8.1 Some remarks on the local level model . . . 32
8.2 Application of the Kalman Filter to Linear Regression . . . 35
8.3 Prediction . . . 36
8.4 Smoothing . . . 36
8.5 Recommended Reading for Today . . . 37

9 Lecture Nine **37**
9.1 Smoothing . . . 37
9.2 Backward Recursion for General State Space Models . . . 37
9.3 Smoothing for Linear Gaussian State Space Models . . . 38
9.4 Dealing with missing data in the context of state space models . . . 41
9.5 Recommended Reading for Today . . . 41

10 Lecture Ten **42**
10.1 Summary: General Filtering and Smoothing . . . 42
10.2 Summary: Kalman Filtering and Smoothing . . . 43
10.3 Special Case: Local Level Model . . . 44
10.4 Numerical Evaluation of $X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ . . . 45
10.5 Recommended Reading for Today . . . 46

11 Lecture Eleven **46**
11.1 Basic Optimization Algorithms . . . 46
11.1.1 Gradient Ascent . . . 46
11.1.2 Newton's Method . . . 47
11.1.3 Quasi-Newton Method: BFGS . . . 47
11.2 Application to Maximum Likelihood Estimation in State Space Models . . . 49
11.2.1 Fisher Identity for the Score . . . 49
11.2.2 $E(\theta, \theta^{(0)})$ for state space models . . . 51
11.3 Recommended Reading for Today . . . 52

12 Lecture Twelve **52**
12.1 Pairwise Smoothing Distributions . . . 52
12.2 Fisher's Identity (from last time) . . . 53
12.3 The Score Function for the Local Level Model . . . 53
12.4 The EM Algorithm . . . 55
12.5 EM for the local level model . . . 55
12.6 Calculation of $E(\theta, \theta^{(0)})$ for general state space models . . . 56
12.7 Recommended Reading for Today . . . 57

13 Lecture Thirteen **57**
13.1 The MM Algorithm . . . 57
13.2 The EM Algorithm as a special case of MM . . . 59
13.2.1 The Kullback-Leibler Divergence . . . 59
13.2.2 EM and MM . . . 60
13.3 Full Smoothing Distribution . . . 60
13.4 Forward Filtering Backward SAMPLING . . . 61
13.5 Recommended Reading for Today . . . 62

14 Lecture Fourteen **63**
14.1 Local Level Model . . . 63
14.2 Gibbs Sampler . . . 64
14.3 Gibbs Sampler for the Local Level Model . . . 64
14.4 Gibbs sampler for general Linear Gaussian state space models . . . 66
14.5 Recommended Reading for Today . . . 66

15 Lecture Fifteen **66**
15.1 Approach One: Gibbs Sampling . . . 67
15.2 Approach Two: Direct Sampling . . . 67
15.3 Approach Three: Posterior Normal Approximation . . . 68
15.4 Approach Four: Importance Sampling . . . 69
15.5 Recommended Reading for Today . . . 72

16 Lecture Sixteen **72**
16.1 Notation for Discrete Distributions . . . 73
16.2 Monte Carlo versions of (99) and (100) . . . 74
16.3 The Bootstrap Particle Filter . . . 75
16.4 Importance Sampling Recalled . . . 76
16.5 Bootstrap Particle Filter as Importance Resampling . . . 77
16.5.1 First Way of Seeing the Connection . . . 77
16.5.2 Second Way of Seeing the Connection . . . 78
16.6 Likelihood Approximation from the Bootstrap Particle Filter . . . 78
16.7 Recommended Reading for Today . . . 79

17 Lecture Seventeen **79**
17.1 Recap: Bootstrap Particle Filter . . . 79
17.2 Unique Values and Particle Degeneracy . . . 80
17.3 The Guided Particle Filter Algorithm . . . 81
17.4 Weights when $q_t(u \mid x, y, \theta) := f_{X_t|X_{t-1}=x,Y_t=y,\theta}(u)$ . . . 83
17.5 Example: Local Level Model . . . 83
17.6 Recommended Reading for Today . . . 84

18 Lecture Eighteen **84**
18.1 Sequential Importance Resampling . . . 84
18.2 Example: Local Level Model with non-Gaussian evolution errors . . . 85
18.3 Recommended Reading for Today . . . 88

19 Lecture Nineteen **89**
19.1 Complete Smoothing . . . 89
19.2 FFBS . . . 91
19.3 Recommended Reading for Today . . . 93

20 Lecture Twenty **94**
20.1 Recap: Complete Smoothing . . . 94
20.2 Complete Smoothing with partial trajectory resampling . . . 94
20.3 Recap: FFBS . . . 95
20.4 Recommended Reading for Today . . . 96

21 Lecture Twenty One **96**
21.1 Model Selection . . . 96
21.2 Akaike Information Criterion (AIC) . . . 96
21.2.1 The simple case of no parameters . . . 97

21.2.2 Models with parameters 98
21.2.3 Digression: MLE asymptotic distribution 98
21.3 Back to AIC 104
21.4 Recommended Reading for Today 105

22 Lecture Twenty Two **106**
22.1 Recap: AIC 106
22.2 Bayesian Model Selection 107
22.3 Two Alternative Expressions for the Evidence 110
22.4 The BIC 111
22.5 Recommended Reading for Today 112

23 Lecture Twenty Three **112**
23.1 Recap: Frequentist and Bayesian Model Selection 112
23.2 Example: Normal Mean 113
23.3 Application: Linear Regression 114
23.4 Recommended Reading for Today 117

1 Lecture One

Time series refers to observations collected sequentially in time. One can have univariate time series (where a single observation is collected at each point in time) or multivariate time series (where a bunch of obserations are collected at each point in time). In this class, we shall denote the observed time series by

$$y_0, y_1, \ldots, y_T.$$

Here y_0 denotes the observed value at the first time point, y_1 denotes the observed value at the second time point etc. Typically the time points where the observations are taken are uniformly spaced but there do exist situations where the time points are not uniformly spaced (if the time points are not uniformly spaced, we shall denote them by $t_0, t_1, \ldots, t_T$ and note that the observation y_i corresponds to the time t_i).

Time series are commonly analyzed through time series models. These models assume first that the observed time series $y_0, \ldots, y_T$ are a realization of random variables $Y_0, Y_1, \ldots, Y_T$, and then proceed to describe the joint distribution of $Y_0, \ldots, Y_T$. We shall focus on *State Space Models* in this class as these are a general class of time series models with wide applicability.

1.1 State Space Models

State space models assume that $\{Y_t, 0 \leq t \leq T\}$ are noisy measurements of a hidden or latent *Markov process* $\{X_t, 0 \leq t \leq T\}$.

Here $\{X_t, 0 \leq t \leq T\}$ is a Markov process means that the conditional distribution of X_t given $X_{t-1} = x_{t-1}, \ldots, X_0 = x_0$ is the same as the conditional distribution of X_t given $X_{t-1} = x_{t-1}$ for every $1 \leq t \leq T$ and $x_0, x_1, \ldots, x_t$. We shall denote the density of X_0 by $p_0(\cdot)$ and the density of X_t given $X_{t-1} = x_{t-1}$ by $p_t(x_t \mid x_{t-1})$ for $t = 1, \ldots, T$. p_0 is called the initial distribution of the Markov process $\{X_t\}$ and $p_t(x_t \mid x_{t-1})$ is called the t^{th} transition density. If the transition densities are the same for all t, we say that $\{X_t\}$ is a time

homogeneous Markov process (otherwise, $\{X_t\}$ is said to be a time inhomogeneous Markov process). Note that the joint density of $X_0, \ldots, X_T$ equals

$$p_0(x_0)p_1(x_1 \mid x_0) \ldots p_T(x_T \mid x_{T-1}) = p_0(x_0) \prod_{t=1}^{T} p_t(x_t \mid x_{t-1}).$$

State space models specify that $\{X_t, 0 \leq t \leq T\}$ is a Markov process and, additionally, that $Y_0, \ldots, Y_T$ are independent conditionally on $X_0, \ldots, X_T$ and, moreoever, that the conditional distribution of Y_t given $X_0 = x_0, \ldots, X_T = x_T$ is the same as the conditional distribuion of Y_t given $X_t = x_t$ for each $0 \leq t \leq T$. We shall denote the conditional density of Y_t given $X_t = x_t$ by $f_t(y_t \mid x_t)$. The conditional joint density of $Y_0, \ldots, Y_T$ given $X_0 = x_0, \ldots, X_T = x_T$ equals

$$\prod_{t=0}^{T} f_t(y_t \mid x_t).$$

To summarize, state space models specify that the joint distribution $X_0, Y_0, \ldots, X_T, Y_T$ equals

$$p_0(x_0) \prod_{t=1}^{T} p_t(x_t \mid x_{t-1}) \prod_{t=0}^{T} f_t(y_t \mid x_t). \tag{1}$$

The random variables $X_0, \ldots X_T$ are known as state variables (or hidden or latent variables) and $Y_0, \ldots, Y_T$ are known as data variables. Observe that the joint density of the data variables $Y_0, \ldots, Y_T$ is given by integrating (1) with respect to $x_0, \ldots, x_T$:

$$\int \cdots \int \left[p_0(x_0) \prod_{t=1}^{T} p_t(x_t \mid x_{t-1}) \prod_{t=0}^{T} f_t(y_t \mid x_t) \right] dx_0 dx_1 \ldots dx_T$$

State space models can also be referred to as Hidden Markov Models although some authors use Hidden Markov Models to refer to models where the state variables X_t are discrete random variables.

1.2 Examples of State Space Models

1.2.1 Direct Examples: Tracking

In tracking problems, the goal is to track the movement of an unknown moving object from noisy measurements $\{Y_t\}$. Here the state space model directly arises with the state variable X_t representing attributes of the moving object (such as position and velocity). To give a concrete example, consider a body moving in the two-dimensional plane. Suppose we discretize time to a resolution δ (so that the time points are $t_0, t_1, \ldots$ with $t_k = k\delta$).

Denote the position of the object at time t_k by (x_{1k}, x_{2k}) (remember we are assuming that the movement is in the two-dimensional plane). Also let the velocity of the object at time t_i is (x_{3k}, x_{4k}). If the velocity in the time period $[t_{k-1}, t_k]$ is assumed to be nearly constant, we would have

$$x_{1k} \approx x_{1,k-1} + \delta x_{3,k-1} \quad \text{and} \quad x_{2k} \approx x_{2,k-1} + \delta x_{4,k-1}.$$

One can assume these equations to be exact (as opposed to approximate) by incorporating error variables:

$$x_{1k} = x_{1,k-1} + \delta x_{3,k-1} + q_{1k} \quad \text{and} \quad x_{2k} = x_{2,k-1} + \delta x_{4,k-1} + q_{2k}$$

Here q_{1k}, q_{2k} denote error variables which can be modeled as i.i.d with a normal distribution. Further the assumption that the velocity is nearly constant in the time period $[t_{k-1}, t_k]$ can be written as

$$x_{3,k} \approx x_{3,k-1} \quad \text{and} \quad x_{4,k} \approx x_{4,k-1}.$$

These two equations can also be assumed to be exact by incorporating error variables:

$$x_{3k} = x_{3,k-1} + q_{3k} \quad \text{and} \quad x_{4k} = x_{4,k-1} + q_{4k}.$$

If we therefore let

$$X_k = \begin{pmatrix} x_{1k} \\ x_{2k} \\ x_{3k} \\ x_{4k} \end{pmatrix}$$

denote the position and velocities of the unknown object, then X_k satisfies the equation

$$X_k = \begin{pmatrix} 1 & 0 & \delta & 0 \\ 0 & 1 & 0 & \delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} X_{k-1} + q_k \quad \text{where } q_k = \begin{pmatrix} q_{1k} \\ q_{2k} \\ q_{3k} \\ q_{4k} \end{pmatrix}$$

If we assume that q_k are i.i.d, then it is easy to check that $\{X_k\}$ is a Markov process.

The observation Y_k here is a noisy measurement of X_k. The exact relationship between Y_k and X_k depends on the nature of the measurements. Suppose that we are obtaining noisy measurements only of the position of the object. Then

$$Y_k = \begin{pmatrix} x_{1k} \\ x_{2k} \end{pmatrix} + \begin{pmatrix} \epsilon_{1k} \\ \epsilon_{2k} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} X_k + \begin{pmatrix} \epsilon_{1k} \\ \epsilon_{2k} \end{pmatrix}$$

Suppose we assume that $\epsilon_k = \begin{pmatrix} \epsilon_{1k} \\ \epsilon_{2k} \end{pmatrix}$ are i.i.d and also that the two error sequences $\{\epsilon_k\}$ and $\{q_k\}$ are independent. Then this represents a state space model (it turns out that this is a linear Gaussian state space model as will be clear soon).

In another measurement model, we could only be measuring the angle that the unknown object makes with the positive x-axis (this is sometimes known as bearings-only tracking). Here we would have

$$Y_k = \arctan\left(\frac{x_{2k}}{x_{1k}}\right) + \epsilon_k.$$

This is again a state-space model (this is a nonlinear state space model).

1.2.2 Trend Estimation

State space models can be used to estimate trend in state space models. Trend in a time series can be generally understood as a smooth function that tracks well the evolution or course of the time series. One way of estimating a smooth trend is via the following state space model. As usual, we let $Y_0, \ldots, Y_T$ to be the data random variables. The idea is that the hidden state variables $X_0, \ldots, X_T$ represent the trend. Because trend is supposed to be smooth, we assume that

$$X_t = X_{t-1} + \eta_t \qquad \text{with } \eta_k \overset{\text{i.i.d}}{\sim} N(0, \sigma_\eta^2). \tag{2}$$

This equation says that X_t is centered around X_{t-1} with an error whose size is controlled by σ_η. If σ_η is small, then $X_t \approx X_{t-1}$ representing a smooth trend. Note that (2) clearly implies that $\{X_t\}$ is a Markov process.

The data variables Y_t are connected to the state variables X_t via

$$Y_t = X_t + \epsilon_t \qquad \text{with } \epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2). \tag{3}$$

This equation captures the intuition that the trend X_t tracks the time series Y_t.

The noise parameters σ_η and σ_ϵ control the twin objectives of smooth trend and tracking the data respectively. If σ_η is small, we would get smoother trends while if σ_ϵ is small, our trend estimate will closely track the data. Note however if the observed time series y_t is not very smooth, then both the objectives cannot be simultaneously achieved. In general, one chooses σ_η and σ_ϵ so as to obtain the best fit to the data (we shall see all this later).

The state space model given by the pair of equations (2) and (3) is called the local level model. It is a common way of estimating trend in time series.

1.3 Recommended Reading for Today

1. Definition of State Space Models: Sections 2.1 and 2.2 of the Chopin-Papaspiliopoulos book.
2. Tracking application of State Space Models: Section 2.4.1 of the Chopin-Papaspiliopoulos book, and Section 1.3.2 of the Triantafyllopoulos book.
3. Local level model: Section 2.1 of the Durbin-Koopman book, and Section 1.2 of the Triantafyllopoulos book.

2 Lecture Two

In the last class, we introduced state space models and looked at two examples (a tracking model and the local level model). To recap, state space models describe the distribution of $Y_0, \ldots, Y_T$ in terms of a hidden set of random variables $X_0, \ldots, X_T$. The joint distribution of $X_0, Y_0, \ldots, X_T, Y_T$ is specified via the joint density:

$$p_0(x_0) \prod_{t=1}^{T} p_t(x_t \mid x_{t-1}) \prod_{t=0}^{T} f_t(y_t \mid x_t). \tag{4}$$

This means that the density of X_0 is p_0, the conditional density of X_t given $X_{t-1} = x_{t-1}$ (as well as given $X_{t-1} = x_{t-1}, \ldots, X_0 = x_0$) equals $p_t(x_t \mid x_{t-1})$ and the conditional density of Y_t given $X_t = x_t$ (as well as given $X_t = x_t, X_s = x_s$ for $s \neq t$) equals $f_t(y_t \mid x_t)$.

Specifying the joint distribution via the joint density (4) requires writing down $p_0(x_0)$, $p_t(x_t \mid x_{t-1})$ as well as $f_t(y_t \mid x_t)$. In practice, people specify state space models via equations involving independent random variables. More precisely, one usually first specifies the distribution p_0 of X_0 (this is often a diffuse density such as a normal with a large variance or a uniform over a large range), and then specify the distribution of X_t via the equation:

$$X_t = K_t(X_{t-1}, U_t) \qquad \text{for } t = 1, 2, \ldots \tag{5}$$

where $\{U_t\}$ are independent random variables that are also independent of X_0. Finally the distribution of Y_t is specified via

$$Y_t = H_t(X_t, V_t) \qquad \text{for } t = 0, 1, 2, \ldots \tag{6}$$

where $\{V_t\}$ are independent random variables that are also independent of $\{U_t\}$ and X_0. The functions K_t and H_t in (5) and (6) can be completely arbitrary.

Linear Gaussian State Space Models form a special case of state space models (inference is particularly easy in linear Gaussian State Space Model because of the Kalman filter; as we shall study in the next few weeks). Specifically, for a linear Gaussian state space model, X_0 is normal, the state evolution equation (5) takes the form

$$X_t = F_{t-1}X_{t-1} + U_t \qquad \text{with } U_t \overset{\text{independent}}{\sim} N(0, Q_t)$$

and the observation equation (6) takes the form

$$Y_t = H_t X_t + V_t \qquad \text{with } V_t \overset{\text{independent}}{\sim} N(0, R_t)$$

Here F_{t-1} and H_t are deterministic matrices, and Q_t and R_t are covariance matrices. Note that for the linear Gaussian state space model, each of the densities $p_0(x_0)$, $p_t(x_t \mid x_{t-1})$ and $f_t(y_t \mid x_t)$ are normal with mean being a linear function of the underlying variable and the covariance being a deterministic matrix.

We shall look at a few additional examples of state space models today.

2.1 Local Level and Local Linear Models

In the last class, we looked at the simple local level model:

$$\begin{aligned} &X_0 \sim N(m_0, \Gamma_0) \\ &X_t = X_{t-1} + \eta_t \qquad \text{with } \eta_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\eta^2) \\ &Y_t = X_t + \epsilon_t \qquad \text{with } \epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2). \end{aligned}$$

This model has the two parameters σ_η^2 and σ_ϵ^2 (the parameters m_0 and Γ_0 of X_0 are usually set to be some standard values corresponding to a diffuse distribution such $m_0 = 0$ and $\Gamma_0 = 10^8$). We have seen simulation examples involving smooth trend estimation where this model does a decent job in recovering the underlying smooth trend function (it does not work however when the underlying trend function is nonsmooth). But often the trend estimate provided by this model is somewhat wiggly and we might want to obtain a smoother fit. This can be achieved by the local *linear* model given by

$$\begin{aligned} &X_0 \sim N(m_0, \Gamma_0) \\ &X_t - X_{t-1} = X_{t-1} - X_{t-2} + \eta_t \qquad \text{with } \eta_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\eta^2) \\ &Y_t = X_t + \epsilon_t \qquad \text{with } \epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2). \end{aligned}$$

The difference between the local level and the local linear models is that the random walk specification in the local linear model is in terms of the slopes $X_t - X_{t-1}$ as opposed to the levels as in the local level model. This generally leads to smoother fits.

Note that the local linear model is also a state space model even though $\{X_t\}$ as defined by $X_t - X_{t-1} = X_{t-1} - X_{t-2} + \eta_t$ is not Markov. This is because we can rewrite the model in terms of the state variable $\tilde{X}_t$ defined by

$$\tilde{X}_t := \begin{pmatrix} X_t \\ X_{t-1} \end{pmatrix}.$$

The equation $X_t - X_{t-1} = X_{t-1} - X_{t-2} + \eta_t$ is easily seen to be equivalent to

$$\tilde{X}_t = \begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix} \tilde{X}_{t-1} + \begin{pmatrix} \eta_t \\ 0 \end{pmatrix}$$

which implies that $\{\tilde{X}_t\}$ is a Markov process. The observation equation $Y_t = X_t + \epsilon_t$ can be written in terms of $\tilde{X}_t$ as

$$Y_t = \begin{pmatrix} 1 & 0 \end{pmatrix} \tilde{X}_t + \epsilon_t.$$

This shows that the local linear model is also a state space model.

This re-writing of a second order Markov process $\{X_t\}$ in terms of the Markov process $\tilde{X}_t$ is reminiscent of a similar argument in Ordinary Differential Equations. For example, the second order differential equation

$$x''(t) = -\omega^2 x(t)$$

can be re-written as the first order differential equation

$$\begin{pmatrix} x_1'(t) \\ x_2'(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

2.2 Stochastic Volatility Models

Consider the model

$$\begin{aligned} X_t &= X_{t-1} + \eta_t \qquad \text{with } \eta_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\eta^2) \\ Y_t &= \exp\left(X_t/2\right) \epsilon_t \qquad \text{with } \epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2). \end{aligned}$$

Data generated from this model exhibits volatility clustering i.e., the variance remains high or low for considerable periods of time. This model is useful for finance data (say for log-returns of stocks) which exhibit volatility clustering. This model is an alternative to volatility time series models such as ARCH or GARCH (which are somewhat less natural even though they are widely used). It is easy to check that this is also a state space model (it is not a linear Gaussian state space model however).

2.3 Dynamic Regression Model

Consider the following model for a response variable Y_t and an explanatory variable x_t (x_t will be treated as deterministic and non-random in the model below) which are both indexed by time $t = 0, 1, \ldots, T$. Dynamic regression models (also known as linear regression with time varying parameters) are of the form:

$$y_t = \alpha_t + \beta_t x_t + \epsilon_t \qquad \text{with } \epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2).$$

The difference with the usual simple linear regression model is that both the intercept and the slope coefficients above are allowed to depend on t. In order to make estimation of this

model feasible, we need further restrictions on $\{\alpha_t\}$ and $\{\beta_t\}$ (otherwise there are just too many parameters in the model). One simple restriction is to assume that:

$$\alpha_t = \alpha_{t-1} + w_{\alpha,t} \qquad \text{with } w_{\alpha,t} \overset{\text{i.i.d}}{\sim} N(0, \sigma_\alpha^2)$$
$$\beta_t = \beta_{t-1} + w_{\beta,t} \qquad \text{with } w_{\beta,t} \overset{\text{i.i.d}}{\sim} N(0, \sigma_\beta^2)$$

Note that this is an example of a state space model with the state variable:

$$S_t = \begin{pmatrix} \alpha_t \\ \beta_t \end{pmatrix}$$

which satisfies

$$S_t = S_{t-1} + \begin{pmatrix} w_{\alpha t} \\ w_{\beta t} \end{pmatrix}$$

which implies that the state process is Markov. Further the observation equation can be written as

$$Y_t = \begin{pmatrix} 1 & x_t \end{pmatrix} S_t + \epsilon_t.$$

Dynamic regression models are used in many regression situations where the response and explanatory variables are collected in time. One example is when y_t gives the returns on a particular stock and x_t gives the average returns of the market. Then the dynamic regression model allows one to study the performance of the stock with respect to the average performance of the market over the course of time.

2.4 Recommended Reading for Today

1. For a description of linear Gaussian state space models, see Section 2.4 of the Petris-Petrone-Campagnoli book and Section 3.1 of the Durbin-Koopman book.
2. Local Linear Model: Section 3.2.1 of the Durbin-Koopman book, Section 11.3 of the Kitagawa book.
3. Stochastic volatility models: page 49 of Petris-Petrone-Campagnoli, Section 2.4.3 of Chopin-Papaspiliopoulos, Section 1.3.3 of Triantafyllopoulos
4. Dynamic linear regression: Section 3.2.7 of Petris-Petrone-Campagnoli and Section 4.1.5 of Triantafyllopoulos.

3 Lecture Three

We are in the midst of looking at different applications of state space models. Our next step is to see that ARMA models are special cases of state space models. We shall first consider the AR(2) model before going to general state space models. Historically, the AR(2) model was introduced in the context of the sunspots data (see the classical 1927 paper titled "On a method of investigating periodicities disturbed series, with special reference to Wolfer's sunspot numbers" by G. Udny Yule or the 2011 book "The Foundations of Modern Time Series Analysis" by T.C. Mills). It is often claimed that (see, for example, `https://en.wikipedia.org/wiki/Sunspot`) the sunspot number varies according to an approximately 11-year cycle. We can verify this by fitting the simple sinusoidal model:

$$Y_i = \mu + \alpha_1 \cos(\omega t_i) + \alpha_2 \sin(\omega t_i) + \epsilon_i \qquad \text{for } i = 1, \ldots, n \tag{7}$$

to the observed data $(t_1, y_1), \ldots, (t_n, y_n)$. Here t_i refers to year i and y_i denotes the average number of sunspots for year t_i. In the dataset (obtained from `https://wwwbis.sidc.be/silso/infosnytot`), we have data for all years from 1700 to 2019. So we are analyzing the whole data, we can take $n = 320$ and $t_1 = 1700, t_2 = 1701, t_3 = 1702, \ldots, t_n = 2019$. In general, it is not necessary to have the observed times t_i to be consecutive (i.e., it is okay for the time series to have some observation gaps).

Today, we shall study the problem of fitting the model (7) and obtaining estimates of the frequency parameter ω from the sunspots data. Note that, if we believe the 11-year cycle for the sunspots data, then we would expect the data to give an estimate of ω (in the model (7)) that is close to $2\pi/11 = 0.5712$. In the next class, we shall see the connection between the model (7) and AR(2).

For the model (7), we shall assume that

$$\epsilon_1, \ldots, \epsilon_n \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$$

which is the most standard distributional assumption for errors. The problem then is to estimate the frequency parameter ω. The other four parameters $\mu, \alpha_1, \alpha_2, \sigma$ are unknown but they are not our main focus (these parameters can be termed *nuisance parameters*). For principled estimation of ω in the presence of the nuisance parameters $\mu, \alpha_1, \alpha_2, \sigma$, we shall take the Bayesian approach with the following natural prior:

$$\omega, \mu, \alpha_1, \alpha_2, \log\sigma \overset{\text{i.i.d}}{\sim} \text{Unif}(-C, C)$$

for a large number C (the exact value of C will not matter in the following calculations). Note that as σ is always positive, we have made the uniform assumption on $\log\sigma$ (by the change of variable formula, we would have $f_\sigma(x) = f_{\log\sigma}(\log x)\frac{1}{x} = \frac{I\{-C<\log x<C\}}{2Cx} = \frac{I\{e^{-C}<x<e^C\}}{2Cx}$.

The posterior for all the unknown parameters $\omega, \mu, \alpha_1, \alpha_2, \log\sigma$ is then (below we write the term "data" for $Y_1 = y_1, \ldots, Y_n = y_n$):

$$f_{\omega,\mu,\alpha_1,\alpha_2,\sigma|\text{data}}(\omega, \mu, \alpha_1, \alpha_2, \sigma) \propto f_{Y_1,\ldots,Y_n|\omega,\mu,\alpha_1,\alpha_2,\sigma}(y_1, \ldots, y_n) f_{\omega,\mu,\alpha_1,\alpha_2,\sigma}(\omega, \mu, \alpha_1, \alpha_2, \sigma).$$

The two terms on the right hand side above are

$$\begin{aligned}
f_{Y_1,\ldots,Y_n|\omega,\mu,\alpha_1,\alpha_2,\sigma}(y_1, \ldots, y_n) &\propto \prod_{i=1}^n f_{Y_i|\omega,\mu,\alpha_1,\alpha_2,\sigma}(y_i) \\
&= \prod_{i=1}^n f_{\epsilon_i|\mu,\sigma,\alpha_1,\alpha_2,\sigma}(y_i - \mu - \alpha_1\cos(\omega t_i) - \alpha_2\sin(\omega t_i)) \\
&= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \mu - \alpha_1\cos(\omega t_i) - \alpha_2\sin(\omega t_i)^2}{2\sigma^2}\right) \\
&\propto \sigma^{-n} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^n (y_i - \mu - \alpha_1\cos(\omega t_i) - \alpha_2\sin(\omega t_i)^2\right),
\end{aligned}$$

and

$$\begin{aligned}
&f_{\omega,\mu,\alpha_1,\alpha_2,\sigma}(\omega, \mu, \alpha_1, \alpha_2, \sigma) = f_\omega(\omega) f_\mu(\mu) f_{\alpha_1}(\alpha_1) f_{\alpha_2}(\alpha_2) f_\sigma(\sigma) \\
&\propto \frac{I\{-C<\omega<C\}}{2C} \frac{I\{-C<\mu<C\}}{2C} \frac{I\{-C<\alpha_1<C\}}{2C} \frac{I\{-C<\alpha_2<C\}}{2C} \frac{I\{e^{-C}<\sigma<e^C\}}{2C\sigma} \\
&\propto \frac{1}{\sigma} I\left\{-C < \omega, \mu, \alpha_1, \alpha_2, \log\sigma < C\right\}.
\end{aligned}$$

We thus obtain

$$f_{\omega,\mu,\alpha_1,\alpha_2,\sigma|\text{data}}(\omega,\mu,\alpha_1,\alpha_2,\sigma)$$
$$\propto \sigma^{-n-1}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu-\alpha_1\cos(\omega t_i)-\alpha_2\sin(\omega t_i)^2\right) I\left\{-C<\omega,\mu,\alpha_1,\alpha_2,\log\sigma<C\right\}.$$

To obtain the posterior density of ω, we simply integrate the above with respect to $\mu,\alpha_1,\alpha_2,\sigma$. Thus for every $\omega\in(-C,C)$,

$$f_{\omega|\text{data}}(\omega)\propto\int_{e^{-C}}^{e^C}\int_{-C}^{C}\int_{-C}^{C}\int_{-C}^{C}\sigma^{-n-1}\exp\left(-\frac{\sum_{i=1}^{n}(y_i-\mu-\alpha_1\cos(\omega t_i)-\alpha_2\sin(\omega t_i)^2}{2\sigma^2}\right)d\mu d\alpha_1 d\alpha_2 d\sigma.$$

When C is large, the above integral is well-approximated by

$$\int_0^\infty\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\sigma^{-n-1}\exp\left(-\frac{\sum_{i=1}^{n}(y_i-\mu-\alpha_1\cos(\omega t_i)-\alpha_2\sin(\omega t_i)^2}{2\sigma^2}\right)d\mu d\alpha_1 d\alpha_2 d\sigma. \tag{8}$$

This integral can be evaluated exactly. The calculation is easiest done using matrix notation. Let

$$Y=\begin{pmatrix}y_1\\ \cdot\\ \cdot\\ \cdot\\ y_n\end{pmatrix}\quad\text{and}\quad X=\begin{pmatrix}1 & \cos(\omega t_1) & \sin(\omega t_1)\\ \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot\\ 1 & cos(\omega t_n) & \sin(\omega t_n)\end{pmatrix}\quad\text{and}\quad\beta=\begin{pmatrix}\mu\\ \alpha_1\\ \alpha_2\end{pmatrix}.$$

With this notation,

$$\sum_{i=1}^{n}(y_i-\mu-\alpha_1\cos(\omega t_i)-\alpha_2\sin(\omega t_i)^2=\|Y-X\beta\|^2.$$

so that (8) is the same as

$$\int_0^\infty\sigma^{-n-1}\int_{\mathbb{R}^3}\exp\left(-\frac{\|Y-X\beta\|^2}{2\sigma^2}\right)d\beta d\sigma \tag{9}$$

Now if $\hat\beta$ is the least squares estimator:

$$\hat\beta:=\underset{\beta}{\text{argmin}}\ \|Y-X\beta\|^2,$$

then

$$\|Y-X\beta\|^2=\|Y-X\hat\beta\|^2+\|X\beta-X\hat\beta\|^2=\|Y-X\hat\beta\|^2+\left(\beta-\hat\beta\right)^T X^TX\left(\beta-\hat\beta\right).$$

The integral (9) then becomes

$$\int_0^\infty\int_{\mathbb{R}^3}\sigma^{-n-1}\exp\left(-\frac{\|Y-X\hat\beta\|^2}{2\sigma^2}\right)\exp\left(-\frac{(\beta-\hat\beta)X'X(\beta-\hat\beta)}{2\sigma^2}\right)d\beta d\sigma$$
$$=\int_0^\infty\sigma^{-n-1}\exp\left(-\frac{\|Y-X\hat\beta\|^2}{2\sigma^2}\right)\int_{\mathbb{R}^3}\exp\left(-\frac{(\beta-\hat\beta)X'X(\beta-\hat\beta)}{2\sigma^2}\right)d\beta d\sigma.$$

We shall now use the formula:

$$\int_{\mathbb{R}^p} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) dx_1 \dots dx_p = (2\pi)^{p/2}\sqrt{\det(\Sigma)}$$

where Σ is a $p \times p$ positive definite matrix and the integral is over $x = (x_1, \dots, x_p)$. This is basically the formula for the normalizing constant for the multivariate normal distribution which we shall study next week.

This formula with $p = 3$ and $\Sigma^{-1} = X'X/(\sigma^2)$ (or equivalently $\Sigma = \sigma^2(X'X)^{-1}$) gives

$$\int_{\mathbb{R}^3} \exp\left(-\frac{(\beta-\hat{\beta})^T X^T X(\beta-\hat{\beta})}{2\sigma^2}\right) d\beta = (2\pi)^{p/2}\sqrt{\det\left(\sigma^2(X'X)^{-1}\right)} = (2\pi)^{p/2}\sigma^p \left(\det(X'X)\right)^{-1/2}.$$

The integral (8) thus equals

$$(2\pi)^{p/2}(\det(X'X))^{-1/2} \int_0^\infty \sigma^{-n+p-1} \exp\left(-\frac{\|Y - X\hat{\beta}\|^2}{2\sigma^2}\right) d\sigma.$$

The change of variable

$$t = \frac{\sigma}{\|Y - X\hat{\beta}\|}$$

then gives

$$\begin{aligned}
&(2\pi)^{p/2}(\det(X'X))^{-1/2} \int_0^\infty \sigma^{-n+p-1} \exp\left(-\frac{\|Y - X\hat{\beta}\|^2}{2\sigma^2}\right) d\sigma \\
&= (2\pi)^{p/2}(\det(X'X))^{-1/2}\|Y - X\hat{\beta}\|^{-n+p} \int_0^\infty t^{-n+p-1} \exp\left(-\frac{1}{2t^2}\right) dt \\
&\propto (\det(X'X))^{-1/2}\|Y - X\hat{\beta}\|^{-n+p}.
\end{aligned}$$

Putting everything together, we have proved that

$$f_{\omega|\text{data}}(\omega) \propto (\det(X'X))^{-1/2}\|Y - X\hat{\beta}\|^{-n+p}.$$

Note that the right hand side depends crucially on ω because X depends on ω. Also $\hat{\beta}$ depends on X as $\hat{\beta} = (X'X)^{-1}X'Y$. To make this explicit, let us write $X(\omega)$ for X and $\hat{\beta}(\omega)$ for $\hat{\beta}$:

$$f_{Y_1,\dots,Y_n|\omega}(y_1, \dots, y_n) \propto \left(\det(X(\omega)'X(\omega))\right)^{-1/2} \|Y - X(\omega)\hat{\beta}(\omega)\|^{-(n-p)}. \tag{10}$$

This function of ω can be plotted on the computer (and normalized so the density integrates to one). Note that $p = 3$. This allows inference on ω based on the data.

3.1 Connection to the Periodogram

It turns out the Bayesian posterior (10) can be related to the periodogram which is a standard object in time series analysis. The periodogram corresponding to the time series data (t_i, y_i) is defined as

$$I(\omega) := \frac{1}{n}\left[\left(\sum_j y_j \cos(\omega t_j)\right)^2 + \left(\sum_j y_j \sin(\omega t_j)\right)^2\right]. \tag{11}$$

This is a function of $\omega \in \mathbb{R}$. Usually, the periodogram is computed for uniformly spaced data (where the time points t_j can be taken to be consecutive integers such as $0, \ldots, n-1$) and when ω is of the form $\frac{2\pi k}{n}$ for some integer $k \in \{1, \ldots, n-1\}$. These values of ω are known as *Fourier Frequencies*. Observe that $I(\omega)$ can also be written as

$$I(\omega) = \frac{1}{n}\left|\sum_j y_j e^{i\omega t_j}\right|^2$$

where $i = \sqrt{-1}$, $e^{i\omega t_j}$ is the complex number $\cos(\omega t_j) + i\sin(\omega t_j)$ and $|z|$ for a complex number z denotes its modulus. The complex number

$$b(\omega) := \sum_j y_j e^{i\omega t_j}$$

is termed the Discrete Fourier Transform of the data when $t_j = j-1$ and ω ranges over the Fourier frequencies. Thus, the periodogram is basically the squared modulus of the DFT (scaled by n).

It is a standard procedure to look at the periodogram of an observed time series in order to determine periodic components present in the data. It turns out that the Bayesian posterior (10) is related to the periodogram as we shall argue below. To see this, first note that the posterior (10) is described in terms of the matrix $X(\omega)$. For this matrix, it is easy to see that

$$X'(\omega)X(\omega) = \begin{pmatrix} n & \sum_{j=1}^n \cos(\omega t_j) & \sum_{j=1}^n \sin(\omega t_j) \\ \sum_{j=1}^n \cos(\omega t_j) & \sum_{j=1}^n \cos^2(\omega t_j) & \sum_{j=1}^n \cos(\omega t_j)\sin(\omega t_j) \\ \sum_{j=1}^n \sin(\omega t_j) & \sum_{j=1}^n \cos(\omega t_j)\sin(\omega t_j) & \sum_{j=1}^n \sin^2(\omega t_j). \end{pmatrix}$$

Quite often, this $X'(\omega)X(\omega)$ matrix can be well-approximated as

$$n\begin{pmatrix} 1 & \frac{1}{n}\sum_{j=1}^n \cos(\omega t_j) & \frac{1}{n}\sum_{j=1}^n \sin(\omega t_j) \\ \frac{1}{n}\sum_{j=1}^n \cos(\omega t_j) & \frac{1}{n}\sum_{j=1}^n \cos^2(\omega t_j) & \frac{1}{n}\sum_{j=1}^n \cos(\omega t_j)\sin(\omega t_j) \\ \frac{1}{n}\sum_{j=1}^n \sin(\omega t_j) & \frac{1}{n}\sum_{j=1}^n \cos(\omega t_j)\sin(\omega t_j) & \frac{1}{n}\sum_{j=1}^n \sin^2(\omega t_j). \end{pmatrix} = n\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/2 \end{pmatrix}$$

To see this, consider the case where the time points are consecutive in which case we take $t_j = j-1$. Then for a wide range of ω, we will argue that

$$\begin{aligned} &\frac{1}{n}\sum_{j=1}^n \cos(\omega t_j) \approx 0 \quad \frac{1}{n}\sum_{j=1}^n \sin(\omega t_j) \approx 0 \\ &\frac{1}{n}\sum_{j=1}^n \cos^2(\omega t_j) \approx \frac{1}{2} \quad \frac{1}{n}\sum_{j=1}^n \sin^2(\omega t_j) \approx \frac{1}{2} \\ &\frac{1}{n}\sum_{j=1}^n \cos(\omega t_j)\sin(\omega t_j) \approx 0 \end{aligned}$$

Let me provide the argument for one of the above assertions. The argument for the others is similar. We shall consider the assertion

$$\frac{1}{n}\sum_{j=1}^n \cos^2(\omega t_j) \approx \frac{1}{2}. \tag{12}$$

To see this, write

$$\frac{1}{n}\sum_{j=1}^{n}\cos^2(\omega t_j) = \frac{1}{n}\sum_{j=1}^{n}\frac{1+\cos(2\omega t_j)}{2}$$
$$= \frac{1}{2} + \frac{1}{2n}\sum_{j=1}^{n}\cos(2\omega t_j) = \frac{1}{2} + \frac{1}{4n}\sum_{j=1}^{n}e^{2i\omega t_j} + \frac{1}{4n}\sum_{j=1}^{n}e^{-2i\omega t_j}$$

The sums above can be evaluated explicitly under the assumption that the times are uniformly spaced $t_j = j - 1$:

$$\sum_{j=1}^{n}e^{2i\omega t_j} = \sum_{j=1}^{n}e^{2i\omega(j-1)} = \sum_{j=1}^{n}(e^{2i\omega})^{j-1} = \frac{e^{2i\omega n}-1}{e^{2i\omega}-1},$$

and similarly

$$\sum_{j=1}^{n}e^{-2i\omega t_j} = \frac{e^{-2i\omega n}-1}{e^{-2i\omega}-1}.$$

Now if ω is a Fourier frequency of the form $\omega = 2\pi k/n$, then $e^{\pm 2i\omega n} = e^{\pm 4i\pi k} = \cos(4\pi k) \pm i\sin(4\pi k) = 1$ so the above displayed sums are zero leading to (12). If ω is not a Fourier frequency, we can write

$$\left|\frac{1}{4n}\sum_{j=1}^{n}e^{2i\omega t_j}\right| = \left|\frac{1}{4n}\frac{e^{2i\omega n}-1}{e^{2i\omega}-1}\right| \leq \frac{1}{2n|e^{2i\omega}-1|}$$

because $|e^{2i\omega n} - 1| \leq |e^{2i\omega n}| + 1 \leq 2$. We can thus ignore this term if $n|e^{2i\omega} - 1|$ is large. Similarly the term

$$\frac{1}{4n}\sum_{j=1}^{n}e^{-2i\omega t_j}$$

can be ignored if $n|e^{-2i\omega} - 1|$ is large. The assertion (12) is therefore justified if $n|e^{\pm 2i\omega} - 1|$ is large (which will often be the case unless ω is too close to zero).

In the rest of this section, we shall assume that

$$X'(\omega)X(\omega) \approx \begin{pmatrix} n & 0 & 0 \\ 0 & n/2 & 0 \\ 0 & 0 & n/2 \end{pmatrix}$$

Under this condition, the integral (9) can be evaluated in the following alternative way. We start with

$$\|Y - X\beta\|^2 = Y'Y - 2Y'X\beta + \beta'X'X\beta$$
$$= \sum_i y_i^2 - 2\sum_{i=1}^{n} y_i(\mu + \alpha_1\cos(\omega t_i) + \alpha - 2\sin(\omega t_i)) + n\mu^2 + \frac{n}{2}\alpha_1^2 + \frac{n}{2}\alpha_2^2$$
$$= \sum_i y_i^2 - 2\mu\sum_i y_i + n\mu^2 - 2\alpha_1\sum_i y_i\cos(\omega t_i) + \frac{n}{2}\alpha_1^2 - 2\alpha_2\sum_i y_i\sin(\omega t_i) + \frac{n}{2}\alpha_2^2$$

Thus the inner integral over $\mathbb{R}^3$ in (9) can be broken down into 3 one dimensional integrals (as opposed to one three-dimensional integral) as

$$\begin{aligned}
&\int_{\mathbb{R}^3} \exp\left(-\frac{\|Y - X\beta\|^2}{2\sigma^2}\right) d\beta \\
&= \exp\left(-\frac{\sum_i y_i^2}{2\sigma^2}\right) \left[\int \exp\left(\frac{\mu \sum_i y_i}{\sigma^2} - \frac{n\mu^2}{2\sigma^2}\right) d\mu\right] \left[\int \exp\left(\frac{\alpha_1 \sum_i y_i \cos(\omega t_i)}{\sigma^2} - \frac{n\alpha_1^2}{4\sigma^2}\right) d\alpha_1\right] \\
&\left[\int \exp\left(\frac{\alpha_2 \sum_i y_i \sin(\omega t_i)}{\sigma^2} - \frac{n\alpha_2^2}{4\sigma^2}\right) d\alpha_2\right]
\end{aligned}$$

can be evaluated in the following alternative way. Each of the above three integrals can be evaluated explicitly using the one-dimensional integration formula:

$$\int_{-\infty}^{\infty} \exp\left(xC_1 - \frac{C_2}{2}x^2\right) dx = \sqrt{\frac{2\pi}{C_2}} \exp\left(\frac{C_1^2}{2C_2}\right).$$

We thus deduce

$$\begin{aligned}
&\int_{\mathbb{R}^3} \exp\left(-\frac{\|Y - X\beta\|^2}{2\sigma^2}\right) d\beta \\
&\propto \exp\left(-\frac{\sum_i y_i^2}{2\sigma^2}\right) \sigma^3 \exp\left(\frac{(\sum_i y_i)^2}{2n\sigma^2}\right) \exp\left(\frac{(\sum_i y_i \cos(\omega t_i))^2}{2\sigma^2}\right) \exp\left(\frac{(\sum_i y_i \sin(\omega t_i))^2}{2\sigma^2}\right).
\end{aligned}$$

Finally the integration over σ can be done as before to obtain

$$\begin{aligned}
f_{\omega|\text{data}}(\omega) &\propto \left[\frac{\sum_i y_i^2}{2} - \frac{(\sum_i y_i)^2}{2n} - \frac{1}{n}\left(\sum_i y_i \cos(\omega t_i)\right)^2 - \frac{1}{n}\left(\sum_i y_i \sin(\omega t_i)\right)^2\right]^{-(n-p)/2} \\
&= \left[\frac{\sum_i (y_i - \bar{y})^2}{2} - \frac{1}{n}\left(\sum_i y_i \cos(\omega t_i)\right)^2 - \frac{1}{n}\left(\sum_i y_i \sin(\omega t_i)\right)^2\right]^{-(n-p)/2}
\end{aligned}$$

Using the periodogram formula (11), we can write the above as

$$\begin{aligned}
f_{\omega|\text{data}}(\omega) &\propto \left[\frac{\sum_i (y_i - \bar{y})^2}{2} - I(\omega)\right]^{-(n-p)/2} \\
&\propto \left[1 - \frac{2I(\omega)}{\sum_{i=1}^n (y_i - \bar{y})^2}\right]^{-(n-p)/2}.
\end{aligned}$$

Thus the Bayesian posterior for ω is essentially a function of the periodogram (and the sample variance of the data). But it is important to note that it is a very specific function which can look quite different from the raw periodogram. For example, for the sunspots dataset, the periodogram has several peaks but the Bayesian posterior is typically quite strongly concentrated. Thus if we are trying to find a single frequency in a time series dataset, the Bayesian posterior will provide that information much more precisely compared to the periodogram.

3.2 Recommended Reading for Today

1. The Bayesian analysis of the model (7) is taken from the book *Bayesian spectrum analysis and parameter estimation* by Larry Bretthorst (available freely online). You can read Chapters 1 and 2 of that book.

2. The periodogram is a standard object in time series analysis and it can be found in many books; see for example Chapter 4 of the book *Time series analysis and its applications* by Shumway and Stoffer (note that some authors use slightly different scaling factors while defining the periodogram).

4 Lecture Four

In the last class, we studied the model

$$Y_i = \mu + \alpha_1 \cos(\omega t_i) + \alpha_2 \sin(\omega t_i) + \epsilon_i \qquad \text{for } i = 1, \ldots, n \tag{13}$$

for the sunspots dataset. We used a Bayesian method to infer the frequency parameter ω (which is the main parameter of interest) and this led to an estimated period of close to 11 (which is often cited as the period of the solar cycle). Note however that (13) is not ideal for the sunspots dataset for at least two reasons: (a) the fit to the data is not very good (some of the oscillations have a much higher amplitude than that explained by the single sinusoid), (b) data generated from the model (13) look much more "noisy" compared to the actual sunspots data. Starting with these observations, Yule (1927) proposed an alternative model that is also based on a single sinuosoid. This is the topic of this lecture.

Yule started with the following basic observation. Let s_t denote the sinusoid:

$$s_t = \mu + \alpha_1 \cos(\omega t) + \alpha_2 \sin(\omega t) \tag{14}$$

The same sinusoid can be understood as the solution to a specific difference equation. To derive the difference equation, let us first note that, in continuous time, $s(t)$ satisfies

$$s''(t) = -\omega^2 \left(\alpha_1 \cos(\omega t) + \alpha_2 \sin(\omega t)\right) = -\omega^2 \left(s(t) - \mu\right). \tag{15}$$

In discrete time (where $t \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$), the sequence (14) satisfies the following difference equation that is analogous to (15):

$$s_{t+2} - 2s_{t+1} + s_t = 2(\cos\omega - 1)\left(s_{t+1} - \mu\right). \tag{16}$$

To see this, note that

$$\begin{aligned} &s_{t+2} - 2s_{t+1} + s_t \\ &= \alpha_1 \left(\cos(\omega(t+2)) - 2\cos(\omega(t+1)) + \cos(\omega t)\right) + \alpha_2 \left(\sin(\omega(t+2)) - 2\sin(\omega(t+1)) + \sin(\omega t)\right) \end{aligned}$$

Writing $A = \omega(t+1)$ and $B = \omega$, we get

$$\begin{aligned} \cos(\omega(t+2)) - 2\cos(\omega(t+1)) + \cos(\omega t) &= \cos(A+B) - 2\cos A + \cos(A-B) \\ &= 2\cos A(\cos B - 1) \\ &= 2(\cos\omega - 1)\cos(w(t+1)) \end{aligned}$$

and similarly

$$\sin(\omega(t+2)) - 2\sin(\omega(t+1)) + \sin(\omega t) = 2(\cos\omega - 1)\sin(\omega(t+1))$$

This proves

$$s_{t+2} - 2s_{t+1} + s_t = 2(\cos\omega - 1)\left(\alpha_1 \cos(\omega(t+1)) + \alpha_2 \sin(\omega(t+1))\right) = 2(\cos\omega - 1)(s_{t+1} - \mu)$$

and this proves (16).

The converse is also true in the sense that every solution $\{s_t\}$ to the difference equation (16) say, for $t = 0, 1, 2, \ldots$, with given values of s_0 and s_1 (initial conditions) is of the form (14) for some α_1 and α_2. To see this, let $g_t = s_t - \mu$ and note that $\{g_t\}$ satisfies

$$g_{t+2} - 2g_{t+1} + g_t = 2(\cos\omega - 1)g_{t+1}.$$

We find α_1 and α_2 such that

$$h_t := \alpha_1 \cos(\omega t) + \alpha_2 \sin(\omega t)$$

matches g_t for $t = 0, 1$. Now if $g_t = h_t$ and $g_{t+1} = h_{t+1}$, then

$$\begin{aligned}
g_{t+2} &= (2\cos\omega)g_{t+1} - g_t \\
&= (2\cos\omega)\left(\alpha_1\cos(\omega(t+1)) + \alpha_2\sin(\omega(t+1))\right) - \left(\alpha_1\cos(\omega t) + \alpha_2\sin(\omega t)\right) \\
&= \alpha_1\left(2\cos\omega\cos(\omega(t+1)) - \cos(\omega t)\right) + \alpha_2\left(2\cos\omega\sin(\omega(t+1)) - \sin(\omega t)\right) \\
&= \alpha_1\left(\cos(\omega t)\left(2\cos^2\omega - 1\right) - \sin(\omega t)2\sin\omega\cos\omega\right) \\
&+ \alpha_2\left(\sin(\omega t)\left(2\cos^2\omega - 1\right) + \cos(\omega t)2\sin\omega\cos\omega\right) \\
&= \alpha_1\left(\cos(\omega t)\cos(2\omega) - \sin(\omega t)\sin(2\omega)\right) + \alpha_2\left(\sin(\omega t)\cos(2\omega) + \cos(\omega t)\sin(2\omega)\right) \\
&= \alpha_1\cos(\omega(t+2)) + \alpha_2\sin(\omega(t+2)) = h_{t+2}.
\end{aligned}$$

Using this for $t = 0, 1, 2, \ldots$ proves that (14) is the unique solution to (16).

To summarize, an alternative way of describing a sinusoid of frequency ω is via the difference equation (16) which is equivalent to

$$s_{t+2} = (2\cos\omega)s_{t+1} - s_t + 2(1 - \cos\omega)\mu.$$

Based on this equation, Yule proposed the model:

$$Y_{t+2} = \theta Y_{t+1} - Y_t + c + Z_{t+2} \tag{17}$$

with two parameters θ and c (and the additional noise parameter σ in $Z_{t+2} \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$). Note that this is also a single sinusoid plus noise model but now the noise is in a different place. To better understand the difference between (17) and the earlier model:

$$Y_t = \mu + \alpha_1\cos(\omega t) + \alpha_2\sin(\omega t) + \epsilon_t, \tag{18}$$

consider the following physical situation where sinusoids naturally arise (see e.g., page 2 of the Fourier Analysis book by Stein and Shakarchi). Consider a mass m that is attached to a horizontal spring, which itself is attached to fixed wall, and assume that the system lies on a frictionless surface. Choose an axis whose origin coincides with the center of the mass when the spring is neither compressed or stretched. When the spring is compressed or stretched and released, the mass undergoes simple harmonic motion.

Let $y(t)$ denote the displacement of the mass at time t. Hooke's law says that the force exerted by the spring on the mass is given by $F = -\kappa y(t)$ where $\kappa > 0$ is the spring constant. By Newton's law (note that the acceleration is given by $y''(t)$), we have

$$-\kappa y(t) = m y''(t)$$

This is same as

$$y''(t) = -\omega^2 y(t) \qquad \text{where } \omega := \sqrt{\frac{k}{m}}$$

whose general solution is the sinusoid $\alpha_1 \cos(\omega t) + \alpha_2 \sin(\omega t)$. In the context of this physical situation, the two different sinusoid plus models ((18) and (17)) can be understood as follows. We are taking measurements of the displacement Y_t at various times t.

Model (18): Here our measurements are noisy and every measurement is corrupted by an unknown noise which we are terming ϵ_t and modeling as $N(0, \sigma^2)$.

Model (17): Here there is no measurement error and our measurement mechanism is perfect. However the actual oscillation of the mass is not perfectly sinusoidal and is affected by noise. For example, imagine, as Yule put it, that some kids are randomly throwing stones at the mass (sometimes from the left and sometimes from the right) while it is oscillating.

It is very interesting to note that observations generated from Model (17) are much smoother compared to observations generated from Model (18). Yule used this to argue that (17) is a better model for the sunspots data compared to (18).

It is natural to wonder if it makes sense to incorporate both kinds of errors simultaneously (measurement errors and errors affecting the oscillation). This leads to the model:

$$\begin{aligned} X_t &= \theta X_{t-1} - X_{t-2} + c + Z_t \\ Y_t &= X_t + \epsilon_t \end{aligned}$$

This is a state space model if we take the state variable to be

$$\tilde{X}_t = \begin{pmatrix} X_t \\ X_{t-1} \end{pmatrix}$$

because the state evolution

$$\tilde{X}_{t+1} = \begin{pmatrix} c \\ 0 \end{pmatrix} + \begin{pmatrix} \theta & -1 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} Z_t \\ 0 \end{pmatrix}$$

is Markov, and the observations are

$$Y_t = \begin{pmatrix} 1 & 0 \end{pmatrix} \tilde{X}_t + \epsilon_t.$$

4.1 The Autoregressive Model

Yule (1927) also fit models to the sunspots dataset that are more complicated compared to (17) and introduced the Autoregressive Model (of order 2) in this process. The AR(2) model is given by

$$Y_{t+2} = \phi_1 Y_{t+1} + \phi_2 Y_t + c + Z_{t+2} \tag{19}$$

Note that (17) can be seen as a simpler version of the above model where the ϕ_2 parameter is set to the value -1. We can fit this model to the observed sunspots data $y_1, \ldots, y_T$ to obtain parameter estimates $\hat{c}, \hat{\phi}_1, \hat{\phi}_2$ and $\hat{\sigma}$ of the model parameters. Using the fitted model, future values can be predicted by recursing the equation:

$$Y_t = \hat{c} + \hat{\phi}_1 Y_{t-1} + \hat{\phi}_2 Y_{t-2} \qquad \text{for } t = T+1, T+2, \ldots$$

with Y_T and Y_{T-1} set to the observed values y_T and y_{T-1} respectively. For the sunspots data, these predictions follow a *damped* sinusoid. Indeed, fitting the AR(2) model to the sunspots data for the time period $1700 - 1969$ led to the model:

$$Y_{t+2} = 23.92 + 1.38 Y_{t+1} - 0.69 Y_t + Z_{t+2}$$

which gives the prediction equation:

$$Y_t = 23.92 + 1.38Y_{t-1} - 0.69Y_{t-2}$$

for the future values of sunspot numbers from 1970 onwards. This equation can also be written as

$$Y_t - 77.16 = 1.38\,(Y_{t-1} - 77.16) - 0.69\,(Y_{t-2} - 77.16)$$

Thus the predictions for $U_t := Y_t - 77.16$ are given by recursing the equation:

$$U_t = 1.38U_{t-1} - 0.69U_{t-2} \tag{20}$$

for $t = T+1, T+2, \ldots$ (note that U_{T-1} and U_T are observed from the data). It follows from the following fact that the general solution of (20) is of the form:

$$c_1\,(1.2)^{-t}\cos\,(0.59t + c_2)$$

for two constants c_1 and c_2. The above is clearly a damped sinusoid (the sinusoid $\cos\,(0.59t + c_2)$ is damped by the factor $(1.2)^{-t}$).

Fact 4.1. *Consider the difference equation*

$$U_t = \phi_1 U_{t-1} + \phi_2 U_{t-2} \qquad \text{for } t \in \{k+1, \ldots\}. \tag{21}$$

with initial conditions $U_{k-1} = \alpha$ and $U_k = \beta$. Suppose that the quadratic polynomial

$$1 - \phi_1 z - \phi_2 z^2$$

has complex roots z_1 and z_2. As ϕ_1 and ϕ_2 are real, z_1 and z_2 must be complex conjugates of each other so they can be written as $re^{i\theta}$ and $re^{-i\theta}$ for some $r > 0$ and $\theta \in \mathbb{R}$ (here $i = \sqrt{-1}$). Then the solution to (21) is of the form:

$$U_t = c_1 r^{-t}\cos\,(\theta t + c_2) \qquad \text{for } t = k+1, k+2, \ldots \tag{22}$$

for some constants c_1 and c_2.

Proof. Let $H_t = c_1 r^{-t}\cos\,(\theta t + c_2)$. We find c_1 and c_2 such that $H_t = U_t$ for $t = k-1$ and $t = k$. Then observe that

$$\begin{aligned} H_t &= c_1 r^{-t}\cos(\theta t + c_2) \\ &= 2c_1 r^{-t}\left(e^{i\theta t}e^{ic_2} + e^{-i\theta t}e^{-ic_2}\right) \\ &= 2c_1 e^{ic_2}(re^{-i\theta})^{-t} + 2c_2 e^{-ic_2}(re^{i\theta})^{-t} \\ &= 2c_1 e^{ic_2} z_1^{-t} + 2c_1 e^{-ic_2} z_2^{-t}. \end{aligned}$$

This gives

$$\begin{aligned} H_t - \phi_1 H_{t-1} - \phi_2 H_{t-2} &= 2c_1 e^{ic_2}\left(z_1^{-t} - \phi_1 z_1^{-t+1} - \phi_2 z_1^{-t+2}\right) + 2c_1 e^{-ic_2}\left(z_2^{-t} - \phi_1 z_2^{-t+1} - \phi_2 z_2^{-t+2}\right) \\ &= 2c_1 e^{ic_2} z_1^{-t}\left(1 - \phi_1 z_1 - \phi_2 z_1^2\right) + 2c_1 e^{-ic_2} z_2^{-t}\left(1 - \phi_1 z_2 - \phi_2 z_2^2\right) = 0 \end{aligned}$$

because $1 - \phi_1 z_1 - \phi_2 z_1^2 = 1 - \phi_1 z_2 - \phi_2 z_2^2 = 0$ as z_1 and z_2 are roots of the polynomial $1 - \phi_1 z - \phi_2 z^2$. Thus H_t satisfies the given difference equation and it matches U_t for $t = k-1, k$ which implies that it matches U_t for all $t \geq k+1$. □

4.2 Recommended Reading for Today

1. A very nice account of Yule's influential 1927 paper is Chapter 6 of the 2011 book "The Foundations of Modern Time Series Analysis" by T. C. Mills. (available for free from the library website).

2. Section 3.4 of the Durbin-Koopman book writes ARMA and ARIMA models in state space form.

5 Lecture Five

We start today with the problem of fitting state space models to observed time series data. This is the main topic of the class. We shall denote the observed time series data by $y_0, y_1, \ldots, y_T$ (note that the number of observations is $T+1$). We shall assume that the observations are realizations of random variables $Y_0, Y_1, \ldots, Y_T$. State space models describe the distribution of $Y_0, \ldots, Y_T$ in terms of a hidden set of state random variables $X_0, \ldots, X_T$. The joint density of $X_0, \ldots, X_T, Y_0, \ldots, Y_T$ is given by

$$f_{X_0}(x_0) \prod_{t=1}^{T} f_{X_t|X_{t-1}=x_{t-1}}(x_t) \prod_{t=0}^{T} f_{Y_t|X_t=x_t}(y_t).$$

As we have seen previously, this means that $X_0, \ldots, X_T$ is Markov, and also that $Y_0, \ldots, Y_T$ are independent conditional on $X_0 = x_0, \ldots, X_T = x_T$ with

$$Y_t \mid X_0 = x_0, \ldots, X_T = x_T \stackrel{\mathrm{d}}{=} Y_t \mid X_t = x_t.$$

Often, in actual specifications of state space models, the description of the conditional densities $f_{X_0}, f_{X_t|X_{t-1}}, f_{Y_t|X_t}$ depends on additional unknown parameters θ (for example, in the local level model, $\theta = (\sigma_\eta^2, \sigma_\epsilon^2)$ where σ_η^2 and σ_ϵ^2 are the state and observation error variances). We shall follow the full Bayesian approach in the treatment of these nuisance parameters θ. Specifically, we shall assume that they are random and employ a (usually diffuse) prior density $f_\theta(\cdot)$. From now on, we shall explicitly acknowledge that $f_{X_0}, f_{X_t|X_{t-1}}, f_{Y_t|X_t}$ depend on θ by using the notation:

$$f_{X_0|\theta}, f_{X_t|X_{t-1},\theta}, f_{Y_t|X_t,\theta}.$$

Our main goal is to fit the state space model to the observed data $y_0, \ldots, y_T$. Fitting a model in the Bayesian context means computing the conditional distribution of the unknown parameters of the model given the observed data $Y_0 = y_0, \ldots, Y_T = y_T$. For the state space model, the unknown parameters are $X_0, \ldots, X_T$ as well as θ. The conditional distribution of $X_0, \ldots, X_T, \theta$ given the observed data $Y_0 = y_0, \ldots, Y_T = y_T$ equals

$$\begin{aligned}
&f_{X_0,\ldots,X_T,\theta|Y_0=y_0,\ldots,Y_T=y_t}(x_0, \ldots, x_t, \theta) \\
&\propto f_{X_0,\ldots,X_T,Y_0,\ldots,Y_T,\theta}(x_0, \ldots, x_T, y_0, \ldots, y_T, \theta) \\
&= f_{X_0,\ldots,X_T,Y_0,\ldots,Y_T|\theta}(x_0, \ldots, x_T, y_0, \ldots, y_T) f_\theta(\theta) \\
&= f_{X_0|\theta}(x_0) \prod_{t=1}^{T} f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) \prod_{t=0}^{T} f_{Y_t|X_t=x_t,\theta}(y_t) f_\theta(\theta).
\end{aligned}$$

With the normalizing constant,

$$\begin{aligned}
&f_{X_0,\ldots,X_T,\theta|Y_0=y_0,\ldots,Y_T=y_t}(x_0, \ldots, x_t, \theta) \\
&= \frac{f_{X_0|\theta}(x_0) \prod_{t=1}^{T} f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) \prod_{t=0}^{T} f_{Y_t|X_t=x_t,\theta}(y_t) f_\theta(\theta)}{\int \cdots \int \int f_{X_0|\theta}(x_0) \prod_{t=1}^{T} f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) \prod_{t=0}^{T} f_{Y_t|X_t=x_t,\theta}(y_t) f_\theta(\theta) dx_0 \ldots dx_T d\theta}
\end{aligned} \tag{23}$$

This is a high-dimensional density which, in principle, answers any inferential question about the unknown parameters $X_0, \ldots, X_T, \theta$ based on the data $Y_0 = y_0, \ldots, Y_T = y_T$. In practice, the quantities of main interest would be the conditional densities of each individual state conditioned on the data $Y_0 = y_0, \ldots, Y_T = y_T$:

$$X_t \mid Y_0 = y_0, \ldots, Y_T = y_T \qquad \text{for } t = 0, 1, \ldots, T.$$

In principle it is possible to deduce

$$f_{X_t|Y_0=y_0,\ldots,Y_T=y_T}(x_t) \tag{24}$$

from the full posterior (23). But a naive way of doing this would involve high dimensional integration that would not be computationally feasible. We shall study principled computationally feasible algorithms for obtaining (24) for $t = 0, \ldots, T$. I will give a high level overview of the main ideas in this class and we shall study full details in the coming classes. The first step is to write (24) as

$$f_{X_t|Y_0=y_0,\ldots,Y_T=y_T}(x_t) = \int f_{X_t|Y_0=y_0,\ldots,Y_T=y_T,\theta}(x_t) f_{\theta|Y_0=y_0,\ldots,Y_T=y_T}(\theta) d\theta.$$

This implies that the task of calculating (24) can be broken down into the following two subtasks:

1. Calculate $f_{X_t|Y_0=y_0,\ldots,Y_T=y_T,\theta}(x_t)$. This is the conditional density of X_t given the entire data as well as θ.

2. Calculate $f_{\theta|Y_0=y_0,\ldots,Y_T=y_T}(\theta)$. This is the conditional density of θ given the entire data. This can be done via calculating the likelihood $f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T)$ because, by Bayes rule,
$$f_{\theta|Y_0=y_0,\ldots,Y_T=y_T}(\theta) \propto f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T) f_\theta(\theta)$$

It is convenient here to introduce some standard terminology. The conditional distributions:

$$X_t \mid Y_0 = y_0, \ldots, Y_T = y_T, \theta \qquad \text{for } t = 0, 1, \ldots, T \tag{25}$$

are called **smoothing distributions**. Thus, the conditional densities (24) can be determined from the smoothing distributions as well as the likelihood $f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T)$.

5.1 Outline of Approach to Calculate Smoothing Distributions

The approach that we will use for efficiently calculating all the smoothing distributions i.e., all the conditional distributions (25) for $t = 0, 1, \ldots, T$ is the following. This is a sequential approach that has the following two steps:

1. The first step calculates the distributions:
$$X_t \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta \tag{26}$$
for each $t = 0, 1, \ldots, T$. Note that the conditioning above is on $Y_0, \ldots, Y_t$ and not on the whole data $Y_0, \ldots, Y_T$. These conditional distributions are known as **Filtering Distributions** and algorithms for calculating them are called **Filtering Algorithms**. We shall study the standard filtering algorithms: Kalman filter (for Linear Gaussian State Space Models) and Partile filter (for arbitrary state space models). These algorithms calculate the filtering densities recursively starting from $t = 0$ and then for $t = 1, \ldots, T$.

2. After the filtering step, the last smoothing distribution:

$$X_T \mid Y_0 = y_0, Y_1 = y_1, \ldots, Y_T = y_T, \theta$$

is already available. From here, the idea is to calculate the rest of the smoothing distributions (25) recursively for $t = T-1, T-2, \ldots, 0$. Note that this is a backward recursion.

This overall approach to calculating the smoothing distributions is known as FFBS (Forward Filtering and Backward Smoothing). We shall study this approach in the case of general state space models. For the special case of linear Gaussian state space models, these recursions can be solved in closed form.

5.2 Linear Gaussian State Space Models

A state space model is specified by the densities $f_{X_0|\theta}(x_0)$, $f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t)$ for $t = 1, \ldots, T$ and $f_{Y_t|X_t=x_t}(y_t)$ for $t = 0, \ldots, T$. We say that a state space model is **Linear Gaussian** if the following three conditions are all satisfied:

1. $f_{X_0|\theta}(\cdot)$ is a Gaussian density.
2. $f_{X_t|X_{t-1}=x_{t-1}}(\cdot)$ is a Gaussian density whose mean is a linear function of x_{t-1} and whose covariance does not depend on x_{t-1}.
3. $f_{Y_t|X_t=x_t}(\cdot)$ is a Gaussian density whose mean is a linear function of x_t and whose covariance does not depend on x_t.

Quite often, linear Gaussian state space models are specified as:

$$\begin{aligned} X_0 &\sim N(0, \Sigma_0) \\ X_t &= A_t X_{t-1} + U_t \\ Y_t &= B_t X_t + V_t \end{aligned}$$

where $X_0, U_1, \ldots, U_T, V_0, \ldots, V_T$ are independent with

$$U_t \sim N(0, \Sigma_t) \quad \text{and} \quad V_t \sim N(0, R_t).$$

It is easy to see that this specification satisfies the three conditions of the Linear Gaussian State Space Model. The quantities $\Sigma_0, A_t, B_t, \Sigma_t, R_t$ all potentially depend on unknown parameters θ.

For the linear Gaussian state space model, all the filtering and smoothing distributions turn out to be Gaussian which means that they are specified by means and covariances. The general approach for filtering and smoothing can be specialized to this case as recursions in terms of means and covariances. This leads to the Kalman Filter and Kalman Smoother algorithms. We shall start our study of these in the next class.

5.3 Recommended Reading for Today

1. Definitions of filtering and smoothing distributions and the general problem of Sequential Analysis of State Space Models is described in Section 2.3 of the Chopin-Papaspiliopoulos book.
2. Chapter 1 of the Särkkä book also describes the main goals in the analysis of state space models and gives a list of the common Filtering and Smoothing algorithms.

6 Lecture Six

The goal of today's lecture is to study the general filtering algorithm and then specialize it to the case of linear Gaussian State Space Models leading to the Kalman Filter.

Let us recall the basic setup. We have a state space model describing the distribution of random variables $X_0, Y_0, X_1, Y_1, \ldots, X_T, Y_T$ as

$$f_{X_0|\theta}(x_0) \prod_{t=1}^{T} f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) \prod_{t=0}^{T} f_{Y_t|X_t=x_t,\theta}(y_t)$$

Here $f_{X_0|\theta}$ is the density of X_0, $f_{X_t|X_{t-1}=x_{t-1},\theta}$ is the conditional density of X_t given $X_{t-1} = x_{t-1}$ and $f_{Y_t|X_t=x_t,\theta}$ is the conditional density of Y_t given $X_t = x_t$. Throughout there is additional conditioning on θ.

Our aim is to calculate the conditional distributions:

$$X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$$

for various values of s and t. These conditional distributions have known by different names depending on the specific values of s and t:

1. **Filtering Distributions**: These correspond to $s = t$.
2. **Smoothing Distributions**: These correspond to $s \leq t$.
3. **Prediction Distributions**: These correspond to $s > t$.

The importance of calculating these three types of conditional distributions varies with the application. In tracking applications, interest mainly lies in filtering and prediction distributions while in applications such as trend estimation, interest mainly lies in smoothing and prediction distributions.

6.1 General Approach for calculating Filtering Distributions

Let us now study the general recursive scheme for calculating the filtering distributions. The main step is to go from the filtering density at time $t-1$:

$$f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1})$$

to the filtering density at time t:

$$f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},Y_t=y_t.\theta}(x_t)$$

This recursion is carried out in two steps:

1. **Step One**: Go from the filtering density $f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1})$ at time $t-1$ to the one-step ahead prediction density $f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t)$ at time $t-1$. This step is known as the *one-step prediction update*.
2. **Step Two**: Go from the one-step ahead prediction density $f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t)$ at time $t-1$ to the filtering density $f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t)$ at time t. This step is known as the *filtering update*.

The one-step ahead prediction update is carried out via the formula:

$$f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t)$$
$$= \int f_{X_t|X_{t-1}=x_{t-1},Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t) f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) dx_{t-1}$$

Now by the Markov nature of the state variables and the independence of the observation random variables conditioned on the state variables, we have

$$f_{X_t|X_{t-1}=x_{t-1},Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t) = f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t).$$

Thus

$$f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t) = \int f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) dx_{t-1} \quad (27)$$

This equation tells us how to go from the filtering density at time $t-1$ to the one-step up ahead prediction density at time $t-1$.

Let us now see the filtering update which specifies how to go from the one-step ahead prediction density at time $t-1$ to the filtering density at time t. By Bayes rule, we can write

$$f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t)$$
$$\propto f_{Y_t|X_t=x_t,Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t) f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t).$$

The Markov nature of the state variables and the independence of the observation random variables conditioned on the state variables implies that

$$f_{Y_t|X_t=x_t,Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t) = f_{Y_t|X_t=x_t,\theta}(y_t).$$

Thus

$$f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t) \propto f_{Y_t|X_t=x_t,\theta}(y_t) f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t).$$

The constant underlying the proportionality symbol $\propto$ above is simply the constant that makes the left hand side integrate to one. We thus get

$$f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t) = \frac{f_{Y_t|X_t=x_t,\theta}(y_t) f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t)}{\int f_{Y_t|X_t=u,\theta}(y_t) f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(u) du} \quad (28)$$

The two steps (27) and (28) together describe the recursion to go from the filtering density at time $t-1$ to the filtering density at time t. The recursion can be initialized by simply using (28) with $t=0$ and replacing $f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t)$ on the right hand by $f_{X_0|\theta}(x_0)$ for $t=0$.

For linear Gaussian state space models, steps (27) and (28) can be implemented in closed form leading to the Kalman Filter which we shall study next.

6.2 The Kalman Filter

Consider the linear Gaussian state space model:

$$\begin{aligned} X_0 &\sim N(\mu_0, \Gamma_0) \\ X_t &= A_t X_{t-1} + U_t \\ Y_t &= B_t X_t + V_t \end{aligned} \quad (29)$$

with $X_0, U_1, \ldots, V_0, V_1, \ldots$ independent and $U_t \sim N(0, \Sigma_t)$ and $V_t \sim N(0, R_t)$. In this case, it turns out every conditional distributions $X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ are Gaussian so we can write

$$X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta \sim N(m_{s|t}, Q_{s|t}). \tag{30}$$

The Kalman filtering algorithm specifies how to compute $m_{t|t}, Q_{t|t}$ for $t = 0, 1, \ldots$ by essentially solving the equations (27) and (28) in closed form. Equation (27) specifies how to calculate $m_{t|t-1}, Q_{t|t-1}$ from $m_{t-1|t-1}, Q_{t-1|t-1}$. For this, we can either explicitly compute the integral in (27) or just use standard properties of normal distributions as:

$$X_t \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta \overset{\mathrm{d}}{=} A_t X_{t-1} + U_t \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta.$$

Because, conditional on $Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta$, the random variables X_{t-1} and U_t are independently distributed as $N(m_{t-1|t-1}, Q_{t-1|t-1})$ and $N(0, \Sigma_t)$ respectively, we obtain

$$X_t \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta \sim N(A_t m_{t-1|t-1}, A_t Q_{t-1|t-1} A_t' + \Sigma_t)$$

Thus

$$m_{t|t-1} = A_t m_{t-1|t-1} \quad \text{and} \quad Q_{t|t-1} = A_t Q_{t-1|t-1} A_t' + \Sigma_t. \tag{31}$$

We next calculate $m_{t|t}, Q_{t|t}$ from $m_{t|t-1}, Q_{t|t-1}$ by calculating filtering update (28). The basic idea behind this calculation is encapsulated in the result below.

Fact 6.1. *Suppose $X \sim N(m_0, Q_0)$ and $Y \mid X = x \sim N(Bx, R)$ (note that the condition $Y \mid X = x \sim N(Bx, R)$ can also be written as $Y = BX + V$ where $V \sim N(0, R)$ with V, X being independent). Then*

$$X \mid Y = y \sim N(m_1, Q_1)$$

where

$$m_1 = \left(Q_0^{-1} + B'R^{-1}B\right)^{-1}\left(Q_0^{-1}m_0 + B'R^{-1}y\right) \quad \text{and} \quad Q_1 = \left(Q_0^{-1} + B'R^{-1}B\right)^{-1}. \tag{32}$$

The following two simple examples can be used to better understand the formula (32).

Example 6.2 (Normal Mean Estimation)**.** *Suppose $\Theta \sim N(\mu, \tau^2)$ and $Y_1, \ldots, Y_n \mid \Theta = \theta \overset{i.i.d}{\sim} N(\theta, \sigma^2)$. Then it is well-known that*

$$\Theta \mid Y_1 = y_1, \ldots, Y_n = y_n \sim N\left(\frac{\mu/\tau^2 + n\bar{y}/\sigma^2}{1/\tau^2 + n/\sigma^2}, \frac{1}{1/\tau^2 + n/\sigma^2}\right).$$

This result is a special case of (32) corresponding to $m_0 = \mu, Q_0 = \tau^2$, $Y = (Y_1, \ldots, Y_n)'$, $y = (y_1, \ldots, y_n)'$, $B = (1, \ldots, 1)'$ and $R = \sigma^2 I_n$.

Example 6.3 (Linear Regression)**.** *Suppose $\beta \sim N(m_0, Q_0)$ and $Y \mid \beta \sim N(Z\beta, \sigma^2 I_n)$ where Z is a deterministic $n \times p$ matrix. The formula (32) then gives*

$$\beta \mid Y = y \sim N\left(\left(Q_0^{-1} + \frac{X'X}{\sigma^2}\right)^{-1}\left(\frac{X'Y}{\sigma^2} + Q_0^{-1}m_0\right), \left(Q_0^{-1} + \frac{X'X}{\sigma^2}\right)^{-1}\right).$$

This result is expected because when $Q_0 = CI$ for a large constant C, we can neglect the effect of Q_0 and this leads to

$$\beta \mid Y = y \approx N\left((X'X)^{-1}X'Y, \sigma^2(X'X)^{-1}\right)$$

which is familiar from usual least squares theory.

The Sherman-Morrison-Woodbury formula:

$$(A+UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1}+VA^{-1}U\right)^{-1}VA^{-1}$$

can be used with $A = Q_0^{-1}$, $U = B'$, $C = R^{-1}$, $V = B$ to obtain the following alternative formulae for m_1 and Q_1:

$$\begin{aligned} m_1 &= m_0 + Q_0B'\left(BQ_0B'+R\right)^{-1}(y - Bm_0) \\ Q_1 &= Q_0 - Q_0B'\left(BQ_0B'+R\right)^{-1}BQ_0 \end{aligned} \tag{33}$$

Note that (32) involves inversion of the matrix $Q_0^{-1}+B'R^{-1}B$ while (53) involves inversion of $BQ_0B'+R$. When the dimension of $BQ_0B'+R$ is much smaller than that of $Q_0^{-1}+B'R^{-1}B$, it is computationally advantageous to work with (53) compared to (32). This will often be the case so we shall mainly use the formula (53).

Now let us get back to the derivation of the filtering updates for the Linear Gaussian State Space Model where we need to calculate $m_{t|t}$ and $Q_{t|t}$ in terms of $m_{t|t-1}$ and $Q_{t|t-1}$. It is easy to check that Fact 9.1 is directly applicable with $m_0 = m_{t|t-1}$, $Q_0 = Q_{t|t-1}$, $B = B_t$, $R = R_t$ and $m_1 = m_{t|t}$, $Q_1 = Q_{t|t}$. The formula (53) then gives

$$\begin{aligned} m_{t|t} &= m_{t|t-1} + Q_{t|t-1}B_t'\left(B_tQ_{t|t-1}B_t'+R_t\right)^{-1}\left(y_t - B_tm_{t|t-1}\right) \\ Q_{t|t} &= Q_{t|t-1} - Q_{t|t-1}B_t'\left(B_tQ_{t|t-1}B_t'+R_t\right)^{-1}B_tQ_{t|t-1} \end{aligned} \tag{34}$$

The equations (51) and (52) together comprise the Kalman Filter. They provide the solution for the filtering problem for linear Gaussian state space models. Here is a formal description of the Kalman Filter including the initialization step: We are given the model (49) and we assume that $\mu_0, \Gamma_0, \{A_t, t \geq 1\}, \{B_t, t \geq 0\}$, $\{\Sigma_t, t \geq 0\}$ and $\{R_t, t \geq 0\}$ are known. The Kalman filter for calculating the conditional distributions (50) for $s = t$ is:

1. **Initialization**: Set $m_{0|-1} = \mu_0$ and $Q_{0|-1} = \Gamma_0$. Implement (52) for $t = 0$ to obtain $m_{0|0}$ and $Q_{0|0}$.
2. **Recursion**: For each $t = 1, 2, \ldots,$ implement (51) and (52).

Note that the Kalman Filter algorithm also computes the one-step ahead prediction means $m_{t|t-1}$ and covariances $Q_{t|t-1}$ in intermediate computations. So the Kalman Filter can also be used to obtain these one-step ahead predictions.

6.3 Recommended Reading for Today

1. The general filtering approach described in Section 6.1 can be found in:
   a) Section 6.2 of the Kitagawa-Gersch book
   b) Section 14.2 of the Kitagawa book.
   c) Section 2.7.1 of the Petris-Petrone-Campagnoli book
2. The Kalman filter is described in the all the books listed in the course outline:
   a) Section 5.2 of the Kitagawa-Gersch book
   b) Section 9.2 of the Kitagawa book
   c) Section 4.3 of the Durbin-Koopman book

d) Section 4.3 of the Särkkä book

e) Section 2.7.2 of the Petris-Petrone-Campagnoli book

f) Section 3.2 of the Triantafyllopoulos book

Section 7.2 of the Chopin-Papaspiliopoulos book also discusses the Kalman filter. They however derive the algorithm from a general Feynman-Kac formalism (see their Chapter 5). I will discuss the Feynman-Kac stuff in class a few weeks later.

7 Lecture Seven

7.1 The Kalman Filter

Consider the linear Gaussian state space model:

$$\begin{aligned} X_0 &\sim N(\mu_0, \Gamma_0) \\ X_t &= A_t X_{t-1} + U_t \\ Y_t &= B_t X_t + V_t \end{aligned} \tag{35}$$

with $X_0, U_1, \ldots, V_0, V_1, \ldots$ independent and $U_t \sim N(0, \Sigma_t)$ and $V_t \sim N(0, R_t)$. Each of the quantities $\mu_0, \Gamma_0, A_t, B_t, \Sigma_t, R_t$ appearing in the model above can depend on an unknown vector of parameters θ. Every conditional distribution $X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ is Gaussian and we can write

$$X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta \sim N(m_{s|t}, Q_{s|t}). \tag{36}$$

The Kalman filtering algorithm specifies how to compute $m_{t|t}, Q_{t|t}$ for $t = 0, 1, \ldots$ using the following equations:

$$m_{t|t-1} = A_t m_{t-1|t-1} \quad \text{and} \quad Q_{t|t-1} = A_t Q_{t-1|t-1} A_t' + \Sigma_t. \tag{37}$$

and

$$\begin{aligned} m_{t|t} &= m_{t|t-1} + Q_{t|t-1} B_t' \left(B_t Q_{t|t-1} B_t' + R_t \right)^{-1} \left(y_t - B_t m_{t|t-1} \right) \\ Q_{t|t} &= Q_{t|t-1} - Q_{t|t-1} B_t' \left(B_t Q_{t|t-1} B_t' + R_t \right)^{-1} B_t Q_{t|t-1} \end{aligned} \tag{38}$$

Equations (51) and (52) together comprise the Kalman Filter. The formal description of the Kalman Filter including the initialization is as follows. We are given the model (49) and we assume that $\mu_0, \Gamma_0, \{A_t, t \geq 1\}, \{B_t, t \geq 0\}$, $\{\Sigma_t, t \geq 0\}$ and $\{R_t, t \geq 0\}$ are known. The Kalman filter for calculating the conditional distributions (50) for $s = t$ is:

1. **Initialization**: Set $m_{0|-1} = \mu_0$ and $Q_{0|-1} = \Gamma_0$. Implement (52) for $t = 0$ to obtain $m_{0|0}$ and $Q_{0|0}$.

2. **Recursion**: For each $t = 1, 2, \ldots,$ implement (51) and (52).

Note that the Kalman Filter algorithm also computes the one-step ahead prediction means $m_{t|t-1}$ and covariances $Q_{t|t-1}$ in intermediate computations. So the Kalman Filter can also be used to obtain these one-step ahead predictions.

7.2 Some Examples

We shall give here some simple examples of linear Gaussian state space models and write the Kalman recursions more explicitly.

7.2.1 Tracking One: Velocity Model

Consider the problem of tracking the position of an object moving on a straight line. We observe the position of the object every Δt seconds but these measurements are imprecise. For $k = 0, 1, 2, , \ldots$, let x_k denote the actual position of the object at time $k(\Delta t)$ and let y_k denote the measurement. We assume that

$$y_k = x_k + \epsilon_k \qquad \text{with } \epsilon_k \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2)$$

for $k = 0, 1, 2, \ldots$. For the state model, in this "velocity model", we assume that the velocity of the particle stays constant at a level u_k in the time interval $[(k-1)(\Delta t), k\Delta t]$ leading to the equation:

$$x_k = x_{k-1} + u_k(\Delta t) \qquad \text{for } k = 1, 2, \ldots.$$

Further, we shall assume that $u_1, u_2, \ldots$ are i.i.d $N(0, \sigma_u^2)$. Finally assume that $x_0 \sim N(0, C)$ for a large positive constant C. This is basically the local level model with the state evolution error variance equal to $\sigma_u^2(\Delta t)^2$.

The Kalman filter for this model for computing

$$x_k \mid y_0, \ldots, y_k, \sigma_\epsilon, \sigma_u, C$$

is easily checked to be given by

$$m_{k|k-1} = m_{k-1|k-1} \quad \text{and} \quad Q_{k|k-1} = Q_{k-1|k-1} + \sigma_u^2(\Delta t)^2$$

and

$$\begin{aligned} m_{k|k} &= m_{k|k-1} + \frac{Q_{k|k-1}}{Q_{k|k-1} + \sigma_\epsilon^2}\left(y_k - m_{k|k-1}\right) \\ Q_{k|k} &= Q_{k|k-1} - \frac{Q_{k|k-1}^2}{Q_{k|k-1} + \sigma_\epsilon^2} = \frac{Q_{k|k-1}\sigma_\epsilon^2}{Q_{k|k-1} + \sigma_\epsilon^2}. \end{aligned} \tag{39}$$

The Kalman Filter is initialized with $m_{0|-1} = 0$ and $Q_{0|-1} = C$ which leads to (via the filter update (39))

$$\begin{aligned} m_{0|0} &= m_{0|-1} + \frac{Q_{0|-1}}{Q_{0|-1} + \sigma_\epsilon^2}\left(y_0 - m_{0|-1}\right) = \frac{C}{C + \sigma_\epsilon^2} y_0 \\ Q_{0|0} &= \frac{Q_{0|-1}\sigma_\epsilon^2}{Q_{0|-1} + \sigma_\epsilon^2} = \frac{C\sigma_\epsilon^2}{C + \sigma_\epsilon^2}. \end{aligned}$$

It is clear that when C is large, the above equations imply that $m_{0|0} \approx y_0$ and $Q_{0|0} \approx \sigma_\epsilon^2$. Thus a commonly used initialization for the local level model is $m_{0|0} = y_0$ and $Q_{0|0} = \sigma_\epsilon^2$.

7.2.2 Tracking Two: Acceleration Model

Consider the same setting as the last subsection. We now consider a different model for the state evolution where we assume that the acceleration (not velocity) remains constant in each time period $[(k-1)(\Delta t), k(\Delta t)]$. Denoting this acceleration by a_k, we see that the velocity at time $(k-1)(\Delta t)$ (which we denote by $x_{k-1,2}$) and the velocity at time $k(\Delta t)$ (which we denote by $x_{k,2}$) are related by the equation:

$$x_{k,2} = x_{k-1,2} + a_k(\Delta t). \tag{40}$$

Further the position at time $(k-1)(\Delta t)$ (which we denote by $x_{k-1,1}$) and the position at time $k(\Delta t)$ (which we denote by $x_{k,1}$) are related by the equation:

$$x_{k,1} = x_{k-1,1} + x_{k-1,2}(\Delta t) + \frac{1}{2}(\Delta t)^2 a_k. \tag{41}$$

Letting the state vector at time k to be both the position and the velocity at time k:

$$x_k = \begin{pmatrix} x_{k,1} \\ x_{k,2} \end{pmatrix},$$

we can write the state evolution as

$$x_k = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} x_{k-1} + a_k \begin{pmatrix} (\Delta t)^2/2 \\ \Delta t \end{pmatrix}.$$

Because the accelerations $a_1, a_2, \ldots$ are unknown, a simple way of dealing with them is to assume that:

$$a_1, a_2, \ldots \overset{\text{i.i.d}}{\sim} N(0, \sigma_a^2).$$

Then the state evolution becomes

$$x_k = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} x_{k-1} + U_k \qquad \text{where } U_k \sim N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \sigma_a^2 \begin{bmatrix} (\Delta t)^4/4 & (\Delta t)^3/2 \\ (\Delta t)^3/2 & (\Delta t)^2 \end{bmatrix} \right)$$

The equation relating the observation and state variables becomes

$$y_k = x_{k,1} + \epsilon_k = \begin{pmatrix} 1 & 0 \end{pmatrix} x_k + \epsilon_k \qquad \text{where } \epsilon_k \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2).$$

The Kalman filter for this model simplifies to the following equations. Note that $m_{s|t}$ is a 2×1 vector and $Q_{s|t}$ is a 2×2 matrix. The one-step prediction update is

$$m_{t|t-1} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} m_{t-1|t-1}$$

and

$$Q_{t|t-1} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} Q_{t-1|t-1} \begin{pmatrix} 1 & 0 \\ \Delta t & 1 \end{pmatrix} + \sigma_a^2 \begin{pmatrix} (\Delta t)^4/4 & (\Delta t)^3/2 \\ (\Delta t)^3/2 & (\Delta t)^2 \end{pmatrix}$$

The filter update is given by the two equations:

$$\begin{aligned} m_{t|t} &= m_{t|t-1} + \frac{\left(y_t - \begin{pmatrix} 1 & 0 \end{pmatrix} m_{t|t-1}\right)}{\begin{pmatrix} 1 & 0 \end{pmatrix} Q_{t|t-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \sigma_\epsilon^2} Q_{t|t-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= m_{t|t-1} + \frac{\left(y_t - m_{t|t-1}[1]\right)}{Q_{t|t-1}[1,1] + \sigma_\epsilon^2} \begin{pmatrix} Q_{t|t-1}[1,1] \\ Q_{t|t-1}[2,1] \end{pmatrix}, \end{aligned}$$

and

$$\begin{aligned} Q_{t|t} &= Q_{t|t-1} - \frac{Q_{t|t-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} Q_{t|t-1}}{\begin{pmatrix} 1 & 0 \end{pmatrix} Q_{t|t-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \sigma_\epsilon^2} \\ &= Q_{t|t-1} - \frac{1}{Q_{t|t-1}[1,1] + \sigma_\epsilon^2} \begin{pmatrix} Q_{t|t-1}^2[1,1] & Q_{t|t-1}[1,1] Q_{t|t-1}[2,1] \\ Q_{t|t-1}[1,1] Q_{t|t-1}[2,2] & Q_{t|t-1}^2[2,1] \end{pmatrix} \end{aligned}$$

In the above, we used $Q_{t|t-1}[i,j]$ for the $(i,j)^{th}$ entry of the matrix $Q_{t|t-1}$ and $m_{t|t-1}[1]$ for the first entry of the 2×1 vector $m_{t|t-1}$.

7.2.3 Tracking Three: Local Linear Model

The local linear model that we saw previously can be seen as an approximation of the acceleration model of the last subsection. Specifically, in the position equation (41), we drop the last term $a_k(\Delta t)^2/2$ the idea being that if Δt is small, then this term will generally be negligible compared to at least one of the other two terms $x_{k-1,1}(\Delta t)$ and $x_{k-1,1}(\Delta t)$. This leads to the equations:

$$x_{k,1} = x_{k-1,1} + x_{k-1,2}(\Delta t) \quad \text{and} \quad x_{k,2} = x_{k-1,2} + a_k(\Delta t).$$

We can combine these two equations into one by using $x_{k,2} = \frac{x_{k+1,1} - x_{k,1}}{\Delta t}$ (which is obtained from the first equation) in the second equation to deduce

$$x_{k,1} - 2x_{k-1,1} + x_{k-2,1} = a_{k-1}(\Delta t)^2 \sim N(0, \sigma_a^2(\Delta t)^4)$$

which is the local linear model with the state evolution error variance equal to $\sigma_a^2(\Delta t)^4$. This can be written as a state space model using $\begin{pmatrix} x_{k,1} \\ x_{k-1,1} \end{pmatrix}$ as the state or as in the previous subsection with $\begin{pmatrix} x_{k,1} \\ (\Delta t)^{-1}(x_{k,1} - x_{k-1,1}) \end{pmatrix}$ as the state. Note that this shows that there can be many different ways to write a model in state space form. The Kalman recursions for the local level model are left as exercise.

7.3 Use of the Kalman Filter for Parameter Estimation by Maximum Likelihood

As mentioned previously, the quantities $\mu_0, \Gamma_0, A_t, B_t, \Sigma_t, R_t$ appearing in the state space model (49) typically depend on an unknown vector of parameters θ which needs to be estimated from the observed data $y_0, \ldots, y_T$. A standard method for parameter estimation is maximum likelihood and the Kalman filter output is useful for writing down the likelihood function. To see this, first note that the likelihood for the observed data $y_0, \ldots, y_T$ is given by

$$f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T) = \prod_{t=0}^{T} f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t) = \prod_{t=0}^{T} f_{B_tX_t+V_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t).$$

Conditionally on $Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta$, the random variables X_t and V_t are independent having the $N(m_{t|t-1}, Q_{t|t-1})$ and $N(0, R_t)$ respectively. Thus

$$B_tX_t + V_t \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta \sim N(B_tm_{t|t-1}, B_tQ_{t|t-1}B_t' + R_t).$$

Thus, for each $t = 0, 1, \ldots, T$, we have

$$f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)$$
$$= \left|2\pi\left(B_tQ_{t|t-1}B_t' + R_t\right)\right|^{-1/2} \exp\left(-\frac{1}{2}\left(y_t - B_tm_{t|t-1}\right)'\left(B_tQ_{t|t-1}B_t' + R_t\right)^{-1}\left(y_t - B_tm_{t|t-1}\right)\right)$$

where $|\cdot|$ denotes determinant. Let

$$\epsilon_t(\theta) := y_t - B_tm_{t|t-1} \quad \text{and} \quad H_t(\theta) := B_tQ_{t|t-1}B_t' + R_t$$

for $t = 0, 1, 2, \ldots$. Then

$$-2\log f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t) = \log|2\pi H_t(\theta)| + \epsilon_t'(\theta)H_t^{-1}(\theta)\epsilon_t(\theta)$$

Thus

$$(-2) \times \text{log-likelihood} = \sum_{t=0}^{T} \left[\log |2\pi H_t(\theta)| + \epsilon_t'(\theta) H_t^{-1}(\theta) \epsilon_t(\theta) \right].$$

For calculating this likelihood, we only need $m_{t|t-1}$ and $Q_{t|t-1}$ for $t = 0, 1, 2, \ldots$ which can be obtained from the Kalman Filter. One can maximize likelihood by minimizing the right hand side above over the parameters θ. Numerical optimization routines can be used for this purpose.

7.4 Recommended Reading for Today

1. The local level model is analyzed in detail in Chapter 2 of the Durbin-Koopman book. In particular, see Section 2.2.1 for the Kalman filter updates in the local level model. Some comments on the initial distribution $X_0 \sim N(0, C)$ (for a large C) can be found in Section 2.9.
2. The acceleration model of Subsection 7.2.2 can be found in `https://en.wikipedia.org/wiki/Kalman_filter` (see Section 7).
3. For likelihood computation using the Kalman filter, see Section 9.6 of the Kitagawa book.

8 Lecture Eight

8.1 Some remarks on the local level model

Consider the local level model:

$$X_0 \sim N(0, C) \qquad X_t = X_{t-1} + Z_t \qquad Y_t = X_t + \epsilon_t$$

where $X_0, Z_1, Z_2, \ldots, \epsilon_0, \epsilon_1, \ldots$ are independent with $Z_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_Z^2)$ and $\epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2)$. The Kalman filter recursions for this model are given by

$$m_{t|t-1} = m_{t-1|t-1} \quad \text{and} \quad Q_{t|t-1} = Q_{t-1|t-1} + \sigma_Z^2$$

and

$$m_{t|t} = m_{t|t-1} + \frac{Q_{t|t-1}}{Q_{t|t-1} + \sigma_\epsilon^2} \left(y_t - m_{t|t-1} \right) = \frac{\sigma_\epsilon^2}{Q_{t|t-1} + \sigma_\epsilon^2} m_{t|t-1} + \frac{Q_{t|t-1}}{Q_{t|t-1} + \sigma_\epsilon^2} y_t$$

$$Q_{t|t} = \frac{Q_{t|t-1} \sigma_\epsilon^2}{Q_{t|t-1} + \sigma_\epsilon^2}.$$

Here $m_{s|t}$ and $Q_{s|t}$ denote the conditional mean and variance of X_s given $Y_0 = y_0, \ldots, Y_t = y_t$ (and $\sigma_Z, \sigma_\epsilon$).

It is interesting to note that $m_{t|t}$ is a weighted linear combination of $m_{t|t-1}$ and y_t. We see in simulations that when the σ_Z parameter is large, then the filtering mean $m_{t|t}$ is close to y_t for each $t \geq 0$. This can be explained as follows. Because $Q_{t|t-1} = Q_{t-1|t-1} + \sigma_Z^2 \geq \sigma_Z^2$, it follows that, when σ_Z is large, each $Q_{t|t-1}$ is also large. As a result, the weight for y_t dominates the weight for $m_{t|t-1}$ in the formula for $m_{t|t}$ leading to $m_{t|t} \approx y_t$ when σ_Z is large.

Note that $m_{t|t} \approx y_t$ does not imply that the model is overfitting the observed data. This is because the log-likelihood multiplied by (-2) is given by

$$\begin{aligned}(-2)\text{log-likelihood} &= \sum_{t=0}^{T}\left[\log\left\{2\pi\left(Q_{t|t-1}+\sigma_\epsilon^2\right)\right\}+\frac{\left(y_t-m_{t|t-1}\right)^2}{Q_{t|t-1}+\sigma_\epsilon^2}\right]\\ &= \sum_{t=0}^{T}\left[\log\left\{2\pi\left(Q_{t-1|t-1}+\sigma_Z^2+\sigma_\epsilon^2\right)\right\}+\frac{\left(y_t-m_{t-1|t-1}\right)^2}{Q_{t-1|t-1}+\sigma_Z^2+\sigma_\epsilon^2}\right]\end{aligned}$$

When σ_Z is large, we would have $m_{t-1|t-1} \approx y_{t-1}$ as remarked above. Thus the above expression for large σ_Z becomes

$$(-2)\text{log-likelihood} \approx \sum_{t=0}^{T}\left[\log\left\{2\pi\left(Q_{t-1|t-1}+\sigma_Z^2+\sigma_\epsilon^2\right)\right\}+\frac{\left(y_t-y_{t-1}\right)^2}{Q_{t-1|t-1}+\sigma_Z^2+\sigma_\epsilon^2}\right]$$

The second term in the sum above is of smaller order compared to the first term when σ_Z is large. Thus the behavior of the whole expression will be similar to the behavior to the first term which is increasing in σ_Z. Thus as σ_Z increases, the log-likelihood decreases (note that the above is the expression for **negative** log-likelihood multiplied by 2). This means that there is no overfitting for large σ_Z (overfitting would happen when the likelihood keeps getting better and better when σ_Z is increased which is not happening here).

In the last class, we mentioned that parameter estimates of σ_Z and σ_ϵ can be obtained by maximizing the likelihood (or equivalently, minimizing negative two times the log-likelihood) over σ_Z and σ_ϵ. The result of this optimization cannot be written in closed because it is a somewhat complicated optimization. This is because it depends on σ_Z and σ_ϵ in a not-so-simple way. To highlight this, let us note that $Q_{t|t-1}$ and $m_{t|t-1}$ depend on σ_Z and σ_ϵ, and also on the initial state variance C. We shall therefore write them as $Q_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)$ and $m_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)$ respectively. We thus have

$$\begin{aligned}\ell(\sigma_Z,\sigma_\epsilon) &:= (-2)\text{log-likelihood}\\ &= \sum_{t=0}^{T}\left[\log\left\{2\pi\left(Q_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)+\sigma_\epsilon^2\right)\right\}+\frac{\left(y_t-m_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)\right)^2}{Q_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)+\sigma_\epsilon^2}\right].\end{aligned}$$

The dependence of this function on $C,\sigma_Z,\sigma_\epsilon$ is not so simple. Numerical routines can be used to optimize this function of σ_Z and σ_ϵ. The following trick reduces this to a one-parameter optimization problem and it can be quite handy. To see this, first note that for $t=0$, we have $Q_{0|-1}=C$ and $m_{0|-1}=0$. Thus

$$\begin{aligned}\ell(\sigma_Z,\sigma_\epsilon) &= \log\left\{2\pi\left(C+\sigma_\epsilon^2\right)\right\}+\frac{y_0^2}{C+\sigma_\epsilon^2}\\ &\quad+\sum_{t=1}^{T}\left[\log\left\{2\pi\left(Q_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)+\sigma_\epsilon^2\right)\right\}+\frac{\left(y_t-m_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)\right)^2}{Q_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)+\sigma_\epsilon^2}\right].\end{aligned}$$

As C is large, the first term is approximately $\log(2\pi C)$ and the second term is zero. Thus

$$\ell(\sigma_Z,\sigma_\epsilon) \approx \log(2\pi C)+\sum_{t=1}^{T}\left[\log\left\{2\pi\left(Q_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)+\sigma_\epsilon^2\right)\right\}+\frac{\left(y_t-m_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)\right)^2}{Q_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)+\sigma_\epsilon^2}\right].$$

The $\log(2\pi C)$ term does not depend on σ_Z or σ_ϵ so it can be removed from the optimization so the goal is to minimize:

$$\sum_{t=1}^{T}\left[\log\left\{2\pi\left(Q_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)+\sigma_\epsilon^2\right)\right\}+\frac{\left(y_t-m_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)\right)^2}{Q_{t|t-1}(C,\sigma_Z,\sigma_\epsilon)+\sigma_\epsilon^2}\right].$$

We shall now take $C = +\infty$. The quantities $Q_{t|t-1}(\infty, \sigma_Z, \sigma_\epsilon)$ and $m_{t|t-1}(\infty, \sigma_Z, \sigma_\epsilon)$ are then obtained for $t = 1, 2, \ldots$ by running the Kalman filter steps with the initialization $m_{0|0} = y_0$ and $Q_{0|0} = \sigma_\epsilon^2$. Our goal is to minimize

$$\ell^*(\sigma_Z, \sigma_\epsilon) := \sum_{t=1}^{T} \left[\log\left\{2\pi\left(Q_{t|t-1}(\infty, \sigma_Z, \sigma_\epsilon) + \sigma_\epsilon^2\right)\right\} + \frac{\left(y_t - m_{t|t-1}(\infty, \sigma_Z, \sigma_\epsilon)\right)^2}{Q_{t|t-1}(\infty, \sigma_Z, \sigma_\epsilon) + \sigma_\epsilon^2} \right].$$

We now note the following useful fact:

$$m_{t|t-1}(\infty, \sigma_Z, \sigma_\epsilon) = m_{t|t-1}(\infty, \frac{\sigma_Z}{\sigma_\epsilon}, 1) \qquad \text{and} \qquad Q_{t|t-1}(\infty, \sigma_Z, \sigma_\epsilon) = \sigma_\epsilon^2 Q_{t|t-1}(\infty, \frac{\sigma_Z}{\sigma_\epsilon}, 1). \tag{42}$$

I will leave the proof of this fact as an exercise. To compute $m_{t|t-1}(\infty, \sigma_Z/\sigma_\epsilon, 1)$ and $Q_{t|t-1}(\infty, \sigma_Z/\sigma_\epsilon, 1)$, we would need to run the Kalman filter with σ_Z and σ_ϵ replaced by σ_Z/σ_ϵ and 1 respectively (the initialization would then be $m_{0|0} = y_0$ and $Q_{0|0} = 1$).

Because of the scaling fact (42), we can write $\ell^*(\sigma_Z, \sigma_\epsilon)$ as

$$\begin{aligned}\ell^*(\sigma_Z, \sigma_\epsilon) &:= \sum_{t=1}^{T} \left[\log\left\{2\pi\left(Q_{t|t-1}(\infty, \sigma_Z, \sigma_\epsilon) + \sigma_\epsilon^2\right)\right\} + \frac{\left(y_t - m_{t|t-1}(\infty, \sigma_Z, \sigma_\epsilon)\right)^2}{Q_{t|t-1}(\infty, \sigma_Z, \sigma_\epsilon) + \sigma_\epsilon^2} \right] \\ &= T\log(2\pi\sigma_\epsilon^2) + \sum_{t=1}^{T} \log\left(Q_{t|t-1}(\infty, \frac{\sigma_Z}{\sigma_\epsilon}, 1) + 1\right) + \frac{1}{\sigma_\epsilon^2}\sum_{t=1}^{T} \frac{\left(y_t - m_{t|t-1}(\infty, \frac{\sigma_Z}{\sigma_\epsilon}, 1)\right)^2}{Q_{t|t-1}(\infty, \frac{\sigma_Z}{\sigma_\epsilon}, 1) + 1}.\end{aligned}$$

The goal is to minimize the above function over all $\sigma_\epsilon > 0$ and $\sigma_Z > 0$. Equivalently, we need to minimize this over all $\sigma_\epsilon > 0$ and $q := \frac{\sigma_Z}{\sigma_\epsilon} > 0$. The advantage of viewing the problem as an optimization over σ_ϵ and q is that it is easy to find the best σ_ϵ for each value of q. Specifically, we need to minimize

$$T\log(2\pi\sigma_\epsilon^2) + \sum_{t=1}^{T} \log\left(Q_{t|t-1}(\infty, q, 1) + 1\right) + \frac{1}{\sigma_\epsilon^2}\sum_{t=1}^{T} \frac{\left(y_t - m_{t|t-1}(\infty, q, 1)\right)^2}{Q_{t|t-1}(\infty, q, 1) + 1} \tag{43}$$

over both $\sigma_\epsilon > 0$ and $q > 0$. For each fixed q, it is easy to find the minimizing σ_ϵ by simply taking the derivative with respect to σ_ϵ^2 and setting it equal to zero. This gives

$$\hat{\sigma}_\epsilon^2(q) := \frac{1}{T}\sum_{t=1}^{T} \frac{\left(y_t - m_{t|t-1}(\infty, q, 1)\right)^2}{Q_{t|t-1}(\infty, q, 1) + 1}. \tag{44}$$

Plugging this value of σ_ϵ^2 in (43), we get

$$T\log(2\pi\hat{\sigma}_\epsilon^2(q)) + \sum_{t=1}^{T} \log\left(Q_{t|t-1}(\infty, q, 1) + 1\right) + T. \tag{45}$$

This function will need to be numerically minimized over $q > 0$ to obtain the minimizer $\hat{q}$. This is an easier optimization problem (compared to minimizing $\ell^*(\sigma_Z, \sigma_\epsilon)$ over both σ_Z and σ_ϵ) for numerical methods as it only depends on the one variable q. After obtaining the minimizer $\hat{q}$, σ_ϵ^2 is estimated by $\hat{\sigma}_\epsilon^2(\hat{q})$ (i.e., the right hand side of (44) with $q = \hat{q}$) and then σ_Z is estimated by $\hat{q}\hat{\sigma}_\epsilon(\hat{q})$. Finally, note that to form the objective (45), we need to calculate $\hat{\sigma}_\epsilon^2(q)$ and, for this, it is necessary to implement the Kalman filter with σ_Z set to q and σ_ϵ set to 1 in order to calculate $m_{t|t-1}(\infty, q, 1)$ and $Q_{t|t-1}(\infty, q, 1)$.

8.2 Application of the Kalman Filter to Linear Regression

Consider the usual linear regression setting where we observe data $(z_0, y_0), \ldots, (z_T, y_T)$ where z_t is the $p \times 1$ covariate and y_t is the scalar response corresponding to index t. The usual linear model for this setting assumes that the covariates $z_0, \ldots, z_T$ are deterministic and the response y_t is related to z_t via

$$y_t = z_t'\beta + \epsilon_t \qquad \text{with } \epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2).$$

In Bayesian treatments of the linear model, one supplements the model above with the prior

$$\beta \sim N(\mu_0, \Gamma_0).$$

If no information on β is available, one can set $\mu_0 = 0$ and $\Gamma_0 = CI$ for a large constant C. Having a prior is a good idea in general as it avoids degeneracy issues. For example, when the matrix Z of covariates (whose rows are $z_0', \ldots, z_T'$) does not have full column rank, the usual least squares estimator is not defined but the Bayesian posterior is well-defined as long as Γ_0 is invertible.

The posterior distribution of β is given by

$$\beta \mid \text{data}, \sigma \sim N\left(\left(\Gamma_0^{-1} + \frac{Z'Z}{\sigma^2}\right)^{-1}\left(\frac{Z'Y}{\sigma^2} + \Gamma_0^{-1}\mu_0\right), \left(\Gamma_0^{-1} + \frac{Z'Z}{\sigma^2}\right)^{-1}\right) \tag{46}$$

Direct computation of mean vector and covariance matrix of the above posterior distribution requires inverting the $p \times p$ matrix $\Gamma_0^{-1} + Z'Z/\sigma^2$ and this can be computationally costly (note that calculating Γ_0^{-1} is usually not hard as Γ_0 is commonly a constant multiple of the identity; the main issue here involves inverting $\Gamma_0^{-1} + Z'Z/\sigma^2$).

The Kalman filter provides an alternative way of computing the posterior mean and variance via a sequential algorithm which does not involve matrix inversion at any step. This is described below. The first step is to write the linear regression model in state space form. We take the state variables to be $\beta_0, \beta_1, \ldots$ with the state evolution as

$$\beta_t = \beta_{t-1} \qquad \text{for } t = 1, 2, \ldots$$

The observation is

$$y_t = z_t'\beta_t + \epsilon_t \qquad \text{for } t = 0, 1, 2, \ldots$$

Finally the initial condition is $\beta_0 \sim N(\mu_0, \Gamma_0)$. This linear Gaussian state space model is exactly the Bayesian linear regression model and so we can apply the Kalman filter. Note that (46) is simply the filtering distribution in this state space model at time T. Thus

$$m_{T|T} = \left(\Gamma_0^{-1} + \frac{Z'Z}{\sigma^2}\right)^{-1}\left(\frac{Z'Y}{\sigma^2} + \Gamma_0^{-1}\mu_0\right) \quad \text{and} \quad Q_{T|T} = \left(\Gamma_0^{-1} + \frac{Z'Z}{\sigma^2}\right)^{-1}.$$

The Kalman filter provides an alternative way of computing $m_{T|T}$ and $Q_{T|T}$ using the following recursions. Because $\beta_t = \beta_{t-1}$, the one-step ahead prediction update is simply $m_{t|t-1} = m_{t-1|t-1}$ and $Q_{t|t-1} = Q_{t-1|t-1}$. The filter update is

$$m_{t|t} = m_{t|t-1} + \frac{y_t - z_t'm_{t|t-1}}{z_t'Q_{t|t-1}z_t + \sigma_\epsilon^2}Q_{t|t-1}z_t$$
$$Q_{t|t} = Q_{t|t-1} - \frac{Q_{t|t-1}z_tz_t'Q_{t|t-1}}{z_t'Q_{t|t-1}z_t + \sigma_\epsilon^2}.$$

These recursions are initialized with $m_{0|-1} = \mu_0, Q_{0|-1} = \Gamma_0$ leading to

$$m_{0|0} = \mu_0 + \frac{y_0 - z_0'\mu_0}{z_0'\Gamma_0 z_0 + \sigma_\epsilon^2}\Gamma_0 z_0$$
$$Q_{0|0} = \Gamma_0 - \frac{\Gamma_0 z_0 z_0' \Gamma_0}{z_0'\Gamma_0 z_0 + \sigma_\epsilon^2}.$$

The main point to be noted here is that, in the Kalman filter, at no point do we need to invert a $p \times p$ matrix. There are matrix vector products and other elementary operations but there is no matrix inversion.

8.3 Prediction

Prediction, in the context of state space models, refers to the problem of finding the distribution $X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ for $s > t$. The prediction problem for linear Gaussian state space models is readily solved by the Kalman filter. To see this, note that we need to find the mean $m_{s|t}$ and covariance $Q_{s|t}$ for each $s > t$. The Kalman filter tells us how to compute $m_{t|t}, Q_{t|t}$. The prediction problem for $s = t+1$ is easily solved via (this is basically the same as the one-step ahead prediction update used in the Kalman filter):

$$m_{t+1|t} = A_{t+1} m_{t|t} \quad \text{and} \quad Q_{t+1|t} = A_{t+1} Q_{t|t} A_{t+1}' + \Sigma_{t+1}. \tag{47}$$

Next for $s = t + 2$, observe that

$$X_{t+2} \mid (Y_0 = y_0, \ldots, Y_t = y_t, \theta) = A_{t+2} X_{t+1} + U_{t+2} \mid (Y_0 = y_0, \ldots, Y_t = y_t, \theta)$$

Note now that

$$X_{t+1} \mid (Y_0 = y_0, \ldots, Y_t = y_t, \theta) \sim N(m_{t+1|t}, Q_{t+1|t})$$
$$U_{t+2} \mid (Y_0 = y_0, \ldots, Y_t = y_t, \theta) \sim N(0, \Sigma_{t+2})$$

and further X_{t+1} and U_{t+2} are independent conditional on $Y_0 = y_0, \ldots, Y_t = y_t, \theta$. Thus

$$X_{t+2} \mid (Y_0 = y_0, \ldots, Y_t = y_t, \theta) \sim N(A_{t+2} m_{t+1|t}, A_{t+2}' Q_{t+1|t} A_{t+2} + \Sigma_{t+2}).$$

Therefore

$$m_{t+2|t} = A_{t+2} m_{t+1|t} \quad \text{and} \quad Q_{t+2|t} = A_{t+2}' Q_{t+1|t} A_{t+2} + \Sigma_{t+2}.$$

Note that the terms $m_{t+1|t}$ and $Q_{t+1|t}$ appearing on the right hand side above have already been calculated in (47).

More generally, one can write $m_{s|t}, Q_{s|t}$ for $s > t$ in terms of $m_{s-1|t}, Q_{s-1|t}$ as

$$m_{s|t} = A_s m_{s-1|t} \quad \text{and} \quad Q_{s|t} = A_s Q_{s-1|t} A_s' + \Sigma_s.$$

This equation can be used recursively for $s = t+1, t+2$, to calculate all prediction distributions.

8.4 Smoothing

Smoothing, in the context of state space models, refers to the problem of finding the distribution of $X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ for $s \leq t$. These are calculated by backward recursion starting for $s = t$ and then decreasing s (note that the smoothing distribution for $s = t$ is a filtering distribution which is given by the Kalman filter). The details for this will be discussed next week.

8.5 Recommended Reading for Today

1. The technique described in Section 8.1 for reducing the likelihood optimization to a one-dimensional optimization problem can be found in Section 2.10.2 of the Durbin-Koopman. This technique holds in some more settings as described in Section 9.6 of the Kitagawa book.
2. See Sections 3.1, 3.2 and 3.3 of the Särkkä book for treatment of linear regression and application of the Kalman filter for recursive linear regression. Also see Section 3.4 for a treatment of the linear regression with drift model.
3. The prediction recursions can be found in Section 9.5 of the Kitagawa book.

9 Lecture Nine

9.1 Smoothing

Smoothing, in the context of state space models, refers to the problem of finding the conditional distribution $X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ for $s \leq t$. The main interest in these conditional distributions is in the case $t = T$ (recall that our observed data is $y_0, \ldots, y_T$).

The algorithm that we shall discuss proceeds by first running the filtering step which calculates the distributions $X_t \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ for $t = 0, 1, 2, \ldots$. Following this, one follows backward recursion starting from $s = t$ and then decreasing s as $t-1, t-2, \ldots$ to calculate the conditional distributions $X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ for $s \leq t$. The overall algorithm is often referred to as FFBS (Forward Filtering Backward Smoothing).

We shall understand the backward recursion in the general case of arbitrary state space models. Subsequently, we shall specialize this to the case of linear Gaussian state space models.

9.2 Backward Recursion for General State Space Models

Fix a value of $t \geq 0$. Assume that we have computed the conditional density:

$$f_{X_{s+1}|Y_0=y_0,\ldots,Y_t=y_t,\theta}$$

for some $s < t$. The goal is then to figure out how to use the above density to calculate

$$f_{X_s|Y_0=y_0,\ldots,Y_t=y_t,\theta}.$$

For this, we write

$$f_{X_s|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_s) = \int f_{X_s|X_{s+1}=x_{s+1},Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_s) f_{X_{s+1}|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{s+1}) dx_{s+1}.$$

The key now is to note that

$$X_s \mid (X_{s+1} = x_{s+1}, Y_0 = y_0, \ldots, Y_t = y_t, \theta) \stackrel{\mathrm{d}}{=} X_s \mid (X_{s+1} = x_{s+1}, Y_0 = y_0, \ldots, Y_s = y_s, \theta).$$

In words, the above means that conditional on $X_{s+1} = x_{s+1}, Y_0 = y_0, \ldots, Y_s = y_s$, the random objects X_s and $(Y_{s+1}, \ldots, Y_t)$ are independent. I will leave the verification of this

property as an exercise. We thus have

$$f_{X_s|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_s) = \int f_{X_s|X_{s+1}=x_{s+1},Y_0=y_0,\ldots,Y_s=y_s,\theta}(x_s) f_{X_{s+1}|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{s+1}) dx_{s+1}.$$

The next step is to calculate $f_{X_s|X_{s+1}=x_{s+1},Y_0=y_0,\ldots,Y_s=y_s,\theta}(x_s)$. For this we use Bayes rule to write

$$\begin{aligned} f_{X_s|X_{s+1}=x_{s+1},Y_0=y_0,\ldots,Y_s=y_s,\theta}(x_s) &\propto f_{X_s|Y_0=y_0,\ldots,Y_s=y_s,\theta}(x_s) f_{X_{s+1}|X_s=x_s,Y_0=y_0,\ldots,Y_s=y_s,\theta}(x_{s+1}) \\ &= f_{X_s|Y_0=y_0,\ldots,Y_s=y_s,\theta}(x_s) f_{X_{s+1}|X_s=x_s,\theta}(x_{s+1}) \\ &= \frac{f_{X_s|Y_0=y_0,\ldots,Y_s=y_s,\theta}(x_s) f_{X_{s+1}|X_s=x_s,\theta}(x_{s+1})}{\int f_{X_s|Y_0=y_0,\ldots,Y_s=y_s,\theta}(u) f_{X_{s+1}|X_s=u,\theta}(x_{s+1}) du}. \end{aligned}$$

We can thus write the backward smoothing recursion in one step as

$$\begin{aligned} &f_{X_s|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_s) \\ &= \int \left[\frac{f_{X_s|Y_0=y_0,\ldots,Y_s=y_s,\theta}(x_s) f_{X_{s+1}|X_s=x_s,\theta}(x_{s+1})}{\int f_{X_s|Y_0=y_0,\ldots,Y_s=y_s,\theta}(u) f_{X_{s+1}|X_s=u,\theta}(x_{s+1}) du} \right] f_{X_{s+1}|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{s+1}) dx_{s+1} \end{aligned} \tag{48}$$

Note that the right hand side above involves the densities $f_{X_{s+1}|Y_0=y_0,\ldots,Y_t=y_t,\theta}$, $f_{X_s|Y_0=y_0,\ldots,Y_s=y_s,\theta}$ and $f_{X_{s+1}|X_s=u,\theta}$. The first of these densities is available to us because we are assuming that we calculated the smoothing density for $s+1$. The second of these densities is a filtering density and will be available after running the forward filtering algorithm. The third of these densities is the transition density of the hidden Markov process that is available from the specification of the state space model.

For the linear Gaussian state space models, the recursion above can be re-written in closed form in terms of the means and covariances of the distributions as we show in the next section.

9.3 Smoothing for Linear Gaussian State Space Models

Consider the linear Gaussian state space model:

$$\begin{aligned} X_0 &\sim N(\mu_0, \Gamma_0) \\ X_t &= A_t X_{t-1} + U_t \\ Y_t &= B_t X_t + V_t \end{aligned} \tag{49}$$

with $X_0, U_1, \ldots, V_0, V_1, \ldots$ independent and $U_t \sim N(0, \Sigma_t)$ and $V_t \sim N(0, R_t)$. Each of the quantities $\mu_0, \Gamma_0, A_t, B_t, \Sigma_t, R_t$ appearing in the model above can depend on an unknown vector of parameters θ. Every conditional distribution $X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ is Gaussian and we can write

$$X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta \sim N(m_{s|t}, Q_{s|t}). \tag{50}$$

We have already seen that the Kalman filter computes $m_{t|t}, Q_{t|t}$ using the following equations:

$$m_{t|t-1} = A_t m_{t-1|t-1} \quad \text{and} \quad Q_{t|t-1} = A_t Q_{t-1|t-1} A_t' + \Sigma_t. \tag{51}$$

and

$$\begin{aligned} m_{t|t} &= m_{t|t-1} + Q_{t|t-1} B_t' \left(B_t Q_{t|t-1} B_t' + R_t \right)^{-1} \left(y_t - B_t m_{t|t-1} \right) \\ Q_{t|t} &= Q_{t|t-1} - Q_{t|t-1} B_t' \left(B_t Q_{t|t-1} B_t' + R_t \right)^{-1} B_t Q_{t|t-1} \end{aligned} \tag{52}$$

The smoothing algorithm described below computes $m_{s|t}, Q_{s|t}$ for a fixed $t \geq 0$ and all $s \leq t$. We shall make use of the following fact that we used previously in the derivation of the Kalman Filter:

Fact 9.1. *Suppose $X \sim N(m_0, Q_0)$ and $Y \mid X = x \sim N(Bx, R)$ (note that the condition $Y \mid X = x \sim N(Bx, R)$ can also be written as $Y = BX + V$ where $V \sim N(0, R)$ with V, X being independent). Then the following assertions hold:*

1. *$X \mid Y = y \sim N(\tilde{m}(y), \tilde{Q})$ where*

$$\begin{aligned} \tilde{m}(y) &= m_0 + Q_0 B' \left(BQ_0B' + R\right)^{-1} (y - Bm_0) \\ \tilde{Q} &= Q_0 - Q_0 B' \left(BQ_0B' + R\right)^{-1} BQ_0 \end{aligned} \tag{53}$$

2. *$Y \sim N(Bm_0, BQ_0B' + R)$*

Note that $\tilde{m}(y)$ depends on y but $\tilde{Q}$ does not depend on y.

Remark 9.1. *Fact 9.1 can be reformulated in terms of densities as follows. Let $\phi(x; \mu, \Sigma)$ denote the multivariate normal density with mean vector μ and covariance matrix Σ evaluated at x i.e.,*

$$\phi(x; \mu, \Sigma) := (2\pi \det(\Sigma))^{-1/2} \exp\left(-\frac{1}{2}(x - \mu)'\Sigma^{-1}(x - \mu)\right).$$

The first conclusion $X \mid Y = y \sim N(\tilde{m}(y), \tilde{Q})$ of Fact 9.1 is equivalent to the identity

$$\frac{\phi(x; m_0, Q_0)\phi(y; Bx, R)}{\int \phi(u; m_0, Q_0)\phi(y; Bu, R)du} = \phi(x; \tilde{m}(y), \tilde{Q}) \tag{54}$$

This is because the left hand side above is simply

$$\frac{f_{Y|X=x}(y) f_X(x)}{\int f_{Y|X=u}(y) f_X(u) du} = f_{X|Y=y}(x).$$

The second conclusion of Fact (9.1) *is equivalent to the identity:*

$$\int \phi(y; Bx, R)\phi(x; m_0, Q_0)dx = \phi(y; Bm_0, BQ_0B' + R). \tag{55}$$

This is because the left hand side above is

$$\int f_{Y|X=x}(y) f_X(x) dx = f_Y(y).$$

It should be easy to see that (55) *is easily extended to the case where the Bx term on the left hand side is replaced by $Bx + c$ for a deterministic vector c:*

$$\int \phi(y; Bx + c, R)\phi(x; m_0, Q_0)dx = \phi(y; Bm_0 + c, BQ_0B' + R). \tag{56}$$

Using the identities (54) and (55), we can rewrite the general backward smoothing recursion (48) as follows.

$$\begin{aligned} &\phi(x_s; m_{s|t}, Q_{s|t}) \\ &= f_{X_s|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_s) \\ &= \int \left[\frac{f_{X_s|Y_0=y_0,\ldots,Y_s=y_s,\theta}(x_s) f_{X_{s+1}|X_s=x_s,\theta}(x_{s+1})}{\int f_{X_s|Y_0=y_0,\ldots,Y_s=y_s,\theta}(u) f_{X_{s+1}|X_s=u,\theta}(x_{s+1})du}\right] f_{X_{s+1}|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{s+1})dx_{s+1} \\ &= \int \left[\frac{\phi(x_s; m_{s|s}, Q_{s|s})\phi(x_{s+1}; A_{s+1}x_s, \Sigma_{s+1})}{\int \phi(u; m_{s|s}, Q_{s|s})\phi(x_{s+1}; A_{s+1}u, \Sigma_{s+1})du}\right] \phi(x_{s+1}; m_{s+1|t}, Q_{s+1|t})dx_{s+1} \end{aligned}$$

Applying (54) with $m_0 = m_{s|s}$, $Q_0 = Q_{s|s}$, $B = A_{s+1}$ and $R = \Sigma_{s+1}$, we deduce that the term inside the square brackets above equals

$$\frac{\phi(x_s; m_{s|s}, Q_{s|s})\phi(x_{s+1}; A_{s+1}x_s, \Sigma_{s+1})}{\int \phi(u; m_{s|s}, Q_{s|s})\phi(x_{s+1}; A_{s+1}u, \Sigma_{s+1})du} = \phi(x_s; \tilde{m}(x_{s+1}), \tilde{Q}) \tag{57}$$

where

$$\tilde{m}(x_{s+1}) = m_{s|s} + Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}\left(x_{s+1} - A_{s+1}m_{s|s}\right)$$
$$\tilde{Q} = Q_{s|s} - Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}A_{s+1}Q_{s|s}$$

As a result

$$\phi(x_s; m_{s|t}, Q_{s|t}) = \int \phi(x_s; \tilde{m}(x_{s+1}), \tilde{Q})\phi(x_{s+1}; m_{s+1|t}, Q_{s+1|t})dx_{s+1} \tag{58}$$

We now apply (56). Note that $\tilde{m}(x_{s+1})$ is a linear function of x_{s+1} and it can be written as $Bx_{s+1} + c$ with

$$B := Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}$$

and

$$c := m_{s|s} - Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}A_{s+1}m_{s|s}.$$

The identity (56) therefore gives that the integral on the right hand side of (58) equals the multivariate normal density with mean $Bm_0 + c$ and covariance $BQ_0B' + R$ evaluated at x_s (here $m_0 = m_{s+1|t}$, $Q_0 = Q_{s+1|t}$ and $R = \tilde{Q}$). Because the left hand side of (58) is the multivariate normal density with mean $m_{s|t}$ and $Q_{s|t}$, we deduce the equations

$$m_{s|t} = m_{s|s} + Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}\left(m_{s+1|t} - A_{s+1}m_{s|s}\right)$$

and

$$\begin{aligned}
&Q_{s|t}\\
&= BQ_0B' + R\\
&= Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}Q_{s+1|t}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}A_{s+1}Q_{s|s}\\
&+ Q_{s|s} - Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}A_{s+1}Q_{s|s}\\
&= Q_{s|s} + Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}Q_{s+1|t}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}A_{s+1}Q_{s|s}\\
&- Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}A_{s+1}Q_{s|s}\\
&= Q_{s|s} + Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}Q_{s+1|t}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}A_{s+1}Q_{s|s}\\
&- Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}A_{s+1}Q_{s|s}\\
&= Q_{s|s}+\\
&Q_{s|s}A'_{s+1}\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}\left(Q_{s+1|t} - A_{s+1}Q_{s|s}A_{s+1} - \Sigma_{s+1}\right)\left(A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}\right)^{-1}A_{s+1}Q_{s|s}
\end{aligned}$$

We shall now write these equations concisely by using the following notation. Recall that from the one-step ahead prediction updates (51), we have

$$m_{s+1|s} = A_{s+1}m_{s|s} \quad \text{and} \quad Q_{s+1|s} = A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}.$$

Replacing the terms $A_{s+1}m_{s|s}$ and $Q_{s+1|s} = A_{s+1}Q_{s|s}A'_{s+1} + \Sigma_{s+1}$ by $m_{s+1|s}$ and $Q_{s+1|s}$ in the smoothing recursion equations, we get

$$m_{s|t} = m_{s|s} + Q_{s|s}A'_{s+1}Q^{-1}_{s+1|s}\left(m_{s+1|t} - m_{s+1|s}\right)$$
$$Q_{s|t} = Q_{s|s} + Q_{s|s}A'_{s+1}Q^{-1}_{s+1|s}\left(Q_{s+1|t} - Q_{s+1|s}\right)Q^{-1}_{s+1|s}A_{s+1}Q_{s|s}$$

Finally using the notation

$$\Gamma_{s+1} := Q_{s|s} A'_{s+1} Q^{-1}_{s+1|s},$$

we get

$$\begin{aligned} m_{s|t} &= m_{s|s} + \Gamma_{s+1} \left(m_{s+1|t} - m_{s+1|s} \right) \\ Q_{s|t} &= Q_{s|s} + \Gamma_{s+1} \left(Q_{s+1|t} - Q_{s+1|s} \right) \Gamma'_{s+1} \end{aligned}$$

These are the Kalman Smoothing equations; alternatively known as the Rauch-Tung-Striebel equations. They allow the calculation of $m_{s|t}, Q_{s|t}$ from knowledge of $m_{s+1|t}, Q_{s+1|t}$ as well as from $m_{s|s}, Q_{s|s}, m_{s+1|s}, Q_{s+1|s}$ (these four quantities are obtained by running the Kalman filter). One runs these smoothing equations starting from $s = t - 1$ and decreasing s all the way to zero.

9.4 Dealing with missing data in the context of state space models

Consider a time series dataset $y_0, y_1, \ldots, y_T$ where observations corresponding to certain time points may be missing. More precisely, the data might look like $y_0, y_1, y_2, \text{miss}, y_4, y_5, y_6, \text{miss}, y_8, \ldots$. How does one analyze this dataset? In the context of state space models, this is quite straightforward. As usual, we use a state space model with a hidden Markov process $\{X_t\}$ and then connect it to the observation random variables $Y_0, Y_1, \ldots$. In contrast to the fully observed setup, we now assume that each Y_t takes an additional value "miss" which means that we should also model:

$$\mathbb{P}\left\{ Y_t = \text{miss} \mid X_t = x \right\}.$$

Modeling this probability requires us to know the missing mechanism which is quite difficult in general. A simplistic assumption is that

$$\mathbb{P}\left\{ Y_t = \text{miss} \mid X_t = x \right\} \text{ does not depend on } x. \tag{59}$$

This is can be viewed as a "missing at random" assumption. Under this assumption, analysis is quite straightforward and the Kalman filter and smoother for the model with missing observations are obtained by a simple modification of the model without missing observations. For example, here is how to run the Kalman filter in the presence of missing observations and the missing at random assumption (59). The Kalman filter tells us about the step:

$$X_{t-1} \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta \quad \text{to} \quad X_t \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta$$

which is the one-step ahead prediction update and then about the

$$X_t \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta \quad \text{and} \quad X_t \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, Y_t = y_t, \theta \tag{60}$$

which is the filter update. When y_t is observed, both these steps are carried out as usual. However when y_t is missing, then there is nothing to do in the filter update so we just take the two conditional distributions in (60) to be the same (observe that the missing at random assumption is crucial here). The smoothing procedure is the same as in the fully observed case. We shall look at specific examples in the next class.

9.5 Recommended Reading for Today

1. The general smoothing approach described in Section 9.2 can be found in:
   a) Section 6.2.1 of the Kitagawa-Gersch book (in particular, see Equation (6.7))

b) Section 14.2 of the Kitagawa book.

c) Section 2.7.4 of the Petris-Petrone-Campagnoli book

d) Section 8.1 of the Särkkä book

2. The Kalman/Rauch-Tung-Striebel smoothing equations are described in all the books listed in the course outline:

   a) Section 5.2 of the Kitagawa-Gersch book (in particular, see equation (5.6)

   b) Section 9.3 of the Kitagawa book

   c) Section 4.4 of the Durbin-Koopman book

   d) Section 8.2 of the Särkkä book

   e) Proposition 2.4 of the Petris-Petrone-Campagnoli book

   f) Theorem 3.4 of the Triantafyllopoulos book

   Section 7.2 of the Chopin-Papaspiliopoulos book also discusses the Kalman smoothing equations. They however derive the algorithm from a general Feynman-Kac formalism (see their Chapter 5). I will discuss the Feynman-Kac stuff in class a few weeks later.

3. For missing data:

   a) See Section 2.7 of the Durbin-Koopman book for a treatment of missing observations for the local level model and Section 4.10 of the Durbin-Koopman book for a more general treatment of missing observations for linear Gaussian state space models.

   b) See Section 9.7 of the Kitagawa book.

   c) Section 2.7.3 of the Petris-Petrone-Campagnoli book for filtering with missing observations (also see page 62 of Petris-Petrone-Campagnoli where it is argued that no changes to the smoothing step is necessary for dealing with missing observations).

10 Lecture Ten

10.1 Summary: General Filtering and Smoothing

Let us use the following simpler notation. Let $f_{s|t}(x_s)$ denote the conditional density of X_s given $Y_0 = y_0, \ldots, Y_t = y_t, \theta$ evaluated at the point x_s.

Filtering recursions for general state space models are (see Lecture Six) the following. The one-step ahead prediction update is

$$f_{t|t-1}(x_t) = \int f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{t-1|t-1}(x_{t-1}) dx_{t-1}, \tag{61}$$

and the filter update is

$$f_{t|t}(x_t) \propto f_{Y_t|X_t=x_t,\theta}(y_t) f_{t|t-1}(x_t). \tag{62}$$

The backward smoothing recursion for general state space models is (see Lecture Nine):

$$f_{s|t}(x_s) = \int \left[\frac{f_{s|s}(x_s) f_{X_{s+1}|X_s=x_s,\theta}(x_{s+1})}{\int f_{s|s}(u) f_{X_{s+1}|X_s=u,\theta}(x_{s+1}) du} \right] f_{s+1|t}(x_{s+1}) dx_{s+1} \tag{63}$$

This can be rewritten in the following way. Note first that the denominator (inside the square brackets) equals:

$$\begin{aligned}
&\int f_{s|s}(u) f_{X_{s+1}|X_s=u,\theta}(x_{s+1}) du \\
&= \int f_{X_s|Y_0=y_0,\ldots,Y_s=y_s,\theta}(u) f_{X_{s+1}|X_s=u,\theta}(x_{s+1}) du \\
&= \int f_{X_s|Y_0=y_0,\ldots,Y_s=y_s,\theta}(u) f_{X_{s+1}|X_s=u,Y_0=y_0,\ldots,Y_s=y_s,\theta}(x_{s+1}) du \\
&= f_{X_{s+1}|Y_0=y_0,\ldots,Y_s=y_s,\theta}(x_{s+1}) = f_{s+1|s}(x_{s+1})
\end{aligned}$$

Thus (63) becomes

$$f_{s|t}(x_s) = \int \frac{f_{s|s}(x_s) f_{X_{s+1}|X_s,\theta}(x_{s+1})}{f_{s+1|s}(x_{s+1})} f_{s+1|t}(x_{s+1}) dx_{s+1}.$$

Note also that the $f_{s|s}(x_s)$ term on the right hand side can be pulled out of the integral (as it does not depend on x_{s+1} which is the variable of integration). Thus the smoothing recursion becomes:

$$f_{s|t}(x_s) = f_{s|s}(x_s) \int \frac{f_{X_{s+1}|X_s,\theta}(x_{s+1}) f_{s+1|t}(x_{s+1})}{f_{s+1|s}(x_{s+1})} dx_{s+1}. \tag{64}$$

10.2 Summary: Kalman Filtering and Smoothing

The general filtering and smoothing recursions can be computed in closed form for the case of the linear Gaussian state space model:

$$\begin{aligned}
X_0 &\sim N(\mu_0, \Gamma_0) \\
X_t &= A_t X_{t-1} + U_t \\
Y_t &= B_t X_t + V_t
\end{aligned}$$

with $X_0, U_1, \ldots, V_0, V_1, \ldots$ independent and $U_t \sim N(0, \Sigma_t)$ and $V_t \sim N(0, R_t)$. Here every density $f_{s|t}$ is normal and we shall write $m_{s|t}$ and $Q_{s|t}$ for the mean and covariance corresponding to the (possibly multivariate) normal density $f_{s|t}$. The one-step ahead prediction update (61) becomes

$$m_{t|t-1} = A_t m_{t-1|t-1} \quad \text{and} \quad Q_{t|t-1} = A_t Q_{t-1|t-1} A_t' + \Sigma_t$$

The filter update (62) becomes

$$\begin{aligned}
m_{t|t} &= m_{t|t-1} + Q_{t|t-1} B_t' \left(B_t Q_{t|t-1} B_t' + R_t \right)^{-1} \left(y_t - B_t m_{t|t-1} \right) \\
Q_{t|t} &= Q_{t|t-1} - Q_{t|t-1} B_t' \left(B_t Q_{t|t-1} B_t' + R_t \right)^{-1} B_t Q_{t|t-1}
\end{aligned}$$

Finally the smoothing backward recursion (63) becomes

$$\begin{aligned}
m_{s|t} &= m_{s|s} + \Gamma_{s+1} \left(m_{s+1|t} - m_{s+1|s} \right) \\
Q_{s|t} &= Q_{s|s} + \Gamma_{s+1} \left(Q_{s+1|t} - Q_{s+1|s} \right) \Gamma_{s+1}'
\end{aligned}$$

where

$$\Gamma_{s+1} := Q_{s|s} A_{s+1}' Q_{s+1|s}^{-1}.$$

10.3 Special Case: Local Level Model

Let us specialize the Kalman filtering and smoothing recursions for the special case of the local level model:

$$\begin{aligned} X_0 &\sim N(0, C) \\ X_t &= X_{t-1} + Z_t \qquad \text{with } U_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_Z^2) \\ Y_t &= X_t + \epsilon_t \qquad \text{with } \epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2) \end{aligned}$$

We have already seen that the Kalman filter for the local level model is:

$$m_{t|t-1} = m_{t-1|t-1} \quad \text{and} \quad Q_{t|t-1} = Q_{t-1|t-1} + \sigma_Z^2$$

and

$$m_{t|t} = \frac{\sigma_\epsilon^2}{Q_{t|t-1} + \sigma_\epsilon^2} m_{t|t-1} + \frac{Q_{t|t-1}}{Q_{t|t-1} + \sigma_\epsilon^2} y_t \quad \text{and} \quad Q_{t|t} = \frac{\sigma_\epsilon^2 Q_{t|t-1}}{Q_{t|t-1} + \sigma_\epsilon^2}.$$

The Kalman smoothing equations become (note that $\Gamma_{s+1} = \frac{Q_{s|s}}{Q_{s+1|s}} = \frac{Q_{s|s}}{Q_{s|s}+\sigma_Z^2}$)

$$\begin{aligned} m_{s|t} &= m_{s|s} + \frac{Q_{s|s}}{Q_{s|s} + \sigma_Z^2} \left(m_{s+1|t} - m_{s+1|s} \right) \\ &= m_{s|s} + \frac{Q_{s|s}}{Q_{s|s} + \sigma_Z^2} \left(m_{s+1|t} - m_{s|s} \right) = \frac{\sigma_Z^2}{Q_{s|s} + \sigma_Z^2} m_{s|s} + \frac{Q_{s|s}}{Q_{s|s} + \sigma_Z^2} m_{s+1|t}, \end{aligned}$$

and

$$\begin{aligned} Q_{s|t} &= Q_{s|s} + \left(\frac{Q_{s|s}}{Q_{s|s} + \sigma_Z^2} \right)^2 \left(Q_{s+1|t} - Q_{s+1|s} \right) \\ &= Q_{s|s} + \left(\frac{Q_{s|s}}{Q_{s|s} + \sigma_Z^2} \right)^2 \left(Q_{s+1|t} - Q_{s|s} - \sigma_Z^2 \right) \\ &= Q_{s|s} - \frac{Q_{s|s}^2}{Q_{s|s} + \sigma_Z^2} + \left(\frac{Q_{s|s}}{Q_{s|s} + \sigma_Z^2} \right)^2 Q_{s+1|t} = \frac{Q_{s|s}\sigma_Z^2}{Q_{s|s} + \sigma_Z^2} + \left(\frac{Q_{s|s}}{Q_{s|s} + \sigma_Z^2} \right)^2 Q_{s+1|t}. \end{aligned}$$

This local level model is useful for estimating smooth trends in time series. However it does not work well for estimating nonsmooth trends such as piecewise constant trend functions. For piecewise constant trend functions, using certain non-Gaussian distributions for the evolution errors $\{Z_t\}$ works well. For example, one can use

$$Z_t \overset{\text{i.i.d}}{\sim} C(0, \sigma_Z^2) \tag{65}$$

or

$$Z_t \overset{\text{i.i.d}}{\sim} \alpha N(0, \text{small}) + (1 - \alpha) N(0, \sigma_2^2). \tag{66}$$

In (65), $C(0, \sigma_Z^2)$ denotes the Cauchy density centered at 0 with scale parameter σ_Z:

$$z \mapsto \frac{\sigma_Z}{\pi(z^2 + \sigma_Z^2)}.$$

(66) is a mixture density with two components: the first component is a normal density centered at zero with small variance and the second component is a normal density centered at zero with variance σ_2^2. For fitting piecewise constant trend functions, we would take α to be close to 1.

When the density of Z_t is not normal (as when it is of the form (65) or (66)), the overall model is not "linear Gaussian" so that Kalman filtering and smoothing are not applicable. We will study two ways of solving the filtering and smoothing problems in such models. The first approach is to numerically evaluate the general recursions of Section 10.1 by discretization. This method is described in the next section. The second approach is to use Monte Carlo approximation and we shall discuss this later (this approach is known as "Sequential Monte Carlo" and is the main focus of the Chopin-Papaspiliopoulos book for example).

10.4 Numerical Evaluation of $X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$

This method works for arbitrary state space models. It is also conceptually simpler (compared to the Kalman recursions) but it is computationally quite intensive because we need to recursively compute entire densities (as opposed to just means and covariances as in the Kalman recursions).

We shall discretize space by placing a dense grid $x^{(g)}, g \in G$ covering the range of X_t. We shall use the same grid for each X_t for simplicity although in principle different grids may be used for different values of t. We shall reduce all densities to probability mass functions over $\{x^{(g)}, g \in G\}$. Let $p_{s|t}(x^{(g)}), g \in G$ denote the probability mass function that approximates the density $f_{s|t}(\cdot)$ i.e.,

$$p_{s|t}(x^{(g)}) \propto f_{s|t}(x^{(g)})$$

or, more precisely,

$$p_{s|t}(x^{(g)}) = \frac{f_{s|t}(x^{(g)})}{\sum_{g' \in G} f_{s|t}(x^{(g')})}.$$

From knowledge of $p_{s|t}(x^{(g)}), g \in G$, the density $f_{s|t}(x)$ cannot be determined precisely for all x but it can be approximated well if the grid G is dense. For example, one can use the approximation

$$f_{s|t}(x) \propto p_{s|t}(x^{(g)}) \qquad \text{for } x^{(g)} \text{ that is closest to } x.$$

We shall discretize the recursions (61), (62) and(63). The discrete equation corresponding to the one-step ahead prediction update (61) is given by

$$p_{t|t-1}(x^{(g)}) \propto \sum_{\tilde{g} \in G} f_{X_t|X_{t-1}=x^{(\tilde{g})}}(x^{(g)}) p_{t-1|t-1}(x^{\tilde{g}})$$

The normalization constant can be written explicitly as

$$p_{t|t-1}(x^{(g)}) = \frac{\sum_{\tilde{g} \in G} f_{X_t|X_{t-1}=x^{(\tilde{g})},\theta}(x^{(g)}) p_{t-1|t-1}(x^{\tilde{g}})}{\sum_{g'} \sum_{\tilde{g} \in G} f_{X_t|X_{t-1}=x^{(\tilde{g})}}(x^{(g')}) p_{t-1|t-1}(x^{\tilde{g}})} \tag{67}$$

The discrete equation corresponding to the filtering update (62) is given by

$$p_{t|t}(x^{(g)}) \propto f_{Y_t|X_t=x^{(g)}}(y_t) p_{t|t-1}(x^{(g)})$$

which becomes the following with proper normalization:

$$p_{t|t}(x^{(g)}) = \frac{f_{Y_t|X_t=x^{(g)}}(y_t) p_{t|t-1}(x^{(g)})}{\sum_{g' \in G} f_{Y_t|X_t=x^{(g')}}(y_t) p_{t|t-1}(x^{(g')})} \tag{68}$$

Finally the discretized version of the smoothing recursion (63) is

$$p_{s|t}(x^{(g)}) = p_{s|s}(x^{(g)}) \sum_{\tilde{g} \in G} \left(\frac{f_{X_{s+1}|X_s=x^{(g)},\theta}(x^{(\tilde{g})})}{\sum_{g' \in G} f_{X_{s+1}|X_s=x^{(g)},\theta}(x^{(g')})} \right) \frac{p_{s+1|t}(x^{(g)})}{p_{s+1|s}(x^{(\tilde{g})})} \tag{69}$$

These three equations (67), (68) and (69) can be implemented to compute $p_{s|t}$ for all $s \leq t$. The recursions can be initialized with

$$p_{0|-1}(x^{(g)}) \propto 1 \qquad \text{for all } g \in G.$$

This corresponds to a diffuse prior on X_0.

These recursions are used, for example, to fit the local level model with evolution errors (65) or (66).

10.5 Recommended Reading for Today

1. Kalman smoothing equations for the local level model can be found in Section 2.4 of the Durbin-Koopman book and Example 8.1 of the Särkkä book
2. The numerical recursions for filtering and smoothing can be found in Section 6.3 of the Kitagawa-Gersch book and Section 14.3 of the Kitagawa book.
3. For estimating smooth trend functions with state space models, see Section 8.2 of the Kitagawa-Gersch book or Chapter 11 of the Kitagawa book.
4. The local level model with Cauchy errors has been used to fit a piecewise constant trend function in Section 14.4 of the Kitagawa book (they actually used the more general Pearson family for the evolution errors). Section 8.4 of the Kitagawa-Gersch book also considers the Pearson family for the evolution errors as well as the normal mixture distribution (66).

11 Lecture Eleven

We shall discuss optimization algorithms for maximum likelihood estimation for state space models. We start with a general discussion of optimization algorithms before specializing to the case of state space models.

11.1 Basic Optimization Algorithms

The goal is to maximize a function $F(\theta)$ over θ. Optimization algorithms are iterative and output a sequence of values $\theta^{(0)}, \theta^{(1)}, \ldots$ which is supposed to converge to a (local) maximizer of F. We shall describe briefly three standard optimization algorithms: Gradient Ascent, Newton's method and BFGS.

11.1.1 Gradient Ascent

Given the current iterate $\theta^{(n)}$, consider the first order Taylor expansion of F near $\theta^{(n)}$:

$$F(\theta) \approx F(\theta^{(n)}) + \left\langle \nabla F(\theta^{(n)}), \theta - \theta^{(n)} \right\rangle.$$

This suggests that $F(\theta) \geq F(\theta^{(n)})$ provided

$$\left\langle \nabla F(\theta^{(n)}), \theta - \theta^{(n)} \right\rangle \geq 0$$

which will be satisfied when

$$\theta - \theta^{(n)} = \alpha_n \nabla F(\theta^{(n)}) \qquad \text{for } \alpha \geq 0.$$

Motivated by this, the gradient ascent update is

$$\theta^{(n+1)} = \theta^{(n)} + \alpha_n \nabla F(\theta^{(n)}).$$

The quantity α_n is called the step-size and the best way to choose it (which guarantees improvement in function values) is to maximize the quantity

$$F\left(\theta^{(n)} + \alpha \nabla F(\theta^{(n)})\right) \qquad \text{over all } \alpha \geq 0.$$

The above one-parameter maximization can be done by a line search.

11.1.2 Newton's Method

Given the current iterate $\theta^{(n)}$, consider the second order Taylor expansion of F near $\theta^{(n)}$:

$$F(\theta) \approx F(\theta^{(n)}) + \left\langle \nabla F(\theta^{(n)}), \theta - \theta^{(n)} \right\rangle + \frac{1}{2}(\theta - \theta^{(n)})' HF(\theta^{(n)})(\theta - \theta^{(n)}). \tag{70}$$

The maximizer of the right hand side above can be calculated in closed form as:

$$\theta - \theta^{(n)} = \left(-HF(\theta^{(n)})\right)^{-1} \left(\nabla F(\theta^{(n)})\right).$$

This motivates setting

$$\theta^{(n+1)} = \theta^{(n)} + \alpha_n \left(-HF(\theta^{(n)})\right)^{-1} \left(\nabla F(\theta^{(n)})\right) \tag{71}$$

where again α_n is chosen to maximize the quantity

$$F\left(\theta^{(n)} + \alpha \left(-HF(\theta^{(n)})\right)^{-1} \left(\nabla F(\theta^{(n)})\right)\right) \qquad \text{over all } \alpha \geq 0.$$

Note that it is important that $-HF(\theta^{(n)}$ must be positive semi-definite for the quadratic approximation (70) to have a well-defined maximum (otherwise, its maximum will be $+\infty$).

Newton's method works very well when initialized reasonably close to the actual maximizer of F. But one needs to calculate the Hessian matrix $HF(\theta^{(n)})$ which may be difficult on impossible in some applications.

11.1.3 Quasi-Newton Method: BFGS

Quasi-Newton methods mimic the Newton update (71) without explicitly including Hessian matrices. Instead the idea is to have approximate Hessians and update them at each step. The most popular of these methods is BFGS (Broyden-Fletcher-Goldfarb-Shanno) and this method works as follows. At each step of the procedure, the current estimate of the maximizer $\theta^{(n)}$ is updated to the next value $\theta^{(n+1)}$ and the current approximate Hessian matrix $H^{(n)}$ is also updated to the next value $H^{(n+1)}$. The update $\theta^{(n)} \to \theta^{(n+1)}$ is exactly the same as (71) with $HF(\theta^{(n)})$ replaced by the current Hessian approximation $H^{(n)}$:

$$\theta^{(n+1)} = \theta^{(n)} + \alpha_n \left(-H^{(n)}\right)^{-1} \left(\nabla F(\theta^{(n)})\right) \tag{72}$$

where, as before, α_n is chosen to maximize the quantity

$$F\left(\theta^{(n)} + \alpha\left(-H^{(n)}\right)^{-1}\left(\nabla F(\theta^{(n)})\right)\right) \qquad \text{over all } \alpha \geq 0.$$

The update for the Hessian is given by (we are assuming that each $-H^{(n)}$ is symmetric and positive definite)

$$H^{(n+1)} = H^{(n)} + \frac{gg'}{g's} - \frac{H^{(n)}ss'H^{(n)}}{s'H^{(n)}s} \tag{73}$$

where

$$s := \theta^{(n+1)} - \theta^{(n)} \quad \text{and} \quad g := \nabla F\left(\theta^{(n+1)}\right) - \nabla F\left(\theta^{(n)}\right).$$

The Hessian update can also be written in terms of $(H^{(n)})^{-1}$:

$$\left(H^{(n+1)}\right)^{-1} = \left(I - \frac{sg'}{g's}\right)\left(H^{(n)}\right)^{-1}\left(I - \frac{gs'}{g's}\right) + \frac{ss'}{g's}. \tag{74}$$

This is useful because the θ-update (72) is written in terms of the inverse of $H^{(n)}$.

Here is some intuition behind the Hessian update (73) (or, equivalently, (74)). It is easy to check that $H^{(n+1)}s = g$ which is same as

$$H^{(n+1)}\left(\theta^{(n+1)} - \theta^{(n)}\right) = \nabla F(\theta^{(n+1)}) - \nabla F(\theta^{(n)}).$$

This is a reasonable condition to insist because $H^{(n+1)}$ is supposed to approximate $HF(\theta^{(n+1)})$. Observe that when the dimension equals 1, the equality $H^{(n+1)}s = g$ is the same as

$$H^{(n+1)} = \frac{F'(\theta^{(n+1)}) - F'(\theta^{(n)})}{\theta^{(n+1)} - \theta^{(n)}}.$$

The matrix $H^{(n+1)}$ defined by (73) is actually the solution to the following optimization problem:

$$H^{(n+1)} = \operatorname*{argmin}_X \left\{D(X\|H^{(n)}) : Xs = g \text{ and } X \text{ is psd}\right\}$$

where

$$D(X\|H) := \frac{1}{2}\left[\operatorname{tr}\left((H^{(n)})^{-1}X\right) - \log\det\left((H^{(n)})^{-1}X\right) - d\right].$$

Here d is the dimension of θ (note that each H is $d \times d$). The above expression $D(X\|H)$ is the Kullback-Leibler divergence between the multivariate normal distribution with covariance X and the multivariate normal distribution with covariance H. At a high level, $H^{(n+1)}$ should be understood as the closed matrix to $H^{(n)}$ (measured in terms of the divergence $D(\cdot\|H^{(n)})$) subject to the condition $H^{(n+1)}s = g$. It is standard to initialize $H^{(0)}$ with the identity matrix.

Observe that in order to apply the gradient ascent and the BFGS methods, it is necessary to be able to compute the gradients of F. To apply the Newton method, one also needs to compute the Hessian of F.

If you want to learn more about these optimization algorithms, you can read standard books on nonlinear optimization; I can recommend *Numerical Optimization* by Nocedal and Wright, or the first chapter of *Introductory Lectures on Convex Optimization* by Nesterov, or *Iterative Methods for Optimization* by Kelley.

11.2 Application to Maximum Likelihood Estimation in State Space Models

We shall apply the optimization algorithms for obtaining maximum likelihood estimates in state space models. The function F in the previous section will now be the log-likelihood function. We have seen that it can be calculated for state space models by filtering (in particular, the Kalman filter can be used for likelihood computation in linear Gaussian state space models). As we saw in the previous section, gradient ascent and BFGS require gradient evaluations. We thus need to calculate the gradient of the log-likelihood function in state space models. One often uses the term *score vector* or *score function* for the gradient of the log-likelihood function.

For calculating the score vector in state space models (and more generally in latent variable models), it is convenient to use the Fisher identity which we shall describe next.

11.2.1 Fisher Identity for the Score

Consider a general latent variable model which describes the joint density $f_{Y,X|\theta}(y,x)$ of two variables Y, X in terms of parameters θ. Here Y denotes the observed variable (the observed data from Y will be denoted by y) and X denotes the hidden or latent variable (we will not be observing any specific realizations x corresponding to X). This setting is quite general and includes the state space model as special case. For state space models, $Y = (Y_0, \ldots, Y_T)$ and $X = (X_0, \ldots, X_T)$.

The likelihood of the observation y is simply equal to the density of Y at y:

$$f_{Y|\theta}(y) = \int f_{Y,X|\theta}(y,x)dx$$

viewed as a function of the parameters θ. Generally in latent variable models, $f_{Y|\theta}(y)$ is harder to evaluate compared to $f_{Y,X|\theta}(y,x)$. Our goal here is to calculate the score function (gradient of the log-likelihood) at a specific parameter value $\theta^{(0)}$. More precisely, we want to calculate:

$$\nabla_\theta \log f_{Y|\theta}(y)\Big|_{\theta=\theta^{(0)}}$$

Fisher's identity provides a formula for the score in terms of $f_{Y,X|\theta}$:

Fact 11.1 (Fisher's Identity)**.** *For every $\theta^{(0)}$ and y, we have*

$$\nabla_\theta \log f_{Y|\theta}(y)\Big|_{\theta=\theta^{(0)}} = \nabla_\theta E\left(\theta, \theta^{(0)}\right)\Big|_{\theta=\theta^{(0)}} \tag{75}$$

where

$$E(\theta, \theta^{(0)}) := \int \left[\log f_{Y,X|\theta}(y,x)\right] f_{X|Y=y,\theta=\theta^{(0)}}(x)dx \tag{76}$$

Proof. Fix $\theta^{(0)}$ and y. Note that for every x,

$$f_{Y|\theta}(y) = \frac{f_{Y,X|\theta}(y,x)}{f_{X|Y=y,\theta}(x)}$$

which can be rewritten as

$$\log f_{Y|\theta}(y) = \log f_{Y,X|\theta}(y,x) - \log f_{X|Y=y,\theta}(x). \tag{77}$$

We now integrate both sides of the above equality with respect to the probability density

$$q(x) := f_{X|Y=y,\theta=\theta^{(0)}}(x).$$

Note that the function $x \mapsto q(x)$ depends on y and $\theta^{(0)}$ but it does not depend on the generic parameter value θ appearing in (77). Integrating both sides of (77) with respect to $q(x)$ (note that the left hand side of (77) does not depend on x), we get

$$\begin{aligned}\log f_{Y|\theta}(y) &= \int \left[\log f_{Y,X|\theta}(y,x)\right] q(x)dx - \int \left[\log f_{X|Y=y,\theta}(x)\right] q(x)dx \\ &= E(\theta,\theta^{(0)}) - \int \left[\log f_{X|Y=y,\theta}(x)\right] q(x)dx.\end{aligned}$$

We now take the gradient on both sides with respect to θ and evaluate the gradient at $\theta = \theta^{(0)}$. This leads to

$$\nabla_\theta \log f_{Y|\theta}(y)\Big|_{\theta=\theta^{(0)}} = \nabla_\theta E(\theta,\theta^{(0)})\Big|_{\theta=\theta^{(0)}} - \nabla_\theta \int \left[\log f_{X|Y=y,\theta}(x)\right] q(x)dx\Big|_{\theta=\theta^{(0)}}.$$

Thus, to complete the proof of (82), it is enough to show that the last term above equals zero. This is true because

$$\begin{aligned}\nabla_\theta \int \left[\log f_{X|Y=y,\theta}(x)\right] q(x)dx\Big|_{\theta=\theta^{(0)}} &= \int \nabla_\theta \left[\log f_{X|Y=y,\theta}(x)\right]\Big|_{\theta=\theta^{(0)}} q(x)dx \\ &= \int \frac{\nabla_\theta f_{X|Y=y,\theta}(x)\Big|_{\theta=\theta^{(0)}}}{f_{X|Y=y,\theta=\theta^{(0)}}(x)} q(x)dx \\ &= \int \frac{\nabla_\theta f_{X|Y=y,\theta}(x)\Big|_{\theta=\theta^{(0)}}}{f_{X|Y=y,\theta=\theta^{(0)}}(x)} f_{X|Y=y,\theta=\theta^{(0)}}(x)dx \\ &= \int \nabla_\theta f_{X|Y=y,\theta}(x)\Big|_{\theta=\theta^{(0)}} dx \\ &= \nabla_\theta \left[\int f_{X|Y=y,\theta}(x)dx\right]\Big|_{\theta=\theta^{(0)}} \\ &= \nabla_\theta [1]\Big|_{\theta=\theta^{(0)}} = 0.\end{aligned}$$

Note that at two places in the above chain of equalities, we interchanged the operations of differentiation (with respect to θ) and integration (with respect to x). □

As we shall see in the next class, the quantity $E(\theta,\theta^{(0)})$ also appears in the EM algorithm. We shall often write it as

$$E(\theta,\theta^{(0)}) = \mathbb{E}_{\theta^{(0)}}\left[\log f_{Y,X|\theta}(y,X) \mid Y = y\right].$$

The notation on the right hand side needs to be understood correctly. The parameter θ appearing in $\log f_{Y,X|\theta}(y,X)$ will remain as θ (i.e., it will not be replaced by $\theta^{(0)}$). $\mathbb{E}_{\theta^{(0)}}$ represents expectation over X with respect to the density $f_{X|Y=y,\theta=\theta^{(0)}}$.

11.2.2 $E(\theta, \theta^{(0)})$ for state space models

For state space models,

$$\begin{aligned}\log f_{Y,X|\theta}(y,x) &= \log f_{X_0,\ldots,X_T,Y_0,\ldots,Y_T|\theta}(x_0,\ldots,x_T,y_0,\ldots,y_T)\\ &= \log f_{X_0|\theta}(x_0) + \sum_{t=1}^{T} \log f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) + \sum_{t=0}^{T} \log f_{Y_t|X_t=x_t,\theta}(y_t).\end{aligned}$$

Observe that the right hand side above involves three kinds of quantities: the observed data $y_0,\ldots,y_T$, the parameters θ and the quantities $x_0,\ldots,x_T$. From here, to obtain $E(\theta,\theta^{(0)})$, we leave $y_0,\ldots,y_T,\theta$ unchanged in the right hand side and take the expectation over $x_0,\ldots,x_T$ conditional on $y_0,\ldots,y_T$. This conditional expectation depends on parameters and we shall fix the parameters at $\theta^{(0)}$ (as opposed to the θ that is already appearing on the right hand side). We can thus write

$$E(\theta,\theta^{(0)}) = I_1(\theta,\theta^{(0)}) + I_2(\theta,\theta^{(0)}) + I_3(\theta,\theta^{(0)}) \tag{78}$$

where

$$\begin{aligned}I_1(\theta,\theta^{(0)}) &= \int \left[\log f_{X_0|\theta}(x_0)\right] f_{X_0|Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}}(x_0)dx_0\\ &= \mathbb{E}\left[\log f_{X_0|\theta}(X_0) \mid Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}\right],\end{aligned}$$

and

$$\begin{aligned}I_2(\theta,\theta^{(0)}) &= \sum_{t=1}^{T} \int\int \left[\log f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t)\right] f_{X_t,X_{t-1}|Y_0=y_0,\ldots Y_T=y_T,\theta^{(0)}}(x_t,x_{t-1})dx_t dx_{t-1}\\ &= \sum_{t=1}^{T} \mathbb{E}\left[\log f_{X_t|X_{t-1}=X_{t-1},\theta}(X_t) \mid Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}\right],\end{aligned}$$

and

$$\begin{aligned}I_3(\theta,\theta^{(0)}) &= \sum_{t=0}^{T} \int \left[\log f_{Y_t|X_t=x_t,\theta}(y_t)\right] f_{X_t|Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}}(x_t)dx_t\\ &= \sum_{t=0}^{T} \mathbb{E}\left[\log f_{Y_t|X_t=X_t,\theta}(y_t) \mid Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}\right].\end{aligned}$$

Note that $I_3(\theta,\theta^{(0)})$ involves expectation with respect to the conditional distribution

$$X_t \mid Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}$$

and $I_1(\theta,\theta^{(0)})$ involves expectation with respect to the above conditional distribution for $t=0$. These conditional distributions are obtained from the smoothing algorithm. Further $I_2(\theta,\theta^{(0)})$ involves expectation with respect to

$$X_t, X_{t-1} \mid Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}$$

which can also be obtained from the smoothing algorithm (we shall see the reasoning behind this in the next class).

For the linear Gaussian state space models, $I_1(\theta,\theta^{(0)}), I_2(\theta,\theta^{(0)}), I_3(\theta,\theta^{(0)})$ can be computed in closed form in terms of the output of the Kalman smoothing algorithm. It is also possible to obtain closed form expressions for the gradient of $E(\theta,\theta^{(0)})$. This is a nice way of computing the score function in linear Gaussian state space models using the Kalman smoother output. We shall see the details in the next class.

11.3 Recommended Reading for Today

1. Some references for an in-depth coverage of optimization algorithms are the books *Numerical Optimization* by Nocedal and Wright, or the first chapter of *Introductory Lectures on Convex Optimization* by Nesterov, or *Iterative Methods for Optimization* by Kelley.

2. For a quick review of optimization algorithms with the goal of applying them to parameter estimation in state space models, see Section 7.3 of the Durbin-Koopman book, Section 14.4 of the Chopin-Papaspiliopoulos book, and Appendix A of the Kitagawa book.

3. Fisher's identity can be found in Section 7.3.3 of the Durbin-Koopman book (although they don't call it the Fisher identity), and Exercise 12.5 of the Chopin-Papaspiliopoulos book, and Equation (12.32) in the Särkkä book.

4. For the formula (87), see equations (12.29) and (12.30) of the Särkkä book.

12 Lecture Twelve

12.1 Pairwise Smoothing Distributions

In our study of smoothing algorithms, we have focussed on calculating the distribution of $X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ for fixed $s \leq t$. For the score vector calculation (as well in the EM algorithm), we would need to calculate the conditional joint distribution of X_s and X_{s+1} given $Y_0 = y_0, \ldots, Y_t = y_t, \theta$. In the general case, this can be done via

$$\begin{aligned} &f_{X_s, X_{s+1} | Y_0 = y_0, \ldots, Y_t = y_t, \theta}(x_s, x_{s+1}) \\ &= f_{X_s | X_{s+1} = x_{s+1}, Y_0 = y_0, \ldots, Y_t = y_t, \theta}(x_s) f_{X_{s+1} | Y_0 = y_0, \ldots, Y_t = y_t, \theta}(x_s) \end{aligned}$$

We have seen in Lecture Nine that the first term in the right hand side above equals

$$\begin{aligned} f_{X_s | X_{s+1} = x_{s+1}, Y_0 = y_0, \ldots, Y_t = y_t, \theta}(x_s) &= f_{X_s | X_{s+1} = x_{s+1}, Y_0 = y_0, \ldots, Y_s = y_s, \theta}(x_s) \\ &= \frac{f_{X_s | Y_0 = y_0, \ldots, Y_s = y_s, \theta}(x_s) f_{X_{s+1} | X_s = x_s, \theta}(x_{s+1})}{\int f_{X_s | Y_0 = y_0, \ldots, Y_s = y_s, \theta}(u) f_{X_{s+1} | X_s = u, \theta}(x_{s+1}) du}. \end{aligned}$$

We thus have

$$\begin{aligned} &f_{X_s, X_{s+1} | Y_0 = y_0, \ldots, Y_t = y_t, \theta}(x_s, x_{s+1}) \\ &= \left[\frac{f_{X_s | Y_0 = y_0, \ldots, Y_s = y_s, \theta}(x_s) f_{X_{s+1} | X_s = x_s, \theta}(x_{s+1})}{\int f_{X_s | Y_0 = y_0, \ldots, Y_s = y_s, \theta}(u) f_{X_{s+1} | X_s = u, \theta}(x_{s+1}) du} \right] f_{X_{s+1} | Y_0 = y_0, \ldots, Y_t = y_t, \theta}(x_s). \end{aligned}$$

The above formula expresses the joint smoothing density of X_s, X_{s+1} in terms of the smoothing density of X_{s+1} as well filtering and transition densities.

For linear Gaussian state space models, explicit calculations can be done leading to the formula:

$$\begin{pmatrix} X_{s+1} \\ X_s \end{pmatrix} \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta \sim N\left(\begin{pmatrix} m_{s+1|t} \\ m_{s|t} \end{pmatrix}, \begin{pmatrix} Q_{s+1|t} & Q_{s+1|t} \Gamma'_{s+1} \\ \Gamma_{s+1} Q_{s+1|t} & Q_{s|t} \end{pmatrix} \right). \tag{79}$$

Here, as before, $m_{s|t}$ and $Q_{s|t}$ denote the mean and covariance of $X_s \mid Y_0 = y_0, \ldots, Y_t, \theta$ respectively. Also Γ_{s+1} equals

$$\Gamma_{s+1} = Q_{s|s} A'_{s+1} Q^{-1}_{s+1|s}. \tag{80}$$

Note that Γ_{s+1} appears in the Kalman smoother recursions. To prove (79), we only need to verify that

$$\mathrm{Cov}\left(X_{s+1}, X_s \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta\right) = Q_{s+1|t}\Gamma'_{s+1}. \tag{81}$$

This is true because (below data_t stands for $Y_0 = y_0, \ldots, Y_t = y_t$)

$$\begin{aligned}\mathrm{Cov}\left(X_{s+1}, X_s \mid \mathrm{data}_t, \theta\right) &= \mathbb{E}\left[\left(X_{s+1} - m_{s+1|t}\right)\left(X_s - m_{s|t}\right)' \mid \mathrm{data}_t, \theta\right] \\ &= \mathbb{E}\left[\left(X_{s+1} - m_{s+1|t}\right)\mathbb{E}\left\{\left(X_s - m_{s|t}\right)' \mid X_{s+1}, \mathrm{data}_t, \theta\right\} \mid \mathrm{data}_t, \theta\right].\end{aligned}$$

We have seen in Lecture Nine that

$$\mathbb{E}\left(X_s \mid X_{s+1}, \mathrm{data}_t\right) = m_{s|s} + \Gamma_{s+1}\left(X_{s+1} - m_{s+1|s}\right)$$

which gives

$$\begin{aligned}\mathbb{E}\left\{\left(X_s - m_{s|t}\right)' \mid X_{s+1}, \mathrm{data}_t, \theta\right\} &= \left(m_{s|s} + \Gamma_{s+1}\left(X_{s+1} - m_{s+1|s}\right) - m_{s|t}\right)' \\ &= X'_{s+1}\Gamma'_{s+1} + \text{non-random}\end{aligned}$$

where "non-random" refers to a quantity which is deterministic. Thus

$$\begin{aligned}\mathrm{Cov}\left(X_{s+1}, X_s \mid \mathrm{data}_t, \theta\right) &= \mathbb{E}\left[\left(X_{s+1} - m_{s+1|t}\right)\left\{X'_{s+1}\Gamma'_{s+1} + \text{non-random}\right\} \mid \mathrm{data}_t, \theta\right] \\ &= \mathbb{E}\left[\left(X_{s+1} - m_{s+1|t}\right)\left\{\left(X_{s+1} - m_{s+1|t}\right)'\Gamma'_{s+1} + \text{non-random}\right\} \mid \mathrm{data}_t, \theta\right] \\ &= \mathrm{Cov}\left(X_{s+1} \mid \mathrm{data}_t, \theta\right)\Gamma'_{s+1} = Q_{s+1|t}\Gamma'_{s+1}.\end{aligned}$$

This proves (81) which completes the proof of (79).

12.2 Fisher's Identity (from last time)

In the last class, we looked at the Fisher identity for the score function. The setting is that of a latent variable model that describes the joint density $f_{Y,X|\theta}(y, x)$ of two variables Y, X in terms of parameters θ. Y is the observed variable (y is the observed data) and X is the hidden or latent variable. Fisher's identity says that

$$\nabla_\theta \log f_{Y|\theta}(y)\Big|_{\theta=\theta^{(0)}} = \nabla_\theta E\left(\theta, \theta^{(0)}\right)\Big|_{\theta=\theta^{(0)}} \tag{82}$$

where

$$E(\theta, \theta^{(0)}) := \int \left[\log f_{Y,X|\theta}(y, x)\right] f_{X|Y=y,\theta=\theta^{(0)}}(x)dx \tag{83}$$

12.3 The Score Function for the Local Level Model

Let us illustrate the Fisher identity for calculating the score function in the local level model:

$$\begin{aligned}&X_0 \sim N(\mu_0, \Gamma_0) \\ &X_t = X_{t-1} + Z_t \qquad \text{with } U_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_Z^2) \\ &Y_t = X_t + \epsilon_t \qquad \text{with } \epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2)\end{aligned}$$

The parameter vector here is $\theta := (\sigma_Z, \sigma_\epsilon)$. Let us calculate $E(\theta, \theta^{(0)})$ to calculate the score vector at $\theta^{(0)} := (\sigma_Z^{(0)}, \sigma_\epsilon^{(0)})$. The log-likelihood of $Y_0, \ldots, Y_T, X_0, \ldots, X_T$ equals

$$\log f_{Y,X}(\theta) := -\frac{1}{2}\log\left(2\pi\Gamma_0\right) - \frac{1}{2\Gamma_0}(x_0 - \mu_0)^2 - \frac{T}{2}\log\left(2\pi\sigma_Z^2\right) - \frac{1}{2\sigma_Z^2}\sum_{t=1}^{T}(x_t - x_{t-1})^2$$
$$- \frac{T+1}{2}\log(2\pi\sigma_\epsilon^2) - \frac{1}{2\sigma_\epsilon^2}\sum_{t=0}^{T}(y_t - x_t)^2$$

Therefore

$$\begin{aligned} E(\theta, \theta^{(0)}) := &-\frac{1}{2}\log\left(2\pi\Gamma_0\right) - \frac{1}{2\Gamma_0}\mathbb{E}\left\{(X_0 - \mu_0)^2 \mid \text{data}, \theta^{(0)}\right\} \\ &- \frac{T}{2}\log\left(2\pi\sigma_Z^2\right) - \frac{1}{2\sigma_Z^2}\mathbb{E}\left\{\sum_{t=1}^{T}(X_t - X_{t-1})^2 \mid \text{data}, \theta^{(0)}\right\} \\ &- \frac{T+1}{2}\log(2\pi\sigma_\epsilon^2) - \frac{1}{2\sigma_\epsilon^2}\mathbb{E}\left\{\sum_{t=0}^{T}(y_t - X_t)^2 \mid \text{data}, \theta^{(0)}\right\} \end{aligned} \tag{84}$$

where "data" represents $Y_0 = y_0, \ldots, Y_T = y_T$ (this is basically data_T in the notation of Section 12.1). As a result

$$\nabla_\theta E(\theta, \theta^{(0)}) = \begin{pmatrix} -\frac{T}{\sigma_Z} + \frac{1}{\sigma_Z^3}\mathbb{E}\left\{\sum_{t=1}^{T}(X_t - X_{t-1})^2 \mid \text{data}, \theta^{(0)}\right\} \\ -\frac{T+1}{\sigma_\epsilon} + \frac{1}{\sigma_\epsilon^3}\mathbb{E}\left\{\sum_{t=0}^{T}(y_t - X_t)^2 \mid \text{data}, \theta^{(0)}\right\} \end{pmatrix} \tag{85}$$

The Fisher identity therefore gives

$$\nabla_\theta \log f_{Y|\theta}(y)\Big|_{\theta=\theta^{(0)}} = \begin{pmatrix} -\frac{T}{\sigma_Z^{(0)}} + \frac{1}{\left(\sigma_Z^{(0)}\right)^3}\mathbb{E}\left\{\sum_{t=1}^{T}(X_t - X_{t-1})^2 \mid \text{data}, \theta^{(0)}\right\} \\ -\frac{T+1}{\sigma_\epsilon^{(0)}} + \frac{1}{\left(\sigma_\epsilon^{(0)}\right)^3}\mathbb{E}\left\{\sum_{t=0}^{T}(y_t - X_t)^2 \mid \text{data}, \theta^{(0)}\right\} \end{pmatrix} \tag{86}$$

The expectations appearing above can be calculated using the output of the Kalman smoother as shown below. Let $m_{s|T}(\theta^{(0)})$ and $Q_{s|T}(\theta^{(0)})$ denote the output of the Kalman smoother when the parameters are set to $\theta^{(0)}$. Then

$$\begin{aligned} &\mathbb{E}\left\{\sum_{t=1}^{T}(X_t - X_{t-1})^2 \mid \text{data}, \theta^{(0)}\right\} \\ &= \sum_{t=1}^{T}\mathbb{E}\left\{(X_t - X_{t-1})^2 \mid \text{data}, \theta^{(0)}\right\} \\ &= \sum_{t=1}^{T}\text{var}\left\{X_t - X_{t-1} \mid \text{data}, \theta^{(0)}\right\} + \sum_{t=1}^{T}\left(m_{t|T}(\theta^{(0)}) - m_{t-1|T}(\theta^{(0)})\right)^2 \\ &= \sum_{t=1}^{T}\left[Q_{t|T}(\theta^{(0)}) + Q_{t-1|T}(\theta^{(0)}) - 2\text{Cov}(X_t, X_{t-1} \mid \text{data}, \theta^{(0)})\right] + \sum_{t=1}^{T}\left(m_{t|T}(\theta^{(0)}) - m_{t-1|T}(\theta^{(0)})\right)^2 \\ &= \sum_{t=1}^{T}\left[Q_{t|T}(\theta^{(0)}) + Q_{t-1|T}(\theta^{(0)}) - 2Q_{t|T}(\theta^{(0)})\Gamma_t(\theta^{(0)})\right] + \sum_{t=1}^{T}\left(m_{t|T}(\theta^{(0)}) - m_{t-1|T}(\theta^{(0)})\right)^2 \end{aligned}$$

where, in the last equation, we used the formula (81) for $s = t - 1$. The quantity $\Gamma_t(\theta^{(0)})$ equals (see (80)):

$$\Gamma_t(\theta^{(0)}) = \frac{Q_{t-1|t-1}(\theta^{(0)})}{Q_{t|t-1}(\theta^{(0)})}$$

which can be calculated by the Kalman filter output.

Also

$$\begin{aligned}\mathbb{E}\left\{\sum_{t=0}^{T}(y_t - X_t)^2 \mid \text{data}, \theta^{(0)}\right\} &= \sum_{t=0}^{T}\mathbb{E}\left\{(y_t - X_t)^2 \mid \text{data}, \theta^{(0)}\right\} \\ &= \sum_{t=0}^{T}\left\{\text{var}\left(X_t \mid \text{data}, \theta^{(0)}\right) + \left(y_t - m_{t|T}(\theta^{(0)})\right)^2\right\} \\ &= \sum_{t=0}^{T}\left\{Q_{t|T}(\theta^{(0)}) + \left(y_t - m_{t|T}(\theta^{(0)})\right)^2\right\}.\end{aligned}$$

Observe that (86) is a closed form expression for the score function (in terms of the Kalman smoother output). Using the expression (86) for the score function, we can use standard optimization methods (such as gradient ascent or BFGS) to obtain the maximum likelihood estimator for $\theta = (\sigma_Z, \sigma_\epsilon)$.

12.4 The EM Algorithm

The EM algorithm is another method for maximizing the log-likelihood $\log f_{Y|\theta}(y)$ over θ in latent variable models. It is also an iterative algorithm. The EM update

$$\theta^{(n)} \to \theta^{(n+1)}$$

consists of the following two steps:

1. **E-Step**: Calculate $E(\theta, \theta^{(n)})$ (this is (83) with $\theta^{(0)}$ replaced by $\theta^{(n)}$).
2. **M-Step**: Take $\theta^{(n+1)}$ to be the maximizer of $E(\theta, \theta^{(n)})$ over θ.

Some intuition behind this algorithm will be provided in the next class.

12.5 EM for the local level model

For the local level model, the expression for $E(\theta, \theta^{(0)})$ as well as $\nabla_\theta E(\theta, \theta^{(0)})\Big|_{\theta=\theta^{(0)}}$ are calculated in Section 12.3 (see (84) and (85)). Using these, we can immediately write down the EM iterate in closed form. Indeed, $\theta^{(n+1)}$ is obtained by maximizing $E(\theta, \theta^{(n)})$ over θ. Setting the gradient of $E(\theta, \theta^{(n)})$ (calculated in (85)) to zero, we can immediately deduce that

$$\sigma_Z^{(n+1)} = \sqrt{\frac{1}{T}\mathbb{E}\left\{\sum_{t=1}^{T}(X_t - X_{t-1})^2 \mid \text{data}, \theta^{(n)}\right\}}$$

and

$$\sigma_\epsilon^{(n+1)} = \sqrt{\frac{1}{T+1}\mathbb{E}\left\{\sum_{t=0}^{T}(y_t - X_t)^2 \mid \text{data}, \theta^{(n)}\right\}}.$$

This is a very easy update (there are no line searches for step size selection) and thus the EM is very popular for state space models.

12.6 Calculation of $E(\theta, \theta^{(0)})$ for general state space models

For a general state space model,

$$\begin{aligned}\log f_{Y,X|\theta}(y,x) &= \log f_{X_0,\ldots,X_T,Y_0,\ldots,Y_T|\theta}(x_0,\ldots,x_T,y_0,\ldots,y_T)\\ &= \log f_{X_0|\theta}(x_0) + \sum_{t=1}^{T} \log f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) + \sum_{t=0}^{T} \log f_{Y_t|X_t=x_t,\theta}(y_t).\end{aligned}$$

Observe that the right hand side above involves three kinds of quantities: the observed data $y_0,\ldots,y_T$, the parameters θ and the quantities $x_0,\ldots,x_T$. From here, to obtain $E(\theta,\theta^{(0)})$, we leave $y_0,\ldots,y_T,\theta$ unchanged in the right hand side and take the expectation over $x_0,\ldots,x_T$ conditional on $y_0,\ldots,y_T$. This conditional expectation depends on parameters and we shall fix the parameters at $\theta^{(0)}$ (as opposed to the θ that is already appearing on the right hand side). We can thus write

$$E(\theta,\theta^{(0)}) = I_1(\theta,\theta^{(0)}) + I_2(\theta,\theta^{(0)}) + I_3(\theta,\theta^{(0)}) \tag{87}$$

where

$$\begin{aligned}I_1(\theta,\theta^{(0)}) &= \int \left[\log f_{X_0|\theta}(x_0)\right] f_{X_0|Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}}(x_0)dx_0\\ &= \mathbb{E}\left[\log f_{X_0|\theta}(X_0) \mid Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}\right],\end{aligned}$$

and

$$\begin{aligned}I_2(\theta,\theta^{(0)}) &= \sum_{t=1}^{T} \int\int \left[\log f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t)\right] f_{X_t,X_{t-1}|Y_0=y_0,\ldots Y_T=y_T,\theta^{(0)}}(x_t,x_{t-1})dx_t dx_{t-1}\\ &= \sum_{t=1}^{T} \mathbb{E}\left[\log f_{X_t|X_{t-1}=X_{t-1},\theta}(X_t) \mid Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}\right],\end{aligned}$$

and

$$\begin{aligned}I_3(\theta,\theta^{(0)}) &= \sum_{t=0}^{T} \int \left[\log f_{Y_t|X_t=x_t,\theta}(y_t)\right] f_{X_t|Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}}(x_t)dx_t\\ &= \sum_{t=0}^{T} \mathbb{E}\left[\log f_{Y_t|X_t=X_t,\theta}(y_t) \mid Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}\right].\end{aligned}$$

Note that $I_3(\theta,\theta^{(0)})$ involves expectation with respect to the conditional distribution

$$X_t \mid Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}$$

and $I_1(\theta,\theta^{(0)})$ involves expectation with respect to the above conditional distribution for $t=0$. These conditional distributions are obtained from the smoothing algorithm. Further $I_2(\theta,\theta^{(0)})$ involves expectation with respect to

$$X_t, X_{t-1} \mid Y_0=y_0,\ldots,Y_T=y_T,\theta^{(0)}$$

which can be obtained from the pairwise smoothing algorithm of Section 12.1.

For linear Gaussian state space models, $I_1(\theta,\theta^{(0)}), I_2(\theta,\theta^{(0)}), I_3(\theta,\theta^{(0)})$ can be computed in closed form in terms of the output of the Kalman smoothing algorithm. The details of this calculation are given in Theorem 12.4 of the Särkkä book. Often maximization of $E(\theta,\theta^{(0)})$ can also be done in closed form for linear Gaussian state space models (see Theorem 12.5 of the Särkkä book).

12.7 Recommended Reading for Today

1. The pairwise smoothing distributions for the linear Gaussian state space model are described in the proof of Theorem 8.2 of the Särkkä book.
2. The EM algorithm is described in Section 12.2.3 of the Särkkä book, Section 2.4.2 of the Triantafyllopoulos book, and Section 14.1.3 of the Chopin-Papaspiliopoulos book.
3. The EM algorithm for the local level model is given in Example 14.1 of the Chopin-Papaspiliopoulos book.
4. More details on the EM algorithm for linear Gaussian state space models are given in Section 12.3.2 of the Särkkä book.

13 Lecture Thirteen

We shall cover the following two topics today:

1. The EM algorithm in the context of the more general MM class of algorithms
2. The Forward Filtering Backward **Sampling** algorithm for sampling from the full posterior of all the states $X_0, \ldots, X_T$ given the data $Y_0 = y_0, \ldots, Y_T = y_T$ and θ.

13.1 The MM Algorithm

The EM algorithm is much easier to understand in the context of a more general class of algorithms called MM. Consider the general problem of maximizing a function $F(\theta)$ over θ. In this setting, MM stands for Minorize-Maximize (if we are studying the problem of *minimizing* $F(\theta)$ as opposed to maximizing, MM would stand for Majorize-Minimize). The MM algorithm for maximizing $F(\theta)$ over θ is iterative and the update from $\theta^{(n)}$ to $\theta^{(n+1)}$ has the following two steps:

1. Construct a function $G(\theta, \theta^{(n)})$ which minorizes $F(\theta)$ for every θ and agrees with $F(\theta)$ at $\theta = \theta^{(n)}$. In other words, $G(\theta, \theta^{(n)})$ must satisfy:

$$G(\theta, \theta^{(n)}) \leq F(\theta) \text{ for every } \theta, \qquad \text{and} \qquad G(\theta^{(n)}, \theta^{(n)}) = F(\theta^{(n)}).$$

2. Take $\theta^{(n+1)}$ to be the maximizer of $G(\theta, \theta^{(n)})$ over θ.

The most important fact about the MM algorithm is that the objective function increases (or stays the same) when going from $\theta^{(n)}$ to $\theta^{(n+1)}$:

$$F(\theta^{(n+1)}) \geq F(\theta^{(n)}). \tag{88}$$

This can be easily proved via:

$$F(\theta^{(n+1)}) \geq G(\theta^{(n+1)}, \theta^{(n)}) \geq G(\theta^{(n)}, \theta^{(n)}) = F(\theta^{(n)}).$$

Note that the first inequality above follows from the fact that $G(\cdot, \theta^{(n)})$ minorizes $F(\theta)$, the second inequality follows because $\theta^{(n+1)}$ maximizes $G(\cdot, \theta^{(n)})$ and the third inequality follows because $G(\theta, \theta^{(n)})$ matches $F(\theta)$ at $\theta = \theta^{(n)}$.

The property (88) is very desirable for a maximization procedure and it is remarkable that the MM algorithm satisfies it without any explicit line search scheme for choosing step sizes.

Before seeing why the EM algorithm is a special case of the MM algorithm, let us first look at two simple examples.

Example 13.1. *Consider the problem of maximizing the function $F(\theta) = \cos\theta$. The MM algorithm can be used for this in the following way. In order to go from $\theta^{(n)}$ to $\theta^{(n+1)}$, the first step is to construct $G(\theta, \theta^{(n)})$ for which we argue as follows. For every θ, we can write*

$$F(\theta) = F(\theta^{(n)}) + F'(\theta^{(n)})(\theta - \theta^{(n)}) + \frac{1}{2}F''(z)(\theta - \theta^{(n)})^2$$

for some z that lies between θ and $\theta^{(n)}$. Thus

$$\begin{aligned} F(\theta) &= \cos\theta^{(n)} - \left(\sin\theta^{(n)}\right)(\theta - \theta^{(n)}) - \frac{1}{2}(\cos z)(\theta - \theta^{(n)})^2 \\ &\geq \cos\theta^{(n)} - \left(\sin\theta^{(n)}\right)(\theta - \theta^{(n)}) - \frac{1}{2}(\theta - \theta^{(n)})^2 \end{aligned}$$

and thus we take

$$G(\theta, \theta^{(n)}) = \cos\theta^{(n)} - \left(\sin\theta^{(n)}\right)(\theta - \theta^{(n)}) - \frac{1}{2}(\theta - \theta^{(n)})^2.$$

It is easy to see that $G(\theta^{(n)}, \theta^{(n)}) = F(\theta^{(n)})$. Thus G satisfies both the requirements of the first step of the MM algorithm. Further as $G(\theta, \theta^{(n)})$ is quadratic in θ, it is easy to maximize it over θ to obtain

$$\theta^{(n+1)} = \theta^{(n)} - \sin\theta^{(n)}.$$

It is an exercise to show that this iterative scheme converges to the true maximizer 0 when initialized anywhere in the open interval $(-\pi, \pi)$.

Example 13.2. *Given m real numbers $y_1, \ldots, y_m$, consider the problem of maximizing*

$$F(\theta) := -\sum_{i=1}^{m} |y_i - \theta|$$

over θ. Any solution of this problem can be termed a median of F. The usual algorithms for computing the median involve sorting the data. MM provides another method for median computation that does not require sorting the data. The key to the MM iterate $\theta^{(n)} \to \theta^{(n+1)}$ is the construction of $G(\theta, \theta^{(n)})$. For this, consider the following inequality:

$$|y_i - \theta| = \frac{|y_i - \theta|}{\sqrt{|y_i - \theta^{(n)}|}}\sqrt{|y_i - \theta^{(n)}|} \leq \frac{1}{2}\frac{(y_i - \theta)^2}{|y_i - \theta^{(n)}|} + \frac{1}{2}|y_i - \theta^{(n)}| \tag{89}$$

where we used the elementary fact: $ab \leq \frac{a^2}{2} + \frac{b^2}{2}$. We thus take

$$G(\theta, \theta^{(n)}) := -\frac{1}{2}\sum_{i=1}^{n}\frac{(y_i - \theta)^2}{|y_i - \theta^{(n)}|} - \frac{1}{2}\sum_{i=1}^{n}|y_i - \theta^{(n)}|.$$

It is easy to check that $G(\theta, \theta^{(n)})$ minorizes $F(\theta)$ (because of (89)) and that $G(\theta^{(n)}, \theta^{(n)}) = F(\theta^{(n)})$. Because $G(\theta, \theta^{(n)})$ is a quadratic function in θ, it is easy to write down its maximizer (over θ) in closed form:

$$\theta^{(n+1)} = \frac{\sum_{i=1}^{n} w_i^{(n)} y_i}{\sum_{i=1}^{n} w_i^{(n)}} \qquad \text{where } w_i^{(n)} := \frac{1}{|y_i - \theta^{(n)}|}.$$

This algorithm clearly does not involve sorting the data. One problem with this algorithm is that it does not work when $\theta^{(n)}$ equals y_i for some i (note then that $w_i^{(n)}$ equals 0). It is difficult (probably impossible) to construct a quadratic $G(\theta, \theta^{(n)})$ satisfying our requirements when $\theta^{(n)}$ equals y_i for some i. A practical fix is to change the weights $w^{(n)}$ slightly by adding a small ϵ to the denominator as: $w_i^{(n)} = \frac{1}{|y_i - \theta^{(n)}| + \epsilon}$.

It should be clear from the above examples that the most important step for the use of the MM algorithm is the construction of the $G(\theta, \theta^{(n)})$ function. There are some general ideas for this (see the book *MM Optimization Algorithms* by Kenneth Lange, or chapter 12 in the book *Numerical Analysis for Statisticians* by Kenneth Lange, or these slides: `https://www.stat.berkeley.edu/~aldous/Colloq/lange-talk.pdf`).

13.2 The EM Algorithm as a special case of MM

The EM algorithm is a special case of MM corresponding to a special choice of $G(\theta, \theta^{(n)})$ in the latent variable model setting. We shall describe this below. Let us first recall the notion of Kullback-Leibler divergence (also known as Relative Entropy).

13.2.1 The Kullback-Leibler Divergence

The Kullback-Leibler divergence $D(p\|q)$ between two densities p and q is defined as

$$D(p\|q) := \int p(x) \log \frac{p(x)}{q(x)} dx.$$

The most important property of $D(p\|q)$ is that it is always nonnegative. This can be proved as a consequence of the elementary inequality:

$$u \log u \geq u - 1 \qquad \text{for all } u \geq 0.$$

Because of this inequality:

$$\begin{aligned} D(p\|q) &= \int p(x) \log \frac{p(x)}{q(x)} dx \\ &= \int q(x) \left(\frac{p(x)}{q(x)} \log \frac{p(x)}{q(x)} \right) dx \\ &\geq \int q(x) \left(\frac{p(x)}{q(x)} - 1 \right) dx = \int q(x) dx - \int p(x) dx = 1 - 1 = 0 \end{aligned}$$

Another way of proving $D(p\|q) \geq 0$ is via the use of the Jensen inequality.

It should also be noted that

$$D(p\|q) = 0 \text{ if and only if } p = q \tag{90}$$

This can be argued using the fact that $x \log x = x - 1$ if and only if $x = 1$.

It is also very important to note that $D(q\|p)$ and $D(p\|q)$ are in general **not equal**.

13.2.2 EM and MM

We are now ready to explain the connection between the EM and MM algorithms. Consider the latent variable setting where the goal is to maximize the log-likelihood:

$$F(\theta) = \log f_{Y|\theta}(y)$$

There is a latent variable X and the model is specified via the full density $f_{Y,X|\theta}(y,x)$. It is important to note that the conditional density of X given $Y = y$ depends on the value of θ. We shall denote this by $f_{X|Y=y,\theta}(x)$.

We shall show below that the EM update for $\theta^{(n)} \to \theta^{(n+1)}$ is exactly equal to the MM update corresponding to

$$G(\theta, \theta^{(n)}) := F(\theta) - D\left(f_{X|Y=y,\theta^{(n)}} \| f_{X|Y=y,\theta}\right) \tag{91}$$

Because the Kullback-Leibler divergence is always nonnegative, it is clear that $G(\theta, \theta^{(n)}) \leq F(\theta)$ for every θ. Further, because of (90), $G(\theta^{(n)}, \theta^{(n)}) = F(\theta^{(n)})$. Thus G satisfies the conditions required for the first step of the MM algorithm. Note that we can write G alternately as

$$\begin{aligned}
G(\theta, \theta^{(n)}) &= F(\theta) - D\left(f_{X|Y=y,\theta^{(n)}} \| f_{X|Y=y,\theta}\right) \\
&= \log f_{Y|\theta}(y) - \int f_{X|Y=y,\theta^{(n)}}(x) \log \frac{f_{X|Y=y,\theta}(x)}{f_{X|Y=y,\theta^{(n)}}(x)} dx \\
&= \int f_{X|Y=y,\theta^{(n)}}(x) \log f_{Y|\theta}(y) dx - \int f_{X|Y=y,\theta^{(n)}}(x) \log \frac{f_{X|Y=y,\theta}(x)}{f_{X|Y=y,\theta^{(n)}}(x)} dx \\
&= \int f_{X|Y=y,\theta^{(n)}}(x) \log \frac{f_{Y|\theta}(y) f_{X|Y=y,\theta}(x)}{f_{X|Y=y,\theta^{(n)}}(x)} dx \\
&= \int f_{X|Y=y,\theta^{(n)}}(x) \log \frac{f_{Y,X|\theta}(y,x)}{f_{X|Y=y,\theta^{(n)}}(x)} dx \\
&= \int f_{X|Y=y,\theta^{(n)}}(x) \log f_{Y,X|\theta}(y,x) dx - \int f_{X|Y=y,\theta^{(n)}}(x) \log f_{X|Y=y,\theta^{(n)}(x)} dx.
\end{aligned}$$

Recall that the first term on the right hand side above is precisely the function $E(\theta, \theta^{(n)})$ that appears in the Expectation Step of the EM algorithm. We have thus proved that

$$G(\theta, \theta^{(n)}) = E(\theta, \theta^{(n)}) - \int f_{X|Y=y,\theta^{(n)}}(x) \log f_{X|Y=y,\theta^{(n)}(x)} dx.$$

The second term above does not depend on θ. Therefore maximizing $G(\theta, \theta^{(n)})$ over θ is equivalent to maximizing $E(\theta, \theta^{(n)})$ over θ. This shows that the second steps of the MM algorithm (with G defined in (91)) and the EM algorithm are identical, which completes the proof of the claim that MM (with G in (91)) is exactly the EM algorithm. The EM algorithm is therefore a special case of MM.

13.3 Full Smoothing Distribution

In our discussion of smoothing, we have so far discussion the computation of the marginal distributions:

$$X_t \mid Y_0 = y_0, \ldots, Y_T = y_T, \theta$$

for each $t = 0, \ldots, T$, as well as the pairwise distributions:

$$X_t, X_{t+1} \mid Y_0 = y_0, \ldots, Y_T = y_T, \theta.$$

It turns out ideas used for the above calculations can also be used to obtain the full conditional joint density:

$$X_0, \ldots, X_T \mid Y_0 = y_0, \ldots, Y_T = y_T, \theta \tag{92}$$

for all the states given the observations (and θ). To see this, first write (below "data" refers to $Y_0 = y_0, \ldots, Y_T = y_T$)

$$f_{X_0,\ldots,X_T|\text{data},\theta}(x_0, \ldots, x_T) = f_{X_T|\text{data},\theta}(x_T) \prod_{t=T-1}^{0} f_{X_t|X_{t+1}=x_{t+1},\ldots,X_T=x_T,\text{data},\theta}(x_t).$$

The Markov property of $\{X_t\}$ and the conditional independence of Y_t given $X_0, \ldots, X_T$ imply

$$f_{X_t|X_{t+1}=x_{t+1},\ldots,X_T=x_T,\text{data},\theta}(x_t) = f_{X_t|X_{t+1}=x_{t+1},Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t). \tag{93}$$

This means that $X_s = x_s$ for $s > t+1$ and $Y_s = y_s$ for $s > t$ can be dropped from the conditioning. As a result

$$f_{X_0,\ldots,X_T|\text{data},\theta}(x_0, \ldots, x_T) = f_{X_T|\text{data},\theta}(x_T) \prod_{t=T-1}^{0} f_{X_t|X_{t+1}=x_{t+1},Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t).$$

By the Bayes rule (note that $f_{X_{t+1}|X_t=x_t,Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{t+1}) = f_{X_{t+1}|X_t=x_t,\theta}(x_{t+1})$), we obtain

$$f_{X_0,\ldots,X_T|\text{data},\theta}(x_0, \ldots, x_T) = f_{X_T|\text{data},\theta}(x_T) \prod_{t=T-1}^{0} \frac{f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t) f_{X_{t+1}|X_t=x_t,\theta}(x_{t+1})}{\int f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(u) f_{X_{t+1}|X_t=u,\theta}(x_{t+1}) du}.$$

This is a formula for the full smoothing joint density in terms of the filtering densities.

For linear Gaussian state space models, the conditional density (93) can be computed in closed form (as we saw in Lecture Nine) as:

$$\begin{aligned} &X_t \mid X_{t+1} = x_{t+1}, Y_0 = y_0, \ldots, Y_t = y_t, \theta \\ &\sim N\left(m_{t|t} + \Gamma_{t+1}\left(x_{t+1} - m_{t+1|t}\right), Q_{t|t} - \Gamma_{t+1} Q_{t+1|t} \Gamma'_{t+1}\right) \end{aligned} \tag{94}$$

where $\Gamma_{t+1} = Q_{t|t} A'_{t+1} Q^{-1}_{t+1|t}$. This gives a closed form expression for the full smoothing joint density.

13.4 Forward Filtering Backward SAMPLING

Suppose we want to generate independent samples

$$X_0^{(i)}, \ldots, X_T^{(i)}$$

for $i = 1, \ldots, N$ from the conditional distribution (92). This can be done using the formulae from the previous section. For linear Gaussian state space models, we use (94) to obtain the following sampling algorithm. Repeat the following steps for each $i = 1, \ldots, N$:

1. Generate $X_T^{(i)}$ from the filtering distribution at time T i.e., we generate $X_T^{(i)}$ from the $N(m_{T|T}, Q_{T|T})$ distribution.

2. Sequentially for $t = T-1, \ldots, 0$, generate $X_t^{(i)}$ from the distribution:

$$N\left(m_{t|t} + \Gamma_{t+1}\left(X_{t+1}^{(i)} - m_{t+1|t}\right), Q_{t|t} - \Gamma_{t+1} Q_{t+1|t} \Gamma_{t+1}'\right).$$

Note that this algorithm requires the quantities $m_{T|T}, Q_{T|T}, m_{t|t}, Q_{t|t}, m_{t+1|t}, Q_{t+1|t}, \Gamma_{t+1}$ which are all obtained from the Kalman Filter. Thus, one would need to implement the Kalman Filter before running the sampling algorithm. Note however that this sampling algorithm does not use any output of the usual Kalman Smoother algorithm.

For a general state space model, sampling can be done by discretization (we shall see other approaches later). The first step is to setup a dense grid $x^{(g)}, g \in G$ covering the range of X_t and perform filtering. This will lead to discrete distributions:

$$p_{t|t}(x^{(g)}), g \in G \tag{95}$$

which approximate the densities $f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}$ for each $t = 0, 1, \ldots, T$. Then the sampling algorithm to generate $X_0^{(i)}, \ldots, X_T^{(i)}$ for $i = 1, \ldots, N$ from the conditional distribution (92) is as follows. Repeat the following steps for each $i = 1, \ldots, N$:

1. Generate $X_T^{(i)}$ from $p_{T|T}(x^{(g)}), g \in G$ (this is the discrete filtering approximation at time T).
2. Sequentially for $t = T-1, \ldots, 0$, repeat the following steps:
   a) Calculate $w_g := p_{t|t}(x^{(g)}) f_{X_{t+1}|X_t=x^{(g)},\theta}(X_{t+1}^{(i)})$ for $g \in G$.
   b) Normalize $w_g, g \in G$ calculated above so they sum to one.
   c) Generate $X_t^{(i)}$ from the discrete distribution which gives probability w_g to the grid point $x^{(g)}$ for $g \in G$.

Note again that this algorithm only uses the filtering approximations (95). It is not necessary to calculate smoothing approximations to run this sampling algorithm.

These sampling algorithms for sampling observations from the full conditional distribution of the states given the data (and the parameters θ) are known as FFBS (Forward Filtering Backward SAMPLING). They should be contrasted with the previous FFBS (Forward Filtering Backward SMOOTHING) algorithms which computed the smoothing densities (exactly or approximately).

13.5 Recommended Reading for Today

1. References for the MM algorithm are the book *MM Optimization Algorithms* by Kenneth Lange, or chapter 12 in the book *Numerical Analysis for Statisticians* by Kenneth Lange, or these slides: `https://www.stat.berkeley.edu/~aldous/Colloq/lange-talk.pdf`.
2. For the FFBSampling algorithm, see Section 5.7.2 of the Triantafyllopoulos book or Section 4.4.1 of the Petris-Petrone-Campagnoli book,

14 Lecture Fourteen

We shall discuss full Bayesian estimation of state space models today. Full Bayesian estimation means that we put a prior on the unknown parameters θ (as opposed to obtaining point estimates for θ and ignoring the uncertainty in their estimation). Let us start by considering the local level model.

14.1 Local Level Model

We have as usual

$$X_0 \sim N(0, C) \quad X_t = X_{t-1} + Z_t \quad Y_t = X_t + \epsilon_t$$

with $Z_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_Z^2)$ and $\epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2)$. σ_Z and σ_ϵ are unknown parameters and C is a large constant. Previously we obtained maximum likelihood estimates for σ_Z and σ_ϵ and then went on to obtain smoothing estimates of $X_0, \ldots, X_T$ ignoring the uncertainty in estimation of σ_Z and σ_ϵ. Now we shall place priors on σ_Z and σ_ϵ. Natural priors on scale parameters reflecting ignorance are:

$$\log \sigma_Z, \log \sigma_\epsilon \overset{\text{i.i.d}}{\sim} \text{Unif}(-C, C).$$

The full joint density of $\theta, X_0, \ldots, X_T, Y_0, \ldots, Y_T$ (here $\theta = (\sigma_Z, \sigma_\epsilon)$) is proportional to

$$\frac{I\{e^{-C} < \sigma_Z, \sigma_\epsilon < e^C\}}{\sigma_Z \sigma_\epsilon} \phi(x_0; 0, C) \prod_{t=1}^{T} \frac{1}{\sigma_Z} \exp\left(-\frac{(x_t - x_{t-1})^2}{2\sigma_Z^2}\right) \prod_{t=0}^{T} \frac{1}{\sigma_\epsilon} \exp\left(-\frac{(y_t - x_t)^2}{2\sigma_\epsilon^2}\right)$$

$$= I\{e^{-C} < \sigma_Z, \sigma_\epsilon < e^C\} \phi(x_0; 0, C) \sigma_Z^{-T-1} \exp\left(-\frac{\sum_{t=1}^{T}(x_t - x_{t-1})^2}{2\sigma_Z^2}\right) \sigma_\epsilon^{-T-2} \exp\left(-\frac{\sum_{t=0}^{T}(y_t - x_t)^2}{2\sigma_\epsilon^2}\right)$$

As a result

$$f_{\theta, X_0, \ldots, X_T | Y_0 = y_0, \ldots, Y_T = y_T}(\theta, x_0, \ldots, x_T)$$

$$\propto I\{e^{-C} < \sigma_Z, \sigma_\epsilon < e^C\} \phi(x_0; 0, C) \sigma_Z^{-T-1} \exp\left(-\frac{\sum_{t=1}^{T}(x_t - x_{t-1})^2}{2\sigma_Z^2}\right) \sigma_\epsilon^{-T-2} \exp\left(-\frac{\sum_{t=0}^{T}(y_t - x_t)^2}{2\sigma_\epsilon^2}\right).$$

Often the main interest is in the conditional distribution of $X_0, \ldots, X_T$ given $Y_0 = y_0, \ldots, Y_T = y_T$ and we can obtain this by integrating over the σ_Z and σ_ϵ. This integration can be done in closed form if we assume that C is large (so that the indicator above can be dropped). We then get

$$f_{X_0, \ldots, X_T | Y_0 = y_0, \ldots, Y_T = y_T}(x_0, \ldots, x_T)$$

$$\propto \phi(x_0; 0, C) \left[\int_0^\infty \sigma_Z^{-T-1} \exp\left(-\frac{\sum_{t=1}^{T}(x_t - x_{t-1})^2}{2\sigma_Z^2}\right) d\sigma_Z\right] \left[\int_0^\infty \sigma_\epsilon^{-T-2} \exp\left(-\frac{\sum_{t=0}^{T}(y_t - x_t)^2}{2\sigma_\epsilon^2}\right) d\sigma_\epsilon\right]$$

$$\propto \phi(x_0; 0, C) \left[\sum_{t=1}^{T}(x_t - x_{t-1})^2\right]^{-T/2} \left[\sum_{t=0}^{T}(y_t - x_t)^2\right]^{-(T+1)/2}.$$

where we used

$$\int_0^\infty \sigma^{-m-1} \exp\left(-\frac{G}{\sigma^2}\right) d\sigma = \frac{\Gamma\left(\frac{m}{2}\right)}{G^{m/2}}.$$

and ignored the $\Gamma(m/2)$ terms in proportionality.

The posterior density:

$$\begin{aligned} & f_{X_0,\ldots,X_T|Y_0=y_0,\ldots,Y_T=y_T}(x_0,\ldots,x_T) \\ & \propto \phi(x_0;0,C)\left[\sum_{t=1}^{T}(x_t-x_{t-1})^2\right]^{-T/2}\left[\sum_{t=0}^{T}(y_t-x_t)^2\right]^{-(T+1)/2} \end{aligned} \tag{96}$$

can, in principle, be used for all inference on the hidden variables $X_0, \ldots, X_T$ given the observed data. The problem is that it does not correspond to any state space model so it is not clear how to derive from it the marginal posterior densities $X_t \mid Y_0 = y_0, \ldots, Y_T = y_T$ in an efficient way. In particular, the Kalman smoother cannot be implemented as this posterior does not correspond to a state space model. As a result, instead of integrating out θ from the joint posterior of $\theta, X_0, \ldots, X_T$, the common approach is to obtain samples $\theta^{(i)}, X_0^{(i)}, \ldots, X_T^{(i)}$ for $i = 1, \ldots, N$ from the joint posterior $\theta, X_0, \ldots, X_T$. Then $X_0^{(i)}, \ldots, X_T^{(i)}$ for $i = 1, \ldots, N$ can be used for posterior inference on the hidden states given the observed data. Also the samples $\theta^{(1)}, \ldots, \theta^{(N)}$ can be used for posterior inference on the parameters θ given the observed data.

For generating the posterior samples $\theta^{(i)}, X_0^{(i)}, \ldots, X_T^{(i)}$ for $i = 1, \ldots, N$, it is convenient to use the Gibbs sampler algorithm.

14.2 Gibbs Sampler

Suppose we want to approximate a joint distribution $f_{A,B}$ over two random variables A and B. The Gibbs sampler algorithm is applicable in situations where the conditional densities $f_{A|B=b}$ and $f_{B|A=a}$ are easy to simulate from for each value of a and b. The algorithm is as follows:

1. Start with $a = a^{(0)}$
2. For each $i = 1, 2, \ldots, N$,
   a) Generate $b^{(i)} \sim f_{B|A=a^{(i-1)}}$.
   b) Generate $a^{(i)} \sim f_{A|B=b^{(i)}}$

When N is large, this method generates samples $(a^{(i)}, b^{(i)})$ for $i = 1, \ldots, N$ having the property that

$$\frac{g(a^{(1)}, b^{(1)}) + \cdots + g(a^{(N)}, b^{(N)})}{N} \approx \int g(a,b) f_{A,B}(a,b) da db$$

for many functions g.

14.3 Gibbs Sampler for the Local Level Model

The Gibbs sampler for generating samples $\theta^{(i)}, X_0^{(i)}, \ldots, X_T^{(i)}$ for $i = 1, \ldots, N$ from the full posterior distribution

$$\theta, X_0, \ldots, X_T \mid Y_0 = y_0, \ldots, Y_T = y_T$$

works as follows:

1. Start with $\theta^{(0)} = (\sigma_Z^{(0)}, \sigma_\epsilon^{(0)})$ for some initial values $\sigma_Z^{(0)}$ and $\sigma_\epsilon^{(0)}$.

2. For each $i = 1, \ldots, N$,

   a) Generate $X_0^{(i)}, \ldots, X_T^{(i)}$ from the conditional joint distribution of $X_0, \ldots, X_T$ given $\theta = \theta^{(i)}$ and $Y_0 = y_0, \ldots, Y_T = y_T$. Because we are conditioning on $\theta = \theta^{(i)}$ here, these samples are obtained by the FFBSampling algorithm discussed in the last class.

   b) Generate $\theta^{(i)}$ from the conditional distribution of θ given $X_0 = X_0^{(i)}, X_1 = X_1^{(i)}, \ldots, X_T = X_T^{(i)}$ and $Y_0 = y_0, \ldots, Y_T = y_T$. The details for doing this are given below.

For the second step above, we need to be able to simuate from the conditional distribution:

$$\theta \mid X_0 = x_0, \ldots, X_T = x_T, Y_0 = y_0, \ldots, Y_T = y_T,$$

and this can be done as follows:

$$\begin{aligned}
&f_{\theta|X_0=x_0,\ldots,X_T=x_T,Y_0=y_0,\ldots,Y_T=y_T}(\theta) \\
&\propto f_{\theta,X_0,\ldots,X_T|Y_0=y_0,\ldots,Y_T=y_T}(\theta, x_0, \ldots, x_T) \\
&\propto I\{e^{-C} < \sigma_Z, \sigma_\epsilon < e^C\}\phi(x_0; 0, C)\sigma_Z^{-T-1} \exp\left(-\frac{\sum_{t=1}^T (x_t - x_{t-1})^2}{2\sigma_Z^2}\right) \sigma_\epsilon^{-T-2} \exp\left(-\frac{\sum_{t=0}^T (y_t - x_t)^2}{2\sigma_\epsilon^2}\right) \\
&\propto I\{e^{-C} < \sigma_Z < e^C\}\sigma_Z^{-T-1} \exp\left(-\frac{\sum_{t=1}^T (x_t - x_{t-1})^2}{2\sigma_Z^2}\right) I\{e^{-C} < \sigma_\epsilon < e^C\}\sigma_\epsilon^{-T-2} \exp\left(-\frac{\sum_{t=0}^T (y_t - x_t)^2}{2\sigma_\epsilon^2}\right)
\end{aligned}$$

as calculated previously. Thus, conditional on $X_0 = x_0, \ldots, X_T = x_0, Y_0 = y_0, \ldots, Y_T = y_T$, the two parameters σ_Z and σ_ϵ are independent with

$$f_{\sigma_Z|X_0=x_0,\ldots,X_T=x_T,Y_0=y_0,\ldots,Y_T=y_T}(\sigma_Z) \propto I\{e^{-C} < \sigma_Z < e^C\}\sigma_Z^{-T-1} \exp\left(-\frac{\sum_{t=1}^T (x_t - x_{t-1})^2}{2\sigma_Z^2}\right)$$

and

$$f_{\sigma_\epsilon|X_0=x_0,\ldots,X_T=x_T,Y_0=y_0,\ldots,Y_T=y_T}(\sigma_\epsilon) \propto I\{e^{-C} < \sigma_\epsilon < e^C\}\sigma_\epsilon^{-T-2} \exp\left(-\frac{\sum_{t=0}^T (y_t - x_t)^2}{2\sigma_\epsilon^2}\right).$$

By changing the indicators to $I\{\sigma_Z > 0\}$ and $I\{\sigma_\epsilon > 0\}$ (which is justified when C is large), and using the standard change of variable formula, we obtain

$$\sigma_Z^{-2} \mid X_0 = x_0, \ldots, X_T = x_0, Y_0 = y_0, \ldots, Y_T = y_T \sim \text{Gamma}\left(\frac{T}{2}, \frac{\sum_{t=1}^T (x_t - x_{t-1})^2}{2}\right),$$

and

$$\sigma_\epsilon^{-2} \mid X_0 = x_0, \ldots, X_T = x_0, Y_0 = y_0, \ldots, Y_T = y_T \sim \text{Gamma}\left(\frac{T+1}{2}, \frac{\sum_{t=0}^T (y_t - x_t)^2}{2}\right).$$

Thus the second step in the iteration for the Gibbs sampler, we simply generate Gamma random variables $G_Z^{(i)}$ and $G_\epsilon^{(i)}$ from the above pair of distributions (with $x_t = X_t^{(i)}$) and then transform them as $\sigma_Z^{(i)} := 1/\sqrt{G_Z^{(i)}}$ and $\sigma_\epsilon^{(i)} := 1/\sqrt{G_\epsilon^{(i)}}$. Thus implementing the Gibbs sampler for the local level model is quite simple.

14.4 Gibbs sampler for general Linear Gaussian state space models

The Gibbs sampler algorithm for general Linear Gaussian state space models is basically the same as the one we saw in the last section:

1. Start with $\theta^{(0)}$.
2. For each $i = 1, \ldots, N$,
   a) Generate $X_0^{(i)}, \ldots, X_T^{(i)}$ from the conditional joint distribution of $X_0, \ldots, X_T$ given $\theta = \theta^{(i)}$ and $Y_0 = y_0, \ldots, Y_T = y_T$. Because we are conditioning on $\theta = \theta^{(i)}$ here, these samples are obtained by the FFBSampling algorithm discussed in the last class.
   b) Generate $\theta^{(i)}$ from the conditional distribution of θ given $X_0 = X_0^{(i)}, X_1 = X_1^{(i)}, \ldots, X_T = X_T^{(i)}$ and $Y_0 = y_0, \ldots, Y_T = y_T$. Unlike the case of the local level model, this step may not always be carried out in closed form. It depends on the specific dependence of the matrices defining the linear Gaussian model on the parameters θ.

14.5 Recommended Reading for Today

1. A general introduction to the Gibbs sampler is in Section 5.7.1 of the Triantafyllopoulos book and in Section 1.6.1 of the Petris-Petrone-Campagnoli book.
2. The Gibbs sampler for the local level model is given in Section 4.4.3 of the Petris-Petrone-Campagnoli book and Section 5.7.3 of the Triantafyllopoulos book.

15 Lecture Fifteen

In the last class, we started discussing Full Bayes estimation of state space models. Full Bayesian estimation means that we put a prior on the unknown parameters θ (as opposed to obtaining point estimates for θ and ignoring the uncertainty in their estimation).

In some applications of state space models such as tracking, θ represents nuisance parameters with the main focus centered on the state variables. In such applications, full Bayesian estimation ensures that uncertainty in estimation of θ is accounted for in our uncertainty quantification of the state variables. In certain other applications of state space models, the main focus is on θ (and the state variables can be considered nuisance parameters). This is for example the case for ARMA models (which can be written in state space form). Here uncertainty quantification for θ is important which is achieved by Full Bayes analysis.

We shall discuss several approaches for Full Bayesian Analysis today. The main starting point is the choice of prior on θ. We shall generally use noninformative (diffuse) priors such as Unif$(-C, C)$ or $N(0, C)$ (with large C) for the components of θ or certain transformations of the components of θ (such as $\log \sigma_Z$ and $\log \sigma_\epsilon$ for $\theta = (\sigma_Z, \sigma_\epsilon)$ in the local level model). When the likelihood is peaked around the MLE (which would generally be the case when the number of observations T is large), it actually does not matter much as to what the prior is. Some heuristic justification for this will be provided today.

Here are some of the ways of doing Full Bayesian Analysis of state space models. We are assuming, from now on, that we have fixed a prior $f_\theta(\theta)$ for the unknown parameters θ.

15.1 Approach One: Gibbs Sampling

We looked at Gibbs sampling in the last class. It proceeds according to the following algorithm.

1. Start with initial values $\theta^{(0)}$.
2. For each $i = 1, \ldots, N$,
   a) Generate $X_0^{(i)}, \ldots, X_T^{(i)}$ from the conditional joint distribution of $X_0, \ldots, X_T$ given $\theta = \theta^{(i-1)}$ and $Y_0 = y_0, \ldots, Y_T = y_T$. Because we are conditioning on a fixed value of θ here, these samples can be obtained by the FFBSampling algorithm discussed previously.
   b) Generate $\theta^{(i)}$ from the conditional distribution of θ given $X_0 = X_0^{(i)}, X_1 = X_1^{(i)}, \ldots, X_T = X_T^{(i)}$ and $Y_0 = y_0, \ldots, Y_T = y_T$.

The last step of the Gibbs sampling algorithm (which involves generating $\theta^{(i)}$ from the conditional distribution of θ given $X_0 = X_0^{(i)}, X_1 = X_1^{(i)}, \ldots, X_T = X_T^{(i)}$ and $Y_0 = y_0, \ldots, Y_T = y_T$) cannot always be carried out in closed form for state space models. In the last class, we saw how this can be done in closed form for the local level model.

The empirical probability measure of the samples $(\theta^{(i)}, X_0^{(i)}, \ldots, X_T^{(i)})$ for $i = 1, \ldots, N$ generated by the Gibbs sampler can be used to approximate the full posterior distribution of $(\theta, X_0, \ldots, X_T)$ given the data $Y_0 = y_0, \ldots, Y_T = y_T$:

$$\frac{1}{N}\sum_{i=1}^{N} \delta_{(\theta^{(i)}, X_0^{(i)}, \ldots, X_T^{(i)})} \approx f_{\theta, X_0, \ldots, X_T | Y_0 = y_0, \ldots, Y_T = y_t}.$$

One implication of this is

$$\int g(\theta, x_0, \ldots, x_T) f_{\theta, X_0, \ldots, X_T | Y_0 = y_0, \ldots, Y_T = y_t}(\theta, x_0, \ldots, x_T) d\theta dx_0 \ldots dx_T \approx \frac{1}{N}\sum_{i=1}^{N} g(\theta^{(i)}, X_0^{(i)}, \ldots, X_T^{(i)}).$$

for arbitrary functions g. For example, the posterior mean of θ given $Y_0 = y_0, \ldots, Y_T = y_T$ is approximated by

$$\frac{1}{N}\sum_{i=0}^{N} \theta^{(i)}.$$

One thing to note that the samples generated by the Gibbs sampler do not correspond to independent draws from the $(\theta, X_0, \ldots, X_T)$ given the data $Y_0 = y_0, \ldots, Y_T = y_T$. Instead, they represent draws from a Markov Chain whose stationary distribution is the full posterior distribution of $(\theta, X_0, \ldots, X_T)$ given the data $Y_0 = y_0, \ldots, Y_T = y_T$. Thus the Gibbs Sampler is an example of a Markov Chain Monte Carlo (MCMC) algorithm.

15.2 Approach Two: Direct Sampling

Direct sampling generates **independent** draws $(\theta^{(i)}, X_0^{(i)}, \ldots, X_T^{(i)})$ for $i = 1, \ldots, N$ via the following algorithm. For each $i = 1, \ldots, N$,

1. Generate $\theta^{(i)}$ according to the conditional distribution of θ given $Y_0 = y_0, \ldots, Y_T = y_T$.
2. Generate $X_0^{(i)}, \ldots, X_T^{(i)}$ from the conditional joint distribution of $X_0, \ldots, X_T$ given $\theta = \theta^{(i)}$ and $Y_0 = y_0, \ldots, Y_T = y_T$. Again, because we are conditioning on a fixed value of θ here, these samples can be obtained by the FFBSampling algorithm.

Notice the very close similarity between the Direct Sampling and the Gibbs Sampling algorithms. The issue with the Direct Sampling algorithm is that the first step is generally difficult and cannot be done in closed form using standard distributions. This is because, in most state space models, the conditional density of θ given $Y_0 = y_0, \ldots, Y_T = y_T$ is a somewhat complicated function given implicitly via the Kalman filter. For some simple models however (such as AutoRegressive Models as we shall discuss in the next class), this method can be carried out.

Note again that, unlike the Gibbs sampler, direct sampling generates independent draws from the full posterior.

15.3 Approach Three: Posterior Normal Approximation

This can be seen as a modification of the Direct Sampling algorithm where the first step is replaced by sampling from a normal approximation: For each $i = 1, \ldots, N$,

1. Generate $\theta^{(i)}$ according to a **normal approximation** to the conditional distribution of θ given $Y_0 = y_0, \ldots, Y_T = y_T$.
2. Generate $X_0^{(i)}, \ldots, X_T^{(i)}$ from the conditional joint distribution of $X_0, \ldots, X_T$ given $\theta = \theta^{(i)}$ and $Y_0 = y_0, \ldots, Y_T = y_T$. This step is the same as in the Direct Sampling algorithm.

Here is one way of obtaining the normal approximation for use in the first step of the above algorithm. Note first that

$$f_{\theta|Y_0=y_0,\ldots,Y_T=y_T}(\theta) \propto f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T) f_\theta(\theta)$$

In the right hand side above, the term $f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T)$ is simply the likelihood (which is given by the Kalman filter for linear Gaussian state space models) and the second term $f_\theta(\theta)$ is the prior. Note that the likelihood $f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T)$ is maximized at the maximum likelihood estimate $\hat{\theta}$. Now generally in state space models, the likelihood is quite peaked around $\hat{\theta}$ which means that the likelihood is very close to 0 outside of a small region around $\hat{\theta}$. On the other hand, the prior $f_\theta(\theta)$ is quite flat which means that it can be well-approximated by a constant (such as $f_\theta(\hat{\theta})$) in the region where the likelihood is significantly different from zero. This leads to the approximation:

$$f_{\theta|Y_0=y_0,\ldots,Y_T=y_T}(\theta) \overset{\bullet}{\propto} f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T) f_\theta(\hat{\theta}) \propto f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T)$$

where $\overset{\bullet}{\propto}$ means "proportional to approximately". In the second relation above, we dropped $f_\theta(\hat{\theta})$ as it is a constant. Thus when the prior is flat in the region where the likelihood is significantly different from zero, the posterior of θ given the data is proportional to the likelihood and does not depend on the exact form of the prior. Writing in terms of the log-likelihood:

$$\ell(\theta) := \log f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T),$$

we get

$$f_{\theta|Y_0=y_0,\ldots,Y_T=y_T}(\theta) \overset{\bullet}{\propto} \exp\left(\ell(\theta)\right).$$

We now do a second order Taylor expansion of $\ell(\theta)$ around the MLE $\hat{\theta}$ to get

$$\begin{aligned} f_{\theta|Y_0=y_0,\ldots,Y_T=y_T}(\theta) &\overset{\bullet}{\propto} \exp\left(\ell(\theta)\right) \\ &\approx \exp\left(\ell(\hat{\theta}) + \left\langle \nabla\ell(\hat{\theta}), \theta - \hat{\theta} \right\rangle + \frac{1}{2}\left(\theta - \hat{\theta}\right)^T H\ell(\hat{\theta})\left(\theta - \hat{\theta}\right)\right). \end{aligned}$$

Note the following about the three terms appearing on the right hand side above. The first term $\exp(\ell(\hat{\theta}))$ is just a constant and will be ignored in proportionality. The second term equals zero because $\nabla \ell(\hat{\theta}) = 0$ as $\hat{\theta}$ is a maximizer of $\ell(\theta)$ (this is, strictly speaking, an assumption because this may not be true if $\hat{\theta}$ is not an interior point in the domain of $\ell(\theta)$). The Hessian $H\ell(\hat{\theta})$ is negative semi-definite (i.e., $-H\ell(\hat{\theta})$ is positive semi-definite) as $\hat{\theta}$ maximizes $\ell(\theta)$ (this also may not always be true but this is generally true). We therefore get

$$f_{\theta|Y_0=y_0,\ldots,Y_T=y_T}(\theta) \overset{\bullet}{\propto} \exp\left(-\frac{1}{2}\left(\theta-\hat{\theta}\right)^T\left(-H\ell(\hat{\theta})\right)\left(\theta-\hat{\theta}\right)\right).$$

The right hand side above is the multivariate normal density (without the normalizing constant). We thus have

$$\theta \mid Y_0 = y_0, \ldots, Y_T = y_T \overset{\bullet}{\sim} N\left(\hat{\theta}, \left(-H\ell(\hat{\theta})\right)^{-1}\right)$$

where $\overset{\bullet}{\sim}$ means "approximately distributed as". This posterior normal approximation is quite popular. Note that for state space models, the log-likelihood is calculated via the Kalman filter and the Hessian of the log-likelihood can be calculated numerically.

Therefore the sampling algorithm using posterior normal approximation is the following. For each $i = 1, \ldots, N$,

1. Generate $\theta^{(i)}$ according to the multivariate normal distribution with mean $\hat{\theta}$ and covariance matrix $\left(-H\ell(\hat{\theta})\right)^{-1}$. Here $\hat{\theta}$ denotes the MLE and $\ell(\theta)$ denotes the log-likelihood (which is obtained by filtering).
2. Generate $X_0^{(i)}, \ldots, X_T^{(i)}$ from the conditional joint distribution of $X_0, \ldots, X_T$ given $\theta = \theta^{(i)}$ and $Y_0 = y_0, \ldots, Y_T = y_T$.

Note that the prior does not appear in the above algorithm at all which can be considered an attractive feature. The posterior normal approximation does not work in the following two situations:

1. The prior $f_\theta(\theta)$ varies considerably in the region of the likelihood. This is generally not an issue as we usually work with flat priors.
2. When θ is far from $\hat{\theta}$, the second order Taylor expansion of $\ell(\theta)$ around $\hat{\theta}$ will not give a good approximation of $\ell(\theta)$. Now if $\ell(\theta)$ is already negligible for such values of θ, this poor approximation will not be a issue. If not, then the normal approximation will not be accurate for the posterior.

15.4 Approach Four: Importance Sampling

Importance sampling can be used when direct sampling is infeasible and posterior normal approximation is not accurate. The basic problem is as follows. We are interested in approximating a distribution P with density p. We cannot sample from P directly but we have the ability to obtain independent samples $X_1, \ldots, X_n$ from a distribution Q (with density q). As explained below, importance sampling provides an approximation for P in terms of $X_1, \ldots, X_n$. In the state space model context, P will be $f_{\theta|Y_0=y_0,\ldots,Y_T=y_T}(\theta)$ and Q will be an approximation (such as the posterior normal approximation).

Importance sampling provides two approximations for P. The first approximation is

$$\hat{P}_1 := \frac{1}{n}\sum_{i=1}^{n} \frac{p(X_i)}{q(X_i)} \delta_{\{X_i\}}.$$

By $\delta_{\{X_i\}}$, we mean a point mass at X_i. Basically $\hat{P}_1$ is a discrete measure giving the weight $\frac{1}{n}\frac{p(X_i)}{q(X_i)}$ to the point X_i. Note that $\hat{P}_1$ is not necessarily a probability measure because the weights $\frac{1}{n}\frac{p(X_i)}{q(X_i)}$ do not necessarily add to 1. It is a approximation to P in the sense that

$$\int g(x) d\hat{P}_1(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{p(X_i)}{q(X_i)} g(X_i) \approx \int g(x) dP(x)$$

for most functions g. This is basically a consequence of the Strong Law of Large Numbers which implies (under a minimal first moment assumption on g) that

$$\frac{1}{n}\sum_{i=1}^{n} \frac{p(X_i)}{q(X_i)} g(X_i) \to \int \frac{p(x)}{q(x)} g(x) q(x) dx = \int g(x) dP(x) \qquad \text{almost surely as } n \to \infty. \quad (97)$$

There are two annoying issues with $\hat{P}_1$:

1. As already mentioned, it is not a probability measure.
2. To use $\hat{P}_1$, we need to know the density $p(x)$ fully. In many situations (including in our setting of state space models), we would only know $p(x)$ upto multiplication by a normalizing constant. This would preclude use of $\hat{P}_1$.

To fix these two issues, importance sampling proposes the following estimator (often known as *self-normalized* importance sampling):

$$\hat{P}_2 := \sum_{i=1}^{n} w_i \delta_{\{X_i\}} \qquad \text{where } w_i := \frac{\frac{1}{n}\frac{p(X_i)}{q(X_i)}}{\sum_{j=1}^{n} \frac{1}{n}\frac{p(X_j)}{q(X_j)}}$$

It is clear that $\hat{P}_2$ is a probability measure. Also to use $\hat{P}_2$, it is enough to know $p(x)$ (and also $q(x)$) up to an unknown multiplicative constant. To see why $\hat{P}_2$ is a good approximation of P, observe that for a function g:

$$\int g(x) d\hat{P}_2(x) = \sum_{i=1}^{n} w_i g(X_i) = \frac{\frac{1}{n}\sum_{i=1}^{n} g(X_i)\frac{p(X_i)}{q(X_i)}}{\frac{1}{n}\sum_{i=1}^{n} \frac{p(X_i)}{q(X_i)}}$$

As we have seen in (97), the numerator above converges to $\int g(x) dP(x)$ almost surely as $n \to \infty$. By another application of the law of large numbers, it can be seen that the denominator converges to 1:

$$\frac{1}{n}\sum_{i=1}^{n} \frac{p(X_i)}{q(X_i)} \to \int \frac{p(x)}{q(x)} q(x) dx = \int p(x) dx = 1 \qquad \text{almost surely as } n \to \infty.$$

Thus

$$\int g(x) d\hat{P}_2(x) \to \int g(x) p(x) dx \qquad \text{almost surely as } n \to \infty.$$

We can do a more precise comparison of the performance of the two estimators:

$$E_1 := \int g(x) d\hat{P}_1(x) = \frac{1}{n}\sum_{i=1}^{n} g(X_i) \frac{p(X_i)}{q(X_i)}$$

and

$$E_2 := \int g(x) d\hat{P}_2(x) = \frac{\frac{1}{n}\sum_{i=1}^n g(X_i)\frac{p(X_i)}{q(X_i)}}{\frac{1}{n}\sum_{i=1}^n \frac{p(X_i)}{q(X_i)}}$$

for estimating

$$\mu := \int g(x) dP(x)$$

by calculation of variances. Note that these estimators are based on data $X_1, \ldots, X_n$ that are independent with common distribution Q. We can also compare these estimators against the simple estimator

$$E_0 := \frac{1}{n}\sum_{i=1}^n g(X_i)$$

based on independent observations $X_1, \ldots, X_n$ having distribution P. E_0 will not be a feasible estimator if we cannot sample from P but we can still use it as a comparison benchmark for the feasible estimators E_1 and E_2. Here are basic observations about these three estimators:

1. **Estimator** E_0: E_0 is clearly an unbiased estimator of μ. Its variance is given by

$$\text{var}(E_0) = \frac{1}{n}\text{var}_P\left(g(X_1)\right) = \frac{1}{n}\int \left(g(x)-\mu\right)^2 p(x)dx. \tag{98}$$

   The subscript P in var_P refers to $X_1 \sim P$. Note also that the mean of $g(X_1)$ under $X_1 \sim P$ is $\int g(x)p(x)dx = \mu$.

2. **Estimator** E_1: E_1 is also an unbiased estimator of μ. Its variance is given by

$$\text{var}(E_1) = \frac{1}{n}\text{var}_Q\left(g(X_1)\frac{p(X_1)}{q(X_1)}\right) = \frac{1}{n}\int \left(g(x)\frac{p(x)}{q(x)} - \mu\right)^2 q(x)dx = \frac{1}{n}\int \frac{\left[g(x)p(x) - \mu q(x)\right]^2}{q(x)}dx.$$

   Note that it is possible that $\text{var}(E_1)$ is much smaller than $\text{var}(E_0)$. This will be the case, for example, when

$$q(x) \approx \frac{g(x)p(x)}{\mu} = \frac{g(x)p(x)}{\int g(x)p(x)dx}.$$

   In the extreme case when $q(x)$ is exactly equal to the right hand side, E_1 is a perfect estimator of μ having zero variance. This also suggests that in situations where g is non-zero only in a tiny part of the support of P, the importance sampling estimator E_1 will work much better than the direct sampling estimator E_0 when $q(x)$ is concentrated on the specific tiny part of the support of P.

3. **Estimator** E_2: This is not an unbiased estimator. But the numerator $\frac{1}{n}\sum_{i=1}^n g(X_i)\frac{p(X_i)}{q(X_i)}$ and denominator $\frac{1}{n}\sum_{i=1}^n \frac{p(X_i)}{q(X_i)}$ of E_2 are very close (by the law of large numbers) to μ and 1 respectively. So we can approximate E_2 by a simple first order Taylor expansion as follows. Let $f(A,B) := \frac{A}{B}$. For fixed points A_0, B_0, we have

$$\begin{aligned}\frac{A}{B} = f(A,B) &\approx f(A_0,B_0) + (A-A_0)\frac{\partial f}{\partial A}\bigg|_{A=A_0,B=B_0} + (B-B_0)\frac{\partial f}{\partial B}\bigg|_{A=A_0,B=B_0} \\ &= \frac{A_0}{B_0} + \frac{A-A_0}{B_0} - \frac{A_0(B-B_0)}{B_0^2}.\end{aligned}$$

   Using this with

$$A := \frac{1}{n}\sum_{i=1}^n g(X_i)\frac{p(X_i)}{q(X_i)} \quad B := \frac{1}{n}\sum_{i=1}^n \frac{p(X_i)}{q(X_i)} \quad A_0 = \mu \quad B_0 = 1,$$

we get

$$E_2 \approx \mu + (A - \mu) - \mu(B-1) = \mu + A - B\mu = \mu + \frac{1}{n}\sum_{i=1}^{n} (g(X_i) - \mu)\frac{p(X_i)}{q(X_i)}.$$

This implies that E_2 is approximately unbiased because

$$\begin{aligned}
\mathbb{E}(E_2) &\approx \mu + \mathbb{E}_Q\left[(g(X_1) - \mu)\frac{p(X_1)}{q(X_1)}\right] \\
&= \mu + \int (g(x) - \mu)\frac{p(x)}{q(x)} q(x)dx \\
&= \mu + \int (g(x) - \mu)\, p(x)dx = \mu + \left(\int g(x)p(x)dx - \mu\right) = \mu.
\end{aligned}$$

Further the variance of E_2 is approximately

$$\mathrm{var}(E_2) \approx \frac{1}{n}\mathrm{var}_Q\left((g(X_1) - \mu)\frac{p(X_1)}{q(X_1)}\right) = \frac{1}{n}\int (g(x) - \mu)^2 \frac{p^2(x)}{q(x)}dx.$$

This variance is more similar to (98) but it can still be smaller than (98). The best possible variance reduction occurs when

$$q^*(x) = \frac{|g(x) - \mu|p(x)}{\int |g(x) - \mu|p(x)dx}$$

when

$$\mathrm{var}(E_2) \approx \frac{1}{n}\left(\int |g(x) - \mu|p(x)dx\right)^2$$

which is definitely smaller than $\mathrm{var}(E_0)$. To see why q^* minimizes $\mathrm{var}(E_2)$, just note, by Cauchy-Schwarz inequality ($\int |a(x)b(x)|dx \le \sqrt{\int a^2(x)dx}\sqrt{\int b^2(x)dx}$) that

$$\begin{aligned}
\int |g(x) - \mu|p(x)dx &= \int \frac{|g(x) - \mu|p(x)}{\sqrt{q(x)}}\sqrt{q(x)}dx \\
&\le \sqrt{\int (g(x) - \mu)^2 \frac{p^2(x)}{q(x)}dx}\sqrt{\int q(x)dx} = \sqrt{\int (g(x) - \mu)^2 \frac{p^2(x)}{q(x)}dx}.
\end{aligned}$$

15.5 Recommended Reading for Today

1. For the importance sampling approach to full Bayesian analysis, see Chapter 13 of the Durbin-Koopman book.
2. A standard MCMC method such as Metropolis-Hastings can also be used in step 1 of the Direct Sampling approach. This method is described in Section 12.2.2 of the Särkkä book or in Section 6.8.1 of the Triantafyllopoulos book.
3. For a general overview of importance sampling, see Chapter 8 of the Chopin-Papaspiliopoulos book or this book chapter: `https://artowen.su.domains/mc/Ch-var-is.pdf`.

16 Lecture Sixteen

Our next topic is Sequential Monte Carlo methods for general state space models. Here the conditional densities $f_{X_t|X_{t-1}=x_{t-1},\theta}(\cdot)$ and $f_{Y_t|X_t=x_t,\theta}(\cdot)$ (as well as the initial density f_{X_0})

can be arbitrary. We shall first look at the problem of filtering. Recall that filtering can be used for writing down the likelihood (which is necessary for inference of θ). Filtering will also be necessary for solving the smoothing problem which we shall study later.

Recall that filtering refers to the problem of determining the conditional distributions:

$$X_t \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta \qquad \text{for } t = 0, 1, 2, \ldots, T.$$

Our approach will be recursive and we shall determine the above distributions sequentially for $t = 0, 1, 2, \ldots$. In Lecture Six, we have seen closed form formulae for obtaining the filtering density at time t using the filtering density at time $t-1$. This involved two steps which we termed *one-step ahead prediction update* and *filtering update*. The one-step ahead prediction update is the following formula for the density of X_t given $Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta$ in terms of the density of X_{t-1} given $Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta$:

$$f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t) = \int f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) dx_{t-1}. \quad (99)$$

The filtering update is the following formula for the density of X_t given $Y_0 = y_0, \ldots, Y_t = y_t, \theta$ in terms of the density of X_t given $Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}$:

$$f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t) = \frac{f_{Y_t|X_t=x_t,\theta}(y_t) f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t)}{\int f_{Y_t|X_t=u,\theta}(y_t) f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(u) du} \quad (100)$$

Formula (100) can be seen as an application of the Bayes rule with the following choices of "prior" and "likelihood":

$$\text{prior} : f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}, \text{ and likelihood} : f_{Y_t|X_t=x_t,\theta} = f_{Y_t|X_t=x_t,Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta} \quad (101)$$

The "posterior" corresponding to the above prior and likelihood is the density of X_t given $Y_0 = y_0, \ldots, Y_t = y_t, \theta$ and is obtained by the Bayes rule leading to the formula (100).

For general state space models, the integral involved in (99) cannot be evaluated in closed form. This would make (100) intractable as well (because (100) needs $f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}$ as input). One approach for dealing with intractibility is to use Monte Carlo which leads to Sequential Monte Carlo methods for state space models. In Monte Carlo methods, the focus is not on evaluating an unknown density f in closed form, and instead, the focus is on obtaining i.i.d samples $X^{(1)}, \ldots, X^{(N)}$ from f. Once these samples are obtained, the distribution corresponding to the density f is approximated by the discrete uniform distribution on $X^{(1)}, \ldots, X^{(N)}$:

$$\text{Unif}\{X^{(1)}, \ldots, X^{(N)}\}. \quad (102)$$

In order to evaluate the expectation of a function g with respect to the density f, the Monte Carlo approach will give

$$\int g(x) f(x) dx \approx \frac{1}{N} \sum_{i=1}^{n} g(X^{(i)}).$$

16.1 Notation for Discrete Distributions

We shall use the following notation in the sequel. A discrete distribution that takes the values $x^{(1)}, \ldots, x^{(N)}$ with probabilities $p^{(1)}, \ldots, p^{(N)}$ will be denoted by

$$p^{(1)} \delta_{\{x^{(1)}\}} + \cdots + p^{(N)} \delta_{\{x^{(N)}\}} = \sum_{i=1}^{n} p^{(i)} \delta_{\{x^{(i)}\}}.$$

For example, the distribution taking the three values $5, 2, -6$ with probabilities $0.3, 0.5, 0.2$ respectively will be written as

$$0.3\delta_{\{5\}} + 0.5\delta_{\{2\}} + 0.2\delta_{\{-6\}}.$$

Note that the uniform distribution (102) is written as

$$\sum_{i=1} \frac{1}{N}\delta_{\{X^{(i)}\}}$$

in this notation.

16.2 Monte Carlo versions of (99) and (100)

In terms of Monte Carlo, the basic question underlying filtering is the following:

Question 16.1. *Suppose we are given i.i.d samples $X_{t-1}^{(1)}, \ldots, X_{t-1}^{(N)}$ from the distribution $X_{t-1} \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta$ (this is the filtering distribution at time $t-1$). How then do we generate i.i.d samples $X_t^{(1)}, \ldots, X_t^{(N)}$ from the distribution $X_t \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ (this is the filtering distribution at time t)?*

We shall solve this question by using Monte Carlo versions of (99) and (100). We start with i.i.d samples $X_{t-1}^{(1)}, \ldots, X_{t-1}^{(N)}$ from the filtering density $f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}$ at time $t-1$. For the one-step ahead prediction update, we need to obtain samples from the density of X_t given $Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta$. This is easily done via:

$$\tilde{X}_t^{(i)} \sim f_{X_t|X_{t-1}=X_{t-1}^{(i)}} \qquad \text{for } i = 1, \ldots, N.$$

This makes sense because the right hand side of (99) is simply the marginal density of X_t under the model:

$$X_t \mid X_{t-1} = x_{t-1} \sim f_{X_t|X_{t-1}=x_{t-1}} \quad \text{and} \quad X_{t-1} \sim f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}.$$

Thus $\tilde{X}_t^{(1)}, \ldots, \tilde{X}_t^{(N)}$ are i.i.d samples from the one-step ahead prediction distrbution $X_t \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta$. One can then approximate the one-step ahead prediction distribution by

$$(X_t \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta) \approx \text{Unif}\left\{\tilde{X}_t^{(1)}, \ldots, \tilde{X}_t^{(N)}\right\} = \sum_{i=1}^{n} \frac{1}{N}\delta_{\{\tilde{X}_t^{(i)}\}}. \tag{103}$$

Let us now come to (100). As noted earlier, this equation arises from the Bayes rule with prior and likelihood given in (101). We do not have access to the prior density $f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}$ as we have not evaulated (99) in closed form. We do, however, have the Monte Carlo approximation (103) for the one-step ahead prediction distribution so it is natural to approximate (100) by applying Bayes rule with

$$\text{prior} : \sum_{i=1}^{n} \frac{1}{N}\delta_{\{\tilde{X}_t^{(i)}\}}, \quad \text{and} \quad \text{likelihood} : f_{Y_t|X_t=x_t,\theta}.$$

The unnormalized posterior corresponding to the prior and likelihood above is given by the weights:

$$w_t^{(i)} := f_{Y_t|X_t=\tilde{X}_t^{(i)},\theta}(y_t).$$

The properly normalized posterior is then given by

$$\sum_{i=1}^{n} W_t^{(i)} \delta_{\{\tilde{X}_t^{(i)}\}} \qquad \text{where } W_t^{(i)} := \frac{w_t^{(i)}}{w_t^{(1)} + \cdots + w_t^{(N)}}.$$

This discrete distribution approximates the filtering distribution at time t:

$$X_t \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta \approx \sum_{i=1}^{n} W_t^{(i)} \delta_{\{\tilde{X}_t^{(i)}\}}.$$

In order to generate i.i.d samples from the filtering distribution at time t, we can simply generate samples from the above discrete distribution:

$$X_t^{(1)}, \ldots, X_t^{(N)} \sim \sum_{i=1}^{n} W_t^{(i)} \delta_{\{\tilde{X}_t^{(i)}\}}.$$

This algorithm for solving the filtering problem in general state space models using Monte Carlo is called *Bootstrap Particle Filter*. We stae the algorithm formally in the next section.

16.3 The Bootstrap Particle Filter

For each $t \geq 0$, this algorithm outputs samples $X_t^{(1)}, \ldots, X_t^{(N)}$ such that

$$\text{Unif}\left\{X_t^{(1)}, \ldots, X_t^{(N)}\right\} \approx X_t \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta.$$

The algorithm proceeds sequentially. At time $t-1$, one has access to the samples $X_{t-1}^{(1)}, \ldots, X_{t-1}^{(N)}$ and using these, one generates the samples $X_t^{(1)}, \ldots, X_t^{(N)}$ by following the three steps given below.

1. **Generation**: For each $i = 1, \ldots, N$, generate independent samples:

   $$\tilde{X}_t^{(i)} \sim f_{X_t | X_{t-1} = X_{t-1}^{(i)}}.$$

   To execute this step, we need to be able to simulate from the state transition density $f_{X_t | X_{t-1} = x_{t-1}}$.

2. **Weights**: For each $i = 1, \ldots, N$, compute

   $$w_t^{(i)} := f_{Y_t | X_t = \tilde{X}_t^{(i)}}(y_t). \tag{104}$$

   Normalize these weights so they sum to one:

   $$W_t^{(i)} = \frac{w_t^{(i)}}{w_t^{(1)} + \cdots + w_t^{(N)}} \qquad \text{for } i = 1, \ldots, N.$$

   To execute this step, we need to be able to evalute the conditional density $f_{Y_t | X_t = x_t}(y_t)$ at least up to a constant that does not depend on x_t.

3. **Resampling**: Generate

   $$X_t^{(1)}, \ldots, X_t^{(N)} \overset{\text{i.i.d}}{\sim} \sum_{i=1}^{N} W_i \delta_{\{\tilde{X}_t^{(i)}\}}$$

This algorithm is initialized by taking

$$\tilde{X}_0^{(1)}, \ldots, \tilde{X}_0^{(N)} \overset{\text{i.i.d}}{\sim} f_{X_0|\theta}$$

and then following the steps 2 (weights) and 3 (resampling) above to generate $X_0^{(1)}, \ldots, X_0^{(N)}$. One can then repeat the recursion for $t = 1, 2, 3, \ldots$. This is similar to the way we initialized the Kalman filter.

This algorithm is called the Bootstrap Particle Filter because: (a) Monte-Carlo samples are called particles in the physics literature, (b) The resampling step is reminiscent of the bootstrap procedure in statistics.

The Bootstrap Particle Filter is very simple and easy to implement. It can also be understood from the point of view of Importance Sampling. Before describing this connection to importance sampling, let us briefly recall importance sampling.

16.4 Importance Sampling Recalled

Consider a probability measure P with density p. Suppose we do not know the formula for p exactly but we only know it up to some unknown multiplicative constant factor c. In other words, we know the explicit formula for the function $x \mapsto cp(x)$ but we do not know c and hence we do not know $p(x)$ explicitly.

Importance sampling attempts to approximate P using i.i.d samples $\tilde{X}^{(1)}, \ldots, \tilde{X}^{(n)}$ drawn from another probability measure Q having density q. The idea is to form weights

$$w^{(i)} := \frac{cp(\tilde{X}^{(i)})}{q(\tilde{X}^{(i)})} \qquad \text{for } i = 1, \ldots, N$$

and the corresponding normalized weights:

$$W^{(i)} := \frac{w^{(i)}}{w^{(1)} + \cdots + w^{(N)}} \qquad \text{for } i = 1, \ldots, N.$$

Then the importance sampling approximation for P is

$$P \approx \sum_{i=1}^{N} W^{(i)} \delta_{\{\tilde{X}^{(i)}\}}$$

Observe that for every function g, this gives the following approximation for $\int g dP$:

$$\int g dP \approx \sum_{i=1}^{N} W^{(i)} g(\tilde{X}^{(i)}) = \frac{\sum_{i=1}^{N} w^{(i)} g(\tilde{X}^{(i)})}{\sum_{i=1}^{N} w^{(i)}}.$$

In Lecture 15, we used the terminology "self-normalized" importance sampling for the above estimator of $\int g dP$.

Note that $w^{(1)}, \ldots, w^{(N)}$ depend on the constant c but the normalized weights $W^{(1)}, \ldots, W^{(N)}$ don't. This means that the approximation $\sum_{i=1}^{N} W^{(i)} \delta_{\{\tilde{X}^{(i)}\}}$ does not depend on c.

It will be helpful to note the following two things before moving on:

1. **Estimating** c: Importance sampling provides the following estimate for the unknown constant c:

$$\hat{c} := \frac{1}{N} \sum_{i=1}^{N} w^{(i)}. \tag{105}$$

To see why this estimator makes sense, just note that

$$\mathbb{E}\hat{c} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left(w^{(i)}\right) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left(\frac{cp(\tilde{X}^{(i)})}{q(\tilde{X}^{(i)})}\right) = \frac{1}{N}\sum_{i=1}^{N}\int\frac{cp(x)}{q(x)}q(x)dx = \int cp(x)dx = c$$

2. **Samples from** P: Importance sampling can be used to obtain approximately i.i.d samples from P. Indeed as the importance sampling approximation for P equals $\sum_{i=1}^{N} W^{(i)}\delta_{\{\tilde{X}^{(i)}\}}$, one can obtain (approximate) samples from P by sampling from the discrete distribution $\sum_{i=1}^{N} W^{(i)}\delta_{\{\tilde{X}^{(i)}\}}$:

$$X^{(1)},\ldots,X^{(N)} \overset{\text{i.i.d}}{\sim} \sum_{i=1}^{N} W^{(i)}\delta_{\{\tilde{X}^{(i)}\}}$$

This method of sample generation is referred to as *Importance Resampling* because $X^{(1)},\ldots,X^{(N)}$ are sampled from $\tilde{X}^{(1)},\ldots,\tilde{X}^{(N)}$ (with weights $W^{(1)},\ldots,W^{(N)}$) which are themselves sampled from Q.

16.5 Bootstrap Particle Filter as Importance Resampling

The Bootstrap Particle Filter algorithm can be understood from the lens of importance resampling. This generalized view is helpful for the creation of other particle filtering algorithms. There are two (very similar) ways of seeing the connection between the Bootstrap Particle Filter and Importance Resampling.

16.5.1 First Way of Seeing the Connection

As explained in Section 16.2, the samples $\tilde{X}_t^{(1)},\ldots,\tilde{X}_t^{(N)}$ generated in the first step of the Bootstrap particle filter recursion (from time $t-1$ to t) can be seen as samples from the one-step ahead prediction distribution $X_t \mid Y_0 = y_0,\ldots,Y_{t-1} = y_{t-1},\theta$:

$$\tilde{X}_t^{(1)},\ldots,\tilde{X}_t^{(N)} \overset{\text{i.i.d}}{\sim} f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}.$$

If we now apply importance sampling to use these samples to approximate the the filtering distribution at time t: $X_t \mid Y_0 = y_0,\ldots,Y_t = y_t,\theta$, we need to use, for some positive constant c, the weights

$$\begin{aligned}&\frac{cf_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(\tilde{X}_t^{(i)})}{f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(\tilde{X}_t^{(i)})} \\ &= \frac{c\frac{f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(\tilde{X}_t^{(i)})f_{Y_t|X_t=\tilde{X}_t^{(i)},\theta}(y_t)}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1}}(y_t)}}{f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(\tilde{X}_t^{(i)})} = \frac{cf_{Y_t|X_t=\tilde{X}_t^{(i)},\theta}(y_t)}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1}}(y_t)}.\end{aligned} \tag{106}$$

It is now clear that the Bootstrap Particle Filter uses the above weights for

$$c = f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1}}(y_t)$$

so that the weights simplify to $f_{Y_t|X_t=\tilde{X}_t^{(i)},\theta}(y_t)$. Therefore each recursion of the Bootstrap Particle Filter can be seen as a version of Importance Resampling.

16.5.2 Second Way of Seeing the Connection

In the first step of the Bootstrap Particle Recursion to go from $t-1$ to t, we generate samples $\tilde{X}_t^{(1)}, \ldots, \tilde{X}_t^{(N)}$ independently according to

$$\tilde{X}_t^{(i)} \sim f_{X_t|X_{t-1}=X_{t-1}^{(i)}}$$

This means that jointly $(X_{t-1}^{(i)}, \tilde{X}_t^{(i)}), i = 1, \ldots, N$ are i.i.d samples from the joint density:

$$f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t)$$

which is just the density of $X_{t-1}, X_t \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, \theta$. We can now employ importance sampling to convert these samples into an approximation of the distribution

$$X_{t-1}, X_t \mid Y_0 = y_0, \ldots, Y_{t-1} = y_{t-1}, Y_t = y_t, \theta.$$

We would need to use weights (for some constant $c > 0$)

$$\frac{c f_{X_{t-1},X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{t-1}, x_t)}{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t)} \tag{107}$$

with $x_{t-1} = X_{t-1}^{(i)}$ and $x_t = \tilde{X}_t^{(i)}$. The above expression can be simplified using Bayes rule as

$$\begin{aligned}
&\frac{c f_{X_{t-1},X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{t-1}, x_t)}{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t)} \\
&= \frac{c \frac{f_{X_{t-1},X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1},x_t) f_{Y_t|X_{t-1}=x_{t-1},X_t=x_t,\theta}(y_t)}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)}}{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t)} \\
&= \frac{c}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)} \frac{f_{X_{t-1},X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}, x_t)}{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t)} f_{Y_t|X_t=x_t,\theta}(y_t) \\
&= \frac{c}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)} \frac{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t)}{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t)} f_{Y_t|X_t=x_t,\theta}(y_t) \\
&= \frac{c}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)} f_{Y_t|X_t=x_t,\theta}(y_t).
\end{aligned}$$

As a result, we can view the weights in the bootstrap particle filter as the weights given by (107) with $c = f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)$.

16.6 Likelihood Approximation from the Bootstrap Particle Filter

In the previous section, we have seen that the recursion (to go from time $t-1$ to time t) in the Bootstrap Particle Filter can be seen as importance sampling with weights (107) (or equivalently (106)) with $c = f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)$. The observation (105) can therefore be used to deduce that:

$$\frac{1}{N} \sum_{i=1}^{N} w_t^{(i)} \approx f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)$$

for each $t = 1, \ldots, T$ (here $w_t^{(i)}$ is as defined in (109)). One can also similarly argue that

$$\frac{1}{N} \sum_{i=1}^{N} w_0^{(i)} = \frac{1}{N} \sum_{i=1}^{N} f_{Y_0|X_0=\tilde{X}_0^{(i)}}(y_0) \approx f_{Y_0}(y_0).$$

The likelihood $f_{Y_0,\ldots,Y_T|\theta}(y_0,\ldots,y_T)$ can thus be approximated as

$$f_{Y_0,\ldots,Y_T|\theta}(y_0,\ldots,y_T) = f_{Y_0|\theta}(y_0)\prod_{t=1}^{T} f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t) \approx \prod_{t=0}^{T}\left(\frac{1}{N}\sum_{i=1}^{N} w_t^{(i)}\right).$$

In this way, the bootstrap particle filter algorithm directly allows likelihood computation.

16.7 Recommended Reading for Today

1. For the Bootstrap Particle Filter Algorithm, I recommend Section 15.2 of the Kitagawa book (Kitagawa refers to the algorithm as simply *The Monte Carlo Filter*).
2. For more details about importance sampling and resampling, I recommend Chapters 8 and 9 of the Chopin-Papaspiliopoulos book.

17 Lecture Seventeen

17.1 Recap: Bootstrap Particle Filter

In the last class, we studied the Bootstrap Particle Filter Algorithm for solving the filtering problem via Monte Carlo in general sequential state space models.

For each $t \geq 0$, this algorithm outputs samples $X_t^{(1)},\ldots,X_t^{(N)}$ such that

$$\text{Unif}\left\{X_t^{(1)},\ldots,X_t^{(N)}\right\} \approx X_t \mid Y_0 = y_0,\ldots,Y_t = y_t,\theta. \tag{108}$$

The algorithm proceeds sequentially. At time $t-1$, one has access to the samples $X_{t-1}^{(1)},\ldots,X_{t-1}^{(N)}$ satisfying (108) for $t-1$ and using these, one generates the samples $X_t^{(1)},\ldots,X_t^{(N)}$ by following the three steps given below.

1. **Generation**: For each $i = 1,\ldots,N$, generate independent samples:

   $$\tilde{X}_t^{(i)} \sim f_{X_t|X_{t-1}=X_{t-1}^{(i)}}.$$

   To execute this step, we need to be able to simulate from the state transition density $f_{X_t|X_{t-1}=x_{t-1}}$.

2. **Weights**: For each $i = 1,\ldots,N$, compute

   $$w_t^{(i)} := f_{Y_t|X_t=\tilde{X}_t^{(i)}}(y_t). \tag{109}$$

   Normalize these weights so they sum to one:

   $$W_t^{(i)} = \frac{w_t^{(i)}}{w_t^{(1)}+\cdots+w_t^{(N)}} \qquad \text{for } i = 1,\ldots,N.$$

   To execute this step, we need to be able to evaluate the conditional density $f_{Y_t|X_t=x_t}(y_t)$.

3. **Resampling**: Generate

   $$X_t^{(1)},\ldots,X_t^{(N)} \overset{\text{i.i.d}}{\sim} \sum_{i=1}^{N} W_i \delta_{\{\tilde{X}_t^{(i)}\}}$$

This algorithm is initialized by taking

$$\tilde{X}_0^{(1)}, \ldots, \tilde{X}_0^{(N)} \overset{\text{i.i.d}}{\sim} f_{X_0|\theta}$$

and then following the steps 2 (weights) and 3 (resampling) above to generate $X_0^{(1)}, \ldots, X_0^{(N)}$. One can then repeat the recursion for $t = 1, 2, 3, \ldots$. This is similar to the Kalman Filter initialization.

The algorithm also allows computation of the likelihood $f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T)$ as:

$$f_{Y_0,\ldots,Y_T|\theta}(y_0, \ldots, y_T) = f_{Y_0|\theta}(y_0) \prod_{t=1}^{T} f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t) \approx \prod_{t=0}^{T} \left(\frac{1}{N} \sum_{i=1}^{N} w_t^{(i)} \right).$$

17.2 Unique Values and Particle Degeneracy

It is clear from the description of the algorithm that the samples $X_t^{(1)}, \ldots, X_t^{(N)}$ output by the Bootstrap Particle Filter are actually sampled from the discrete distribution:

$$\sum_{i=1}^{N} W_t^{(i)} \delta_{\{\tilde{X}_t^{(i)}\}}$$

An immediate implication of this is that $X_t^{(1)}, \ldots, X_t^{(N)}$ will not all be distinct and there will be repeats among them. A useful diagnostic here is the number of unique values N_t among $X_t^{(1)}, \ldots, X_t^{(N)}$. If N_t is particularly small for some t, the Monte Carlo approximation (108) will not be accurate. If N_t is small for some time indices t, then one says that the particle filter algorithm suffers from the problem of *Particle Degeneracy*.

The Bootstrap particle filter can suffer from particle degeneracy. To understand when this problem is particularly serious, observe first that, in the generation step, $\tilde{X}_t^{(1)}, \ldots, \tilde{X}_t^{(N)}$ can be seen as i.i.d samples from the one-step ahead prediction density:

$$f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}.$$

The weights $w_t^{(i)} = f_{Y_t|X_t=\tilde{X}_t^{(i)}}(y_t)$ satisfy

$$w_t^{(i)} \propto \frac{f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},Y_t=y_t,\theta}(\tilde{X}_t^{(i)})}{f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(\tilde{X}_t^{(i)})}$$

The two densities in play here are the proposal density given by $f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}$ and the target density given by $f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}$. The algorithm will not work well if these two densities are far from each other. Specifically, particle degeneracy occurs if the target density $f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t)$ is quite small when x_t belongs to the high-density regions of the proposal density $f_{X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_t)$. Note that the only difference between the proposal and target densities is the additional conditioning on $Y_t = y_t$ in the target density. Thus the proposal and target densities will be different if Y_t provides significantly more and different information about X_t beyond that already provided by $Y_0, \ldots, Y_{t-1}$. This tends to happen, for example, if the observation model relating Y_t to X_t has small errors. For example, in the local level model $Y_t = X_t + \epsilon_t$, if ϵ_t is small (e.g., when σ_ϵ is small), then Y_t is quite informative for X_t and, in such situations, the bootstrap particle filter algorithm suffers from particle degeneracy.

This also tends to happen for $t = 0$ when the proposal density is $f_{X_0|\theta}$ and the target density is $f_{X_0|Y_0=y_0,\theta}$. The proposal density is usually quite diffuse and the target density is relatively informative leading to small weights for most of the samples (and, consequently, small N_0).

In such situations where Y_t is quite informative about X_t, a natural fix is to change the proposal distribution by including information on Y_t. This is the idea underlying the Guided Particle Filter Algorithm.

17.3 The Guided Particle Filter Algorithm

The guided particle filter algorithm uses more general proposal distributions. For each time point $t \geq 0$, each value x in the space of the hidden variables $\{X_t\}$, and each value y in the space of the observation variables $\{Y_t\}$, let

$$u \mapsto q_t(u \mid x, y, \theta)$$

be an arbitrary density. The general algorithm described below works for any such set of densities $q_t(\cdot \mid x, y, \theta)$. The only requirement is that it should be possible simulate from this density. This general algorithm is known as the guided particle filter algorithm and an alternative name for the same algorithm is the *Sequential Importance Resampling* (SIR) algorithm. In order to apply this algorithm in an actual problem, it is necessary to specify $q_t(\cdot \mid x, y, \theta)$. For this, two choices are commonly used:

1. $q_t(u \mid x, y, \theta) := f_{X_t|X_{t-1}=x,\theta}(u)$. The following algorithm for this choice of q_t reduces to the Bootstrap Particle Filter algorithm. Therefore the SIR algorithm is a generalization of the Bootstrap Particle Filter. Note that this choice of $q_t(\cdot \mid, x, y, \theta)$ does not depend on y (it only depends on x).

2. $q_t(u, \mid x, y, \theta) := f_{X_t|X_{t-1}=x,Y_t=y,\theta}(u)$. This is commonly used as an alternative to the Bootstrap particle filter when the latter suffers from particle degeneracy. The use of this density in the SIR algorithm requires one to be able to simulate from the conditional density of X_t given $X_{t-1} = x, Y_t = y, \theta$.

The following is the SIR algorithm. As the Bootstrap particle filter algorithm, the goal is to output, for each $t \geq 0$, samples $X_t^{(1)}, \ldots, X_t^{(N)}$ such that

$$\text{Unif}\left\{X_t^{(1)}, \ldots, X_t^{(N)}\right\} \approx X_t \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta. \tag{110}$$

The algorithm proceeds sequentially. At time $t-1$, one has access to the samples $X_{t-1}^{(1)}, \ldots, X_{t-1}^{(N)}$ satisfying (114) for $t-1$ and using these, one generates the samples $X_t^{(1)}, \ldots, X_t^{(N)}$ by following the three steps given below.

1. **Generation**: For each $i = 1, \ldots, N$, generate independent samples:

$$\tilde{X}_t^{(i)} \sim q(\cdot \mid x = X_{t-1}^{(i)}, y = y_t, \theta)$$

   To execute this step, we obviously need to be able to simulate from $q(\cdot \mid x = X_{t-1}^{(i)}, y = y_t)$.

2. **Weights**: For each $i = 1, \ldots, N$, compute

$$w_t^{(i)} := \frac{f_{X_t|X_{t-1}=X_{t-1}^{(i)},\theta}(\tilde{X}_t^{(i)}) f_{Y_t|X_t=\tilde{X}_t^{(i)},\theta}(y_t)}{q_t(\tilde{X}_t^{(i)} \mid x = X_{t-1}^{(i)}, y = y_t, \theta)} \tag{111}$$

Normalize these weights so they sum to one:

$$W_t^{(i)} = \frac{w_t^{(i)}}{w_t^{(1)} + \cdots + w_t^{(N)}} \qquad \text{for } i = 1, \ldots, N.$$

To execute this step, we need to be able to evaluate $f_{X_t|X_{t-1}=x_{t-1}}(x_t)$ and $f_{Y_t|X_t=x_t}(y_t)$.

3. **Resampling**: Generate

$$X_t^{(1)}, \ldots, X_t^{(N)} \overset{\text{i.i.d}}{\sim} \sum_{i=1}^{N} W_i \delta_{\{\tilde{X}_t^{(i)}\}}$$

This algorithm is initialized by taking

$$X_0^{(1)}, \ldots, X_0^{(N)} \overset{\text{i.i.d}}{\sim} f_{X_0|Y_0=y_0}$$

and then repeating the three steps described above for $t = 1, 2, \ldots$.

The justification for the weights (115) is as follows. Note first that $(X_{t-1}^{(i)}, \tilde{X}_t^{(i)})$ for $i = 1, \ldots, N$ are i.i.d samples from the joint density:

$$(x_{t-1}, x_t) \mapsto f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) q_t(x_t \mid x_{t-1}, y_t, \theta).$$

The target should have, as its second marginal, the filtering density at time t. This suggests the target density:

$$(x_{t-1}, x_t) \mapsto f_{X_{t-1},X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{t-1}, x_t).$$

The importance weights will then be given by

$$\frac{c f_{X_{t-1},X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{t-1}, x_t)}{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) q_t(x_t \mid x_{t-1}, y_t, \theta)} \tag{112}$$

with $x_{t-1} = X_{t-1}^{(i)}$ and $x_t = \tilde{X}_t^{(i)}$. The above expression can be simplified using Bayes rule as

$$\begin{aligned}
&\frac{c f_{X_{t-1},X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{t-1}, x_t)}{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) q_t(x_t \mid x_{t-1}, y_t, \theta)} \\
&= \frac{c \frac{f_{X_{t-1},X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1},x_t) f_{Y_t|X_{t-1}=x_{t-1},X_t=x_t,\theta}(y_t)}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)}}{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) q_t(x_t \mid x_{t-1}, y_t, \theta)} \\
&= \frac{c}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)} \frac{f_{X_{t-1},X_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}, x_t)}{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) q_t(x_t \mid x_{t-1}, y_t, \theta)} f_{Y_t|X_t=x_t,\theta}(y_t) \\
&= \frac{c}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)} \frac{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t)}{f_{X_{t-1}|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(x_{t-1}) q_t(x_t \mid x_{t-1}, y_t, \theta)} f_{Y_t|X_t=x_t,\theta}(y_t) \\
&= \frac{c}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)} \frac{f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{Y_t|X_t=x_t,\theta}(y_t)}{q_t(x_t \mid x_{t-1}, y_t, \theta)}.
\end{aligned}$$

As a result, we can view the weights in the SIR algorithm as the weights given by (112) with $c = f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)$. This justifies the choice of weights in the SIR algorithm. Note also that because $c = f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)$, the average of the unnormalized weights provides an approximation of $f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)$:

$$\frac{1}{n} \sum_{i=1}^{N} w_t^{(i)} \approx f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t) \qquad \text{for each } t = 1, \ldots, T.$$

The product of these averages for $t = 1, \ldots, T$ (additionally multiplied by $f_{Y_0}(y_0)$) gives an approximation for the likelihood.

17.4 Weights when $q_t(u \mid x, y, \theta) := f_{X_t|X_{t-1}=x,Y_t=y,\theta}(u)$

As already remarked, the two most common choices of q_t in the SIR algorithm are $q_t(u \mid x, y, \theta) = f_{X_t|X_{t-1}=x}(u)$ (which corresponds to the bootstrap filter) and $q_t(u \mid x, y, \theta) = f_{X_t|X_{t-1}=x,Y_t=y}(u)$. The weights for the latter choice can be simplified (using Bayes rule in the denominator) as:

$$\begin{aligned}\frac{f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{Y_t|X_t=x_t,\theta}(y_t)}{q_t(x_t \mid x_{t-1}, y_t, \theta)} &= \frac{f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{Y_t|X_t=x_t,\theta}(y_t)}{f_{X_t|X_{t-1}=x_{t-1},Y_t=y_t,\theta}(x_t)} \\ &= \frac{f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{Y_t|X_t=x_t,\theta}(y_t)}{\frac{f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{Y_t|X_t=x_t,X_{t-1}=x_{t-1},\theta}(y_t)}{f_{Y_t|X_{t-1}=x_{t-1},\theta}(y_t)}} \\ &= \frac{f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{Y_t|X_t=x_t,\theta}(y_t)}{\frac{f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{Y_t|X_t=x_t,\theta}(y_t)}{f_{Y_t|X_{t-1}=x_{t-1},\theta}(y_t)}} = f_{Y_t|X_{t-1}=x_{t-1},\theta}(y_t).\end{aligned}$$

In other words, the weight corresponding to $(X_{t-1}^{(i)}, \tilde{X}_t^{(i)})$ for SIR with $q_t(u \mid x, y, \theta) = f_{X_t|X_{t-1}=x,Y_t=y}(u)$ is given by

$$w_t^{(i)} = f_{Y_t|X_{t-1}=X_{t-1}^{(i)},\theta}(y_t).$$

It is interesting to contrast this weight with the weight $f_{Y_t|X_t=\tilde{X}_t^{(i)},\theta}(y_t)$ used in the bootstrap particle filter.

17.5 Example: Local Level Model

As already remarked, the bootstrap particle filter is widely applicable because, in order to use it, one only needs to be able to simulate from the state transition density $f_{X_t|X_{t-1}=x_{t-1}}$ and be able to compute the density $f_{Y_t|X_t=x_t}(y_t)$. On the other hand, in order to apply the Guided Particle Filter algorithm with

$$q_t(u \mid x, y, \theta) := f_{X_t|X_{t-1}=x,Y_t=y,\theta}(u) \tag{113}$$

one should be able to simulate from $f_{X_t|X_{t-1}=x,Y_t=y,\theta}$ and evaluate $f_{Y_t|X_{t-1}=x_{t-1}}(y_t)$. While this may not always possible, here is a simple setting where the method can be easily applied. This is the case of the local level model:

$$X_0 \sim N(0, C) \qquad X_t = X_{t-1} + Z_t \qquad Y_t = X_t + \epsilon_t$$

where $X_0, Z_1, Z_2, \ldots, \epsilon_0, \epsilon_1, \ldots$ are independent with $Z_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_Z^2)$ and $\epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2)$. For this model, we have

$$X_t \mid X_{t-1} = x_{t-1}, \theta \sim N(x_{t-1}, \sigma_Z^2) \quad \text{and} \quad Y_t \mid X_t = x_t, X_{t-1} = x_{t-1}, \theta \sim N(x_t, \sigma_\epsilon^2)$$

from which it readily follows that

$$X_t \mid X_{t-1} = x_{t-1}, Y_t = y_t, \theta \sim N\left(\frac{\frac{x_{t-1}}{\sigma_Z^2} + \frac{y_t}{\sigma_\epsilon^2}}{\frac{1}{\sigma_Z^2} + \frac{1}{\sigma_\epsilon^2}}, \frac{1}{\frac{1}{\sigma_Z^2} + \frac{1}{\sigma_\epsilon^2}}\right).$$

Thus the Guided Particle Filter Algorithm with (113) is feasible in this case and the generation step simulates observations as:

$$\tilde{X}_t^{(i)} \sim N\left(\frac{\frac{X_{t-1}^{(i)}}{\sigma_Z^2} + \frac{y_t}{\sigma_\epsilon^2}}{\frac{1}{\sigma_Z^2} + \frac{1}{\sigma_\epsilon^2}}, \frac{1}{\frac{1}{\sigma_Z^2} + \frac{1}{\sigma_\epsilon^2}}\right).$$

We also have

$$Y_t \mid X_{t-1} = x_{t-1}, \theta \sim N(x_{t-1}, \sigma_Z^2 + \sigma_\epsilon^2)$$

so that the weights are computed as

$$w_t^{(i)} = \phi(y_t; X_{t-1}^{(i)}, \sigma_Z^2 + \sigma_\epsilon^2)$$

where $\phi(y; \mu, \sigma^2)$ denotes the normal density with mean μ and variance σ^2 evaluated at y. Initialization is done by generating observations from the distribution:

$$X_0 \mid Y_0 = y_0, \theta \sim N\left(\frac{y_0/\sigma_\epsilon^2}{1/C + 1/\sigma_\epsilon^2}, \frac{1}{1/C + 1/\sigma_\epsilon^2}\right).$$

It can easily be seen (in simulations) that when σ_ϵ^2 is small, the bootstrap particle filter suffers from particle degeneracy. The performance of the guided particle filter with (113) is much better.

17.6 Recommended Reading for Today

1. Good references for the SIR or Guided Particle Filter algorithms are:
   a) Section 5.1 of the Petris-Petrone-Campagnoli book
   b) Section 7.4 of the Särkkä book
   c) Section 6.7.3 of the Triantafyllopoulos book
   d) Sections 10.3.1 and 10.3.2 of the Chopin-Papaspiliopoulos (they derive these algorithms from a slightly more general viewpoint involving Feynman-Kac models which are described in Chapter 5 of their book)

18 Lecture Eighteen

18.1 Sequential Importance Resampling

In the last class, we looked at the Sequential Importance Resampling (SIR) algorithm (we also used the term "Guided Particle Filter") which generates, for each $t \geq 0$, samples $X_t^{(1)}, \ldots, X_t^{(N)}$ such that

$$\text{Unif}\left\{X_t^{(1)}, \ldots, X_t^{(N)}\right\} \approx X_t \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta. \tag{114}$$

The algorithm proceeds sequentially. At time $t-1$, one has access to the samples $X_{t-1}^{(1)}, \ldots, X_{t-1}^{(N)}$ satisfying (114) for $t-1$ and using these, one generates the samples $X_t^{(1)}, \ldots, X_t^{(N)}$ by following the three steps given below.

1. **Generation**: For each $i = 1, \ldots, N$, generate independent samples:

   $$\tilde{X}_t^{(i)} \sim q(\cdot \mid x = X_{t-1}^{(i)}, y = y_t, \theta)$$

   To execute this step, we obviously need to be able to simulate from $q(\cdot \mid x = X_{t-1}^{(i)}, y = y_t)$.

2. **Weights**: For each $i = 1, \ldots, N$, compute

$$w_t^{(i)} := \frac{f_{X_t|X_{t-1}=X_{t-1}^{(i)},\theta}(\tilde{X}_t^{(i)}) f_{Y_t|X_t=\tilde{X}_t^{(i)},\theta}(y_t)}{q_t(\tilde{X}_t^{(i)} \mid x = X_{t-1}^{(i)}, y = y_t, \theta)} \tag{115}$$

Normalize these weights so they sum to one:

$$W_t^{(i)} = \frac{w_t^{(i)}}{w_t^{(1)} + \cdots + w_t^{(N)}} \qquad \text{for } i = 1, \ldots, N.$$

To execute this step, we need to be able to evaluate $f_{X_t|X_{t-1}=x_{t-1}}(x_t)$ and $f_{Y_t|X_t=x_t}(y_t)$.

3. **Resampling**: Generate

$$X_t^{(1)}, \ldots, X_t^{(N)} \overset{\text{i.i.d}}{\sim} \sum_{i=1}^{N} W_i \delta_{\{\tilde{X}_t^{(i)}\}}$$

This algorithm is initialized by taking

$$X_0^{(1)}, \ldots, X_0^{(N)} \overset{\text{i.i.d}}{\sim} f_{X_0|Y_0=y_0}$$

and then repeating the three steps described above for $t = 1, 2, \ldots$.

The density $q(\cdot \mid x_{t-1}, y_t)$ appearing above is often referred to as a proposal density. The SIR algorithm works for pretty much any proposal density (the only requirement is the ability to simulate from it). The two most common choices of the proposal density are:

1. **Bootstrap Particle Filter**: $q(x_t \mid x_{t-1}, y_t) = f_{X_t|X_{t-1}=x_{t-1}}(x_t)$. Note that this choice of q does not depend on y_t. The weights corresponding to this proposal are $f_{Y_t|X_t=x_t}(y_t)$ i.e., $w_t^{(i)} = f_{Y_t|X_t=\tilde{X}_t^{(i)}}(y_t)$.

2. **"Optimal" Guided Particle Filter**: $q(x_t \mid x_{t-1}, y_t) = f_{X_t|X_{t-1}=x_{t-1},Y_t=y_t}(x_t)$. This algorithm is only feasible if it is possible to simulate from the conditional density of X_t given $X_{t-1} = x_{t-1}$ and $Y_t = y_t$. The reason why this choice of q for the Guided Particle Filter algorithm is called "optimal" can be found, for example, in Theorem 10.1 of the Chopin-Papaspiliopoulos book. We shall not make any use of this optimality criterion. The weights corresponding to this proposal are $f_{Y_t|X_{t-1}=x_{t-1}}(y_t)$ i.e., $w_t^{(i)} = f_{Y_t|X_{t-1}=X_{t-1}^{(i)}}(y_t)$. Note that these weights do not depend on the particles $\tilde{X}_t^{(i)}$ generated in this iterate of the algorithm.

The second algorithm above (optimal guided particle filter) usually suffers from less particle degeneracy compared to the Bootstrap particle filter because the function

$$x \mapsto f_{Y_t|X_{t-1}=x}(y_t)$$

is less concentrated compared to the function

$$x \mapsto f_{Y_t|X_t=x}(y_t).$$

18.2 Example: Local Level Model with non-Gaussian evolution errors

Consider the local level model:

$$X_t = X_{t-1} + Z_t \qquad \text{with } Z_t \overset{\text{i.i.d}}{\sim} (1-\alpha)N(0, \sigma_0^2) + \alpha N(0, \sigma_a^2)$$
$$Y_t = X_t + \epsilon_t \qquad \text{with } \epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2).$$

This model can be used to model trend functions that are piecewise constant. The parameter α and the variance σ_0^2 will both be small in such applications.

The Kalman filter is obviously not applicable here as the evolution error is non-Gaussian. The bootstrap filter algorithm is quite easy to implement: in the generation step, the challenge is to simulate

$$\tilde{x}_t \sim (1-\alpha)N(x_{t-1}, \sigma_0^2) + \alpha N(x_{t-1}, \sigma_a^2)$$

for $x_{t-1} = X_{t-1}^{(i)}$. This is a mixture of Gaussian distributions. One can simulate from this mixture by first simulating a Bernoulli random variable B with success probability α. If $\alpha = 1$, then one would simulate $\tilde{x}_t$ from $N(x_{t-1}, \sigma_a^2)$ and if $\alpha = 1$, then one would simulate $\tilde{x}_t$ from $N(x_{t-1}, \sigma_0^2)$. The weights are given by

$$w_t = f_{Y_t|X_t=\tilde{x}_t}(y_t) = \phi(y_t; \tilde{x}_t, \sigma_\epsilon^2) \tag{116}$$

where $\phi(x; \mu, \sigma^2)$ stands for the normal density with mean μ and variance σ^2. Note that if y_t is far from $\tilde{x}_t$ and σ_ϵ is relatively small, then there will be particle degeneracy. Also note that, when the parameters α and σ_0^2 are small, most ($(1-\alpha)$ fraction) of the generated observations $\tilde{x}_t$ will be close to x_{t-1}.

Now let us consider applying the optimal guided particle filter for this model. Particle generation will have to be done from the conditional density $f_{X_t|X_{t-1}=x_{t-1},Y_t=y_t}$. By Bayes rule:

$$\begin{aligned}
f_{X_t|X_{t-1}=x_{t-1},Y_t=y_t}(x_t) &\propto f_{X_t|X_{t-1}=x_{t-1}}(x_t) f_{Y_t|X_t=x_t,X_{t-1}=x_{t-1}}(y_t) \\
&= f_{X_t|X_{t-1}=x_{t-1}}(x_t) f_{Y_t|X_t=x_t}(y_t) \\
&= \left\{(1-\alpha)\phi(x_t; x_{t-1}, \sigma_0^2) + \alpha\phi(x_t; x_{t-1}, \sigma_a^2)\right\} \phi(y_t; x_t, \sigma_\epsilon^2) \\
&= (1-\alpha)\phi(x_t; x_{t-1}, \sigma_0^2)\phi(y_t; x_t, \sigma_\epsilon^2) + \alpha\phi(x_t; x_{t-1}, \sigma_a^2)\phi(y_t; x_t, \sigma_\epsilon^2).
\end{aligned}$$

We now use the following elementary identity (which can be proved by direct calculation): For every $\theta, x, \mu \in (-\infty, \infty)$ and $\tau, \sigma > 0$, we have

$$\phi(\theta; \mu, \tau^2)\phi(x; \theta, \sigma^2) = \phi\left(\theta; \frac{x/\sigma^2 + \mu/\tau^2}{1/\sigma^2 + 1/\tau^2}, \frac{1}{1/\sigma^2 + 1/\tau^2}\right) \phi(x; \mu, \tau^2 + \sigma^2). \tag{117}$$

We get

$$\begin{aligned}
f_{X_t|X_{t-1}=x_{t-1},Y_t=y_t}(x_t) \propto\ & (1-\alpha)\phi\left(x_t; \frac{y_t/\sigma_\epsilon^2 + x_{t-1}/\sigma_0^2}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}, \frac{1}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}\right) \phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2) \\
& + \alpha\phi\left(x_t; \frac{y_t/\sigma_\epsilon^2 + x_{t-1}/\sigma_a^2}{1/\sigma_\epsilon^2 + 1/\sigma_a^2}, \frac{1}{1/\sigma_\epsilon^2 + 1/\sigma_a^2}\right) \phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_a^2).
\end{aligned}$$

The integral of the right hand side above with respect to x_t is

$$(1-\alpha)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2) + \alpha\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_a^2).$$

As a result

$$\begin{aligned}
& f_{X_t|X_{t-1}=x_{t-1},Y_t=y_t}(x_t) \\
&= \frac{(1-\alpha)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2)}{(1-\alpha)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2) + \alpha\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_a^2)} \phi\left(x_t; \frac{y_t/\sigma_\epsilon^2 + x_{t-1}/\sigma_0^2}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}, \frac{1}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}\right) \\
&+ \frac{\alpha\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_a^2)}{(1-\alpha)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2) + \alpha\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_a^2)} \phi\left(x_t; \frac{y_t/\sigma_\epsilon^2 + x_{t-1}/\sigma_a^2}{1/\sigma_\epsilon^2 + 1/\sigma_a^2}, \frac{1}{1/\sigma_\epsilon^2 + 1/\sigma_a^2}\right).
\end{aligned}$$

This is just a mixture of two normal distributions so one can simulate observations from it by first generating a Bernoulli random variable B with success probability:

$$\frac{\alpha\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_a^2)}{(1-\alpha)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2) + \alpha\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_a^2)}$$

If $B = 1$, one simulates from

$$N\left(\frac{y_t/\sigma_\epsilon^2 + x_{t-1}/\sigma_a^2}{1/\sigma_\epsilon^2 + 1/\sigma_a^2}, \frac{1}{1/\sigma_\epsilon^2 + 1/\sigma_a^2}\right),$$

and if $B = 0$, one simulates from

$$N\left(\frac{y_t/\sigma_\epsilon^2 + x_{t-1}/\sigma_0^2}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}, \frac{1}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}\right).$$

Finally note that

$$f_{Y_t|X_{t-1}=x_{t-1}}(y_t) = (1-\alpha)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2) + \alpha\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_a^2)$$

which is useful for weight calculation. Note that, as a function of x_{t-1}, the right hand side above is more diffuse compared to the weight function in the Bootstrap filter (116). This implies that, in this example, the optimal guided particle filter suffers from less particle degeneracy compared to the Bootstrap particle filter.

It should be noted that it is not always possible to implement the optimal guided particle filter in closed form (as in the above example). For example, consider the local level model again where the $N(0, \sigma_a^2)$ distribution in the evolution error is replaced by the standard Cauchy distribution:

$$\begin{aligned} X_t &= X_{t-1} + Z_t \qquad \text{with } Z_t \overset{\text{i.i.d}}{\sim} (1-\alpha)N(0, \sigma_0^2) + \alpha C(0,1) \\ Y_t &= X_t + \epsilon_t \qquad \text{with } \epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_\epsilon^2) \end{aligned}$$

where $C(0,1)$ denotes the standard Cauchy distribution with density proportional to $(1 + x^2)^{-1}$. In this case, $f_{X_t|X_{t-1}=x_{t-1}, Y_t=y_t}$ is given by

$$\begin{aligned} f_{X_t|X_{t-1}=x_{t-1}, Y_t=y_t}(x_t) &\propto f_{X_t|X_{t-1}=x_{t-1}}(x_t) f_{Y_t|X_t=x_t}(y_t) \\ &= \left\{(1-\alpha)\phi(x_t; x_{t-}, \sigma_0^2) + \alpha\gamma(x_t; x_{t-1})\right\}\phi(y_t; x_t, \sigma_\epsilon^2) \\ &= (1-\alpha)\phi(x_t; x_{t-}, \sigma_0^2)\phi(y_t; x_t, \sigma_\epsilon^2) + \alpha\gamma(x_t; x_{t-1})\phi(y_t; x_t, \sigma_\epsilon^2) \end{aligned}$$

where $\gamma(x_t; x_{t-1})$ is the density of a Cauchy random variable centered at x_{t-1} (and scale parameter equal to 1):

$$\gamma(x; \mu) = \frac{1}{\pi}\frac{1}{1 + (x-\mu)^2}.$$

Using the fact (117), we obtain

$$\begin{aligned} &f_{X_t|X_{t-1}=x_{t-1}, Y_t=y_t}(x_t) \\ &\propto (1-\alpha)\phi\left(x_t; \frac{y_t/\sigma_\epsilon^2 + x_{t-1}/\sigma_0^2}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}, \frac{1}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}\right)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2) + \alpha\gamma(x_t; x_{t-1})\phi(y_t; x_t, \sigma_\epsilon^2). \end{aligned}$$

Letting

$$V(y; \mu, \sigma^2) := \int \gamma(x; \mu)\phi(y; x, \sigma^2)dx,$$

we can write

$$
\begin{aligned}
&f_{X_t|X_{t-1}=x_{t-1},Y_t=y_t}(x_t) \\
&\propto (1-\alpha)\phi\left(x_t; \frac{y_t/\sigma_\epsilon^2 + x_{t-1}/\sigma_0^2}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}, \frac{1}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}\right)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2) \\
&+ \alpha\left[\frac{\gamma(x_t; x_{t-1})\phi(y_t; x_t, \sigma_\epsilon^2)}{V(y_t; x_{t-1}, \sigma_\epsilon^2)}\right] V(y_t; x_{t-1}, \sigma_\epsilon^2).
\end{aligned}
$$

The integral of the right hand side above with respect to x_t equals

$$(1-\alpha)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2) + \alpha V(y_t; x_{t-1}, \sigma_\epsilon^2)$$

so that

$$
\begin{aligned}
&f_{X_t|X_{t-1}=x_{t-1},Y_t=y_t}(x_t) \\
&= \frac{(1-\alpha)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2)}{(1-\alpha)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2) + \alpha V(y_t; x_{t-1}, \sigma_\epsilon^2)}\phi\left(x_t; \frac{y_t/\sigma_\epsilon^2 + x_{t-1}/\sigma_0^2}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}, \frac{1}{1/\sigma_\epsilon^2 + 1/\sigma_0^2}\right) \\
&+ \frac{\alpha V(y_t; x_{t-1}, \sigma_\epsilon^2)}{(1-\alpha)\phi(y_t; x_{t-1}, \sigma_\epsilon^2 + \sigma_0^2) + \alpha V(y_t; x_{t-1}, \sigma_\epsilon^2)}\left[\frac{\gamma(x_t; x_{t-1})\phi(y_t; x_t, \sigma_\epsilon^2)}{V(y_t; x_{t-1}, \sigma_\epsilon^2)}\right].
\end{aligned}
$$

The density function $y \mapsto V(y; \mu, \sigma^2)$ is known as the Voigt profile (see `https://en.wikipedia.org/wiki/Voigt_profile`) and efficient algorithms exist for its computation. In order to simulate $\tilde{x}_t$ from the above conditional density, the main challenge is to simulate from the conditional density:

$$x_t \mapsto \frac{\gamma(x_t; x_{t-1})\phi(y_t; x_t, \sigma_\epsilon^2)}{V(y_t; x_{t-1}, \sigma_\epsilon^2)}. \tag{118}$$

It is not clear if this can be done in closed form. One can use some numerical techniques for this. For example, a straightforward approach is to use discretization: one can discretize the domain and approximate the continuous distribution with density given by (118) by a discrete distribution supported on the discrete set of values. One can then simulate from the discrete distribution.

Note that the filter algorithms can be used for obtaining the likelihood (which is the joint density of $Y_0, \ldots, Y_T$ given the parameters) which can be used for maximum likelihood estimation of the parameters.

18.3 Recommended Reading for Today

1. Good references for the SIR or Guided Particle Filter algorithms are:
   a) Section 5.1 of the Petris-Petrone-Campagnoli book
   b) Section 7.4 of the Särkkä book
   c) Section 6.7.3 of the Triantafyllopoulos book
   d) Sections 10.3.1 and 10.3.2 of the Chopin-Papaspiliopoulos (they derive these algorithms from a slightly more general viewpoint involving Feynman-Kac models which are described in Chapter 5 of their book)
2. The local level model with non-Gaussian errors for estimating piecewise constant trend functions is discussed in Section 15.2.6 of the Kitagawa book, and in Section 8.4 of the Kitagawa-Gersch book.

19 Lecture Nineteen

The goal today is to study Monte Carlo methodology for smoothing in general state space models. We shall focus on two smoothing algorithms:

1. Complete Smoothing: This method is simple and is just an extension of the particle filter algorithm. However it generally suffers from particle degeneracy.
2. FFBS: This is based on general smoothing ideas that we previously saw in the context of linear Gaussian models. The method works well but is computationally intensive.

19.1 Complete Smoothing

This algorithm is an extension of particle filtering (and can also be termed complete filtering). It generates, for each $t \geq 0$, samples

$$(X_{0|t}^{(i)}, X_{1|t}^{(i)}, \ldots, X_{t|t}^{(i)}), 1 \leq i \leq N$$

such that the discrete uniform distribution over these samples approximates the smoothing distribution at time t:

$$\frac{1}{N}\sum_{i=1}^{N} \delta_{\{(X_{0|t}^{(i)}, X_{1|t}^{(i)}, \ldots, X_{t|t}^{(i)})\}} \approx (X_0, \ldots, X_t) \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta \tag{119}$$

for each $t = 0, \ldots, T$.

The algorithm proceeds sequentially over time $t = 0, 1, \ldots, T$. At time $t-1$, one has access to samples $(X_{0|t-1}^{(i)}, X_{1|t-1}^{(i)}, \ldots, X_{t-1|t-1}^{(i)}), 1 \leq i \leq N$ satisfying (119) for time $t-1$ and using these, one generates the samples $(X_{0|t}^{(i)}, X_{1|t}^{(i)}, \ldots, X_{t|t}^{(i)}), 1 \leq i \leq N$ by following the three steps given below.

1. **Generation**: For each $i = 1, \ldots, N$, let

   $$\tilde{X}_{0|t}^{(i)} = X_{0|t-1}^{(i)}, \ldots, \tilde{X}_{t|t}^{(i)} = X_{t-1|t-1}^{(i)}$$

   and

   $$\tilde{X}_{t|t}^{(i)} \sim q(\cdot \mid x = X_{t-1|t-1}^{(i)}, y = y_t, \theta).$$

2. **Weights**: For each $i = 1, \ldots, N$, compute

   $$w_t^{(i)} := \frac{f_{X_t|X_{t-1}=X_{t-1|t-1}^{(i)},\theta}(\tilde{X}_{t|t}^{(i)}) f_{Y_t|X_t=\tilde{X}_{t|t}^{(i)},\theta}(y_t)}{q_t(\tilde{X}_{t|t}^{(i)} \mid x = X_{t-1|t-1}^{(i)}, y = y_t, \theta)} \tag{120}$$

   Normalize these weights so they sum to one:

   $$W_t^{(i)} = \frac{w_t^{(i)}}{w_t^{(1)} + \cdots + w_t^{(N)}} \qquad \text{for } i = 1, \ldots, N.$$

3. **Resampling**: Generate

   $$(X_{0|t}^{(i)}, X_{1|t}^{(i)}, \ldots, X_{t|t}^{(i)}), 1 \leq i \leq N \overset{\text{i.i.d}}{\sim} \sum_{i=1}^{N} W_t^{(i)} \delta_{\{(\tilde{X}_{0|t}^{(i)}, \ldots, \tilde{X}_{t|t}^{(i)})\}}.$$

The following are useful things to know about this algorithm:

1. The weights in (120) can be deduced from importance sampling. To see this, note that the generation of $(\tilde{X}_{0|t}^{(i)}, \ldots, \tilde{X}_{t|t}^{(i)})$ is from (approximately) the density:

$$f_{t-1|t-1}(x_0, \ldots, x_{t-1}) q(x_t \mid x_{t-1}, y_t, \theta).$$

   where we are using the notation:

$$f_{s|t}(x_0, \ldots, x_s) := f_{X_0,\ldots,X_s|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_0, \ldots, x_s).$$

   On the other hand, the target density equals

$$\begin{aligned} f_{t|t}(x_0, \ldots, x_t) &= f_{X_0,\ldots,X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_0, \ldots, x_t) \\ &= \frac{f_{t|t-1}(x_0, \ldots, x_{t-1}) f_{Y_t|X_t=x_t,\theta}(y_t)}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)} \\ &= \frac{f_{t-1|t-1}(x_0, \ldots, x_{t-1}) f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{Y_t|X_t=x_t,\theta}(y_t)}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)} \end{aligned}$$

   where we used Bayes rule in the second step. The importance weights are therefore given by

$$\begin{aligned} &\frac{f_{t|t}(x_0, \ldots, x_t)}{f_{t-1|t-1}(x_0, \ldots, x_{t-1}) q(x_t \mid x_{t-1}, y_t, \theta)} \\ &= \frac{f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{Y_t|X_t=x_t,\theta}(y_t)}{q(x_t \mid x_{t-1}, y_t, \theta)} \frac{1}{f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)} \end{aligned}$$

   Note that second term above (inverse of $f_{Y_t|Y_0=y_0,\ldots,Y_{t-1}=y_{t-1},\theta}(y_t)$) is a constant as it does not depend on x_{t-1} or x_t. We can thus view the importance weight as simply

$$\frac{f_{X_t|X_{t-1}=x_{t-1},\theta}(x_t) f_{Y_t|X_t=x_t,\theta}(y_t)}{q(x_t \mid x_{t-1}, y_t, \theta)}.$$

   This is exactly the importance weight $w_t^{(i)}$ in (120) with $x_{t-1} = X_{t-1|t-1}^{(i)}$ and $x_t = \tilde{X}_{t|t}^{(i)}$.

2. This algorithm needs to be initialized with samples from $X_0 \mid Y_0 = y_0, \theta$:

$$X_{0|0}^{(1)}, \ldots, X_{0|0}^{(N)}.$$

   After the first iteration, it outputs samples:

$$\begin{aligned} &X_{0|1}^{(1)}, \ldots, X_{0|1}^{(N)} \\ &X_{1|1}^{(1)}, \ldots, X_{1|1}^{(N)}. \end{aligned}$$

   Note that $X_{0|1}^{(1)}, \ldots, X_{0|1}^{(N)}$ is a resample drawn from the initial sample $X_{0|0}^{(1)}, \ldots, X_{0|0}^{(N)}$. After the second iteration, the algorithm outputs samples:

$$\begin{aligned} &X_{0|2}^{(1)}, \ldots, X_{0|2}^{(N)} \\ &X_{1|2}^{(1)}, \ldots, X_{1|2}^{(N)} \\ &X_{2|2}^{(1)}, \ldots, X_{2|2}^{(N)} \end{aligned}$$

Here $X_{0|2}^{(1)}, \ldots, X_{0|2}^{(N)}$ is a resample of $X_{0|1}^{(1)}, \ldots, X_{0|1}^{(N)}$ which was already a resample from $X_{0|0}^{(1)}, \ldots, X_{0|0}^{(N)}$. Also $X_{1|2}^{(1)}, \ldots, X_{1|2}^{(N)}$ is a resample from $X_{1|1}^{(1)}, \ldots, X_{1|1}^{(N)}$. One then proceeds iteratively ending in the final step where the algorithm ouputs:

$$\begin{array}{c} X_{0|T}^{(1)}, \ldots, X_{0|T}^{(N)} \\ X_{1|T}^{(1)}, \ldots, X_{1|T}^{(N)} \\ \ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots \\ X_{T|T}^{(1)}, \ldots, X_{T|T}^{(N)} \end{array} \tag{121}$$

The columns of the above output (121) in the final iteration of the algorithm are samples from the smoothing distribution of interest: $(X_0, \ldots, X_T) \mid Y_0 = y_0, \ldots, Y_T = y_T, \theta$.

3. This algorithm is very similar to the Sequential Importance Resampling (SIR) algorithm from the last couple of lectures for filtering. Indeed, if we keep track of only the filtering samples i.e., the samples

$$X_{t|t}^{(1)}, \ldots X_{t|t}^{(N)}$$

for $t = 0, 1, \ldots, T$, and ignore the set of time indices $s \mid t$ for $s < t$, we get back the SIR algorithm.

4. **Particle Degeneracy**: This algorithm suffers from serious particle degeneracy. Specifically, the number of unique values N_t among $X_{t|T}^{(1)}, \ldots, X_{t|T}^{(N)}$ can be much smaller than N and this is especially true for small values of t. This is because the samples $X_{t|t}^{(1)}, \ldots, X_{t|t}^{(N)}$ are created in the t^{th} iteration and the subsequent samples

$$X_{t|s}^{(1)}, \ldots, X_{t|s}^{(N)} \qquad \text{s = t+1, } \ldots, \text{ T}$$

are all obtained by resampling from $X_{t|t}^{(1)}, \ldots, X_{t|t}^{(N)}$ with various choices of weights. This means that $X_{t|T}^{(1)}, \ldots, X_{t|T}^{(N)}$ are obtained after resampling $T-t$ times from $X_{t|t}^{(1)}, \ldots, X_{t|t}^{(N)}$. Every resampling leads to a decrease in the effective sample size, and thus if $T - t$ is large (which will be the case for small t), the number of unique samples will be much smaller than N.

Computational Complexity: The complexity of this algorithm is $O(NT)$. This is because in each iteration of the algorithm, $O(N)$ computations are done (note that weights need to be calculated for each of the generated samples). The final complexity is therefore $O(NT)$ as there are T iterations.

19.2 FFBS

This algorithm is similar to the FFBS (Forward Filtering Backward Sampling) algorithms that we studied previously for linear Gaussian state space models (we also previously looked at a numerical version of FFBS for general state space models).

The goal of FFBS is to generate M samples:

$$(X_{0|T}^{(i)}, X_{1|T}^{(i)}, \ldots, X_{T|T}^{(i)}) \qquad \text{for } i = 1, \ldots, M$$

form the conditional distribution

$$(X_0, \ldots, X_T) \mid Y_0 = y_0, \ldots, Y_T = y_T, \theta.$$

Note that we are using the notation M for the number of smoothing samples.

The first step in FFBS is to run a particle filtering algorithm. This will result in samples:

$$\mathcal{X}_{t|t}^{(1)}, \ldots, \mathcal{X}_{t|t}^{(N)} \tag{122}$$

for each $t = 0, \ldots, T$. The discrete uniform distribution over the samples (122) approximates the filtering distribution $X_t \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ for each $t = 0, 1, \ldots, T$. Note that, because of the resampling steps that are used in particle filtering algorithms, there need not be any connection between $\mathcal{X}_{t-1|t-1}^{(i)}$ and $\mathcal{X}_{t|t}^{(i)}$ for fixed i (in the notation of our filtering algorithms, $\mathcal{X}_{t-1|t-1}^{(i)}$ and $\tilde{\mathcal{X}}_t^{(i)}$ would be related but there won't be any connection between $\tilde{\mathcal{X}}_t^{(i)}$ and $\mathcal{X}_{t|t}^{(i)}$ because of resampling). I am using the notation $\mathcal{X}$ (instead of the usual X) for the filtering particles to distinguish them from the smoothing samples which will be subsequently generated by FFBS.

Observe that we have N filtering samples for each time t. This N can be distinct from the desired number M of smoothing samples.

The step of running a particle filter algorithm represents the "FF" part of FFBS. We shall now describe the "BS" (Backward Sampling) part of the algorithm. Here, for each $i = 1, \ldots, M$, we shall generate samples

$$X_{T|T}^{(i)}, X_{T-1|T}^{(i)}, \ldots, X_{0|T}^{(i)}$$

in backward order for $t = T, \ldots, 0$. The first sample $X_{T|T}^{(i)}$ is just drawn from the discrete filtering approximation for $t = T$ (this is because the filtering and smoothing marginal distributions for $t = T$ coincide):

$$X_{T|T}^{(i)} \sim \text{Unif}\left\{\mathcal{X}_{T|T}^{(1)}, \ldots, \mathcal{X}_{T|T}^{(N)}\right\} = \frac{1}{N}\sum_{j=1}^{N} \delta_{\{\mathcal{X}_{T|T}^{(j)}\}}.$$

The recursive process for obtaining the subsequent samples $X_{T-1|T}^{(i)}, \ldots, X_{0|T}^{(i)}$ is described next. To go from $X_{t+1|T}^{(i)}$ to $X_{t|T}^{(i)}$, we would need to generate from the conditional density (below $x_{t+1} := X_{t+1|T}^{(i)}$)

$$\begin{aligned} f_{X_t|X_{t+1}=x_{t+1},Y_0=y_0,\ldots,Y_T=y_T,\theta}(x_t) &= f_{X_t|X_{t+1}=x_{t+1},Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t) \\ &= \frac{f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t) f_{X_{t+1}|X_t=x_t,\theta}(x_{t+1})}{f_{X_{t+1}|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{t+1})}. \end{aligned}$$

A natural idea of generating $X_{t|T}^{(i)}$ is to therefore use importance sampling where we first generate from a proposal density $q(x_t \mid x_{t+1}, \text{data}, \theta)$ and then use the weight:

$$\begin{aligned} \text{weight}(x_t) &= \frac{f_{X_t|X_{t+1}=x_{t+1},Y_0=y_0,\ldots,Y_T=y_T,\theta}(x_t)}{q(x_t \mid x_{t+1}, \text{data}, \theta)} \\ &= \frac{f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t) f_{X_{t+1}|X_t=x_t,\theta}(x_{t+1})}{q(x_t \mid x_{t+1}, \text{data}, \theta)} \frac{1}{f_{X_{t+1}|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_{t+1})} \\ &\propto \frac{f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t) f_{X_{t+1}|X_t=x_t,\theta}(x_{t+1})}{q(x_t \mid x_{t+1}, \text{data}, \theta)}. \end{aligned} \tag{123}$$

The proportionality sign above is in terms of x_t (factors not depending on x_t can be taken as part of the proportionality).

Any proposal density $q(x_t \mid x_{t+1}, \text{data}, \theta)$ can be used for this purpose. However, it is especially convenient to take it as the filtering density at time t:

$$q(x_t \mid x_{t+1}, \text{data}, \theta) = f_{X_t|Y_0=y_0,\ldots,Y_t=y_t,\theta}(x_t) \tag{124}$$

for the following two reasons:

1. We already have access to samples $\mathcal{X}_{t|t}^{(1)}, \ldots, \mathcal{X}_{t|t}^{(N)}$ from the filtering density at time t because of implementing a filtering algorithm in the first step of FFBS.
2. With the choice (124), the weights in (123) become quite simple:

$$\text{weight}(x_t) \propto f_{X_{t+1}|X_t=x_t,\theta}(x_{t+1}).$$

With the choice (124), the method for generating $X_{t|T}^{(i)}$ from $X_{t+1|T}^{(i)}$ becomes:

$$X_{t|T}^{(i)} \sim \sum_{j=1}^{N} W_j \delta_{\{\mathcal{X}_{t|t}^{(j)}\}} \tag{125}$$

where

$$W_j = \frac{w_j}{w_1 + \ldots w_N} \qquad \text{and } w_j = f_{X_{t+1}|X_t=\mathcal{X}_{t|t}^{(j)},\theta}(X_{t+1|T}^{(i)}).$$

(125) just means that $X_{t|T}^{(i)}$ is just sampled from the discrete distribution that is concentrated on the filtering samples $\mathcal{X}_{t|t}^{(1)}, \ldots, \mathcal{X}_{t|t}^{(N)}$ with weights $w_1, \ldots, w_N$.

The overall FFBS algorithm is therefore:

1. **Filtering**: Run a particle filter algorithm to generate samples $\mathcal{X}_{t|t}^{(1)}, \ldots, \mathcal{X}_{t|t}^{(N)}$ which approximate the filtering distribution $X_t \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ at each time $t = 0, 1, \ldots, T$.
2. Repeat the following for $i = 1, \ldots, M$
   a) **Initialization for Backward Recursion**: Draw one sample $X_{T|T}^{(i)}$ from the discrete uniform distribution on $\mathcal{X}_{T|T}^{(1)}, \ldots, \mathcal{X}_{T|T}^{(N)}$.
   b) **Backward Recursion**: Repeat the following $t = T-1, \ldots, 0$:
      i. Calculate weights $w_j = f_{X_{t+1}|X_t=\mathcal{X}_{t|t}^{(j)},\theta}(X_{t+1|T}^{(i)})$ for each $j = 1, \ldots, N$. Normalize these weights to obtain $W_1, \ldots, W_N$ which sum to one.
      ii. Generate $X_{t|T}^{(i)}$ from the discrete distribution on $\mathcal{X}_{t|t}^{(1)}, \ldots, \mathcal{X}_{t|t}^{(N)}$ with probabilities $W_1, \ldots, W_N$.

19.3 Recommended Reading for Today

1. The complete smoothing algorithm is described in Section 15.3 of the Kitagawa book, Section 11.1 of the Särkkä book, and Section 12.1.2 of the Chopin-Papaspiliopoulos book.

2. The FFBS algorithm is described in Section 12.3.2 of the Chopin-Papaspiliopoulos book, and in Section 11.2 of the Särkkä book (Särkkä calls it the Backward-Simulation Particle Smoother algorithm).

20 Lecture Twenty

20.1 Recap: Complete Smoothing

In the last class, we studied two algorithms for smoothing in general state space models. These algorithms produce samples

$$(X_{0|T}^{(i)}, \ldots, X_{T|T}^{(i)}), 1 \leq i \leq M$$

which approximate the smoothing distribution $(X_0, \ldots, X_T) \mid Y_0 = y_0, \ldots, Y_T = y_T, \theta$. The first of these algorithms was the complete smoothing algorithm (also known as the SIR-PS: Sequential Importance Resampling Particle Smoother). This algorithm works in the following way:

1. Draw M samples $X_0^{(1)}, \ldots, X_0^{(M)}$ from the conditional distribution of $X_0 \mid Y_0 = y_0, \theta$.
2. Repeat the following for each $t = 1, \ldots, T$:
   a) Draw M new samples $X_t^{(1)}, \ldots, X_t^{(M)}$ from the importance distribution:
   $$X_t^{(i)} \overset{\text{independent}}{\sim} q(\cdot \mid x_{t-1} = X_{t-1}^{(i)}, Y_t = y_t, \theta) \qquad \text{for } i = 1, \ldots, M.$$
   b) Calculate weights
   $$w_t^{(i)} = \frac{f_{Y_t|X_t=X_t^{(i)}}(y_t) f_{X_t|X_{t-1}=X_{t-1}^{(i)}}(X_t^{(i)})}{q(X_t^{(i)} \mid X_{t-1}^{(i)}, y_t)} \qquad \text{for } i = 1, \ldots, M.$$
   Renormalize these weights to obtain $W_t^{(i)}, i = 1, \ldots, M$ which sum to one.
   c) Append the samples to the state histories:
   $$X_{0:t}^{(i)} = \left(X_{0:t-1}^{(i)}, X_t^{(i)}\right).$$
   d) Resample from state trajectories $X_{0:t}^{(1)}, \ldots, X_{0:t}^{(M)}$ with probabilities $W_t^{(1)}, \ldots, W_t^{(M)}$.

The above description of the complete smoothing algorithm is slightly different from that given in the previous class but algorithm is exactly the same. Its computational complexity is $O(MT)$.

20.2 Complete Smoothing with partial trajectory resampling

The main problem with the complete smoothing algorithm is particle degeneracy. Specifically, for the samples $(X_0^{(i)}, \ldots, X_T^{(i)}), 1 \leq i \leq M$ obtained in the final iteration, the number of unique values among $X_t^{(1)}, \ldots, X_t^{(M)}$ will be quite small (compared to M) especially for

small values of t. This is because a fixed number of particles (M) are repeatedly being resampled. One, somewhat adhoc, fix to this problem is to resample, instead of the full state trajectories $X_{0:t}^{(1)}, \ldots, X_{0:t}^{(M)}$, just the trajectories

$$X_{t-L:t}^{(1)}, \ldots, X_{t-L:t}^{(M)}$$

for some fixed L. Of course, here we are assuming that $t \geq L$ (if $t < L$, the resamping is done as before from the full trajectories). This method fixes the particle degeneracy issue and the marginal samples $X_t^{(1)}, \ldots, X_t^{(M)}$ will have distribution which is nearly the same as $X_t \mid Y_0 = y_0, \ldots, Y_T = y_T, \theta$ (if L is not too small). However, the final trajectories $X_{0:T}^{(1)}, \ldots, X_{0:T}^{(M)}$ can no longer be treated as samples from $X_0, \ldots, X_T \mid Y_0 = y_0, \ldots, Y_T = y_T, \theta$.

Another way of thinking about this partial trajectory resampling fix is the following. The main problem with the complete smoothing algorithm is that it gives poor approximation, due to particle degeneracy, to the smoothing distributions of X_s (given $Y_0 = y_0, \ldots, Y_T = y_T, \theta$) when s is small. More generally, the samples will provide a poor approximation to the joint distribution:

$$X_{s_1}, X_{s_2}, \ldots, X_{s_k} \mid Y_0 = y_0, \ldots, Y_T = y_T, \theta \tag{126}$$

when $s_1 < \cdots < s_k$ and s_k is small. One heuristic way to obtain better approximation of this joint density is as follows. First reason that

$$\begin{aligned} &X_{s_1}, X_{s_2}, \ldots, X_{s_k} \mid Y_0 = y_0, \ldots, Y_T = y_T, \theta \\ &\approx X_{s_1}, X_{s_2}, \ldots, X_{s_k} \mid Y_0 = y_0, \ldots, Y_{s_k+L} = y_{s_k+L}, \theta \end{aligned} \tag{127}$$

for some fixed L that is much smaller than $T - s_k$. The idea is that the observation values $Y_t = y_t$ for t larger than $s_k + L$ probably do not have much influence on the distribution of $X_{s_1}, \ldots, X_{s_k}$. Under the approximation (127), the full smoothing distribution (126) can therefore be obtained by the right hand side of (127) which is well-approximated by the complete smoothing algorithm at iteration $s_k + L$. In other words, we don't need to run the complete smoothing algorithm till iteration T to approximate (126). The amount of resampling at iteration $s_k + L$ will be much smaller than until time T and this will lead to much less particle degeneracy. It is tricky however to choose an appropriate value of L (one usually just takes an arbitrary value such as $L = 30$).

20.3 Recap: FFBS

The FFBS algorithm is:

1. **Filtering**: Run a particle filter algorithm to generate samples $\mathcal{X}_{t|t}^{(1)}, \ldots, \mathcal{X}_{t|t}^{(N)}$ which approximate the filtering distribution $X_t \mid Y_0 = y_0, \ldots, Y_t = y_t, \theta$ at each time $t = 0, 1, \ldots, T$.
2. Repeat the following for $i = 1, \ldots, M$
   a) **Initialization for Backward Recursion**: Draw one sample $X_{T|T}^{(i)}$ from the discrete uniform distribution on $\mathcal{X}_{T|T}^{(1)}, \ldots, \mathcal{X}_{T|T}^{(N)}$.
   b) **Backward Recursion**: Repeat the following $t = T-1, \ldots, 0$:
      i. Calculate weights $w_j = f_{X_{t+1}|X_t = \mathcal{X}_{t|t}^{(j)}, \theta}(X_{t+1|T}^{(i)})$ for each $j = 1, \ldots, N$. Normalize these weights to obtain $W_1, \ldots, W_N$ which sum to one.

ii. Generate $X_{t|T}^{(i)}$ from the discrete distribution on $\mathcal{X}_{t|t}^{(1)}, \ldots, \mathcal{X}_{t|t}^{(N)}$ with probabilities $W_1, \ldots, W_N$.

The level of particle degeneracy of this algorithm can be checked by looking at the number of unique values among $X_{t|T}^{(1)}, \ldots, X_{t|T}^{(M)}$ for each $t = 0, \ldots, T$. Generally, particle degeneracy is not a problem for FFBS. The issue however is speed. Notice the double loop present in the algorithm (an outer loop over $i = 1, \ldots, M$ and then an inner loop over $t = T-1, \ldots, 0$). In the inner loop, there is a calculation of N weights. The total computational complexity is therefore $O(MNT)$. The FFBS algorithm will therefore be much slower compared to the complete smoothing algorithms. However, as there is not much particle degeneracy, one can afford to choose M to be much smaller than N (e.g., M can be of the order of a few hundreds while N is in the order of tens of thousands).

20.4 Recommended Reading for Today

1. The complete smoothing algorithm and the partial trajectory resampling variant are described in Section 15.3 of the Kitagawa book (see also Section 12.1 of the Chopin-Papaspiliopoulos book).
2. The FFBS algorithm is described in Section 12.3.2 of the Chopin-Papaspiliopoulos book, and in Section 11.2 of the Särkkä book (Särkkä calls it the Backward-Simulation Particle Smoother algorithm). A technique for making FFBS faster is described in Section 12.3.3 of the Chopin-Papaspiliopoulos book.

21 Lecture Twenty One

21.1 Model Selection

We shall next look at the topic of model selection. This important problem appears in almost every data analysis. In our context of state space models, consider, for example, the problem of deciding between a local level model or a local linear model. This is the problem of Model Selection. There are Frequentist and Bayesian approaches to Model Selection. One popular frequentist approach is the AIC (Akaike Information Criterion) and one popular Bayesian approach is the BIC (Bayesian Information Criterion). We shall study these procedures.

21.2 Akaike Information Criterion (AIC)

The AIC for a model M is defined as:

$$AIC(M) := -2 \times (\text{Maximized log-likelihood for } M) + 2 \times (\text{number of parameters in } M)\,.$$

This can be calculated for any model for which we can maximize likelihood. Let us look at the logic behind this criterion in the case of i.i.d models. State Space Models are not of this i.i.d kind but the analysis can be extended to them. Consider a dataset $y_1, \ldots, y_n$. By an i.i.d model M, we mean a model which postulates that $y_1, \ldots, y_n$ are realizations of random variables $Y_1, \ldots, Y_n$ which satisfy

$$Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} p_\theta$$

for some family of densities p_θ with a p-dimensional parameter θ. The log-likelihood for this model is:

$$\sum_{i=1}^{n} \log p_\theta(y_i).$$

The maximizer of this log-likelihood is the MLE (Maximum Likelihood Estimator) $\hat{\theta}_n$. The AIC for this model is thus:

$$AIC(M) = -2\sum_{i=1}^{n} \log p_{\hat{\theta}_n}(y_i) + 2p \tag{128}$$

The AIC for a different model $\tilde{M}$ which says that $Y_1, \ldots, Y_n$ are i.i.d q_α with a q-dimensional parameter α is

$$AIC(\tilde{M}) = -2\sum_{i=1}^{n} \log q_{\hat{\alpha}_n}(y_i) + 2q \tag{129}$$

The logic behind the AIC formulae (128) and (129) is explained below.

21.2.1 The simple case of no parameters

Consider first the case where we consider models with no parameters. Specifically, M is the model $Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} p$ and $\tilde{M}$ is the model $Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} q$. In this case, the AIC is simply the negative log-likelihood (multiplied by 2). In other words, we prefer the model with the higher loglikelihood. This makes sense and one of the explanations for looking at the loglikelihood is the following. Suppose that the true data generating process is given by:

$$Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} f^*. \tag{130}$$

It then makes sense to pick p or q depending on how close they are to f^*. This obviously depends on the specific way in which "closeness" is measured. One common choice is the Kullback-Leiber divergence:

$$D(f^*\|p) := \int f^* \log \frac{f^*}{p} = \int f^* \log f^* - \int f^* \log p,$$

and similarly

$$D(f^*\|q) := \int f^* \log \frac{f^*}{q} = \int f^* \log f^* - \int f^* \log q,$$

Note that the first term $\int f^* \log f^*$ is the same for both $D(f^*\|p)$ and $D(f^*\|q)$. Thus comparing $D(f^*\|p)$ and $D(f^*\|q)$ is equivalent to comparing $\int f^* \log p$ and $\int f^* \log q$. However f^* is unknown so we cannot directly compare $\int f^* \log p$ and $\int f^* \log q$. But a simple unbiased estimate of $\int f^* \log p$ is simply:

$$\frac{1}{n}\sum_{i=1}^{n} \log p(Y_i)$$

because the true data generating mechanism is (130). Similarly

$$\frac{1}{n}\sum_{i=1}^{n} \log q(Y_i)$$

is unbiased for $\int f^* \log q$. We thus compare

$$\frac{1}{n}\sum_{i=1}^{n} \log p(Y_i) \quad \text{and} \quad \frac{1}{n}\sum_{i=1}^{n} \log q(Y_i)$$

and pick the model with the higher value. This is clearly the same as comparing likelihoods.

21.2.2 Models with parameters

Now suppose that the two models are given by

$$\text{Model } M : Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} p_\theta$$

and

$$\text{Model } \tilde{M} : Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} q_\alpha.$$

The true data generating process is still (130). One can again consider the accuracy of estimating f^* under the Kullback-Leibler divergence. Model M would provide the estimate $p_{\hat{\theta}_n}$ and Model $\tilde{M}$ would provide the estimate $q_{\hat{\alpha}_n}$ for f^*. Here $\hat{\theta}_n$ is the MLE of θ under Model M and $\hat{\alpha}_n$ is the MLE of α under Model $\tilde{M}$. The Kullback-Leibler divergences are

$$D(f^* \| p_{\hat{\theta}_n}) = \int f^* \log f^* - \int f^* \log p_{\hat{\theta}_n}$$

and

$$D(f^* \| q_{\hat{\alpha}_n}) = \int f^* \log f^* - \int f^* \log q_{\hat{\alpha}_n}$$

Thus comparing the Kullback-Leibler divergences is equivalent to comparing

$$\int f^* \log p_{\hat{\theta}_n} \quad \text{and} \quad \int f^* \log q_{\hat{\alpha}_n}.$$

As we do not know f^*, we would need to estimate the above integrals from the data $y_1, \ldots, y_n$ generating according to (130). Natural estimators are given by

$$\frac{1}{n} \sum_{i=1}^{n} \log p_{\hat{\theta}_n}(Y_i) \quad \text{and} \quad \frac{1}{n} \sum_{i=1}^{n} \log q_{\hat{\alpha}_n}(Y_i).$$

However, unlike in the case where there are no parameters, these are no longer unbiased estimators of $\int f^* \log p_{\hat{\theta}_n}$ and $\int f^* \log q_{\hat{\alpha}_n}$ respectively. Indeed, we would expect $\frac{1}{n} \sum_{i=1}^{n} \log p_{\hat{\theta}_n}(Y_i)$ to be larger than $\int f^* \log p_{\hat{\theta}_n}$ and this will be especially true if p_θ is a complicated model which overfits the data. In order to correct the bias, we need to understand the quantity:

$$\int f^* \log p_{\hat{\theta}_n} - \frac{1}{n} \sum_{i=1}^{n} \log p_{\hat{\theta}_n}(Y_i), \tag{131}$$

and the analogous quantity for the second model. In order to estimate the above quantity, we need to know something about the behaviour of the maximum likelihood estimator $\hat{\theta}_n$.

21.2.3 Digression: MLE asymptotic distribution

Given data $y_1, \ldots, y_n$ and a candidate model which stipulates $Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} p_\theta$, consider the behaviour of the MLE:

$$\hat{\theta}_n := \operatorname*{argmax}_\theta \left(\sum_{i=1}^{n} \log p_\theta(y_i) \right).$$

The asymptotic behaviour of the MLE is usually studied under two assumptions: well-specified model and misspecified model.

MLE Asymptotics when model is correctly specified: By "model is correctly specified", we assume that the observed data are realizations of random variables $Y_1, \ldots, Y_n$ which

are independent and identically distributed according to p_{θ^*} for some θ^*. In other words, the true data generating distribution belongs to the candidate model class $\{p_\theta\}$. In this case, the MLE $\hat{\theta}_n$ is an accurate estimator of θ^*. More precisely, it can be shown that

$$\sqrt{n}\left(\hat{\theta}_n - \theta^*\right) \xrightarrow{\text{L}} N\left(0, (I(\theta^*))^{-1}\right) \tag{132}$$

where $I(\theta^*)$ is the Fisher information matrix:

$$\begin{aligned} I(\theta^*) &:= \mathbb{E}_{\theta^*}\left\{\left(\nabla_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right)\left(\nabla_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right)^T\right\} \\ &= \mathbb{E}_{\theta^*}\left\{\frac{\left(\nabla_\theta p_\theta(Y)\Big|_{\theta=\theta^*}\right)\left(\nabla_\theta p_\theta(Y)\Big|_{\theta=\theta^*}\right)^T}{p^2_{\theta^*}(Y)}\right\} = \int \frac{\left(\nabla_\theta p_\theta(y)\Big|_{\theta=\theta^*}\right)\left(\nabla_\theta p_\theta(y)\Big|_{\theta=\theta^*}\right)^T}{p_{\theta^*}(y)} dy. \end{aligned}$$

Here $\mathbb{E}_{\theta^*}$ denotes Expectation taken under the assumption $Y \sim p_{\theta^*}$. $I(\theta^*)$ is a $p \times p$ matrix where p is the dimension of θ^*. According to the above definition, the $(i,j)^{th}$ entry of $I(\theta^*)$ is given by

$$\mathbb{E}_{\theta^*}\left\{\frac{\partial(\log p_\theta(Y))}{\partial\theta_i}\Big|_{\theta=\theta^*} \frac{\partial(\log p_\theta(Y))}{\partial\theta_j}\Big|_{\theta=\theta^*}\right\} \tag{133}$$

A sketch of the proof of (132) can be found in the next subsection in the more general setting of model misspecification.

It is important to note that the Fisher Information Matrix has two alternative formulae in this correctly specified case. The first is that

$$I(\theta^*) = \text{Cov}_{\theta^*}\left(\nabla_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right) \tag{134}$$

where Cov_{θ^*} denotes covariance taken under the assumption $Y \sim p_{\theta^*}$. To see this, note first that $\text{Cov}(Z) = \mathbb{E}(ZZ^T) - (\mathbb{E}Z)(\mathbb{E}Z)^T$. Thus to see why this alternative formula of $I(\theta^*)$ is true, we only need to show that

$$\mathbb{E}_{\theta^*}\left(\nabla_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right) = 0$$

This is true because

$$\begin{aligned} \mathbb{E}_{\theta^*}\left(\nabla_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right) &= \mathbb{E}_{\theta^*} \frac{\nabla_\theta p_\theta(Y)\Big|_{\theta=\theta^*}}{p_{\theta^*}(Y)} \\ &= \int \frac{\nabla_\theta p_\theta(y)\Big|_{\theta=\theta^*}}{p_{\theta^*}(y)} p_{\theta^*}(y)dy \\ &= \int \nabla_\theta p_\theta(y)\Big|_{\theta=\theta^*} dy = \nabla_\theta\left(\int p_\theta(y)dy\right)\Big|_{\theta=\theta^*} = \nabla_\theta(1)\Big|_{\theta=\theta^*} = 0. \end{aligned}$$

Note that we have interchanged the two operations of integation with respect to y and differentiation with respect to θ. Some regularity conditions are necessary for such an interchange which we are ignoring in this treatment. This mean zero property validates the alternative formula (134).

The second alternative formula of Fisher Information is:

$$I(\theta^*) = -\mathbb{E}_{\theta^*}\left\{ H_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*} \right\} \tag{135}$$

where H_θ denotes Hessian. The $(i,j)^{th}$ entry of $I(\theta^*)$ according to this formula is

$$-\mathbb{E}_{\theta^*}\left\{ \frac{\partial^2 \log p_\theta(Y)}{\partial\theta_i \partial\theta_j}\Big|_{\theta=\theta^*} \right\} \tag{136}$$

To verify the validity of this alternative formula, we need to prove that (133) and(136) are equal. For this, observe first that

$$\begin{aligned}
\frac{\partial^2 \log p_\theta(y)}{\partial\theta_i\partial\theta_j} &= \frac{\partial}{\partial\theta_i}\left[\frac{\partial \log p_\theta(y)}{\partial\theta_j}\right] \\
&= \frac{\partial}{\partial\theta_i}\left[\frac{1}{p_\theta(y)}\frac{\partial p_\theta(y)}{\partial\theta_j}\right] \\
&= \frac{1}{p_\theta(y)}\frac{\partial^2 p_\theta(y)}{\partial\theta_i\partial\theta_j} - \frac{1}{(p_\theta(y))^2}\left[\frac{\partial p_\theta(y)}{\partial\theta_i}\right]\left[\frac{\partial p_\theta(y)}{\partial\theta_j}\right] \\
&= \frac{1}{p_\theta(y)}\frac{\partial^2 p_\theta(y)}{\partial\theta_i\partial\theta_j} - \left[\frac{\partial \log p_\theta(y)}{\partial\theta_i}\right]\left[\frac{\partial \log p_\theta(y)}{\partial\theta_j}\right].
\end{aligned}$$

As a result

$$\begin{aligned}
&-\mathbb{E}_{\theta^*}\left\{ \frac{\partial^2 \log p_\theta(Y)}{\partial\theta_i\partial\theta_j}\Big|_{\theta=\theta^*} \right\} \\
&= -\int \frac{1}{p_{\theta^*}(y)}\frac{\partial^2 p_\theta(y)}{\partial\theta_i\partial\theta_j}\Big|_{\theta=\theta^*} p_{\theta^*}(y)dy + \mathbb{E}_{\theta^*}\left\{ \frac{\partial(\log p_\theta(Y))}{\partial\theta_i}\Big|_{\theta=\theta^*} \frac{\partial(\log p_\theta(Y))}{\partial\theta_j}\Big|_{\theta=\theta^*} \right\}
\end{aligned}$$

The first term in the right hand side above equals zero because

$$\begin{aligned}
&-\int \frac{1}{p_{\theta^*}(y)}\frac{\partial^2 p_\theta(y)}{\partial\theta_i\partial\theta_j}\Big|_{\theta=\theta^*} p_{\theta^*}(y)dy \\
&= -\int \frac{\partial^2 p_\theta(y)}{\partial\theta_i\partial\theta_j}\Big|_{\theta=\theta^*} dy = -\frac{\partial^2}{\partial\theta_i\partial\theta_j}\left(\int p_\theta(y)dy\right)\Big|_{\theta=\theta^*} = -\frac{\partial^2}{\partial\theta_i\partial\theta_j}(1)\Big|_{\theta=\theta^*} = 0
\end{aligned}$$

and this proves that (133) and (136) are equal.

Here is some popular terminology that is used to describe these results:

1. The quantity $\theta \mapsto \nabla_\theta \log p_\theta(y)$ is called the score function corresponding to the model $\{p_\theta\}$.
2. The Fisher Information Matrix is defined as the second moment of the score function evaluated at the true parameter value.
3. When the model is correctly specified, The Fisher Information Matrix equals the covariance matrix of the score function evaluated at the true parameter value.
4. When the model is correctly specified, the Fisher Information Matrix equals the negative of the Hessian of the log-likelihood evaluated at the true parameter value.

MLE Asymptotics when model is misspecified: Here we assume that the data $y_1, \ldots, y_n$ are generated according to the model $Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} f^*$ where f^* does **not** necessarily belong to the class p_θ. In other words, f^* may not equal p_θ for any parameter value

θ. This means that there is no "true" parameter value θ^* anymore. So what exactly is the MLE $\hat{\theta}_n$ estimating? It turns out that the MLE $\hat{\theta}_n$ is really estimating the parameter value θ^* for which p_{θ^*} is closest to f^* in Kullback-Leibler divergence:

$$\theta^* := \operatorname*{argmin}_{\theta} D(f^* \| p_\theta)$$

Because $D(f^* \| p_\theta) = \int f^* \log f^* - \int f^* \log p_\theta$, we can also define θ^* as

$$\theta^* := \operatorname*{argmax}_{\theta} \int f^*(y) \log p_\theta(y) dy.$$

In other words, θ^* can also be thought of as the maximizer of the average loglikelihood (averaged with respect to the true data generating density).

In this misspecified case, it again turns out that $\sqrt{n}\left(\hat{\theta}_n - \theta^*\right)$ converges to a zero mean multivariate normal distribution with some covariance matrix. However the covariance matrix now is not simply the inverse of the Fisher Information Matrix. To understand this, first let us consider the following simple example.

Example 21.1 (Normal Mean Model)**.** *Suppose $Y_1, \ldots, Y_n \overset{i.i.d}{\sim} f^*$ for some density f^*. Consider the model $N(\theta, 1)$ i.e.,*

$$p_\theta(y) := (2\pi)^{-1/2} \exp\left(-\frac{(y-\theta)^2}{2}\right).$$

Let us consider the misspecified setting where f^* is not equal to $N(\theta, 1)$ for any θ. What is θ^* in this case? The loglikelihood averaged with respect to f^* is:

$$\begin{aligned}\int f^*(y) \log p_\theta(y) dy &= \int f^*(y) \left\{-\frac{(y-\theta)^2}{2} - \frac{1}{2}\log(2\pi)\right\} dy \\ &= -\frac{1}{2}\int (y-\theta)^2 f^*(y) dy - \frac{1}{2}\log(2\pi).\end{aligned}$$

It is clear that the minimizer of $\int f^* \log p_\theta$ over all $\theta \in \mathbb{R}$ equals the mean corresponding to the density f^*. We thus take

$$\theta^* = \int y f^*(y) dy.$$

On the other hand, given data $Y_1, \ldots, Y_n$, the MLE of θ is easily seen to be

$$\hat{\theta}_n := \bar{Y}_n := \frac{Y_1 + \cdots + Y_n}{n}.$$

By the Central Limit Theorem (assuming that the variance corresponding to f^* is finite), we have

$$\sqrt{n}\left(\hat{\theta} - \theta^*\right) = \sqrt{n}\left(\bar{Y}_n - \theta^*\right) \xrightarrow{L} N(0, V^*) \tag{137}$$

where V^* is the variance corresponding to f^*. What is the Fisher Information Matrix in this case? The loglikelihood and the score function equal respectively

$$\log p_\theta(y) = -\frac{(y-\theta)^2}{2} - \frac{1}{2}\log(2\pi),$$

and

$$\frac{d}{d\theta} \log p_\theta(y) = y - \theta.$$

The second moment of the score function evaluated at $\theta = \theta^*$ is therefore:

$$I(\theta^*) = \mathbb{E}_{f^*}(Y - \theta^*)^2 = V^*.$$

Thus the asymptotic variance of the MLE does not equal the inverse of $I(\theta^*)$ (in this case, it equals exactly the Fisher Information).

Let us now state the result for the asymptotic distribution of the MLE $\hat{\theta}_n$ in the misspecified case. We need some definitions. First the Fisher Information Matrix as before is defined as the second moment of the score function:

$$\begin{aligned} I(\theta^*) &:= \mathbb{E}_{f^*}\left\{\left(\nabla_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right)\left(\nabla_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right)^T\right\} \\ &= \mathbb{E}_{f^*}\left\{\frac{\left(\nabla_\theta p_\theta(Y)\Big|_{\theta=\theta^*}\right)\left(\nabla_\theta p_\theta(Y)\Big|_{\theta=\theta^*}\right)^T}{p^2_{\theta^*}(Y)}\right\} \\ &= \int \frac{\left(\nabla_\theta p_\theta(y)\Big|_{\theta=\theta^*}\right)\left(\nabla_\theta p_\theta(y)\Big|_{\theta=\theta^*}\right)^T}{p^2_{\theta^*}(y)} f^*(y)dy. \end{aligned}$$

The crucial difference from the correctly specified case is that the Expectation is taken to be with respect to the true density f^* (and not p_{θ^*}). As in the well-specified case, $I(\theta^*)$ also equals the covariance matrix of the score function evaluated at θ^*. This is because

$$\mathbb{E}_{f^*}\left(\nabla_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right) = \int \nabla_\theta \log p_\theta(y)\Big|_{\theta=\theta^*} f^*(y)dy = \nabla_\theta\left(\int \log p_\theta(y) f^*(y)dy\right)\Big|_{\theta=\theta^*} = 0. \tag{138}$$

The last equality above is because the gradient of $\int \log p_\theta(y) f^*(y)dy$ equals zero at $\theta = \theta^*$ as θ^* maximizes the average loglikelihood (with respect to f^*) over θ (this is the definition of θ^*). Therefore

$$I(\theta^*) = \mathrm{Cov}_{f^*}\left(\nabla_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right). \tag{139}$$

In the well-specified setting, we have seen that the Fisher Information Matrix also equals the negative of the Expected Hessian of the loglikelihood evaluated at $\theta = \theta^*$ (see the formula (135)). This is no longer in the case of misspecification. Specifically here $I(\theta^*)$ is not necessarily the same as $J(\theta^*)$ where

$$J(\theta^*) := -\mathbb{E}_{f^*}\left\{H_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right\}. \tag{140}$$

That $I(\theta^*)$ and $J(\theta^*)$ can be distinct is seen in the simple normal example.

Example 21.2 (Normal Mean Model continued)**.** *Consider the same setting of Example* (21.1)*. Here the Hessian of the loglikelihood is easily seen to be $\frac{d^2}{d\theta^2}\log p_\theta(y) = -1$ so that $J(\theta^*) = 1$. On the other hand, we saw in Example* (21.1) *that $I(\theta^*) = V^*$ where V^* is the variance corresponding to f^*. Thus, unless $V^* = 1$, the two quantities $I(\theta^*)$ and $J(\theta^*)$ will be different. Note that if we insist on correct specification, $f^* = N(\theta^*, 1)$, then the variance corresponding to f^* will be 1 so that $I(\theta^*)$ and $J(\theta^*)$ will be the same.*

Here is the correct asymptotic distribution result for the MLE in the misspecified setting:

$$\sqrt{n}\left(\hat{\theta}_n - \theta^*\right) \xrightarrow{\text{L}} N\left(0, J(\theta^*)^{-1} I(\theta^*) J(\theta^*)^{-1}\right). \tag{141}$$

The formula $J(\theta^*)^{-1} I(\theta^*) J(\theta^*)^{-1}$ for the covariance is sometimes called the "Sandwich Formula" (see e.g., `http://www.econ.uiuc.edu/~roger/courses/476/lectures/L10.pdf`). In the case of correct specification, we have $J(\theta^*) = I(\theta^*)$ as we saw in the previous section so that (141) is identical to (132).

It is easy to see that (141) gives the correct answer in the simple normal mean example.

Example 21.3 (Normal Mean Model Continued)**.** *Here* $I(\theta^*) = V^*$ *and* $J(\theta^*) = 1$ *so that* (141) *gives*

$$\sqrt{n}\left(\hat{\theta}_n - \theta^*\right) \xrightarrow{L} N\left(0, J(\theta^*)^{-1} I(\theta^*) J(\theta^*)^{-1}\right) = N(0, V^*)$$

which coincides with (137)*.*

Here is a sketch of the proof of (141).

Proof of (141)*.* By definition, the MLE $\hat{\theta}_n$ maximizes the loglikelihood function:

$$\ell(\theta) := \sum_{i=1}^{n} \log p_\theta(Y_i).$$

Thus the gradient of the loglikelihood evaluated at the MLE $\hat{\theta}_n$ will be zero:

$$\nabla_\theta \ell(\theta)\big|_{\theta=\hat{\theta}_n} = \nabla \ell(\hat{\theta}_n) = 0.$$

Now intuitively, $\hat{\theta}_n$ should be close to θ^*. So we do a Taylor expansion of $\nabla \ell(\hat{\theta}_n)$ around θ^*:

$$0 = \nabla \ell(\hat{\theta}_n) \approx \nabla \ell(\theta^*) + H\ell(\theta^*)\left(\hat{\theta}_n - \theta^*\right)$$

which immediately gives

$$\hat{\theta}_n - \theta^* \approx -\left(H\ell(\theta^*)\right)^{-1} \left[\nabla \ell(\theta^*)\right].$$

We rewrite the above as

$$\sqrt{n}\left(\hat{\theta}_n - \theta^*\right) \approx \left(-\frac{1}{n} H\ell(\theta^*)\right)^{-1} \left[\frac{1}{\sqrt{n}} \nabla \ell(\theta^*)\right].$$

Now

$$\frac{1}{\sqrt{n}} \nabla \ell(\theta^*) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \nabla_\theta \log p_\theta(Y_i)\Big|_{\theta=\theta^*}$$

By (138), each random variable $\nabla_\theta \log p_\theta(Y_i)$ (note $Y_i \overset{\text{i.i.d}}{\sim} f^*$) has mean zero. Thus by the Central Limit Theorem (and (139)),

$$\frac{1}{\sqrt{n}} \nabla \ell(\theta^*) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \nabla_\theta \log p_\theta(Y_i)\Big|_{\theta=\theta^*} \xrightarrow{\text{L}} N\left(0, \text{Cov}_{f^*}\left(\nabla_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right)\right) = N\left(0, I(\theta^*)\right).$$

Further, by the law of large numbers (and (140)),

$$-\frac{1}{n} H\ell(\theta^*) = \frac{1}{n} \sum_{i=1}^{n} \left(-H_\theta \log p_\theta(Y_i)\Big|_{\theta=\theta^*}\right) \xrightarrow{\mathbb{P}} -\mathbb{E}_{f^*}\left\{H_\theta \log p_\theta(Y)\Big|_{\theta=\theta^*}\right\} = J(\theta^*).$$

Thus

$$\sqrt{n}\left(\hat{\theta}_n - \theta^*\right) \approx \left(-\frac{1}{n} H\ell(\theta^*)\right)^{-1} \left[\frac{1}{\sqrt{n}} \nabla \ell(\theta^*)\right] \xrightarrow{\text{L}} N\left(0, J(\theta^*)^{-1} I(\theta^*) J(\theta^*)^{-1}\right)$$

which proves (141). □

21.3 Back to AIC

Let us now get back to the setting of Section 21.2.2. Our goal is to understand the quantity (131). This is a random variable (as it is a function of $Y_1, \ldots, Y_n$ which are independently distributed according to f^*). We shall concentrate on finding the expectation of (131):

$$Q := \mathbb{E}\left\{\int f^* \log p_{\hat{\theta}_n} - \frac{1}{n}\sum_{i=1}^{n} \log p_{\hat{\theta}_n}(Y_i)\right\}$$

We write

$$Q = Q_1 + Q_2 + Q_3$$

where

$$Q_1 := \mathbb{E}\left\{\int f^* \log p_{\hat{\theta}_n} - \int f^* \log p_{\theta^*}\right\}, \quad Q_2 := \mathbb{E}\left\{\int f^* \log p_{\theta^*} - \frac{1}{n}\sum_{i=1}^{n} \log p_{\theta^*}(Y_i)\right\}$$

and

$$Q_3 := \mathbb{E}\left\{\frac{1}{n}\sum_{i=1}^{n} \log p_{\theta^*}(Y_i) - \frac{1}{n}\sum_{i=1}^{n} \log p_{\hat{\theta}_n}(Y_i)\right\}$$

It is clear that $Q_2 = 0$ so we only need to focus on Q_1 and Q_3. For Q_1, Taylor expansion around θ^* gives

$$\begin{aligned} Q_1 &= \mathbb{E}\left\{\int f^* \log p_{\hat{\theta}_n} - \int f^* \log p_{\theta^*}\right\} \\ &\approx \mathbb{E}\left\{\left\langle \nabla_\theta \int f^* \log p_\theta \Big|_{\theta=\theta^*}, \hat{\theta}_n - \theta^* \right\rangle + \frac{1}{2}\left(\hat{\theta}_n - \theta^*\right)^T H_\theta \int f^* \log p_\theta \Big|_{\theta=\theta^*} \left(\hat{\theta}_n - \theta^*\right)\right\} \end{aligned}$$

The gradient in the first term above equals zero (because of (138)). The Hessian equals

$$H_\theta \int f^* \log p_\theta \Big|_{\theta=\theta^*} = \int f^* H_\theta \log p_\theta \Big|_{\theta=\theta^*} = -J(\theta^*)$$

because of the definition (140) of $J(\theta^*)$. Thus

$$\begin{aligned} Q_1 &\approx -\frac{1}{2}\mathbb{E}\left\{\left(\hat{\theta}_n - \theta^*\right)^T J(\theta^*)\left(\hat{\theta}_n - \theta^*\right)\right\} \\ &= -\frac{1}{2}\mathbb{E}\left[\text{trace}\left\{J(\theta^*)\left(\hat{\theta}_n - \theta^*\right)\left(\hat{\theta}_n - \theta^*\right)^T\right\}\right] \\ &= -\frac{1}{2}\text{trace}\ \mathbb{E}\left\{J(\theta^*)\left(\hat{\theta}_n - \theta^*\right)\left(\hat{\theta}_n - \theta^*\right)^T\right\} \\ &= -\frac{1}{2}\text{trace}\left\{J(\theta^*)\mathbb{E}\left(\hat{\theta}_n - \theta^*\right)\left(\hat{\theta}_n - \theta^*\right)^T\right\} \end{aligned}$$

Because of (141), we take

$$\mathbb{E}\left(\hat{\theta}_n - \theta^*\right)\left(\hat{\theta}_n - \theta^*\right)^T = \frac{1}{n} J(\theta^*)^{-1} I(\theta^*) J(\theta^*)^{-1}$$

to get

$$Q_1 \approx -\frac{1}{2n}\text{trace}\left\{J(\theta^*)J(\theta^*)^{-1}I(\theta^*)J(\theta^*)^{-1}\right\} = -\frac{1}{2n}\text{trace}\left\{I(\theta^*)J(\theta^*)^{-1}\right\}$$

For Q_3, we use Taylor expansion around the MLE $\hat{\theta}_n$ to get

$$Q_3 = \mathbb{E}\left\{\frac{1}{n}\sum_{i=1}^{n}\log p_{\theta^*}(Y_i) - \frac{1}{n}\sum_{i=1}^{n}\log p_{\hat{\theta}_n}(Y_i)\right\}$$
$$\approx \mathbb{E}\left\{\left\langle \nabla_\theta \frac{1}{n}\sum_{i=1}^{n}\log p_\theta(Y_i)\Big|_{\theta=\hat{\theta}_n}, \hat{\theta}_n - \theta^*\right\rangle + \frac{1}{2}\left(\hat{\theta}_n - \theta^*\right)^T H_\theta \frac{1}{n}\sum_{i=1}^{n}\log p_\theta(Y_i)\Big|_{\theta=\hat{\theta}_n}\left(\hat{\theta}_n - \theta^*\right)\right\}.$$

The gradient in the first term above equals zero because $\hat{\theta}_n$ maximizes log-likelihood. The Hessian for large n can be approximated by $-J(\theta^*)$ (this is because $\hat{\theta}_n$ will be close to θ^*). Thus

$$Q_3 \approx -\frac{1}{2}\mathbb{E}\left\{\left(\hat{\theta}_n - \theta^*\right)^T J(\theta^*)\left(\hat{\theta}_n - \theta^*\right)\right\}$$

which (just as in the computation of Q_1) leads to

$$Q_3 \approx -\frac{1}{2n}\text{trace}\left\{I(\theta^*)J(\theta^*)^{-1}\right\}.$$

We have thus proved

$$Q \approx -\frac{1}{2n}\text{trace}\left\{I(\theta^*)J(\theta^*)^{-1}\right\} - \frac{1}{2n}\text{trace}\left\{I(\theta^*)J(\theta^*)^{-1}\right\} = -\frac{1}{n}\text{trace}\left\{I(\theta^*)J(\theta^*)^{-1}\right\}$$

This suggests the estimator:

$$\frac{1}{n}\sum_{i=1}^{n}\log p_{\hat{\theta}_n}(Y_i) - \frac{1}{n}\text{trace}\left\{I(\theta^*)J(\theta^*)^{-1}\right\} \tag{142}$$

for

$$\int f^* \log p_{\hat{\theta}_n}. \tag{143}$$

(142) is not really an estimator because the second term depends on θ^*. However if we assume that the model is well-specified, then $I(\theta^*) = J(\theta^*)$ so that the second term equals p (note p is the dimension of θ^*). We then get

$$\frac{1}{n}\sum_{i=1}^{n}\log p_{\hat{\theta}_n}(Y_i) - \frac{p}{n}$$

as the estimate for (143). The quantity (142) is simply the AIC multiplied by the constant $-\frac{1}{2n}$. This motivates the use of AIC for model selection.

21.4 Recommended Reading for Today

1. A description of the AIC can be found in Chapter 4 (especially Section 4.5) of the Kitagawa book.
2. Various applications of the AIC for model selection in state space models can be found throughout the Kitagawa-Gersch and the Kitagawa books.
3. More details on the AIC and other related model selection criteria can be found in the book *Information Criteria and Statistical Modeling* by Konishi and Kitagawa (accessible through the Berkeley library website).

22 Lecture Twenty Two

22.1 Recap: AIC

In the last class, we looked at frequentist model selection using the Akaike Information Criterion (AIC) which is defined as:

$$AIC(M) := -2 \times (\text{Maximized log-likelihood for } M) + 2 \times (\text{number of parameters in } M)\,. \tag{144}$$

This criterion arises in the process of estimation of the out-of-sample accuracy of the model. More precisely, suppose that the data is $y_1, \ldots, y_n$ and the model is $Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} p_\theta$ with parameter θ. The in-sample accuracy of this model is

$$\frac{1}{n}\sum_{i=1}^{n} \log p_{\hat{\theta}_n}(y_i) \tag{145}$$

where $\hat{\theta}_n$ is the MLE. Its out-of-sample accuracy is defined as

$$\int f^*(y) \log p_{\hat{\theta}_n}(y) dy \tag{146}$$

where f^* denotes the true data generating density. As we discussed last class, asymptotics (under a bunch of assumptions) justify

$$\frac{1}{n}\sum_{i=1}^{n} \log p_{\hat{\theta}_n}(y_i) - \frac{p}{n} \tag{147}$$

as an estimator of (146) where p is the dimension of the parameter θ. The AIC is just the above quantity multiplied by the constant factor $-2n$.

It can be noted that the out-of-sample accuracy can also be estimated more directly (without using any asymptotics) if additional independent data $\tilde{y}_1, \ldots, \tilde{y}_m \overset{\text{i.i.d}}{\sim} f^*$ is available. In this case, we can use

$$\frac{1}{m}\sum_{j=1}^{m} \log p_{\hat{\theta}_n}(\tilde{y}_j) \tag{148}$$

as an estimate of (146). If additional data is not available, one can split the existing dataset $y_1, \ldots, y_n$ into two parts, and use one part to calculate $\hat{\theta}_n$ and the other part as $\tilde{y}_j$ for the calculation of (148). To summarize, AIC and related test-data out-of-sample accuracy evaluations have the following issues:

1. AIC is popular but it uses many difficult to verify assumptions for obtaining the simple estimate (147) for (146).
2. Heldout/Test-set methodology is more popular but requires additional data. In the absence of additional data, one needs to construct training and test datasets whose choices can be adhoc. If the test dataset is too small, the estimate (148) will be noisy. On the other hand, if the tranining dataset set is too small, the MLE calculated from the training dataset will be quite different from the MLE calculated on the full dataset and which create bias in the estimation of (146).

We shall next study Bayesian Model Selection.

22.2 Bayesian Model Selection

Bayesian model selection works for comparing Bayesian models. By a Bayesian model, I mean a model in which both the likelihood as well as the prior are specified. For example, given data $y_1, \ldots, y_n$, consider the two models:

$$Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} N(\theta, 1) \qquad \text{with } \theta \in [-5, 5], \tag{149}$$

and

$$Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} N(\theta, 1) \qquad \text{with } \theta \sim \text{unif}[-5, 5]. \tag{150}$$

Model (149) is not a Bayesian model because the prior is not specified. The constraint $\theta \in [-5, 5]$ does not precisely say how θ is distributed on $[-5, 5]$. On the other hand, the model (150) is a Bayesian model.

An important advantage of Bayesian models is that they allow calculation of the probability of the observed data under the model. For example, the Bayesian model (150) would calculate the probability of the observed data $y_1, \ldots, y_n$ as

$$\frac{1}{10} \int_{-5}^{5} (2\pi)^{-n/2} \exp\left(-\frac{1}{2} \sum_{i=1}^{n} (y_i - \theta)^2 \right) d\theta.$$

On the other hand, the non-Bayesian model (149) would not allow computation of the probability of the observed dataset. Indeed, under the model (149), one can write the probability of the observed data as

$$(2\pi)^{-n/2} \exp\left(-\frac{1}{2} \sum_{i=1}^{n} (y_i - \theta)^2 \right)$$

for some $\theta \in [-5, 5]$. But this not give a precise answer to the probability of the observed data as it involves the unknown value θ about which we only know that $\theta \in [-5, 5]$.

Note the slight abuse of terminology here. By probability of the observed data under a model, I actually mean the joint density:

$$f_{Y_1, \ldots, Y_n}(y_1, \ldots, y_n)$$

when the underlying random variables are continuous. In the case where the random variables are discrete, probability of the observed data will mean

$$\mathbb{P}\{Y_1 = y_1, \ldots, Y_n = y_n\}.$$

In the continuous case, one really should think of an observation 1.29 as not being exactly equal to the number 1.29 but rather as $[1.29 - \delta, 1.29 + \delta]$ for some very small number δ which represents recording precision. The observed dataset $y_1, \ldots, y_n$ is then really $[y_1 - \delta, y_1 + \delta], \ldots, [y_n - \delta, y_n + \delta]$. In such a case, the probability of the observed dataset will be represented by

$$\mathbb{P}\{Y_1 \in [y_1 - \delta, y_1 + \delta], \ldots, Y_n \in [y_n - \delta, y_n + \delta]\} \approx f_{Y_1, \ldots, Y_n}(y_1, \ldots, y_n)(2\delta)^n.$$

Thus, up to multiplication by the constant factor $(2\delta)^n$ (which will be the same across different models), the probability of the observed dataset is proportional to the joint density. This justifies the abuse of notation referring to the joint density as the probability of the observed dataset.

Consider now a generic dataset y (y could be a vector or matrix or something even more general). We have two Bayesian models for y:

$$M_1 : Y \mid \theta \sim p_\theta \qquad \text{with } \theta \sim f_\theta(\cdot) \tag{151}$$

and

$$M_2 : Y \mid \alpha \sim q_\alpha \qquad \text{with } \alpha \sim f_\alpha(\cdot) \tag{152}$$

Bayesian Model Selection compares M_1 and M_2 by simply calculating the probabiliity of the observed data y under both M_1 and M_2. Specifically, we compare

$$f_{Y|M_1}(y) = \int p_\theta(y) f_\theta(\theta) d\theta \quad \text{and} \quad f_{Y|M_2}(y) = \int q_\alpha(y) f_\alpha(\alpha) d\alpha.$$

Preference will be given to the model for which the probability of observed data is higher. The following are alternative terms for $f_{Y|M_1}(y)$:

1. **Marginal or Integrated Likelihood**: $f_{Y|M_1}(y)$ is simply the integration of the likelihood $p_\theta(y)$ with respect to the prior density $f_\theta(\theta)$.
2. **Evidence**: $f_{Y|M_1}(y)$ is often referred to as the Evidence of the model M_1 under the observed data y.

Thus Bayesian Model Selection compares the Integrated Likelihoods or Evidences of models. The following simple example is a good illustration of the basic idea behind Bayesian Model Selection.

Example 22.1 (MacKay)**.** *This example is from Chapter 28 of David MacKay's book titled Information Theory, Inference, and Learning Algorithms. We have the dataset* $-1, 3, 7, 11$. *Consider the following two Bayesian models for this dataset:*

1. ***Model 1 (linear)****:* $Y_1 = \alpha$ *and* $Y_{n+1} = Y_n + \beta$ *for* $n \geq 1$. *This model has the two parameters* α *and* β. *We assume that* α *and* β *are integer-valued that they are independently uniformly distributed over the set* $\{-50, -49, \ldots, 49, 50\}$ *which has cardinality 101.*
2. ***Model 2 (cubic)****:* $Y_1 = a$ *and* $Y_{n+1} = bY_n^3 + cY_n^2 + d$. *This model has the four parameters* a, b, c, d. *We assume that these four parameters are independent with* a *having the uniform on* $\{-50, -49, \ldots, 49, 50\}$ *and* b, c, d *each having the distribution of* x/y *where* $x \sim$ *Unif*$\{-50, -49, \ldots, 49, 50\}$ *and* $y \sim$ *Unif*$\{1, \ldots, 50\}$ *are independent.*

Which of these two models would you use for the data? Bayesian model selection is readily applicable here as both the models are Bayesian. We only need to calculate the probability of the observed data for the two models. For the linear model (M1):

$$\begin{aligned}
&\mathbb{P}\{Y_1 = -1, Y_2 = 3, Y_3 = 7, Y_4 = 11 \mid \text{M1}\} \\
&= \sum_{i,j} \mathbb{P}\{Y_1 = -1, Y_2 = 3, Y_3 = 7, Y_4 = 11 \mid \alpha = i, \beta = j, \text{M1}\} \mathbb{P}\{\alpha = i, \beta = j \mid \text{M1}\} \\
&= \mathbb{P}\{Y_1 = -1, Y_2 = 3, Y_3 = 7, Y_4 = 11 \mid \alpha = -1, \beta = 4, \text{M1}\} \mathbb{P}\{\alpha = -1, \beta = 4 \mid \text{M1}\} \\
&= (1)\mathbb{P}\{\alpha = -1 \mid \text{M1}\} \mathbb{P}\{\beta = 4 \mid \text{M1}\} = \left(\frac{1}{101}\right)^2 = 9.8 \times 10^{-5}.
\end{aligned}$$

For the cubic model:

$$\begin{aligned}
&\mathbb{P}\{Y_1 = -1, Y_2 = 3, Y_3 = 7, Y_4 = 11 \mid \text{M2}\} \\
&= \sum_{a,b,c,d} \mathbb{P}\{Y_1 = -1, Y_2 = 3, Y_3 = 7, Y_4 = 11 \mid a, b, c, d, \text{M2}\} \mathbb{P}\{a = a, b = b, c = c, d = d \mid \text{M2}\}
\end{aligned}$$

It turns out that the cubic model explains the given data perfectly if and only if its four parameters a, b, c, d are chosen as $a = -1, b = -1/11, c = 9/11, d = 23/11$. As a result

$$\begin{aligned}
&\mathbb{P}\{Y_1 = -1, Y_2 = 3, Y_3 = 7, Y_4 = 11 \mid \text{M2}\} \\
&= \mathbb{P}\left\{a = -1, b = -1/11, c = 9/11, d = 23/11 \mid M2\right\} \\
&= \mathbb{P}\{a = -1\}\mathbb{P}\{b = -1/11\}\mathbb{P}\{c = 9/11\}\mathbb{P}\{d = 23/11\} \\
&= \left(\frac{1}{101}\right)\left(4 \cdot \frac{1}{101} \cdot \frac{1}{50}\right)\left(4 \cdot \frac{1}{101} \cdot \frac{1}{50}\right)\left(2 \cdot \frac{1}{101} \cdot \frac{1}{50}\right) = 2.5 \times 10^{-12}.
\end{aligned}$$

Clearly the probability of the observed data is much smaller for the cubic model compared to the simpler linear model. Bayesian model selection here will prefer the linear model and this would align with common sense. Note here both the models explain the data equally well. The cubic model gets downgraded however because the prior in the cubic model gives a much smaller probability to the correct parameter values compared to the linear model. We shall come back to this point later.

Bayesian model selection can also be understood from the perspective of hierarchical modeling. Specifically consider the following hierarchical model which converts the two models M_1 and M_2 (defined as in (151) and (152) respectively) into a *single Bayesian model*.

$$\begin{aligned}
&\mathcal{I} \text{ takes the values 1 and 2 with probabilities } \rho \text{ and } 1 - \rho \\
&Y \mid \mathcal{I} = 1, \theta \sim p_\theta \quad \text{and} \quad \theta \mid \mathcal{I} = 1 \sim f_\theta \\
&Y \mid \mathcal{I} = 2, \theta \sim q_\alpha \quad \text{and} \quad \alpha \mid \mathcal{I} = 2 \sim f_\alpha
\end{aligned} \tag{153}$$

The random variable $\mathcal{I}$ represents one of the two models M_1 and M_2. More precisely $\mathcal{I} = 1$ represents M_1 and $\mathcal{I} = 2$ represents M_2. ρ and $1 - \rho$ represent the prior probabilities of M_1 and M_2. Under this single Bayesian model, we can calculate the posterior distribution of $\mathcal{I}$ given the data $Y = y$ as:

$$\mathbb{P}\left\{\mathcal{I} = 1 \mid Y = y\right\} = \frac{f_{Y|M_1}(y)\mathbb{P}\{\mathcal{I} = 1\}}{f_{Y|M_1}(y)\mathbb{P}\{\mathcal{I} = 1\} + f_{Y|M_2}(y)\mathbb{P}\{\mathcal{I} = 2\}}$$

and

$$\mathbb{P}\left\{\mathcal{I} = 2 \mid Y = y\right\} = \frac{f_{Y|M_2}(y)\mathbb{P}\{\mathcal{I} = 2\}}{f_{Y|M_1}(y)\mathbb{P}\{\mathcal{I} = 1\} + f_{Y|M_2}(y)\mathbb{P}\{\mathcal{I} = 2\}}.$$

These are the posterior probabilities of the two models given the data $Y = y$. Model M_1 will be preferred compared to Model M_2 if and only if

$$\mathbb{P}\left\{\mathcal{I} = 1 \mid Y = y\right\} > \mathbb{P}\left\{\mathcal{I} = 2 \mid Y = y\right\}.$$

As the denominators of the above probabilities are the same, this is equivalent to

$$f_{Y|M_1}(y)\mathbb{P}\{\mathcal{I} = 1\} > f_{Y|M_2}(y)\mathbb{P}\{\mathcal{I} = 2\}.$$

Now if $\mathbb{P}\{\mathcal{I} = 1\} = \mathbb{P}\{\mathcal{I} = 2\}$ i.e., if the two models are *a priori* equally likely, then the above comparison is equivalent to comparing $f_{Y|M_1}(y)$ and $f_{Y|M_2}(y)$. Thus Bayesian model selection in terms of evidences is equivalent to looking at posterior probabilities of the two models in a hierarchical model where the prior probabilities are the same. When the prior model probabilities are not the same, we need to multiply the evidences by the prior probabilities before evaluating the models.

22.3 Two Alternative Expressions for the Evidence

The evidence $f_{Y|M_1}(y)$ satisfies the following two alternative expressions which bear some similarities to the AIC formula (144). Both these formulae are consequences of the following expression for posterior density of the parameter θ in the model M_1:

$$\text{posterior}(\theta) = \frac{\text{prior}(\theta)p_\theta(y)}{f_{Y|M_1}(y)} \qquad \text{for every } \theta.$$

Here $\text{prior}(\theta) = f_\theta(\theta)$ and $\text{posterior}(\theta)$ is the density of θ conditional on $Y = y$ in the model M_1. As a result, we have

$$f_{Y|M_1}(y) = \frac{\text{prior}(\theta)p_\theta(y)}{\text{posterior}(\theta)} \qquad \text{for every } \theta. \tag{154}$$

Taking θ to be the MLE $\hat{\theta}$ in the model M_1, we obtain

$$f_{Y|M_1}(y) = \frac{\text{prior}(\hat{\theta})p_{\hat{\theta}}(y)}{\text{posterior}(\hat{\theta})}.$$

This immediately gives the formula:

$$-2\log f_{Y|M_1}(y) = -2\log p_{\hat{\theta}}(y) + 2\log\left[\frac{\text{posterior}(\hat{\theta})}{\text{prior}(\hat{\theta})}\right]$$

$\log p_{\hat{\theta}}(y)$ is simply the maximized log-likelihood for the model M_1. Thus

$$-2\log\left(\text{Evidence}(M_1)\right) = -2 \times (\text{Maximized log-likelihood for } M_1) + 2\log\left[\frac{\text{posterior}(\hat{\theta})}{\text{prior}(\hat{\theta})}\right] \tag{155}$$

Note the similarity of (155) with (144). The first term above measures the fit of the best model in M_1 to the observed data, while the second term measures model complexity. The model complexity term is more complicated compared to (144). The posterior evaluated at the MLE will generally be larger than the prior evaluated at the MLE which means that the model complexity term in (155) will be positive.

Example 22.2 (Example 22.1 continued)**.** *Here both the models M_1 (linear) and M_2 (cubic) perfectly explain the observed data. Therefore the maximized log-likelihood value is the same for both M_1 and M_2. Also both the models have exactly one parameter setting which explains the data perfectly, and every other setting gives zero probability to the observed data. This means that posterior($\hat{\theta}$) equals 1 for both the models. The only difference in the models will be in the prior evaluated at the best parameter setting. This term is much higher for the linear model compared to the cubic model. The reason is that the prior for the cubic model is supported on a much larger set (compared to the prior for the linear model) and consequently the prior mass assigned to each individual element of the large set is much smaller.*

For the second alternative formula, take logarithms on both sides of (154) to get

$$\log f_{Y|M_1}(y) = \log p_\theta(y) - \log\left[\frac{\text{posterior}(\theta)}{\text{prior}(\theta)}\right] \qquad \text{for every } \theta.$$

Integrating both sides of the above equation with respect to $\text{posterior}(\theta)$, we get (note left hand side does not depend on θ):

$$\begin{aligned}\log f_{Y|M_1}(y) &= \int \text{posterior}(\theta)\log p_\theta(y)d\theta - \int \text{posterior}(\theta)\log\left[\frac{\text{posterior}(\theta)}{\text{prior}(\theta)}\right]d\theta \\ &= \mathbb{E}_{\theta\sim\text{posterior}}\log p_\theta(y) - D\left(\text{posterior}\|\text{prior}\right)\end{aligned}$$

where $D(\cdot\|\cdot)$ denotes Kullback-Leibler divergence. In other words

$$-2\log\left(\text{Evidence}(M_1)\right) = \mathbb{E}_{\theta\sim\text{posterior}}\left[-2\log p_\theta(y)\right] + 2D\left(\text{posterior}\|\text{prior}\right) \tag{156}$$

This is similar to (156) except that maximized log-likelihood is replaced by the expected log-likelihood where the expectation is taken with respect to the posterior, and the complexity term is replaced by the Kullback-Leibler divergence between the posterior and the prior. Generally, for complex models, the posterior will be quite different from the prior leading to greater penalization (for a concrete example, consider the setting of Example 22.1).

22.4 The BIC

The BIC (Bayesian Information Criterion) is obtained as an approximation for (155) when the posterior is replaced by its normal approximation. As we have seen previously, in some cases, the posterior distribution is well approximated by a normal distribution $N_p(\hat{\theta}, \Sigma/n)$ where $\hat{\theta}$ is the MLE, n denotes sample size and Σ is a $p \times p$ covariance matrix (generally Σ is related to the Hessian of the log-likelihood evaluated at $\hat{\theta}$). In such cases,

$$\text{posterior}(\theta) = (2\pi)^{-p/2}\left(\det(\Sigma/n)\right)^{-1/2}\exp\left(-\frac{n}{2}(\theta-\hat{\theta})'\Sigma^{-1}(\theta-\hat{\theta})\right)$$

which implies that

$$\text{posterior}(\hat{\theta}) = (2\pi)^{-p/2}\left(\det(\Sigma/n)\right)^{-1/2}.$$

As a result

$$\begin{aligned}
\log\left[\frac{\text{posterior}(\hat{\theta})}{\text{prior}(\hat{\theta})}\right] &= \log\frac{(2\pi)^{-p/2}\left(\det(\Sigma/n)\right)^{-1/2}}{\text{prior}(\hat{\theta})} \\
&= \log\frac{(2\pi)^{-p/2}n^{p/2}\left(\det(\Sigma)\right)^{-1/2}}{\text{prior}(\hat{\theta})} \\
&= \frac{p}{2}(\log n) - \frac{p}{2}(\log(2\pi)) - \frac{1}{2}\log\det\Sigma - \log\text{prior}(\hat{\theta}) \\
&= \frac{p}{2}(\log n)\left\{1 - \frac{\frac{p}{2}(\log(2\pi)) + \frac{1}{2}\log\det\Sigma + \log\text{prior}(\hat{\theta})}{\frac{p}{2}(\log n)}\right\}.
\end{aligned}$$

Now if the sum of the terms $\frac{p}{2}(\log(2\pi))$, $\frac{1}{2}\log\det\Sigma$ and $\log\text{prior}(\hat{\theta})$ is small compared to $\frac{p}{2}\log n$:

$$\left|\frac{\frac{p}{2}(\log(2\pi)) + \frac{1}{2}\log\det\Sigma + \log\text{prior}(\hat{\theta})}{\frac{p}{2}(\log n)}\right| \ll 1, \tag{157}$$

then we can approximate the term in the parantheses by just 1 leading to

$$\log\left[\frac{\text{posterior}(\hat{\theta})}{\text{prior}(\hat{\theta})}\right] \approx \frac{p}{2}(\log n).$$

The formula (156) then simplifies to

$$-2\log\left(\text{Evidence}(M_1)\right) \approx -2 \times (\text{Maximized log-likelihood for } M_1) + p\log n. \tag{158}$$

The right hand side above is called the BIC (Bayesian Information Criterion). It is similar to the AIC with a more stringent penality for model complexity. As a result, BIC leads to smaller models compared to the AIC. Also note that because of (157), the formula (158) does not depend on the prior π making this convenient to use in practice.

22.5 Recommended Reading for Today

1. For a very good treatment of Bayesian Model Comparison, see Chapter 28 of the book *Information Theory, Inference and Learning Algorithms* by David MacKay, or Chapter 20 of the book *Probability Theory: the logic of science* by E. T. Jaynes.
2. The formulae (155) and (156) can be found in the 2010 paper titled *Bayesian system identification based on probability logic* by James L. Beck.

23 Lecture Twenty Three

23.1 Recap: Frequentist and Bayesian Model Selection

We studied frequentist and Bayesian methods for model selection in the last couple of classes. Frequentist methods aim to estimate the generalization accuracy of each model with the best parameter choices. For example, in the case of a model $Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} p_\theta$, frequentist methods aim to estimate generalization accuracy defined as:

$$\int f^*(y) \log p_{\hat{\theta}_n}(y) dy \tag{159}$$

where f^* denotes the true data generating density (it is assumed here that data are actually generated independently from the density f^*), and $\hat{\theta}_n$ denotes the MLE of θ. Several estimators exist for the generalization error. The AIC is constructed based on the following estimate of the generalization error:

$$\frac{1}{n} \sum_{i=1}^{n} \log p_{\hat{\theta}_n}(y_i) - \frac{p}{n}. \tag{160}$$

In fact, the AIC for the model is simply the above generalization accuracy estimate multiplied by constant factor $-2n$.

In practice, people often estimate the generalization accuracy (159) by splitting the observed dataset into two parts called training data and test data respectively, and then using the estimator:

$$\frac{1}{m} \sum_{j=1}^{m} \log p_{\hat{\theta}_{n-m}}(\tilde{y}_j) \tag{161}$$

where $\tilde{y}_1, \ldots, \tilde{y}_m$ denotes the test data and $\hat{\theta}_{n-m}$ is the MLE of θ computed from the training data. While this methodology is popular, there don't exist principled ways of doing the test-training split.

Bayesian Model Selection compares models in terms of the total probability each model assigns to the observed data. In the above context where the data is $y_1, \ldots, y_n$ and the model is $Y_1, \ldots, Y_n \overset{\text{i.i.d}}{\sim} p_\theta$, the model is evaluated via

$$f_{Y_1,\ldots,Y_n}(y_1, \ldots, y_n) = \int \left(\prod_{i=1}^{n} p_\theta(y_i) \right) f_\theta(\theta) d\theta$$

where f_θ denotes the prior distribution of θ. This marginal probability $f_{Y_1,\ldots,Y_n}(y_1, \ldots, y_n)$ is often referred to as the Evidence of the model. In the last class, we looked at the following

alternative formula for the evidence:

$$-2\log\left(\text{Evidence}(M)\right) = -2 \times (\text{Maximized log-likelihood for } M) + 2\log\left[\frac{\text{posterior}(\hat{\theta})}{\text{prior}(\hat{\theta})}\right] \tag{162}$$

We remarked that this formula bears some resemblance to the formula for the AIC.

The Evidence also has some connection to estimates of generalization accuracy such as (161). This is because we can decompose the Evidence as

$$\begin{aligned}\text{Evidence}(M) &= f_{Y_1,\ldots,Y_n}(y_1,\ldots,y_n) \\ &= f_{Y_1}(y_1) f_{Y_2|Y_1=y_1}(y_2) f_{Y_3|Y_1=y_1,Y_2=y_2}(y_3)\cdots f_{Y_n|Y_1=y_1,\ldots,Y_{n-1}=y_{n-1}}(y_n).\end{aligned} \tag{163}$$

Thus the Evidence is simply the product of all the predictive probabilities for each of the data points, using the model "trained" on the previous data points. Note that

$$f_{Y_i|Y_1=y_1,\ldots,Y_{i-1}=y_{i-1}}(y_i) = \int p_\theta(y_i) f_{\theta|Y_1=y_1,\ldots,Y_{i-1}=y_{i-1}}(\theta) d\theta.$$

When i is not too small, the posterior density $f_{\theta|Y_1=y_1,\ldots,Y_{i-1}=y_{i-1}}(\theta)$ should be peaked near the MLE based on the data $Y_1,\ldots,Y_{i-1}$ so this can be viewed as measuring the generalization accuracy of the MLE similar to (161).

The main issue that people have with Bayesian Model Selection is the reliance on the priors. The next section contains a simple example where the dependence on the prior can be seen explicitly.

23.2 Example: Normal Mean

We use Bayesian Model Selection to evaluate the following two models for the observed data $y_1,\ldots,y_n$:

$$\text{Model 1}: Y_1,\ldots Y_n \overset{\text{i.i.d}}{\sim} N(0,1),$$

and

$$\text{Model 2}: Y_1,\ldots,Y_n \overset{\text{i.i.d}}{\sim} N(\theta,1) \qquad \text{with } \theta \sim \text{unif}(-C,C).$$

The prior in Model 2 depends on the quantity C. We assume that C is large. The Evidence for M_1 is

$$\text{Evidence}(M_1) = \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(-\frac{1}{2}\sum_{i=1}^n y_i^2\right).$$

The Evidence for M_2 is

$$\begin{aligned}\text{Evidence}(M_2) &= \frac{1}{2C}\int_{-C}^{C}\left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(-\frac{1}{2}\sum_{i=1}^n (y_i-\theta)^2\right) d\theta \\ &= \frac{1}{2C}\left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(-\frac{1}{2}\sum_{i=1}^n (y_i-\bar{y})^2\right)\int_{-C}^{C}\exp\left(-\frac{n}{2}(\bar{y}-\theta)^2\right) d\theta.\end{aligned}$$

Because C is large, the limits $-C$ and C can be replaced by $-\infty$ and ∞ respectively without nontrivially changing the value of the integral above. Thus

$$\begin{aligned}\text{Evidence}(M_2) &\approx \frac{1}{2C}\left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(-\frac{1}{2}\sum_{i=1}^n (y_i-\bar{y})^2\right)\int_{-\infty}^{\infty}\exp\left(-\frac{n}{2}(\bar{y}-\theta)^2\right) d\theta \\ &= \frac{1}{2C}\left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(-\frac{1}{2}\sum_{i=1}^n (y_i-\bar{y})^2\right)\sqrt{\frac{2\pi}{n}}.\end{aligned}$$

The ratio of the two Evidences is thus:

$$\frac{\text{Evidence}(M_1)}{\text{Evidence}(M_2)} = 2C\sqrt{\frac{n}{2\pi}}\exp\left(\frac{1}{2}\sum_{i=1}^{n}(y_i-\bar{y})^2 - \frac{1}{2}\sum_{i=1}^{n}y_i^2\right)$$

which can be simplified to

$$\frac{\text{Evidence}(M_1)}{\text{Evidence}(M_2)} = 2C\sqrt{\frac{n}{2\pi}}\exp\left(\frac{-n\bar{y}^2}{2}\right) \qquad \text{where } \bar{y} := \frac{y_1+\cdots+y_n}{n}.$$

Note that if $\bar{y}$ is exactly equal to zero or is close to zero, then the factors C and $\sqrt{n}$ appearing in the formula above make the ratio of evidences quite large. Thus, when $\bar{y}$ is close to zero, the simpler model M_1 will be preferred. On the other hand, if $\bar{y}$ is far from zero, the factor of n appearing in the exponent of $\exp(-n\bar{y}^2/2)$ will make the evidence small. This, of course, is in line with intuition. If $\bar{y}$ is neither very close to zero nor very far from zero, then the value of C will be crucial for determining whether the ratio of evidences is larger or smaller than 1. This example shows how Bayes model selection based on evidences depends on the priors chosen in the individual models.

The ratio of the Evidence of model M_1 to the Evidence of model M_2 is often referred to as the *Bayes Factor* especially in the statistics literature (see, for example, `https://en.wikipedia.org/wiki/Bayes_factor`).

23.3 Application: Linear Regression

Let us now calculate the Evidence for a linear regression model under a natural choice of prior. These evidences can be used for comparing various linear regression models (such as those obtained by different choices of covariates) for the same dataset.

The observed dataset is $y_1, \ldots, y_n$. For each response value y_i, we also associate a $p \times 1$ covariate vector x_i. The response values $y_1, \ldots, y_n$ are usually placed in a $n \times 1$ vector denoted by Y. The covariate vectors are placed as rows of the $n \times p$ matrix X. We shall consider the matrix X to be deterministic. The linear model is given by

$$Y \sim N_n(X\beta, \sigma^2 I_n)$$

for two parameters β and σ^2. The parameter vector is $\theta = (\beta, \sigma)$. The Maximum Likelihood Estimate of θ is $\hat{\theta} := (\hat{\beta}, \hat{\sigma})$ where

$$\hat{\beta} := \operatorname*{argmin}_{\beta} \|Y - X\beta\|^2 = (X'X)^{-1}X'Y \quad \text{and} \quad \hat{\sigma} := \sqrt{\frac{1}{n}\|Y - X\hat{\beta}\|^2} = \frac{\|Y - X\hat{\beta}\|}{\sqrt{n}}$$

To make this into a Bayesian model, we need a prior on θ. Let us consider a generic prior $f_\theta(\theta)$ for now which will be specified shortly. The Evidence is then given by

$$\text{Evidence} = \int (2\pi)^{-n/2}\sigma^{-n}\exp\left(-\frac{\|Y - X\beta\|^2}{2\sigma^2}\right) f_\theta(\theta)d\theta$$

The likelihood function

$$\theta = (\beta, \sigma) \mapsto (2\pi)^{-n/2}\sigma^{-n}\exp\left(-\frac{\|Y - X\beta\|^2}{2\sigma^2}\right)$$

will have a single peak at $\hat{\theta}$ and usually the likelihood is concentrated around $\hat{\theta}$. The prior $f_\theta(\theta)$, on the other hand, will be quite diffuse. As a result, we can approximate the Evidence as

$$\begin{aligned}\text{Evidence} &= \int (2\pi)^{-n/2}\sigma^{-n}\exp\left(-\frac{\|Y-X\beta\|^2}{2\sigma^2}\right) f_\theta(\theta)d\theta \\ &\approx f_\theta(\hat{\theta})\int (2\pi)^{-n/2}\sigma^{-n}\exp\left(-\frac{\|Y-X\beta\|^2}{2\sigma^2}\right) d\theta.\end{aligned}$$

The integral above can be evaluated explicitly as

$$\begin{aligned}&\int\int (2\pi)^{-n/2}\sigma^{-n}\exp\left(-\frac{\|Y-X\beta\|^2}{2\sigma^2}\right) d\beta d\sigma \\ &= \int (2\pi)^{-n/2}\sigma^{-n}\exp\left(-\frac{\|Y-X\hat{\beta}\|^2}{2\sigma^2}\right)\left\{\int \exp\left(-\frac{\|X\hat{\beta}-X\beta\|^2}{2\sigma^2}\right) d\beta\right\} d\sigma \\ &= \int (2\pi)^{-n/2}\sigma^{-n}\exp\left(-\frac{\|Y-X\hat{\beta}\|^2}{2\sigma^2}\right)\left\{\int \exp\left(-\frac{(\beta-\hat{\beta})'X'X(\beta-\hat{\beta})}{2\sigma^2}\right) d\beta\right\} d\sigma \\ &= \int (2\pi)^{-n/2}\sigma^{-n}\exp\left(-\frac{\|Y-X\hat{\beta}\|^2}{2\sigma^2}\right)\left\{(\sqrt{2\pi}\sigma)^p|X'X|^{-1/2}\right\} d\sigma \\ &= (\sqrt{2\pi})^{-(n-p)}|X'X|^{-1/2}\int_0^\infty \sigma^{-(n-p)}\exp\left(-\frac{\|Y-X\hat{\beta}\|^2}{2\sigma^2}\right) d\sigma.\end{aligned}$$

Using the change of variable $\sigma = t^{-1/2}$, the above integral can be checked to equal:

$$\int_0^\infty \sigma^{-(n-p)}\exp\left(-\frac{\|Y-X\hat{\beta}\|^2}{2\sigma^2}\right) d\sigma = 2^{(n-p-3)/2}\frac{\Gamma(\frac{n-p-1}{2})}{\|Y-X\hat{\beta}\|^{n-p-1}}.$$

We have thus proved

$$\begin{aligned}\text{Evidence} &\approx f_\theta(\hat{\theta})(\sqrt{2\pi})^{-(n-p)}|X'X|^{-1/2}2^{(n-p-3)/2}\frac{\Gamma(\frac{n-p-1}{2})}{\|Y-X\hat{\beta}\|^{n-p-1}} \\ &= f_\theta(\hat{\theta})2^{-3/2}\pi^{-(n-p)/2}|X'X|^{-1/2}\frac{\Gamma(\frac{n-p-1}{2})}{\|Y-X\hat{\beta}\|^{n-p-1}}.\end{aligned}$$

We shall now specify the prior $f_\theta(\theta)$. We take β and σ to be independent with

$$\beta \sim N(0, \tau^2(X'X)^{-1}) \quad \text{and} \quad \log\sigma \sim \text{Unif}(-C, C).$$

This prior depends on the two hyperparameters τ and σ. The normality assumption for β is standard and facilitates computation. Note that we have taken the covariance to be proportional to $(X'X)^{-1}$ as opposed to the identity matrix. This is because usually the different components of β correspond to widely different covariates (e.g., X_1 might be age, X_2 might be current weight in pounds, X_3 might be weight a year ago in kilograms etc.). In such cases, we should not treat the different components in the same footing and $\beta \sim N(0, \tau^2(X'X)^{-1})$ is a more sensible assumption than $\beta \sim N(0, \tau^2 I_p)$. The prior $\beta \sim N(0, \tau^2(X'X)^{-1})$ is usually referred to as the Zellner prior. The uniform prior for $\log\sigma$ is quite standard. Thus

$$f_\theta(\theta) = (2\pi)^{-p/2}\tau^{-p}|X'X|^{1/2}\exp\left(-\frac{\beta'X'X\beta}{2\tau^2}\right)\frac{I\{e^{-C} < \sigma < e^C\}}{2C\sigma}.$$

Our formula for the Evidence then becomes

$$\text{Evidence} \approx 2^{-(p+3)/2}\pi^{-n/2}\tau^{-p}\exp\left(-\frac{\hat{\beta}'X'X\hat{\beta}}{2\tau^2}\right)\frac{\Gamma(\frac{n-p-1}{2})}{\|Y-X\hat{\beta}\|^{n-p-1}}\frac{I\{e^{-C}<\hat{\sigma}<e^{C}\}}{2C\hat{\sigma}}.$$

Plugging in the value of $\hat{\sigma} = n^{-1/2}\|Y - X\hat{\beta}\|$, we get

$$\text{Evidence}(\tau) \approx 2^{-(p+3)/2}\pi^{-n/2}\tau^{-p}\exp\left(-\frac{\hat{\beta}'X'X\hat{\beta}}{2\tau^2}\right)\frac{\sqrt{n}\Gamma(\frac{n-p-1}{2})}{\|Y-X\hat{\beta}\|^{n-p}}\frac{I\{e^{-C}<\hat{\sigma}<e^{C}\}}{2C}.$$

This quantity depends on the two prior hyperparameters τ and C. The dependence on C is not very problematic because the indicator term $I\{e^{-C} < \hat{\sigma} < e^{C}\}$ will always be positive as C is large, and the other factor $(1/(2C))$ will be common across the various linear regression models provided we choose the same value of C in every model. The dependence on τ is more sensitive however. This means that the probability assigned to the data by the Bayesian linear regression model depends sensitively on the parameter τ. For some values of τ, the probability of the observed data will be high and for some other values of τ, the probability of the observed data will be low. Furthermore, the values of τ where the probability of the observed data will be high (or low) will depend on the specific regression model (i.e., they will be different from one regression model to another, and this will have a bearing on the model selection problem).

To deal with this, the sensible way (from a Bayes perspective) is to take a prior on τ and then integrate the evidence formula above with respect to that prior. For a prior $f_\tau(\tau)$ on τ, the integrated evidence (with respect to f_τ) equals

$$\text{Evidence} = 2^{-(p+3)/2}\pi^{-n/2}\frac{\sqrt{n}\Gamma(\frac{n-p-1}{2})}{\|Y-X\hat{\beta}\|^{n-p}}\frac{I\{e^{-C}<\hat{\sigma}<e^{C}\}}{2C}\int_0^\infty \tau^{-p}\exp\left(-\frac{\hat{\beta}'X'X\hat{\beta}}{2\tau^2}\right)f_\tau(\tau)d\tau.$$

We take the prior

$$\log\tau \sim \text{Unif}(-C_1, C_1) \implies f_\tau(\tau) = \frac{I\{e^{-C_1}<\tau<e^{C_1}\}}{2C_1\tau}.$$

This leads to

$$\begin{aligned}\int_0^\infty \tau^{-p}\exp\left(-\frac{\hat{\beta}'X'X\hat{\beta}}{2\tau^2}\right)f_\tau(\tau)d\tau &= \frac{1}{2C_1}\int_{e^{-C_1}}^{e^{C_1}}\tau^{-p-1}\exp\left(-\frac{\|X\hat{\beta}\|^2}{2\tau^2}\right)d\tau \\ &\approx \frac{1}{2C_1}\int_0^\infty \tau^{-p-1}\exp\left(-\frac{\|X\hat{\beta}\|^2}{2\tau^2}\right)d\tau \\ &= \frac{2^{(p-2)/2}}{2C_1}\frac{\Gamma\left(\frac{p}{2}\right)}{\|X\hat{\beta}\|^p}.\end{aligned}$$

We thus have

$$\text{Evidence} \approx \frac{\pi^{-n/2}}{2^{7/2}}\frac{\sqrt{n}\Gamma(\frac{n-p-1}{2})}{\|Y-X\hat{\beta}\|^{n-p}}\frac{\Gamma\left(\frac{p}{2}\right)}{\|X\hat{\beta}\|^p}\frac{I\{e^{-C}<\hat{\sigma}<e^{C}\}}{4CC_1}$$

This formula depends on C and C_1. The indicator will usually equal 1. The rest of the formula is proportional to CC_1. If these constants are chosen to be equal across the different linear regression models, then all the evidences will be affected by C and C_1 in the same way. In that case, we can write (ignoring terms that do not depend on the particular regression model):

$$\text{Evidence} \propto \frac{\Gamma(\frac{n-p-1}{2})}{\|Y-X\hat{\beta}\|^{n-p}}\frac{\Gamma\left(\frac{p}{2}\right)}{\|X\hat{\beta}\|^p}$$

23.4 Recommended Reading for Today

1. For more comments on the relation between the Bayesian Evidence (163) and generalization accuracy estimates via cross validation, see David MacKay's Bayes FAQ webpage `http://www.inference.org.uk/mackay/Bayes_FAQ.html#gcv`. In particular, see MacKay's response to the question on the relation between Bayes and GCV.

2. The simple normal mean example in Section 23.2 is taken from Section 5.3 of the 1990 paper titled *From Laplace to Supernova SN 1987A: Bayesian Inference in Astrophysics* by Tom Loredo.

3. Model selection via calculations similar to Section 23.3 can be found in Chapter 5 of the book *Bayesian spectrum analysis and parameter estimation* by Larry Bretthorst.