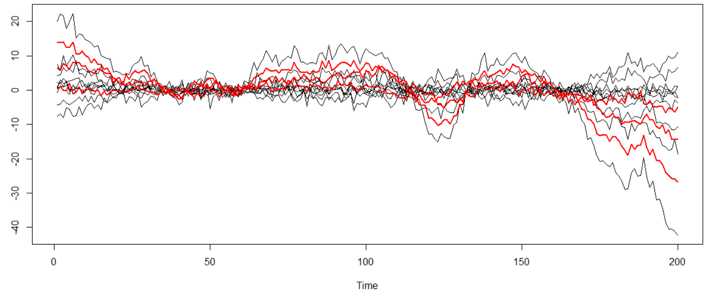
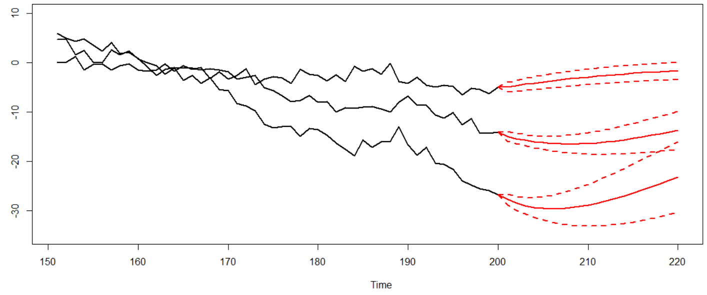
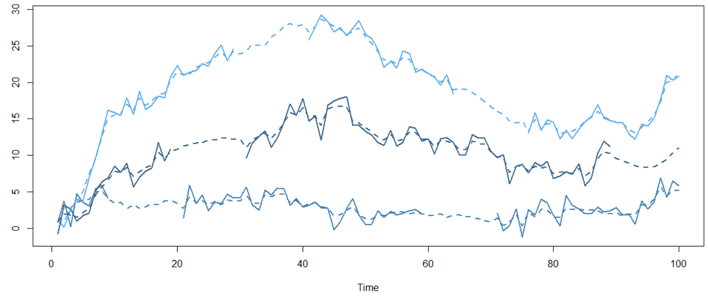
Practical Implementation of Dynamic Factor Models







Seton Leonard

Preliminary Draft

Contents

Introduction

This short book is primarily concerned with estimating factor models and is accompanied by R and C++/Armadillo code to implement the routines described herein. These estimation routines are: (1) maximum likelihood estimation of factor models following Watson and Engle (1983) and (2) Bayesian estimation of factor models by simulation. All routines are designed to accept noisy and/or missing data; extracting meaningful information from noisy data is, after all, the primary purpose of the filtering and smoothing algorithms upon which factor models are based. Dedicating a book to dynamic factor models may sound narrow, so it is worth noting at the outset that these models encompass a wide array of time series tools, and every model can be adapted to backcasting (predicting past information that was not observed), nowcasting (predicting contemporaneous unobserved information), and forecasting (predicting future realizations).

In its simplest form a dynamic factor model is described by two equations: a measurement equation

$$y_t = Hx_t + \varepsilon_t$$

and a transition equation

$$x_t = Bx_{t-1} + e_t$$

where y_t is observed noisy data, x_t are (typically) unobserved factors, H is a matrix of factor loadings, and B is a matrix of parameters that determines factor dynamics.

With a slight generalization on the above every model in this book can be described by these two equations: denote the measurement equation as

$$y_t = \tilde{H}z_t + \varepsilon_t \tag{1}$$

and the transition equation as

$$z_t = Az_{t-1} + e_t \tag{2}$$

In the above z_t is a vector of stacked factors such that $z_t = \begin{bmatrix} x'_t & x'_{t-1} & \dots & x'_{t-p+1} \end{bmatrix}'$ where p is the number of lags in the model. $\tilde{H} = HJ$ where H is a matrix of loadings and J is a helper matrix. A is the companion form of the more familiar VAR model

$$x_t = B \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ \vdots \\ x_{t-p} \end{bmatrix} + e_t \tag{3}$$

Thus equipped we could, for example, write down a VAR model for noisy and/or missing data. Taking the a simple three variable, three lag example we would have the observation equation

$$\underbrace{\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ y_{3,t} \end{bmatrix}}_{y_t} = \underbrace{\begin{bmatrix} I_3 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} I_3 & 0_3 & 0_3 \end{bmatrix}}_{J} \underbrace{\begin{bmatrix} x_t \\ x_{t-1} \\ x_{t-2} \end{bmatrix}}_{z_t} + \varepsilon_t$$

and the transition equation

$$\underbrace{\begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{bmatrix}}_{x_t} = \underbrace{\begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} x_{t-1} \\ x_{t-2} \\ x_{t-3} \end{bmatrix}}_{z_{t-1}} + e_t$$

Since we already know H in this case the parameters to estimate are B and the covariance matrices $\varepsilon_t \sim \mathcal{N}\Big([0], R\Big)$ and $e_t \sim \mathcal{N}\Big([0], Q\Big)$.[1]

As a second example, one way to write an ARMA(1,1) model

$$y_t - a_1 y_{t-1} = \varepsilon_t + \theta\varepsilon_{t-1}$$

in state space form would be with the observation equation

$$y_t = x_t$$

and the transition equation

$$\begin{bmatrix} x_t \\ \varepsilon_t \end{bmatrix} = \begin{bmatrix} a & \theta \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ \varepsilon_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_t \\ \varepsilon_t \end{bmatrix}$$

though the state space form of any model is never unique. As an aside, this formulation of the ARMA(1,1) would make it easy to allow for measurement error in y_t by simply including an error term in the observation equation, that is, letting $y_t = x_t + \varepsilon_t$.

[1] Note that I am a bit loose with the dimensionality of e_t. In the case we're using the companion form of the transition equation e_t just has a bunch of zeros ($m \times (p-1)$ of them) under the first m elements.

This book begins with some basic Bayesian statistics in chapter 1 that form the building blocks of our state space framework. Chapter 2 then introduces basic Kalman filtering and smoothing, followed in chapter 3 by Durbin and Koopman (2012)'s much more computationally efficient disturbance smoothing. Chapter 4 on deals with estimating state space models. Chapter 4 briefly discusses estimation using principal components, though other approaches (maximum likelihood or Bayesian simulation) tend to yield better results. Chapter 5 presents Watson and Engle (1983)'s maximum likelihood approach (I do not cover simply plugging the likelihood function into a numerical maximization (minimization) routine. It's easy to do but very slow compared with Watson and Engle (1983)'s EM algorithm). Chapter 6 presents Bayesian estimation by simulation for uniform frequency data. Finally, chapter 7 introduces mixed frequency models.

All of the notation in this book matches that in my R and C++ code so that hopefully the code is easy to follow. This text is intentionally short and too the point, geared towards implementation of state space models as opposed to derivations or theory. To get a better handle and more depth on factor models, Bayesian filtering and smoothing, and time series analysis more generally Sarkka (2013), Durbin and Koopman (2012), Hamilton (1994), Lutkepohl (2007), and Koop et al. (2007) are good resources. As is no doubt already obvious I assume the reader is familiar with matrix algebra. I also assume some familiarity with statistics, econometrics, and calculus.

Chapter 1

The Necessary Bayesian Statistics

Bayesian econometrics is founded on Bayes' theorem, a simple equation which states that the probability of an event A conditional on B equals the probability of B conditional on A times the unconditional probability of A over the unconditional probability of B, that is,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1.1}$$

This result can be easily derived from the definition of conditional probability, which states that the probability of A conditional on B equals the probability of A and B occurring together over the probability of B, that is

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{1.2}$$

Using equation 1.2 we have

$$P(A|B)P(B) = P(A \cap B)$$

and

$$P(B|A)P(A) = P(A \cap B)$$

which, when combined, yields equation 1.1.

We can also write Bayes' theorem in terms of probability density functions (or probability mass functions for discrete distributions) by again using the definition of conditional probability

$$f(y|x) = \frac{f(x,y)}{f(x)} \tag{1.3}$$

That is, the distribution of y conditional on x equals the joint distribution of x and y over the distribution of x. The derivation of Bayes' theorem, which we use in estimating the distribution of parameters conditional on observed data (called the posterior distribution of the parameters), is the same as above:

$$f(y|x)f(x) = f(x, y)$$

and

$$f(x|y)f(y) = f(x, y)$$

thus

$$f(y|x) = \frac{f(x|y)f(y)}{f(x)} \tag{1.4}$$

In Bayesian jargon, $f(y)$ is our prior distribution (or just prior) for y and $f(y|x)$ is our posterior distribution (or just posterior) for y. The distribution $f(x|y)$ is the conditional distribution of x given y (for example the distribution of the data x conditional on the parameters y) and $f(x)$ is the marginal distribution of x, typically calculated as $f(x) = \int f(x, y)dy = \int f(x|y)f(y)dy$.

1.1 Basic Examples

Example 1: Testing for Illnesses

Testing for illnesses provides a morbid but illustrative example of Bayes rule. Suppose 1% of the population has an illness (but that there are no observable symptoms), that if an individual has the illness he always tests positive, and that the probability of a false positive is 10%. Denoting $A = 1$ as having the illness and $B = 1$ as testing positive then

$$A = \begin{cases} 1 & \textit{with prob. } 0.01 \\ 0 & \textit{with prob. } 0.99 \end{cases}$$

and

$$(B|A = 1) = \begin{cases} 1 & \textit{with prob. } 1 \\ 0 & \textit{with prob. } 0 \end{cases} \qquad (B|A = 0) = \begin{cases} 1 & \textit{with prob. } .1 \\ 0 & \textit{with prob. } .9 \end{cases}$$

Then

$$P(B = 1|A = 1)P(A = 1) = 1 \times 0.01$$

and

$$\begin{aligned} P(B = 1) &= P(B = 1|A = 1)P(A = 1) + P(B = 1|A = 0)P(A = 0) \\ &= 1 \times 0.01 + 0.1 \times 0.99 \\ &= 0.109 \end{aligned}$$

thus the probability that an individual who tests positive once is in fact ill is

$$\begin{aligned} P(A = 1|B = 1) &= \tfrac{P(B=1|A=1)P(A=1)}{P(B=1)} \\ &= \tfrac{0.01}{0.109} \\ &= 0.0917 \end{aligned}$$

If an individual tests positive twice, the probability that he is in fact ill is

$$\begin{aligned} P(A=1|B=1,1) &= \tfrac{P(B=1,1|A=1)P(A=1)}{P(B=1,1)} \\ &= \tfrac{0.01}{0.1\times0.1\times0.99+1\times1\times0.01} \\ &= 0.5025 \end{aligned}$$

Example 2: Normal data with a Bernoulli prior

This question was on Mark Watson's Gerzensee midterm.[1] Suppose some data Y follows a normal distribution with mean θ and standard deviation 1 and that θ follows a Bernoulli distribution with $p = 0.5$, that is, we know θ is either 1 or 0 and we attach equal probability to each outcome. Suppose we observe a single draw of $Y = 0$. What is our posterior for θ?

To answer this question we need the fact that

$$f_Y(Y|\theta) = \frac{1}{\sqrt{2\pi}} \exp\Big\{ -\frac{1}{2}(Y-\theta)^2 \Big\}$$

$$f_\theta(\theta) = \begin{cases} 0 & \textit{with prob. } 0.5 \\ 1 & \textit{with prob. } 0.5 \end{cases}$$

Then evaluating $f_Y(0|\theta=0) = \frac{1}{\sqrt{2\pi}}$, $f_Y(0|\theta=1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}}$ we have that the denominator in Bayes rule for $Y = 0$ is

$$\begin{aligned} f_Y(0) &= f_Y(0|\theta=0)f_\theta(0) + f_Y(0|\theta=1)f_\theta(0) \\ &= \tfrac{1}{\sqrt{2\pi}}\tfrac{1}{2} + \tfrac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}}\tfrac{1}{2} \end{aligned}$$

The numerator (the conditional distribution times our prior) is, for $\theta = 0$ (recall θ can only take values 0 or 1)

$$f_Y(0|\theta=0)f(\theta) = \frac{1}{\sqrt{2\pi}}\frac{1}{2}$$

thus for $\theta = 0$

$$f_\theta(0|Y=0) = \frac{\frac{1}{\sqrt{2\pi}}\frac{1}{2}}{\frac{1}{\sqrt{2\pi}}\frac{1}{2} + \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}}\frac{1}{2}} = \frac{1}{1+e^{-\frac{1}{2}}}$$

For $\theta = 1$

$$f_\theta(1|Y=0) = \frac{\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}}\frac{1}{2}}{\frac{1}{\sqrt{2\pi}}\frac{1}{2} + \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}}\frac{1}{2}} = \frac{e^{-\frac{1}{2}}}{1+e^{-\frac{1}{2}}}$$

thus our posterior distribution for θ is

$$f(\theta|Y=0) = \begin{cases} 0 & \textit{with prob. } \frac{1}{1+e^{-\frac{1}{2}}} \\ 1 & \textit{with prob. } \frac{e^{-\frac{1}{2}}}{1+e^{-\frac{1}{2}}} \end{cases}$$

[1] Gerzensee is an economics program for first year Swiss PhDs.

Example 3: Normal-Normal Conjugate Density

Suppose we believe a parameter θ follows a normal distribution

$$f_p(\theta) = \frac{\lambda}{\sqrt{2\pi}} \exp\left\{-\frac{\lambda^2}{2}(\theta - \theta_0)^2\right\}$$

where θ_0 is our prior mean and $\frac{1}{\lambda}$ our prior standard deviation. Suppose also that we observe a vector X that follows a normal distribution

$$f_X(X|\theta) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left\{-\frac{1}{2\sigma^2}\sum_i (x_i - \theta)^2\right\}$$

which we could alternatively write as

$$f_X(X|\theta) = (2\pi)^{-\frac{N}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(X-\theta)'\Sigma^{-1}(X-\theta)\right\}$$

where $\Sigma = \sigma^2 I_N$ and N is the number of observations or elements of X (Note here that I have assumed the standard deviation of the data σ is known. This assumption facilitates deriving and computing the posterior for θ, but we will properly specify a prior for both θ and σ in section 1.2). To derive the posterior distribution of θ it is sufficient to show that this distribution also has the form of a normal; we need not worry about the constant $f_X(X)$ in the denominator of Bayes rule as this only ensures that the resulting distribution sums to one, that is, $\int_{-\infty}^{\infty} f_\theta(\theta|X)d\theta = 1$. Looking only at the numerator of Bayes rule, $f_X(X|\theta)f_p(\theta)$, we have

$$\begin{aligned} f_\theta(\theta|X) &\propto \exp\left\{-\tfrac{\lambda^2}{2}(\theta-\theta_0)^2\right\}\exp\left\{-\tfrac{1}{2\sigma^2}\textstyle\sum_i (x_i-\theta)^2\right\} \\ &\propto \exp\left\{-\tfrac{1}{2}\Big[\lambda^2(\theta^2 - 2\theta\theta_0 + \theta_0^2) + \tfrac{1}{\sigma^2}(\textstyle\sum_i x_i^2 - 2\sum_i x_i\theta + N\theta^2)\Big]\right\} \end{aligned}$$

In the second term above the only parts we care about are those which relate to the parameter θ; the rest goes into the constant of integration which ensures $\int_{-\infty}^{\infty} f(\theta|X)d\theta = 1$, thus we can simply write the above term as

$$\begin{aligned} f_\theta(\theta|X) &\propto \exp\left\{-\tfrac{1}{2}\Big[\lambda^2(\theta^2 - 2\theta\theta_0) + \tfrac{1}{\sigma^2}(-2\textstyle\sum_i x_i\theta + N\theta^2)\Big]\right\} \\ &= \exp\left\{-\tfrac{1}{2\sigma^2}\Big[(\lambda\sigma)^2(\theta^2 - 2\theta\theta_0) - 2\textstyle\sum_i x_i\theta + N\theta^2\Big]\right\} \end{aligned} \tag{1.5}$$

The trick now is to try and re-write equation 1.5 as a normal distribution. The easiest way to proceed is to guess and check (assuming, of course, that we can make a good guess of the posterior). To this end suppose

$$f_\theta(\theta|X) \propto \exp\left\{-\tfrac{N+(\lambda\sigma)^2}{2\sigma^2}\Big[\theta - \big(N + (\lambda\sigma)^2\big)^{-1}\big(\textstyle\sum_i x_i + (\lambda\sigma)^2\theta_0\big)\Big]^2\right\} \tag{1.6}$$

Multiplying out equation 1.6 we have

$$f_\theta(\theta|X) \quad\propto\quad \exp\Big\{-\tfrac{1}{2\sigma^2}\Big[\big(N+(\lambda\sigma)^2\big)\theta^2 - 2\theta\big(\textstyle\sum_i x_i + (\lambda\sigma)^2\theta_0\big) + \frac{\sum_i x_i+\theta_0}{N+(\lambda\sigma)^2}\Big]\Big\}$$

Again, since the term $\frac{\sum_i x_i+\theta_0}{N+(\lambda\sigma)^2}$ does not contain our parameter of interest θ we can dump it into the constant of integration and keep only the relevant part of the above term,

$$\begin{aligned} f_\theta(\theta|X) &\propto \exp\Big\{-\tfrac{1}{2\sigma^2}\Big[\big(N+(\lambda\sigma)^2\big)\theta^2 - 2\theta\big(\textstyle\sum_i x_i + (\lambda\sigma)^2\theta_0\big)\Big]\Big\} \\ &= \exp\Big\{-\tfrac{1}{2\sigma^2}\Big[(\lambda\sigma)^2(\theta^2-2\theta\theta_0) - 2\textstyle\sum_i x_i\theta + N\theta^2\Big]\Big\} \end{aligned}$$

which is equation 1.5. Thus our posterior distribution of θ given the realization of the data X is also normal with mean $\big(N+\big(\lambda\sigma\big)^2\big)^{-1}\big(\sum_i x_i + (\lambda\sigma)^2\theta_0\big)$ and variance $\frac{\sigma^2}{N+(\lambda\sigma)^2}$. Notice that as the standard deviation of our prior becomes smaller our posterior variance also shrinks (λ is therefore sometimes referred to as a "shrinkage parameter") but our posterior mean becomes biased. In the extreme case that $\lambda \to \infty$ our posterior mean is our prior mean θ_0 and our posterior variance is zero. Thus by specifying a prior we are reducing the variance of our parameter estimates at the cost of introducing a bias.

1.2 Bayesian Linear Regression

Bayesian Linear Regressions will become a sort of all purpose tool with which we will estimate both the observation equation and transition equation of our factor model when we come to estimation by simulation in section 6. Our model in this section is a simple linear regression. In the univariate case we have

$$y = X\beta + \varepsilon \tag{1.7}$$

where y, the dependent variable, is a $T \times 1$ vector, X is a $T \times m$ matrix of explanatory variables, β is a $m \times 1$ vector of unknown parameters, and ε is a $T \times 1$ vector of shocks with distribution[2]

$$\varepsilon_t \sim \mathcal{N}(0,\sigma)$$

In the multivariate case our model is

$$Y = XB' + \varepsilon \tag{1.8}$$

in which Y is a $T \times k$ matrix of k dependent variables (observations t are indexed by row though the index here need not necessarily be time), X is a $T \times m$ matrix of m explanatory

[2] I use σ here to denote variance *not* standard deviation as is often the case since for the multivariate model Σ also denotes variance.

variables, B a $m \times k$ matrix of unknown parameters, and ε_t is a $T \times k$ matrix of shocks with distribution

$$\varepsilon_t \sim \mathcal{N}\Big([0], \Sigma\Big)$$

In each section below I begin with the derivation of the posterior when we assume the covariance of shocks is know before moving to normal-inverse gamma or normal-inverse Wishart conjugate priors which include posteriors for the covariance of shocks.

Simple Univariate Linear Regression

Assuming initially that the distribution of shocks is known and beginning with the univariate case our prior for β is[3]

$$\pi(\beta) \propto \exp\left\{-\frac{1}{2\sigma}(\beta-\beta_0)'\Lambda_0(\beta-\beta_0)\right\}$$

where β_0 is our prior for β and Λ_0, our prior covariance of β, defines the strength of our prior beliefs. The distribution for our model in (1.7) is

$$f(y|\beta,\sigma) \propto \exp\left\{-\frac{1}{2\sigma}(y-X\beta)'(y-X\beta)\right\}$$

so that our posterior is

$$f(\beta|y,X,\sigma) \propto \exp\left\{-\frac{1}{2\sigma}\left[(y-X\beta)'(y-X\beta)+(\beta-\beta_0)'\Lambda_0(\beta-\beta_0)\right]\right\}$$

Multiplying out the terms in square brackets above an ignoring those which do not contain β and thus become part of the constant of integration we have

$$f(\beta|y,X,\sigma) \propto -2y'X\beta + \beta'X'X\beta + \beta'\Lambda_0\beta - 2\beta'\Lambda_0\beta_0 \tag{1.9}$$

Letting $\Lambda_n = (X'X + \Lambda_0)$ and again only keeping track of terms which do not form part of our constant of integration we have

$$\begin{aligned} f(\beta|y,X,\sigma) &\propto \exp\left\{-\tfrac{1}{2\sigma}\left[\left(\beta-\Lambda_n^{-1}(X'y+\Lambda_0\beta_0)\right)'\Lambda_n\left(\beta-\Lambda_n^{-1}(X'y+\Lambda_0\beta_0)\right)\right]\right\} \\ &\propto \exp\left\{-\tfrac{1}{2\sigma}\left[\beta'X'X\beta+\beta'\Lambda_0\beta-2\beta'X'y-2\beta'\Lambda_0\beta_0\right]\right\} \end{aligned}$$

These terms match those in equation (1.9), thus our posterior for β is distributed

$$\beta \sim \mathcal{N}\Big((X'X+\Lambda_0)^{-1}(X'y+\Lambda_0\beta_0), \sigma(X'X+\Lambda_0)^{-1}\Big) \tag{1.10}$$

This derivation assumes we know σ already, which of course will not be the case in practice.

[3] Specifying the variance of our prior in terms of the variance of shocks keeps the algebra much cleaner but is not strictly necessary. Also note that like Σ, σ denotes variance, not standard deviation as is often the case.

Simple Multivariate Linear Regression

To derive the posterior for our multivariate linear regression when we assume the distribution of shocks to the model $\varepsilon_t \sim \mathcal{N}\Big(0, \sigma\Big)$ is known begin by writing the model for an observation t as

$$y_t = Bx_t + \varepsilon_t$$

and define the vectorization of B as

$$\beta = vec(B)$$

Our prior for β is

$$\pi(\beta|\sigma) \propto \exp\Bigg\{ -\frac{1}{2}(\beta - \beta_0)'(V_0)^{-1}(\beta - \beta_0)\Bigg\}$$

where $V_0 = \Lambda \otimes \sigma$ and Λ determines our prior tightness, that is, the strength of our prior beliefs. The distribution of our model requires a few more definitions. Let y be the vector of observations y_t stacked over time (that is, $y = vec(Y')$) and $\mathbf{X}$ be the associated matrix of explanatory variables. Explicitly, $\mathbf{X} = X \otimes I_k$ where k is the number of observed variables in y_t. Then

$$f(y|\beta, \sigma, X) \propto \exp\left\{-\frac{1}{2}(y - \mathbf{X}\beta)'\Sigma^{-1}(y - \mathbf{X}\beta)\right\}$$

where $\Sigma^{-1} = I_T \otimes \sigma^{-1}$. Our posterior is then

$$f(\beta|X, y, \sigma) \propto \exp\Bigg\{ -\frac{1}{2}\Big[(\beta - \beta_0)'(V_0)^{-1}(\beta - \beta_0) + (y - \mathbf{X}\beta)'\Sigma^{-1}(y - \mathbf{X}\beta)\Big]\Bigg\}$$

Multiplying this expression out (and ignoring terms that don't contain β) leaves us with

$$\begin{aligned} f(\beta|X, y, \sigma) \quad &\propto \quad \exp\Big\{ -\tfrac{1}{2}\Big[\beta'(V_0)^{-1}\beta - 2y'\Sigma^{-1}\mathbf{X}\beta \\ &+ \quad \beta'\mathbf{X}'\Sigma^{-1}\mathbf{X}\beta - 2\beta'(V_0)^{-1}\beta_0\Big]\Big\} \end{aligned}$$

Using the property for Kronecker products that for conformable matrices $(A \otimes B)(C \otimes D) = AC \otimes BD$ this simplifies to

$$f(\beta|X, y, \sigma) \propto \exp\Bigg\{ -\frac{1}{2}\Big[\beta'(V_0)^{-1}\beta - 2y'(X \otimes \sigma^{-1})\beta + \beta'(X'X \otimes \sigma^{-1}) - 2\beta'(V_0)^{-1}\beta_0\Big]\Bigg\}$$

in which case we can write the posterior as

$$f(\beta|X, y, \sigma) \propto \exp\left\{-\frac{1}{2}\Big(\beta - V_1^{-1}\big((X' \otimes \sigma^{-1})y + V_0\beta_0\big)\Big)' V_1 \Big(\beta - V_1^{-1}\big((X' \otimes \sigma^{-1})y + V_0\beta_0\big)\Big)\right\}$$

where

$$V_1 = (X'X \otimes \sigma^{-1} + V_0^{-1})$$

or, using our definition of $V_0 = \Lambda \otimes \sigma$,

$$V_1 = (X'X + \Lambda) \otimes \sigma^{-1}$$

That is, our posterior for β is normally distributed

$$\beta \sim \mathcal{N}\Big((X'X + \Lambda)^{-1}(X'y + \Lambda\beta_0), V_1^{-1}\Big)$$

In the future we can use a matrix normal distribution for our multivariate normal models which avoids the hassle of having to vectorize and then simplify our model, though it is hopefully useful to illustrate the vectorized version at least once.

Univariate Linear Regression with Normal-Inverse Gamma Prior

The previous two subsections have assumed the variances of shocks to our models are known. This simplifies the derivations but is not a very realistic assumption. These variances and, in the multivariate case, covariance matrices are particularly important when we come to filtering and smoothing in chapter 3 as they determine how aggressively we should update our estimates of unobserved factors based on observables. Getting good estimates of the scale of shocks is therefore a high priority. My conjugate prior (meaning the prior and posterior have the same form) for the univariate case models the parameters β as normally distributed while my prior for σ follows an inverse gamma distribution. Explicitly,

$$\pi(\beta|\sigma) \propto (\sigma)^{-\frac{k}{2}} \exp\Bigg\{ -\frac{1}{2\sigma}(\beta - \beta_0)'\Lambda_0(\beta - \beta_0)\Bigg\}$$

and

$$\pi(\sigma) \propto \sigma^{\frac{\nu_0}{2}-1} \exp\left\{-\frac{s_0}{2\sigma}\right\}$$

In the above inverse gamma distribution s_0 is our prior scale parameter. Loosely interpreted, this is our prior for the residual sum of squares in our model. ν_0 is our prior scale parameter, which we can interpret as the number of "prior observations". Increasing ν_0 will force our posterior for σ towards $\frac{1}{\nu_0}s_0$.

As we model innovations ε_t as normally distributed, our model has the form

$$f(y|\beta, \sigma, X) \propto \sigma^{-\frac{T}{2}} \exp\left\{-\frac{1}{2\sigma}(y - X\beta)'(y - X\beta)\right\}$$

so our posterior is

$$\begin{aligned} f(\beta, \sigma|y, X) \quad \propto \quad & \underbrace{\sigma^{-\frac{T}{2}} \exp\left\{-\frac{1}{2\sigma}(y - X\beta)'(y - X\beta)\right\}}_{f(y|\beta,\sigma,X)} \times \\ & \underbrace{(\sigma)^{-\frac{k}{2}} \exp\Bigg\{ -\frac{1}{2\sigma}(\beta - \beta_0)'\Lambda_0(\beta - \beta_0)\Bigg\}}_{\pi(\beta|\sigma)} \underbrace{\sigma^{\frac{\nu_0}{2}-1} \exp\left\{-\frac{s_0}{2\sigma}\right\}}_{\pi(\sigma)} \end{aligned}$$

Beginning by multiplying out the exponential terms we have

$$-\frac{1}{2\sigma}\Big[y'y - 2y'X\beta + \beta'X'X\beta + \beta'\Lambda_0\beta - 2\beta'\Lambda_0\beta_0 + \beta_0'\Lambda_0\beta_0 + s_0\Big] \tag{1.11}$$

If we define $\Lambda_T \equiv X'X + \Lambda_0$ then

$$\begin{array}{lr} \big(\beta - \Lambda_T^{-1}(X'y + \Lambda_0\beta_0)\big)'\Lambda_T\big(\beta - \Lambda_T^{-1}(X'y + \Lambda_0\beta_0) & = \\ \beta'\Lambda_T\beta - 2\beta'(X'y + \Lambda_0\beta_0) + (X'y + \Lambda_0\beta_0)'\Lambda_T^{-1}(X'y + \Lambda_0\beta_0) & \end{array} \tag{1.12}$$

Comparing terms in 1.12 and 1.11 shows that we can re-write the exponential terms of the posterior as

$$\begin{array}{lr} -\frac{1}{2\sigma}\Big[\big(\beta - \Lambda_T^{-1}(X'y + \Lambda_0\beta_0)\big)'\Lambda_T\big(\beta - \Lambda_T^{-1}(X'y + \Lambda_0\beta_0)\big) & + \\ y'y - (X'y + \Lambda_0\beta_0)'\Lambda_T^{-1}(X'y + \Lambda_0\beta_0) + \beta_0'\Lambda_0\beta_0 + s_0\Big] & \end{array} \tag{1.13}$$

The first line in 1.13 forms the normal part of our normal-inverse gamma posterior; the second line contains terms that will go into the posterior scale parameter of the inverse gamma distribution. We can re-write the scale parameter for the posterior by noting that the posterior mean for β is $\beta_T = \Lambda_T^{-1}(X'y + \Lambda_0\beta_0)\big)$. Thus

$$\begin{array}{lr} (y - X\beta_T)'(y - X\beta_T) + (\beta_T - \beta_0)'\Lambda_0(\beta_T - \beta_0) & = \\ y'y - 2y'X\beta_T + \beta_T X'X\beta_T + \beta_T\Lambda_0\beta_T - 2\beta_T'\Lambda_0\beta_0 + \beta_0'\Lambda_0\beta_0 & = \\ y'y - 2(y'X + \beta_0'\Lambda_0)\beta_T + \beta_T'\underbrace{(X'X + \Lambda_0)}_{\Lambda_T}\beta_T + \beta_0'\Lambda_0\beta_0 & \end{array} \tag{1.14}$$

Using the definitions of β_T and Λ_T

$$\beta_T'\Lambda_T\beta_T - (y'X + \beta_0'\Lambda_0)\beta_T = 0$$

so that collecting all the terms in our posterior we have

$$\begin{array}{rcl} f(\beta, \sigma|y, X) & \propto & \sigma^{-\frac{k}{2}} \exp\Big\{-\frac{1}{2\sigma}\big(\beta - \Lambda_T^{-1}(X'y + \Lambda_0\beta_0)\big)'\Lambda_T\big(\beta - \Lambda_T^{-1}(X'y + \Lambda_0\beta_0)\big)\Big\} \\ & \times & \sigma^{-\frac{\nu_0+T}{2}-1} \exp\big\{-\frac{1}{2\sigma}\big((y - X\beta_T)'(y - X\beta_T) + (\beta_T - \beta_0)'\Lambda_0(\beta_T - \beta_0) + s_0\big)\big\} \end{array}$$

That is, our posterior is a normal-inverse gamma distribution such that

$$f(\beta|\sigma, X, y) \sim \mathcal{N}\Big(\Lambda_T^{-1}(X'y + \Lambda_0\beta_0), \sigma\Lambda_T^{-1}\Big)$$

where $\Lambda_T = (X'X + \Lambda_0)$ and $f(\sigma|X, y)$ follows an inverse gamma distribution with scale parameter $s_T = (y - X\beta_T)'(y - X\beta_T) + (\beta_T - \beta_0)'\Lambda_0(\beta_T - \beta_0) + s_0$ and $\nu_T = T + \nu_0$.

Multivariate Linear Regression with Normal-Inverse Wishart Prior

To derive the posterior for our multivariate normal model

$$Y_t = B'X_t + \varepsilon_t$$

or, for all observations,

$$Y = XB + \varepsilon$$

we will use the multivariate equivalent of our normal-inverse gamma conjugate prior for the univariate case, that is, a matrix normal-inverse Wishart distribution.[4] Accordingly, our priors are

$$\pi(\Sigma) \sim \mathcal{IW}\big(V_0, \nu_0\big)$$

and

$$\pi(B|\Sigma) \sim \mathcal{MN}\big(B, \Lambda_0, \Sigma\big)$$

which is equivalent to the vectorised form where $\beta = vec(B')$[5]

$$\pi(\beta|\Sigma) \sim N(\beta_0, \Lambda_0 \otimes \Sigma)$$

Given these priors and the distribution for our model

$$f(Y|B, \Sigma, X) \sim \mathcal{MN}\big(XB, I_T, \Sigma\big)$$

we can write our posterior as

$$\begin{aligned} f(B, \Sigma|Y, X) \quad \propto \quad & \underbrace{|\Sigma|^{-(\nu_0+k+1)/2} \exp\Big\{-\frac{1}{2}tr(V_0\Sigma^{-1})\Big\}}_{\pi(\Sigma)} \\ \times \quad & \underbrace{|\Sigma|^{-m/2} \exp\Big\{-\frac{1}{2}tr\big((B - B_0)'\Lambda_0(B - B_0)\Sigma^{-1}\big)\Big\}}_{\pi(B|\Sigma)} \\ \times \quad & \underbrace{|\Sigma|^{-T/2} \exp\Big\{-\frac{1}{2}tr\big((Y - XB)'(Y - XB)\Sigma^{-1}\big)\Big\}}_{f(Y|B,\Sigma,X)} \end{aligned}$$

As previously, the trick is to write our posterior as a sum of squares. Using the fact that $tr(A)tr(B) = tr(A + B)$ and dealing first with exponential terms, our result is the matrix equivalent to equation (1.11):

$$-\frac{1}{2}\Big[tr\Big(\big(Y'Y - 2Y'XB + B'X'XB + B'\Lambda_0 B - 2B_0'\Lambda_0 B + B_0'\Lambda_0 B_0 + V_0\big)\Sigma^{-1}\Big)\Big] \tag{1.15}$$

[4] Note that in this section I will write B as its transpose for ease of notation. That is, B in this section corresponds to its transpose elsewhere. This is due to the fact that we will write our stacked model of observations as $Y = XB + \varepsilon$.

[5] I use the vectorized form $\beta = vec(B')$ as this stacks B over rows, thus the resulting vector β_0 corresponds to the covariance matrix $\Lambda_0 \otimes \Sigma$ and to $\mathbf{X} = X \otimes I_k$ as defined in the derivation for a simple normal-normal multivariate linear regression.

As previously, define $\Lambda_T = X'X + \Lambda_0$. We can again propose the sum of squares portion of our solution as

$$\begin{aligned}(1.16) \qquad & \big(B - \Lambda_T^{-1}(X'Y + \Lambda_0 B_0)\big)'\Lambda_T\big(B - \Lambda_T^{-1}(X'Y + \Lambda_0 B_0)\big) \qquad = \\ & B'\Lambda_T B - 2B'(X'Y + \Lambda_0 B_0) + (X'Y + \Lambda_0 B_0)'\Lambda_T^{-1}(X'Y + \Lambda_0 B_0)\end{aligned}$$

Again, comparing terms in 1.16 and 1.15 shows that we can re-write the exponential terms of the posterior as

$$\begin{aligned}(1.17) \qquad & -\tfrac{1}{2}tr\Bigg[\Big(\big(B - \Lambda_T^{-1}(X'Y + \Lambda_0 B_0)\big)'\Lambda_T\big(B - \Lambda_T^{-1}(X'Y + \Lambda_0 B_0)\big) \qquad + \\ & Y'Y - (X'Y + \Lambda_0 B_0)'\Lambda_T^{-1}(X'Y + \Lambda_0 B_0) + B_0'\Lambda_0 B_0 + V_0\Big)\Sigma^{-1}\Bigg]\end{aligned}$$

so that our final result is the matrix equivalent to our result in the univariate case, that is,

$$\begin{aligned}(1.18) \qquad f(B,\Sigma|Y,X) \;\propto\; & |\Sigma|^{-m/2}\exp\Big\{-\tfrac{1}{2}tr\Big[\Big(\big(B - \Lambda_T^{-1}(X'Y + \Lambda_0 B_0)\big)'\Lambda_T\big(B - \Lambda_T^{-1}(X'Y + \Lambda_0 B_0)\big)\Big)\Sigma^{-1}\Big]\Big\} \\ \times\; & \Sigma^{-\frac{1}{2}(\nu_0+k+T+1)}\Big\{tr\Big[\Big(\big((Y - XB_T)'(Y - XB_T) + (B_T - B_0)'\Lambda_0(B_T - B_0) + V_0\big)\Big)\Sigma^{-1}\Big]\Big\}\end{aligned}$$

where $B_T = \Lambda_T^{-1}(X'Y + \Lambda_0 B_0)$. Thus our posterior for B conditional on Σ and the data is

$$f(B|\Sigma, X, Y) \sim \mathcal{MN}\Big(\Lambda_T^{-1}(X'Y + \Lambda_0 B_0), \Lambda_T, \Sigma\Big)$$

and our posterior for Σ conditional on the data is

$$f(\Sigma|X,Y) \sim \mathcal{IW}\Big((Y - XB_T)'(Y - XB_T) + (B_T - B_0)'\Lambda_0(B_T - B_0) + V_0, \nu_0 + T\Big)$$

where $\Lambda_T = X'X + \Lambda_0$ and $B_t = \Lambda_T^{-1}(X'Y + \Lambda_0 B_0)$. Note that for the practical purpose of drawing B in programs to simulate these posterior distributions we will need to use the vectorized form $\beta = vec(B')$ of

$$f(\beta|\Sigma, X, Y) \sim \mathcal{N}\Big(vec\big([\Lambda_T^{-1}(X'Y + \Lambda B_0)]'\big), \Lambda_T^{-1} \otimes \Sigma\Big)$$

1.3 Summing Up

These results will form the basis of our Bayesian estimation routines for dynamic factor models. In chapter six we will construct posterior densities for parameters of our model by simulation. These simulations will use the conditional densities we have derived above. In summary, for a normal-inverse gamma conjugate prior our prior for β is

$$\pi(\beta|\sigma) \propto (\sigma)^{-\frac{k}{2}}\exp\Big\{ -\frac{1}{2\sigma}(\beta - \beta_0)'\Lambda_0(\beta - \beta_0)\Big\}$$

and our prior for sigma is

$$\pi(\sigma) \propto \sigma^{\frac{\nu_0}{2}-1} \exp\left\{-\frac{s_0}{2\sigma}\right\}$$

With the model distribution

$$f(y|\beta,\sigma,X) \propto \sigma^{-\frac{T}{2}} \exp\left\{-\frac{1}{2\sigma}(y - X\beta)'(y - X\beta)\right\}$$

the posterior distribution conditional on σ and the data X and y for β is

$$f(\beta|\sigma,X,y) \sim \mathcal{N}\Big(\Lambda_T^{-1}(X'y + \Lambda_0\beta_0), \sigma\Lambda_T^{-1}\Big)$$

where $\Lambda_T = (X'X + \Lambda_0)$ and $f(\sigma|X,y)$ follows an inverse gamma distribution with scale parameter $s_T = (y - X\beta_T)'(y - X\beta_T) + (\beta_T - \beta_0)'\Lambda_0(\beta_T - \beta_0) + s_0$ and $\nu_T = T + \nu_0$.

In the case of the multivariate model with a normal-inverse Wishart conjugate prior our prior for β conditional on σ is,[6] in vectorized form,

$$\pi(\beta|\sigma) \propto \exp\bigg\{-\frac{1}{2}(\beta - \beta_0)'(\mathbf{\Lambda}_0)^{-1}(\beta - \beta_0)\bigg\}$$

where $\mathbf{\Lambda}_0 = \Lambda_0 \otimes \sigma$ and Λ_0 determines our prior tightness. Using a prior scale parameter of I_k our prior for $\sigma \sim \mathcal{IW}(I_k, \nu_0)$ is

$$\pi(\sigma) \propto |\sigma|^{-\frac{\nu_0+k+1}{2}} \exp\left\{-\frac{1}{2}tr\big(\sigma^{-1}\big)\right\}$$

Our model is [7]

$$y_t = BX_t + \varepsilon_t$$

so that $\beta = vec(B)$ and thus the model distribution (again in vectorized form) is

$$f(y|\beta,\sigma,X) \propto \exp\left\{-\frac{1}{2}(y - \mathbf{X}\beta)'\Sigma^{-1}(y - \mathbf{X}\beta)\right\}$$

where $\mathbf{X} = X \otimes I_k$. Then the posterior for β conditional on σ is

$$f(\beta|\sigma,X,y) \sim \mathcal{N}\Big(vec\Big[\big(\Lambda_T^{-1}(X'Y + \Lambda_0 B_0')\big)'\Big], \Lambda_T^{-1} \otimes \sigma\Big)$$

where $\Lambda_T = X'X + \Lambda_0$ and $f(\sigma|X,y)$ follows an inverse-Wishart distribution with scale parameter $S_T = I_n + (Y - XB_T')'(Y - XB_T') + (B_T - B_0)\Lambda_0(B_T - B_0)'$ and $\nu_T = \nu_0 + T$. Note here that $B_T = \big(\Lambda_T^{-1}(X'Y + \Lambda_0 B_0')\big)'$ is the $k \times m$ matrix formed by stacking β_T and similarly B_0 is the stacked form of β_0.

[6] In using the vectorized form of the problem I use σ to denote the $k \times k$ covariance matrix of shocks ε_t and Σ to denote $I_T \otimes \sigma$. In using the matrix normal distribution Σ is sufficient to denote the $k \times k$ covariance matrix of shocks as using the Kronecker product is not necessary.

[7] Note here that B is back to its normal $k \times m$ orientation as opposed its definition in the previous section.

Chapter 2

The Kalman Filter

Kalman filtering, named for Kalman (1960), is a means of estimating the time series model

$$y_t = Hx_t + \varepsilon_t \tag{2.1}$$

$$x_t = Ax_{t-1} + e_t \tag{2.2}$$

where x_t is an unobserved state or states, y_t observed outcomes, and ε_t and e_t are normally distributed error terms with the covariance matrix $Cov \begin{bmatrix} e_t \\ \varepsilon_t \end{bmatrix} = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$. States, observed outcomes, and error terms can be either scalars or vectors. The Kalman filter works by first predicting an outcome $x_{t|t-1}$ where the subscripts indicate the prediction of x_t based on all observations from the initial period until period $t-1$ and then updating this prediction using the outcome from period t. Formally, we first predict

$$p(x_t|y_{1:t-1}) = \int p(x_t,\ x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{2.3}$$

and then update this prediction based on the current observation y_t

$$p(x_t|y_t) \propto p(y_t|x_t)p(x_t|y_{1:t-1}) \tag{2.4}$$

In this way equation 2.4 is a Bayesian estimate of our unobserved states x_t using equation 2.3 as our prior. Before writing down what the prediction and updating equations will in fact be when our model follows 2.2 and 2.1 it's worth looking at a few features of the multivariate normal distribution.

2.1 Preliminaries

Suppose we know that the scalars x and y follow the multivariate normal distribution

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu \\ m \end{bmatrix}, \Sigma\right) \tag{2.5}$$

where $\Sigma = \begin{bmatrix} P & C \\ C & S \end{bmatrix}$ and we're interested in determining the distribution $f(x|y)$. Using the definition of a conditional distribution we know that

$$f(x|y) = \frac{f(x,y)}{f(y)}$$

or, since $f(y)$ is a normalizing constant,

$$f(x|y) \propto |\Sigma|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \left(\begin{bmatrix} x-\mu \\ y-m \end{bmatrix}' \Sigma^{-1} \begin{bmatrix} x-\mu \\ y-m \end{bmatrix} \right) \right\}$$

The inverse of the covariance matrix is $\Sigma^{-1} = \frac{1}{PS-C^2} \begin{bmatrix} S & -C \\ -C & P \end{bmatrix}$ thus the exponential terms are

$$-\frac{1}{2(PS-C^2)}[(x-\mu)^2 S + (y-m)^2 P - 2(x-\mu)(y-m)C]$$

We can re-write this expression as

$$-\frac{1}{2(PS-C^2)}[(y-m)^2 P + (x-\mu-(y-m)CS^{-1})S(x-\mu-(y-m)CS^{-1}) - (y-m)^2 C^2 S^{-1}]$$

or, dumping the terms which don't contain our parameter of interest x into the normalizing constant,

$$-\frac{1}{2}[(x-(\mu+(y-m)CS^{-1}))\tilde{P}^{-1}(x-(\mu+(y-m)CS^{-1}))] \tag{2.6}$$

where $\tilde{P} = P - C^2 S^{-1}$. $f(x|y)$ is therefore normally distributed with mean $(\mu + (y - m)CS^{-1})$ and variance $\tilde{P} = P - C^2 S^{-1}$. This result generalizes to the case in which x and y are vectors with covariance matrix $\Sigma = \begin{bmatrix} P & C \\ C' & S \end{bmatrix}$ as

$$E(x|y) = \mu + CS^{-1}(y-m)$$

and

$$Var(x|y) = P - CS^{-1}C'$$

These results are essentially all we need to derive the Kalman filter.

2.2 The Kalman Filter

To use the results from section 2.1 requires two elements. The first is our model, equations 2.2 and 2.1. The second is the distribution for $\begin{bmatrix} x_{t|t-1} \\ y_t \end{bmatrix}$; this is not as obvious as it may seem since we never in fact observe x_t (or x_{t-1}). Therefore we need to define a new variable, call it $\mu_{t|t}$, which is our expected value of x_t given observations $y_{1:t}$. Define the variance of $x_{t|t}$ as $P_{t|t}$ and the variance of $x_{t|t-1}$ (x_t given observations $1 : t-1$) as $P_{t|t-1}$. Then the joint distribution for $x_{t|t-1}$ and y_t is

$$\begin{bmatrix} x_{t|t-1} \\ y_t \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} A\mu_{t-1|t-1} \\ H(A\mu_{t-1|t-1}) \end{bmatrix}, \Sigma_t\right)$$

where

$$\Sigma_t = \begin{bmatrix} P_{t|t-1} & C_t \\ C_t' & S_t \end{bmatrix}$$

From the results in section 2.1 we can immediately calculate the expected value of $x_{t|t-1}$ (which is unobserved) given y_t (which is observed), $E(x_{t|t-1}|y_t) = A\mu_{t-1|t-1} + C_t S_t^{-1}(y_t - H(A\mu_{t-1|t-1}))$, as well as $Var(x_{t|t-1}|y_t) = P_{t|t-1} - C_t S_t^{-1} C_t'$. However, we still need to derive the values for Σ. For notational convenience write $x_{t|t-1}$ as simply x_t which is distinct from $x_{t|t}$ Writing

$$\Sigma = E_{t|t-1}\begin{bmatrix} \underbrace{(x_t - A\mu_{t-1|t-1})(x_t - A\mu_{t-1|t-1})'}_{P_{t|t-1}} & \underbrace{(x_t - A\mu_{t-1|t-1})(y_t - HA\mu_{t-1|t-1})'}_{C_t} \\ \underbrace{(y_t - HA\mu_{t-1|t-1})(x_t - A\mu_{t-1|t-1})'}_{C_t'} & \underbrace{(y_t - HA\mu_{t-1|t-1})(y_t - HA\mu_{t-1|t-1})'}_{S_t} \end{bmatrix}$$

we then have

$$\begin{aligned} P_{t|t-1} &= E_{t|t-1}\left[x_t x_t' - 2x_t\mu_{t-1|t-1}'A' + A\mu_{t-1|t-1}\mu_{t-1|t-1}'A'\right] \\ &= E_{t|t-1}\left[(Ax_{t-1}+v_t)(Ax_{t-1}+v_t)' - 2(Ax_{t-1}+v)\mu_{t-1|t-1}'A' + A\mu_{t-1|t-1}\mu_{t-1|t-1}'A'\right] \\ &= E_{t|t-1}(Ax_{t-1}x_{t-1}'A') + E_{t|t-1}(v_t v_t') - A\mu_{t-1|t-1}\mu_{t-1|t-1}'A' \\ &= A'P_{t-1|t-1}A' + Q \end{aligned}$$

since $P_{t-1|t-1} = E_{t-1|t-1}(x_{t-1}x_{t-1}') - \mu_{t-1|t-1}\mu_{t-1|t-1}'$. Similarly

$$S = HP_{t|t-1}H' + R$$

And finally

$$\begin{aligned} C_t &= E_{t|t-1}\left[x_t y_t' - x_t\mu_{t-1|t-1}'A'H' - A\mu_{t-1|t-1}y_t + A\mu_{t-1|t-1}\mu_{t-1|t-1}'A'H'\right] \\ &= E_{t|t-1}\left[(Ax_{t-1}+v_t)(H(Ax_{t-1}+v_t)w_t)' - 2(Ax_{t-1}+v)\mu_{t-1|t-1}'A'H' + A\mu_{t-1|t-1}\mu_{t-1|t-1}'A'H'\right] \\ &= E_{t|t-1}(Ax_{t-1}x_{t-1}'A'H') + v_t v_t' H' - A\mu_{t-1|t-1}\mu_{t-1|t-1}'A'H' \\ &= P_{t|t-1}H' \end{aligned}$$

Thus equipped we can write the Kalman filter as follows. Our prediction for the mean and variance of x_t (without conditioning on y_t) is

$$\begin{aligned} \mu_{t|t-1} &= A\mu_{t-1|t-1} \\ P_{t|t-1} &= AP_{t-1|t-1}A' + Q \end{aligned}$$

Our prediction for the mean and variance of y_t given observations $1 : t-1$ (before y_t is observed), denoted $y_{t|t-1}$, is

$$\begin{aligned} m_{t|t-1} &= H\mu_{t|t-1} \\ S_t &= HP_{t|t-1}H' + R \end{aligned}$$

Our prediction for the covariance between x_t (again, without using y_t) and $y_{t|t-1}$ is

$$C_t = P_{t|t-1}H'$$

Note that the Kalman gain combines this covariance and the estimated variance of $y_{t|t-1}$ and is typically written as $K_t = C_t S_t^{-1}$. The above equations, called the prediction step, give us our prior. Our posterior estimates for the mean and variance of x_t given y_t (recall y_t is observed), called the updating step, are

$$\begin{aligned} \mu_{t|t} &= \mu_{t|t-1} + C_t S_t^{-1}(y_{t|t} - m_{t|t-1}) \\ P_{t|t} &= P_{t|t-1} - C_t S_t^{-1} C_t' \end{aligned}$$

2.3 The Likelihood Function

We can write the likelihood of observing $\{y_1 \quad y_2 \quad \ldots \quad y_T\}$ as

$$f(y_{1:T}) = f(y_T|y_{1:T-1})f(y_{1:T-1}) = f(y_T|y_{1:T-1})f(y_{T-1}|y_{1:T-2})f(y_{1:T-2}) = \prod_{t=1}^{T} f(y_t|y_{1:t-1})$$

where $f(y_t|y_{1:t-1})$ is normally distributed with mean $m_{t|t-1}$ and variance S_t. Denoting the predictive error calculated by the Kalman filter in each period as $\nu_t = y_{t|t} - m_{t|t-1}$ we can thus write the likelihood of our observables as

$$\mathcal{L} = \prod_{t=1}^{T} (2\pi)^{k/2} |S_t|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2}\nu_t' S_t^{-1} \nu_t \right\}$$

so that the log likelihood, which we typically use for any maximization problem, is

$$l = \kappa - \sum_{t=1}^{T} \frac{1}{2} \log(|S_t|) - \frac{1}{2} \sum_{t=1}^{T} \nu_t' S_t^{-1} \nu_t \tag{2.7}$$

where κ does not contain any parameters of interest and can thus be ignored in the maximization problem. The log likelihood is remarkably easy to calculate in practice as both ν_t and S_t are calculated in each period by the Kalman filter.

2.4 A Kalman Smoother

What the Kalman filter of the previous section delivers are estimates of $x_{t|t}$, that is, an estimate of x_t given observations from period 1 through t. However, if the states of the model are autocorrelated then presumably observations realized after period t also contain information about states in period t. The Kalman smoothers provide a means of employing this information so that our final estimate of states becomes $x_{t|T}$, that is, an estimate of x_t given observations from period 1 through T. There are several approaches to Kalman smoothing; I'll outline the simple and popular Rauch-Tung-Striebel smoother. The process begins by running the Kalman filter and saving the values for $P_{t|t-1}$, $P_{t|t}$, $\mu_{t|t-1}$, and of course $\mu_{t|t}$. The key to the smoother is the fact that $f(x_t|x_{t+1}, y_{1:T}) = f(x_t|x_{t+1}, y_{1:t})$, which states that if we know x_{t+1} then further realizations of observables after period t do not add any additional information. We can summarize the relationship between $x_{t|t}$ and $x_{t+1|t}$ as

$$\begin{bmatrix} x_t|y_{1:t} \\ x_{t+1}|y_{1:t} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{t|t} \\ \mu_{t+1|t} \end{bmatrix}, \begin{bmatrix} P_{t|t} & P_{t|t}A' \\ AP_{t|t} & P_{t+1|t} \end{bmatrix}\right) \tag{2.8}$$

where $P_{t+1|t} = AP_{t|t}A' + Q$. Using the same results for a joint normal distribution we used to derive the Kalman filter we then have[1]

$$\begin{aligned} E(x_t|x_{t+1}, y_{1:t}) &= \mu_{t|T} = \mu_{t|t} + g_t(x_{t+1} - \mu_{t+1|t}) \\ Var(x_t|x_{t+1}, y_{1:t}) &= P_{t|T} = P_{t|t} - g_t P_{t+1|t} g_t' \end{aligned}$$

where $g_t = P_{t|t}A'P^{-1}_{t+1|t}$. Since we do not in fact observe x_{t+1} we need to slightly modify the above equations. Using the law of iterated expectations for the first

$$E(x_t|y_{1:T}) = E(E(x_t|x_{t+1}, y_{1:t})|y_{1:T}) = \mu_{t|t} + g_t(\mu_{t+1|T} - \mu_{t+1|t}) \tag{2.9}$$

Using the law of iterated variance for the second

$$\begin{aligned} Var(x_t|y_{1:T}) &= E(Var(x_t|x_{t+1}, y_{1:t})|y_{1:T}) + Var(E(x_t|x_{t+1}, y_{1:t})|y_{1:T}) \\ &= P_{t|t} - g_t P_{t+1|t} g_t' + g_t P_{t+1|T} g_t' \\ &= P_{t|t} - g_t(P_{t+1|t} - P_{t+1|T}) g_t' \end{aligned} \tag{2.10}$$

Equations (2.9) and (2.10) form our smoother. We begin using our last filtered value for $\mu_{T|T}$ and $P_{T|T}$ and iterate backwards to the first period.

[1] The result for $P_{t|T}$ comes from the fact that

$$\begin{aligned} Var(x_t|x_{t+1}, y_{1:t}) &= P_{t|t} - P_{t|t}A'P^{-1}_{t+1|t}AP_{t|t} \\ &= P_{t|t} - P_{t|t}A'P^{-1}_{t+1|t}P_{t+1|t}P^{-1}_{t+1|t}AP_{t|t} \\ &= P_{t|t} - g_t P_{t+1|t} g_t' \end{aligned}$$

2.5 The Steady State Filter

Note that in the above Kalman filter neither $P_{t|t}$, S_t, nor C_t depend on the realization of y_t or the expected values of x_t (they do, however, depend on the number of series observed each period). Thus, if our series is covariance stationary, the number of observations remains the same each period, and if we happen to know the long run value of C_t, call it C, and S_t, call it S, we could simplify our Kalman filter as

$$\begin{array}{lcl} \mu_{t|t-1} & = & A\mu_{t-1|t-1} \\ m_{t|t-1} & = & H\mu_{t|t-1} \\ \mu_{t|t} & = & \mu_{t|t-1} + CS^{-1}(y_{t|t} - m_{t|t-1}) \end{array} \tag{2.11}$$

This is the steady state Kalman filter. To obtain these steady state values, we can simply run the system of difference equations that determine the relevant variances and covariances until convergence. This system is

$$\begin{array}{lcl} P_{t|t-1} & = & AP_{t-1|t-1}A' + Q \\ S_t & = & HP_{t|t-1}H' + R \\ C_t & = & P_{t|t-1}H' \\ P_{t|t} & = & P_{t|t-1} - C_t S_t^{-1} C_t' \end{array} \tag{2.12}$$

2.6 An Example

Suppose we have a model described by

$$x_t = \begin{bmatrix} 1 & -.5 \\ .1 & .7 \end{bmatrix} x_{t-1} + e_t$$

$$y_t = \begin{bmatrix} .5 & 1 \\ -1 & 2 \\ 1 & -1 \\ 1 & -.5 \end{bmatrix} x_t + \varepsilon_t$$

where x_t is a 2×1 vector of unobserved factors, y_t is a 4×1 vector of observed data, $e_t \sim N(0, I_2)$, and $\epsilon_t \sim N(0, I_4)$ and we would like to construct the unobserved factors x_t from the observed data.

In this case I already know the parameters A, Q, H, and R. Thus all that remains is to specify initial conditions. $x_0 = [0]$ is a natural but arbitrary choice. Thus, to this initial guess has little influence on the model I begin with a large initial factor variance, in this case

$$P_0 = \begin{bmatrix} 10^5 & 0 \\ 0 & 10^5 \end{bmatrix}$$

Figure 2.1 plots the results for 200 observations.

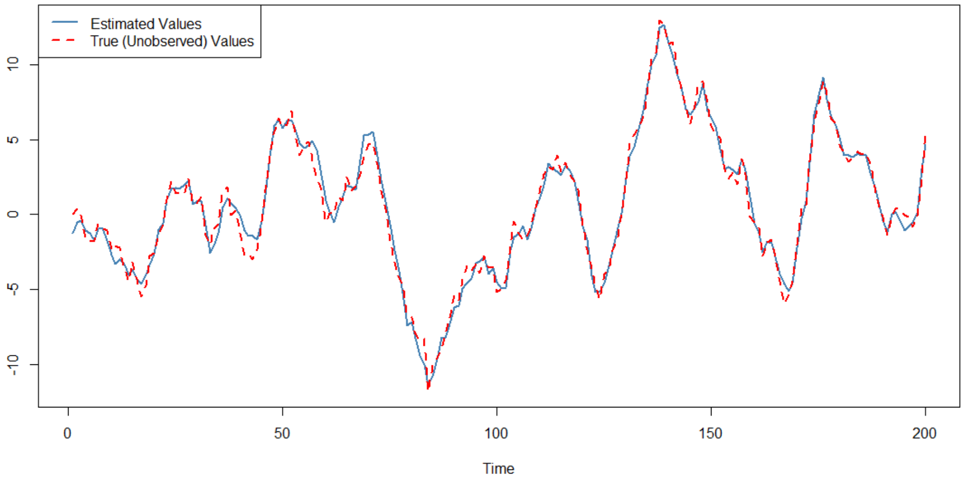


Figure 2.1: True series $x_{1,t}$ versus the series estimated from observations y_t

2.7 Missing Observations

There are several ways to handle missing observations for the filtering and smoothing algorithms in this book. I describe here the approach used in the accompanying R and C++ code.

Using the example from the previous section, suppose the second series in y_t is not observed in a given period t. We can simply re-write our model for the data we do observe by dropping the rows of H and rows and columns of R corresponding to the missing data. The dimensions of the unobserved factors x_t remain the same, thus this does not present any problems in updating our factor predictions from one period to the next. Explicitly, if the second series of x_t were missing in period t then our transition equation would become

$$y_t = \underbrace{\begin{bmatrix} .5 & 1 \\ 1 & -1 \\ 1 & -.5 \end{bmatrix}}_{H_t} x_t + \varepsilon_t$$

and R becomes a 3 × 3 identity matrix.

By re-casting the dimensions of H_t and R_t depending on which series of data are observed, our calculation of the gain

$$K_t = P_{t|t-1} H_t' \left(H_t P_{t|t-1} H_t' + R_t \right)^{-1}$$

will automatically correspond to the observed series. The same is true of the dimensionality of terms used in the disturbance smoother in chapter 3. From a programming point of view, the only downside to the changing dimensions of H, R and the terms calculated using each is storage. For example, if we wish to save the gain K_t and the prediction error $\nu_t = y_t - Hm_{t|t-1}$ at each period t we will need a data type that can handle different dimensions for each observations. In C++/Armadillo this means using `field<mat>`, and in R a `list` as opposed to a `cube` or `array`, but this disadvantage is slight and does not appreciably detract from the speed or efficiency of calculations.

Note that in the rest of this text I do not include time subscripts on parameters such as H and R that may change dimension from one period to the next due to missing data in order to keep the notation clean and avoid confusion with objects such as observations or factors with values that change over time.

Chapter 3

The Durbin-Koopman Disturbance Smoother

I begin this chapter by re-stating our basic factor model with a slight generalization, allowing for exogenous predetermined variables n_t. In practice, these will typically take the form of seasonal factors; Chapter 5 includes an example of estimating seasonal adjustments for covariance stationary data by maximum likelihood. Thus our measurement equation becomes

$$y_t = Hx_t + Mn_t + \varepsilon_t \tag{3.1}$$

The transition equation (in companion form) is unchanged,[1] that is

$$z_t = Az_{t-1} + e_t \tag{3.2}$$

where ε_t and e_t are normally distributed error terms with the covariance matrix

$$Cov \begin{bmatrix} e_t \\ \varepsilon_t \end{bmatrix} = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$$

In the above y_t are noisy observations, z_t stacked factors $z_t = \begin{bmatrix} x_t & x_{t-1} & \dots & x_{t-p} \end{bmatrix}$ with p lags, and n_t predetermined, exogenous variables.

Given the parameters H, A, M, Q, and R estimates of factors or estimates of missing series in y_t derive from the Kalman filter and smoother. I re-state the Kalman filter here

[1] One could also include exogenous variables in the transition equation, but I do not here as what I have in mind are seasonal adjustments to observations.

as

$$\begin{aligned}
z_{t|t-1} &= Az_{t-1|t-1} \\
P_{t|t-1} &= AP_{t-1|t-1}A' + Q \\
y_{t|t-1} &= \tilde{H}z_{t|t-1} + Mn_t \\
S_t &= \tilde{H}P_{t|t-1}\tilde{H}' + R \\
C_t &= P_{t|t-1}\tilde{H}' \\
z_{t|t} &= z_{t|t-1} + C_t S_t^{-1}(y_{t|t} - y_{t|t-1}) \\
P_{t|t} &= P_{t|t-1} - C_t S_t^{-1} C_t'
\end{aligned}$$

and the Kalman smoother as

$$\begin{aligned}
z_{t|T} &= z_{t|t} + g_t(x_{t+1|T} - z_{t+1|t}) \\
P_{t|T} &= P_{t|t} - g_t(P_{t+1|t} - P_{t+1|T})g_t'
\end{aligned} \tag{3.3}$$

where $g_t = P_{t|t}A'P_{t+1|t}^{-1}$. $\tilde{H}$ in the above incorporates a helper matrix J to extract, in the simplest example, contemporaneous factors from z_t. That is, $J = \begin{bmatrix} I_m & 0 & 0 & \ldots \end{bmatrix}$ and $\tilde{H} = HJ$. In the above notation $x_{t|t-1}$ refers to our estimate of x_t given observations through period $t-1$ while $x_{t|t}$ is our estimate of x_t given observations through t, and $x_{t|T}$ is our estimate of x_t conditional on all available data through period T. Note that in the above $K_t = C_t S_t^{-1}$ is the Kalman gain, $\nu_t = y_{t|t} - y_{t|t-1}$ is the prediction error, and thus $K_t\nu_t$ is our forecast update.

Smoothing here requires we store the following matrices (or vectors) from the filter at every period t: $P_{t|t}$, $P_{t+1|t}$, $z_{t|t}$, and $z_{t+1|t}$. Additionally, we must invert the matrix $P_{t+1|t}$ at each iteration of the smoother. This can be computationally burdensome for even moderately sized models. Consider an example with four factors and five lags. In this case we will be inverting the 20 × 20 matrix $P_{t+1|t}$ in each time period. For mixed frequency models (see chapter 7) inverting $P_{t+1|t}$ can become prohibitive. For a daily/weekly/monthly model with three factors and three lags at a monthly (30 day) frequency $P_{t+1|t}$ will be a 270 × 270 matrix. The Durban-Koopman disturbance smoother allows us to avoid these cumbersome calculations resulting in much faster and more efficient model estimation.

3.1 Durban Koopman (2001) Filtering and Smoothing

Sticking with the above notation, Durbin and Koopman (2001) write the Kalman filter as

$$\nu_t = y_t - \tilde{H}x_{t|t-1} - Mn_t \tag{3.4}$$

$$S_t = \tilde{H}P_{t|t-1}\tilde{H}' + R \tag{3.5}$$

$$K_t = AP_{t|t-1}\tilde{H}'S_t^{-1} \tag{3.6}$$

$$z_{t+1|t} = Az_{t|t} + K_t\nu_t \tag{3.7}$$

$$L_t = A - K_t\tilde{H} \tag{3.8}$$

$$P_{t+1|t} = AP_{t|t}L_t' + Q \tag{3.9}$$

Note that in this notation the Kalman gain in (3.7) updates the forecast for *next period* factors; factoring A out of equation (3.8) yields the same forecast update as in chapter 2's more standard Kalman filter. Similarly, multiplying out the elements of (3.9) yields $P_{t+1|t} = AP_{t|t-1}A' - AC_tS_t^{-1}C_t'A' + Q$. For the above notation, the disturbance smoother is

$$r_t = \tilde{H}'S_t^{-1}\nu_t + L_t'r_{t+1} \tag{3.10}$$

with $r_T = 0$ from which we can recover the vector of errors (disturbances) as

$$\begin{bmatrix} \varepsilon_{t|T} \\ e_{t|T} \end{bmatrix} = \begin{bmatrix} RS_t^{-1} & -RK_t' \\ 0 & Q \end{bmatrix} \begin{bmatrix} \nu_t \\ r_t \end{bmatrix} \tag{3.11}$$

Note that in order to recover smoothed factors we must iterate r_t back to r_0. Equipped with $e_{t|T}$ we can recover smoothed estimates of the factors by initializing our smoothed factors as

$$z_1 = z_{1|0} + P_{1|0}r_0$$

or, since we typically set z_0 to zero simply $z_1 = P_{1|0}r_0$, and iterating forward again using

$$z_{t|T} = Az_{t-1|T} + e_{t|T}$$

In this case we must store the matrices (or vectors) ν_t, $\tilde{H}$, S_t^{-1}, L_t, and potentially K_t. Since filtering requires solving $\tilde{H}S_t^{-1}$ disturbance smoothing requires no additional matrix inversions and will thus be much more computationally efficient.

Chapter 4

Estimation via Principal Components

By far the simplest and fastest way to estimate the factor model outlined in equations (1) and (2) is by principal components, though this approach has several major drawbacks. Perhaps most importantly, principal components solves a static problem. That is, when using principal components we find factors by looking at the measurement equation in isolation ignoring the inter-temporal correlations implied by the transition equation. Additional problems with principal components estimates are (1) data must be square and complete, a large drawback when using a methodology particularly adept at handling missing and noisy observations and (2) principal components limits our ability to fix parameters or apply prior beliefs to parameter estimates. Despite these drawbacks, when we have a data set without missing observations principal components provides a very quick and easy way to estimate a factor model. I begin this chapter by examining reduced rank regressions before moving on to principal components as a special case. Once we have our principal component estimates of factors we can estimate the transition equation by OLS and simply plug these results into the Kalman filter and, if desired, smoother to obtain our final factor estimates.

4.1 Reduced Rank Regression

Suppose we observe a $k \times t$ set of data Y that we wish to explain using a $s \times t$ set of data X, but that the number of series in X (denoted by s) is very large and that we think the relationship between X and Y can in fact be summarized by the data in an $m \times t$ matrix γX. That is, we would like to reduce the rank of X before estimating its relationship with Y. Assuming we want to model a linear relationship, we can write this problem succinctly as

$$Y = B\gamma X + \epsilon \tag{4.1}$$

where both B and γ are parameter matrices to estimate. If we want the least squares estimates of B and γ then we need to minimize

$$\min\Big\{tr\big(\underbrace{(Y-B\gamma X)}_{\epsilon}\underbrace{(Y-B\gamma X)'}_{\epsilon'}\big)\Big\} \tag{4.2}$$

where tr denotes the trace of the covariance matrix for ϵ. Note that equation (4.1) will be observationally equivalent for

$$Y=\underbrace{B\theta}_{B^a}\underbrace{\theta^{-1}\gamma}_{\gamma^a}X+\epsilon$$

where B^a and γ^a are some alternative B and γ, thus we need to impose some sort of normalization on γ. It is convenient to impose

$$\gamma XX'\gamma' = I_m \tag{4.3}$$

From a simple OLS model we already know that given γ our estimate for B that minimizes our loss function (4.2) will be

$$B = YX'\gamma'(\gamma XX'\gamma')^{-1}$$

which, given our normalization (4.3) is just $YX'\gamma'$. Plugging this into our loss function we have

$$\min\Big\{tr\big(YY'-2YX'\gamma'\gamma XY'+YX'\gamma'\gamma XX'\gamma'\gamma XY'\big)\Big\}$$

which, given (4.3) is just

$$\min\Big\{tr\big(YY'-YX'\gamma'\gamma XY'\big)\Big\} \tag{4.4}$$

Since the first term does not include γ we can re-write this minimization problem as a constrained maximization, using the property of a matrix trace that $tr(ABC)=tr(CAB)=tr(BCA)$ so that $tr(YX'\gamma'\gamma XY')=tr(\gamma XY'YX'\gamma')$,

$$\mathcal{L}=tr\big(\gamma XY'YX'\gamma'+\Lambda\big(I_m-\gamma XX'\gamma'\big)\big)$$

where the constraint comes from our normalization (4.3) and Λ is a diagonal matrix of Lagrange multipliers. The first order condition for this problem (see the appendix for useful matrix derivatives) is

$$2\gamma XY'YX'-2\Lambda\gamma XX'=0$$

or

$$\gamma XY'YX'(XX')^{-1}-\Lambda\gamma=0$$

so that the solution for γ is the left eigenvectors of $XY'YX'(XX')^{-1}$. Since left eigenvectors are a bit annoying (Matlab automatically computes right eigenvectors) we can simply take the transpose of this result

$$(XX')^{-1}XY'YX'\gamma' - \gamma'\Lambda = 0 \tag{4.5}$$

so that γ' is given by the (right) eigenvectors of $(XX')^{-1}XY'YX'$ associated with the largest m eigenvalues, the diagonal elements of Λ (Λ is a diagonal matrix). Recall that the Lagrange multiplier is a measure of how strongly our constraint binds.[1] We chose the eigenvectors associated with the largest eigenvalues because those vectors give the factors γX which will have the largest impact on our objective function $tr\big(YX'\gamma'\gamma XY'\big)$. Once we have γ we can simply compute $B = YX'\gamma'$.

4.2 Principal Components as a Special Case

Principal components is a special case of reduced rank regressions in which $X = Y$. That is, γY is a collection of $m < k$ series that we believe is sufficient to explain Y. If $X = Y$ then equation (4.5) simply becomes

$$YY'\gamma' - \gamma'\Lambda = 0 \tag{4.6}$$

in which case the solution for γ' is the (right) eigenvectors of YY' associated with the largest m eigenvalues. We can actually derive principal components from a slightly more general problem. Suppose we want to minimize the loss function

$$\min\{tr\big((Y - \gamma F)(Y - \gamma F)'\big)\} \tag{4.7}$$

where F can be any set of explanatory variables. Thus, unlike the problem in section 4.1, we are not restricting what the set of explanatory variables can be (previously it was X). We again need to impose a normalization of γ since $Y = \gamma\theta\theta^{-1}F + \epsilon$ will be observationally equivalent to the model in (4.7). In this case the normalization

$$I_m = \gamma'\gamma \tag{4.8}$$

is convenient. Minimizing first over F, the solution for the factors given γ is

$$F = (\gamma'\gamma)^{-1}\gamma'Y$$

or, using (4.8),

$$F = \gamma'Y$$

[1] Specifically Λ is the partial derivative of our objective function with respect to the constraint, in this case I_k.

Plugging this back into our minimization problem, equation (4.7), we have

$$\min\{tr\big(YY' - 2\gamma\gamma' YY' + \gamma\gamma' YY'\gamma\gamma'\big)\}$$

Using the property of a matrix trace that $tr(ABC) = tr(CBA) = tr(ACB)$ and our normalization, equation (4.8) this result becomes

$$\min\{tr\big(YY' - Y'\gamma\gamma' Y\big)\}$$

which, again using our normalization, we can write as the constrained maximization problem

$$\mathcal{L} = tr\big(\gamma' YY'\gamma + \Lambda(I - \gamma'\gamma)\big)$$

with first order condition

$$YY'\gamma - \gamma\Lambda = 0$$

thus γ is the eigenvectors of YY' associated with the m largest eigenvalues (note that this γ is the transpose of the previous one we derived for reduced rank regressions). It is fairly simple in this context to calculate how much each principal component, in this case each row of γY, reduces our original loss function, equation (4.7). If we scale our loss function by $1/T$ and assume Y has zero mean our loss function is simply the trace of the covariance matrix of Y, denoted $tr(\Sigma_{YY})$. We can re-write this trace using the eigendecomposition of Σ_{YY} as[2]

$$tr\big(\Gamma\Lambda\Gamma'\big) = tr(\Gamma'\Gamma\Lambda) = tr(\Lambda)$$

where again we have used the properties of a matrix trace. Thus the trace of Σ_{YY} is the sum of its eigenvalues. With one principal component, that associated with the largest eigenvalue denoted λ_1, our scaled loss function is

$$\begin{aligned}
& tr\left(\tfrac{1}{T}(Y - \gamma\gamma' Y)(Y - \gamma\gamma' Y)'\right) \\
= \;& tr\left(\tfrac{1}{T}(YY' - 2\gamma\gamma' YY' + \gamma\gamma' YY'\gamma'\gamma\right) \\
= \;& tr\left(\tfrac{1}{T}(YY' - \gamma\gamma' YY')\right) \\
= \;& tr\left(\tfrac{1}{T}(YY' - \gamma' YY'\gamma)\right) \\
= \;& tr\left(\Gamma\Lambda\Gamma' - \gamma'\Gamma\Lambda\Gamma'\gamma\right) \\
= \;& tr\left(\Lambda - \begin{bmatrix} \lambda_1 & 0 & \dots \\ 0 & 0 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}\right)
\end{aligned}$$

[2] From our first order condition for m factors using the scaled loss function, $\Sigma_{YY}\gamma - \gamma\Lambda$, we can write the system of equations for the complete set of factors when $m = k$ as $\Sigma_{YY}\Gamma = \Gamma\Lambda$ where Λ is a diagonal matrix of the eigenvalues of Σ_{YY} and the columns of Γ are the associated eigenvectors. Thus $\Sigma_{YY} = \Gamma\Lambda\Gamma'$ where we maintain our normalization $\Gamma'\Gamma = I_k$, and since Γ is now square it is also true that $\Gamma\Gamma' = I_k$

where the last equality comes from the fact that $\gamma'\Gamma$ is a matrix of zeros with a one in the upper left corner, as is $\Gamma'\gamma$. Thus the first principal component reduces our scaled loss function by λ_1. Similarly, the first two principal components will reduce our scaled loss function by $\lambda_1 + \lambda_2$. The usual interpretation of this result is that the first m principal components explain $\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{k} \lambda_i}$ of the variance of Y, where by variance of Y we mean the sum of the diagonal of the covariance matrix.

4.3 Summing Up

Beginning with our transition equation

$$Y_t = Hx_t + \varepsilon_t$$

if our set of k covariance stationary observables Y_t does not have missing observations then we can estimate the $k \times m$ matrix of loadings H by principal components as the (right) eigenvectors associated with the m largest eigenvalues of Σ_{YY} where Σ_{YY} is the (typically scaled to have ones on the diagonal) covariance matrix for Y_t, that is, $\Sigma_{YY} = (Y - \bar{Y})'(Y - \bar{Y})'$. Due to the normalizing assumption $I_m = H'H$ in equation (4.8) our estimate of the factors x_t is

$$H'Y_t = \hat{x}_t$$

and residuals are given by

$$\hat{\varepsilon}_t = Y_t - H\hat{x}_t$$

allowing us to estimate R. Typically we will use only the diagonal elements of R as Doz et al. (2012) show that this model, the approximate factor model, even if misspecified still consistently estimates factors. Using the factors $\hat{x}_t$ we can estimate the transition equation by OLS for parameters A and Q. These are nearly all we will need to run the Kalman filter and, if desired, smoother. The only thing that remains are initial values x_0 and P_0. Because x_0 is not known, it is good practice to initiate the filter with diffuse values, that is, specifying a large initial factor variance P_0. For example, we might begin our filter with $x_0 = 0_m$ and $P_0 = 10^5 I_m$.[3]

[3] In the case of more than one lag p in the transition equation we would have $x_0 = 0_{m\times p}$ and $P_0 = 10^5 I_{m\times p}$.

Chapter 5

Maximum Likelihood Estimation via Watson and Engle (1983)

Maximum likelihood estimation of state space models using numerical techniques such as the function `optim()` in R provide an easy to program method for estimating model parameters. To use a numerical method we can simply write a function that returns the log likelihood described in equation (2.7) and search for parameters that maximize it. The great drawback of numerical methods is that they are computationally intensive and thus slow to converge and unfeasible for larger models. Watson and Engle (1983) provide an alternative approach that overcomes these shortcomings of numerical routines.

5.1 The Algorithm

Watson and Engle (1983) propose an iterative scheme that describes an expectation-maximization (EM) algorithm for estimating state space models. The insight Watson and Engle (1983) for the purpose of estimating the model described in equations (1) and (2) is that, though we do not observe the true factors, we can still calculate the necessary moment matrices for A, H, Q, and R using an appropriate adjustment and, moreover, we can use the standard Kalman smoother to calculate this adjustment. I begin by re-stating the model for clarity. The measurement equation for k observables and m factors is

$$y_t = Hx_t + \varepsilon_t$$

and the transition equation is

$$x_t = Bz_{t-1} + e_t$$

where z_t is defined as

$$z_t = \begin{bmatrix} x_t \\ x_{t-1} \\ \vdots \\ x_{t-p+1} \end{bmatrix}$$

and p denotes the number of lags in the transition equation. Then the moment matrices we need to estimate are, for B: $E(z_{t-1}z'_{t-1})$ and $E(x_t z'_{t-1})$; for H: $E(x_t x'_t)$ and $E(y_t x'_t)$; for Q: $E(e_t e'_t)$; and for R: $E(\varepsilon_t \varepsilon'_t)$. Proper estimation of the above moment matrices allows us to then compute maximum likelihood estimates of the model parameters.

In brief, the approach works by noting that, where $x_{t|T}$, $P_{t|T}$, $C_{t|T}$, and so on are the values calculated at the j^{th} iteration of the algorithm (I suppress the superscript j to keep the notation cleaner and use $z_{t|T}$ to denote our estimate of the factors z_t given observations $1:T$)

$$E(z_{t-1}z'_{t-1}) \quad = \quad E\Big((z_{t-1|T} + (z_{t-1} - z_{t-1|T}))(z_{t-1|T} + (z_{t-1} - z_{t-1|T}))'\Big) \tag{5.1}$$

which we can estimate as

$$\frac{1}{T}\Big[\sum_t z_{t-1|T}z'_{t-1|T} + \sum_t P_{t-1|T}\Big]$$

$$E(x_t z'_{t-1}) \quad = \quad E\Big((x_{t|T} + (x_t - x_{t|T}))(z_{t-1|T} + (z_{t-1} - z_{t-1|T}))'\Big) \tag{5.2}$$

which we can estimate as

$$\frac{1}{T}\Big[\sum_t (x_{t|T}(z_{t-1|T})') + \sum_t C_{t|T}\Big]$$

giving the moment matrices for B;

$$E(x_t x'_t) \quad = \quad E\Big((x_{t|T} + (x_t - x_{t|T}))(x_{t|T} + (x_t - x_{t|T}))'\Big) \tag{5.3}$$

which we estimate as

$$\frac{1}{T}\Big[\sum_t x_{t|T}x'_{t|T} + \sum_t P^x_{t|T}\Big]$$

$$\begin{aligned} E(y_t x'_t) \quad &= \quad E\Big(y_t\big(x_{t|T} + (x_t - x_{t|T})\big)'\Big) \\ &= \quad E(y_t x'_{t|T}) \end{aligned} \tag{5.4}$$

giving the moment matrices for H; and for the covariance matrices Q and R

$$e_t = (x_{t|T} - Bz_{t-1|T}) + \big((x_t - x_{t|T}) - B(z_{t-1} - z_{t-1|T})\big) \tag{5.5}$$

so that we estimate $E(e_t e_t')$ as

$$\frac{1}{T}\Big[\sum_t v_{t|T}v_{t|T} + \sum_t P^x_{t|T}\sum_t BP_{t-1|T}B' + \sum_t BC_{t|T} + \sum_t C'_{t|T}B'\Big]$$

and

$$\varepsilon_t = y_t - Hx_{t|T} - H(x_t - x_{t|T}) \tag{5.6}$$

so that we estimate $E(\varepsilon_t \varepsilon_t')$ as

$$\frac{1}{T}\Big[\sum_t \varepsilon_{t|T}\varepsilon_{t|T} + \sum_t HP^x_{t|T}H'\Big]$$

As suggested by Watson and Engle (1983) we can calculate the correlations $E\big((x_t - x_{t|T})(z_t - z_{t-1|T})'\big)$ by including an extra lag of x_t in the state vector of the Kalman filter and smoother. For example, if we wanted to estimate a model with three lags we would write the transition equation as

$$\begin{bmatrix} x_t \\ x_{t-1} \\ x_{t-2} \\ x_{t-3} \end{bmatrix} = \begin{bmatrix} B_1 & B_2 & B_3 & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ x_{t-3} \\ x_{t-4} \end{bmatrix} + \begin{bmatrix} v_t \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Denote the variance of this augmented vector of states in period t as

$$P^{WE}_{t|T} = \begin{bmatrix} \sigma_{t,t} & \sigma_{t,t-1} & \sigma_{t,t-2} & \sigma_{t,t-3} \\ \sigma_{t-1,t} & \sigma_{t-1,t-1} & \sigma_{t-1,t-2} & \sigma_{t-1,t-3} \\ \sigma_{t-2,t} & \sigma_{t-2,t-1} & \sigma_{t-2,t-2} & \sigma_{t-2,t-3} \\ \sigma_{t-3,t} & \sigma_{t-3,t-1} & \sigma_{t-3,t-2} & \sigma_{t-3,t-3} \end{bmatrix} \tag{5.7}$$

For this three lag example $P_{t|T}$ is the upper left[1] 3×3 submatrix of $P^{WE}_{t|T}$ in equation (5.7); $C_{t|T}$ refers to the upper right 1×3 submatrix of $P^{WE}_{t|T}$, and $P^x_{t|T}$ refers to element $\sigma_{t,t}$. This gives all the results needed to estimate the moment matrices above. The EM algorithm proceeds by alternately estimating the unobserved factors via the Kalman filter and smoother and estimating the parameters of the model as outlined above until the log likelihood function converges.

[1] Of course each element $\sigma_{i,j}$ is itself a matrix so in fact $P_{t|T}$ will have dimension $3m \times 3m$.

5.2 Practical Issues

Identification

An important practical issue for either maximum likelihood or Bayesian estimation of factor models is identifying the model. The issue arises as the factors x_t are never observed. Thus the model

$$\begin{aligned} y_t &= Hx_t + \varepsilon_t \\ x_t &= Bx_{t-1} + e_t \end{aligned} \tag{5.8}$$

is observationally equivalent to the model

$$\begin{aligned} y_t &= \underbrace{H\theta^{-1}}_{\mathsf{H}}\underbrace{\theta x_t}_{\mathsf{x_t}} + \varepsilon_t \\ \underbrace{\theta x_t}_{\mathsf{x_t}} &= \underbrace{\theta B\theta^{-1}}_{\mathsf{B}}\underbrace{\theta x_{t-1}}_{\mathsf{x_{t-1}}} + \underbrace{\theta e_t}_{\mathsf{e_t}} \end{aligned}$$

Factor estimates and the likelihood function for the model in (5.8) would be identical to those for the model

$$\begin{aligned} y_t &= \mathsf{H}\mathsf{x_t} + \varepsilon_t \\ \mathsf{x_t} &= \mathsf{B}\mathsf{x_{t-1}} + \mathsf{e_t} \end{aligned} \tag{5.9}$$

Identification may be important for interpreting our model.[2] It is, for maximum likelihood and Bayesian estimation, also important for estimation.[3] Identification requirements for maximum likelihood estimation are less stringent than for Bayesian estimation. Bayesian estimation by simulation relies on the assumption we are drawing factors and parameters from the same distribution at each iteration. On the other hand, because our convergence criteria for maximum likelihood is in the likelihood function, parameter estimates can drift between the models in (5.8) and (5.9) as the likelihood function for the two will be identical. What is important is that we scale the model at each iteration of the algorithm; otherwise our parameter estimates may become arbitrarily large or small thereby breaking the algorithm. There is no one way to scale or identify our model; for a discussion of several identification possibilities see Stock and Watson (2016). To scale the model we could enforce $H'H = I$ as with principal components or even $diag(H'H) = I$. Perhaps the simplest identification technique, however, it set the top $m \times m$ sub-matrix of H, where m is the number of factors, to be an identity matrix (this is called "naming factors" identification). In this case our model will not just be scaled, but be fully identified (thus

[2] See Stock and Watson (2016) for a more comprehensive discussion of interpretation and identification issues.

[3] For principal component estimation identification is dealt with by the normalization for H in equation (4.8).

we could and will use it for Baysian estimation as well). This strategy has the added advantage of giving our model a simple interpretation: this first factor describes filtered movements of the first variable in common with the entire data set, the second factor describes filtered movements of the second variable in common with the entire data set, and so on.

Initial Values

As with principal components we will not know initial values z_0 with which to initiate the algorithm. As smoothing produces estimates of z_0 one possibility is to begin with $z_0 = [0]$ and then update z_0 at each iteration of the algorithm. However, there is no particularly good theoretical justification for this approach and it may result in the likelihood function getting worse (that is, smaller) from one iteration to the next. A better approach is to again begin with diffuse values, for example setting $z_0 = [0]$ and P_0 to be the identify matrix times a large number (large relative to the actual variance of shocks).

5.3 An Example: Seasonally Adjusting Stationary Data

As an example of using maximum likelihood via Watson and Engle (1983) we could seasonally adjust noisy data using our factor model with exogenous regressors. This example will specifically consider month on month percent changes is Spanish industrial production (IP). Our observation equation is then

$$y_t = x_t + MN_t + \varepsilon_t \tag{5.10}$$

where y_t is observed raw IP data, x_t is unobserved adjusted industrial production, and M is a matrix of parameters to estimate multiplying a dummy for each month contained in N. Thus N is a $T \times 12$ matrix of predetermined variables where T is the number of observations of Spanish IP. To keep the example simple, I will use only one lag in the transition equation, that is,

$$x_t = bx_{t-1} + e_t$$

to use the Watson and Engle (1983) algorithm we will need:

1. An initial guess for M and $r = var(\varepsilon_t)$
2. An initial guess for b and $q = var(e_t)$
3. Diffuse initial conditions for x_0 and P_0

Furthermore, we will need to add a lag to the transition equation in order to estimate the adjustment factors for the transition equation. One possible initial guess for M is to regress N_t on y_t by OLS and to calculate our initial guess for r from the residuals. We can

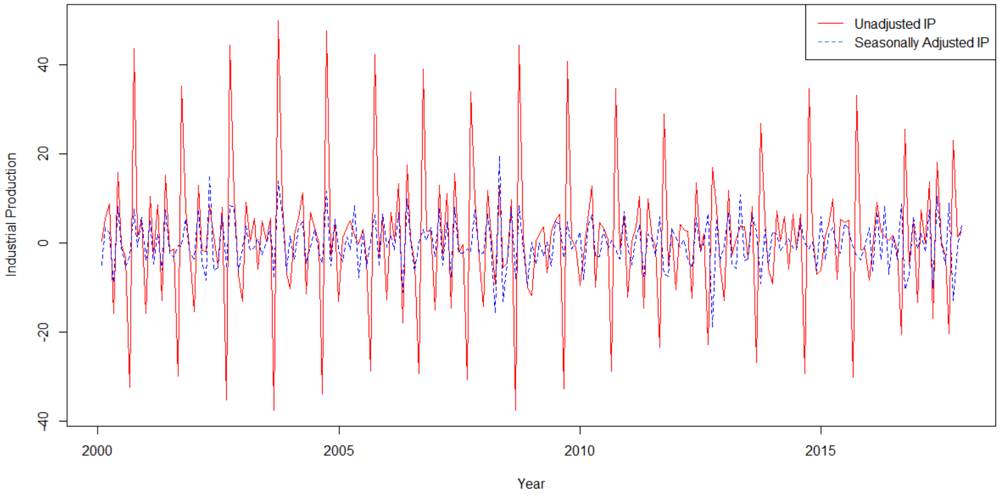


Figure 5.1: Factor Model Seasonally Adjusted Spanish IP, MoM percent change

also form initial estimates for b and r by OLS using the observed data y_t. Finally, we will need to write our transition equation with an extra lag in companion form as

$$\begin{bmatrix} x_t \\ x_{t-1} \end{bmatrix} = \begin{bmatrix} b & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ x_{t-2} \end{bmatrix} + \begin{bmatrix} e_t \\ 0 \end{bmatrix} \tag{5.11}$$

Equipped with our initial guesses for parameters and the initial conditions $[x_0 \;\; x_{-1}]' = [0 \;\; 0]'$ and $P_0 = 10^6 \times I_2$ the algorithm proceeds as follows:

1. Run the Kalman smoother for the current iteration value of parameters. If the likelihood does not improve from the last iteration then the algorithm has converged; otherwise continue.

2. Using y_t, $z_{t|T} = \left[x_{t|T} \quad x_{t-1|T}\right]'$, and $P_{t|T}$ for the augmented model re-estimate the parameters following equations (5.1) through (5.6)

These two steps alternate until the likelihood converges. Figure 5.1 illustrates seasonally adjusted IP estimated as outlined above against the raw data. Programs to estimate the model are contained in the file ML_Seasonal_Example.

Chapter 6

Bayesian Estimation by Simulation

Bayesian estimation by simulation is a third approach to estimating factor models which for many applications will be the best option. There are of course theoretical reasons why one might prefer Bayesian statistics generally — humans typically never encounter a situation in life in which they have no prior beliefs, even if these beliefs are strongly biased. Specifically to our purpose of estimating factor models, using prior beliefs may have practical applications. For example, we may believe unobserved factors to transition smoothly from one period to the next despite noisy observations. We could incorporate this belief into our model by biasing the parameters of the transition equation B towards zero and/or increasing the value of ν_0, the degrees of freedom in our inverse-Wishart prior for the variance of shocks in the transition equation.

I should note at the outset that this chapter only discusses one approach to estimation by simulation. In the general case, one can simulate posterior distributions using a range of mixing distributions — distributions which determine the probability of accepting a draw from our sampling distribution. We will use the simplest possible mixing distribution and accept draws from sampling distributions with probability one. This approach, Gibbs sampling, may not always be the most efficient but is simple to code and generally works well for the purpose of estimating the models in this book. For a much richer discussion of Monte Carlo methods see Robert and Casella (2010).

6.1 Sampling

Chapters 1, 2 and 3 present the main results we will use to estimate our model. Estimation begins with an initial guess for parameters. Given this initial guess, we would like to draw factors from our model[1]

$$y_t = Hx_t + \varepsilon_t \tag{6.1}$$

$$x_t = Bx_{t-1} + e_t \tag{6.2}$$

where

$$\begin{bmatrix} e_t \\ \varepsilon_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \right)$$

One possible approach is to

1. Use the standard Kalman filter to obtain an estimate of $x_{T|T}$ and $P_{T|T}$ describing our posterior distribution for factors in the final period.

2. Draw $\tilde{x}_T \sim \mathcal{N}\big(x_{T|T}, P_{T|T}\big)$

3. Iterate backwards one period using the standard Kalman smoother. That is, calculate

$$\begin{aligned} x_{T-1|T} &= x_{T-1|T-1} + g_{T-1}(\tilde{x}_T - x_{T|T-1}) \\ P_{T-1|T} &= P_{T-1|T-1} - g_{T-1}(P_{T|T-1} - P_{T|T})g'_{T-1} \end{aligned}$$

4. Draw $\tilde{x}_{T-1} \sim \mathcal{N}\big(x_{T-1|T}, P_{T-1|T}\big)$

5. Repeat these iterations back to the initial period to generate draws of $\tilde{x}_t$ for every period.

This approach is simple. However, as

$$g_t = P_{t|t}A'P^{-1}_{t+1|t}$$

we must invert $P_{t+1|t}$ at each step of the iteration. For a steady state model, that is, one in which the variance of factors does not change, this is not a big drawback as we can calculate g once (it will be the same in every period for a steady state model) and proceed. If the factor variance is not constant (this will always be the case if y_t has missing observations), calculating $P^{-1}_{t+1|t}$ becomes computationally burdensome, particularly in

[1] I use the case of one lag in the transition equation for explanatory purposes here as the notation is simple. For more lags our model becomes

$$y_t = \tilde{H}z_t + \varepsilon_t$$

$$z_t = Az_{t-1} + e_t$$

where typically we will have $\tilde{H} = \begin{bmatrix} H & 0 & 0 & \ldots \end{bmatrix}$, $z_t = \begin{bmatrix} x_t & x_{t-1} & \ldots & x_{t-p+1} \end{bmatrix}$ where p is the number of lags, and A is the companion form of B in the more familiar vector autoregression $x_t = Bz_t + e_t$. Note also that I am somewhat lose with the definition of e_t as when we have more than one lag this vector includes zeros as "shocks" to lagged variables in z_t to make dimensions agree.

models with many lags as P has dimension $m \times p$ where m is the number of factors and p the number of lags. For this more general case, Durbin and Koopman (2012) propose a much more computationally efficient method for drawing factors. Their key result for normally distributed data, presented earlier in Durbin and Koopman (2001), is a method for drawing from the conditional distribution $f(x|y)$ when doing so directly is burdensome but drawing from the joint normal distribution $f(x, y)$ and calculating the expected values $E(x|y)$ is computationally more efficient. To borrow their notation, we can draw x^+, y^+ from $f(x, y)$, calculate $\hat{x} = E(x|y)$ (the filtered and smoothed factors from our true observations y), and calculate $\hat{x}^+ = E(x^+|y^+)$ (the filtered and smoothed observations from our simulated observations y^+). Then the draw from our desired distribution $f(x|y)$ is $\tilde{x} = \hat{x} + x^+ - \hat{x}^+$. The values $\hat{x}$ are the posterior means for the factors conditional on the data y, and $x^+ - \hat{x}^+$ are the simulated zero mean shocks. For our purposes the authors point out that this approach can be made even more efficient as follows:

Algorithm 1

1. Begin with a draw for x_0, ε_t at every period t, and e_t at every period t and iterate equations (6.1) and (6.2) forward to obtain a draw for factors x_t^+ and a draw for observations y_t^+.

2. Obtain $y_t^* = y_t - y_t^+$.

3. Obtain $\hat{x}_t^*$ by filtering and smoothing y_t^* using the disturbance smoother described in chapter 3.

4. We can then calculate our simulated factors as

$$\tilde{x}_t = \hat{x}_t^* + x_t^+$$

An important question is what to use as our draw for x_0. As previously, the best way to avoid issues of this initial vector of factors is to begin our algorithm with a large factor variance, that is, allow for uncertainty regarding x_0. Using diffuse values to initialize the filter and smoother allows us to simply plug in $x_0 = 0$ and proceed with algorithm 1.

This excellent algorithm provides a quick, efficient, and easy to code method for sampling factors. Once we have a draw for factors, we can then draw parameters based on these simulated factors. Posterior distributions for parameters given factors are presented in chapter 1. Because we will be using an exact factor model, that is, we will treat the covariance matrix for shocks to observations R as diagonal, we can estimate each diagonal element of R and each row of H using a normal-inverse gamma conjugate prior.

Because our draw for R_{jj}, the j^{th} diagonal of the covariance matrix for shocks to the observation equation and thus the variance of shocks to the j^{th} observed series, is conditional on the observations and factors only, while our draw for H_j, the j^{th} row of H, is conditional on R_{jj}, we will begin by drawing variances. Specifically, for series j we draw R_{jj} from

$$f(R_{jj}|\tilde{X}, y) \sim \mathcal{IG}\Big((y - X\beta_T)'(y - X\beta_T) + (\beta_T - h_0)'\Lambda_0(\beta_T - h_0) + s_0, T + \nu_0\Big) \tag{6.3}$$

In the above $\beta_T = (\tilde{X}'X + \Lambda_0)^{-1}(\tilde{X}'y_j + \Lambda_0 h_0)$. s_0 is our prior scale parameter — our prior for the variance of shocks when multiplied by ν_0. ν_0 is our prior "degrees of freedom", that is, ν_0 determines how aggressively we shrink R_{jj}. Λ_0 is a diagonal matrix determining tightness on our prior for H_j, denoted h_0. Typically Λ_0 will be an identity matrix times a (scalar) tightness parameter, with larger values corresponding to a tighter prior, and h_0 will be zero for all rows of H, though we could use different priors for each row if desired. Once we have a draw for R_{jj}, we draw row j of H from

$$f(H_j|R_{jj}, \tilde{X}, y_j) \sim \mathcal{N}\Big(\Lambda_T^{-1}(\tilde{X}'y_j + \Lambda_0 h_0), R_{jj}\Lambda_T^{-1}\Big) \tag{6.4}$$

where $\Lambda_T = (\tilde{X}'X + \Lambda_0)$. Note that the posterior mean for H_j is just β_T as defined above.

Our draws for Q (the covariance of shocks to the observation equation) and B are nearly identical except that, because Q contains off-diagonal elements, we will use a normal-inverse Wishard conjugate prior. Beginning again with covariances, we draw Q from

$$f(Q|\tilde{X}, Y) \sim \mathcal{IW}\Big((Y - \tilde{X}B_T)'(Y - \tilde{X}B_T) + (B_T - B_0)'\Lambda_0(B_T - B_0) + V_0, \nu_0 + T\Big) \tag{6.5}$$

where $B_T = (\tilde{X}'\tilde{X} + \Lambda_0)^{-1}(X'Y + \Lambda_0 B_0)$. Again, Λ_0 determines the tightness on our prior for B, denoted B_0. Typically we will use a zero prior for B, though using a random walk prior is also popular. V_0 is our prior scale parameter (our prior for the covariance to shocks when multiplied by ν_0) and ν_0 determines how aggressively we shrink Q. In most programming languages, including the C++ and R code associated with this book, we will need to use the vectorized form of our posterior for B. Thus we draw $\beta = vec(B')$ from

$$f(\beta|Q, \tilde{X}, Y) \sim \mathcal{N}\Big(vec\big([\Lambda_T^{-1}(\tilde{X}'Y + \Lambda B_0)]'\big), \Lambda_T^{-1} \otimes \Sigma\Big) \tag{6.6}$$

where $\Lambda_T = \tilde{X}'\tilde{X} + \Lambda_0$ and our posterior mean (in matrix format) is just B_T as defined above.

Once we have a draw for parameters we can again draw factors. We continue these iterations until our posterior distributions converge to a stationary distribution, and then draw a sufficient number of parameters to suit our needs, that is, estimating posterior means and variances of the parameters themselves. This whole process is re-stated in the following algorithm:

Algorithm 2

1. Begin with a guess for parameters.
2. Given parameters, sample factors following Algorithm 1.
3. Given factors obtained in step (2), draw:
    a) R_{jj} from the distribution in (6.3) and H_j from the distribution in (6.4) for every series j.
    b) Q from the distribution in (6.5) and B from the distribution in (6.6).
4. Repeat steps 2 and 3 until distributions converge to stationary distributions.
5. Once distributions converge repeat steps 2 and 3 storing draws for R, H, Q, and B at each iteration.

The question of when the distribution has converged is not straight forward — see Robert and Casella (2010) for a discussion of convergence criteria. The software associated with this book draws a default 500 samples in a burn loop before moving on to step 5 above and storing output. Convergence criteria can then be applied to the stored output to determine whether burn time must be increased.

6.2 Identification

Bayesian estimation of dynamic factor models is complicated by the fact that factors are not identified. Identifying factors ensures that we are in fact drawing from the same distributions in each step of Algorithm 2. The issue of identification is described in section 5.2. The simplest, though not the only, solution to identification of Bayesian dynamic factor models is what Stock and Watson (2011) call naming factors identification, that is, setting the top $m \times m$ submatrix of H (where m is the number of factors) to be the identify matrix. Employing this identification scheme in our sampling iterations requires that we create a temporary $m \times m$ H matrix for the first m series in Y, which I will call M. Based on our draw for factors $\tilde{X}$, draw the elements of R_{jj}, $j \in (1, m)$ and M for the first m series from distributions (6.3) and (6.4) respectively. To enforce that the top $m \times m$ submatrix of H is I_m we then use M to normalize our draw for the factors $\tilde{X}$. This turns out to be quite simple. Using the observation equation, our normalization is

$$y_{t,1:m} = \underbrace{MM^{-1}}_{I_m} \underbrace{M\tilde{x}_{t,old}}_{\tilde{x}_{t,new}} + \varepsilon_t$$

Thus for a model with one lag, our normalization is just[2]

$$\tilde{X}_{new} = \tilde{X}_{old} M$$

[2] We postmultiply by M here as X_{old} is the matrix of simulated factors with time indexed by rows

Because we have not yet estimated the transition equation (we do that after normalizing factors) or any other elements of H and R, this is all that is needed to identify our model. For a model with p lags where

$$z_t = \begin{bmatrix} x_t \\ x_{t-1} \\ \vdots \\ x_{t-p+1} \end{bmatrix}$$

This normalization becomes

$$\tilde{Z}_{new} = \tilde{Z}_{old}(I_p \otimes M) \tag{6.7}$$

6.3 An Example: Bayesian vs. ML Estimation for Simulated Data

As a first example I simulate ten series ($k = 10$) from a factor model with observation equation

$$y_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ .5 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 0.5 & -1 \\ 0 & .5 & 1 \end{bmatrix} x_t + \varepsilon_t \tag{6.8}$$

and transition equation

$$x_t = \begin{bmatrix} 0.4 & 0 & 0 & 0.3 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0.3 & 0.2 & 0 & 0.2 & 0.1 & 0 & 0.1 & 0 \\ 0 & 0.2 & 0.4 & 0 & 0 & 0.2 & 0 & 0.1 & 0.1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ x_{t-3} \end{bmatrix} + e_t \tag{6.9}$$

where

$$\varepsilon_t \sim \mathcal{N}\big(0, I_{10}\big)$$

and

$$e_t \sim \mathcal{N}\left(0, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -0.5 \\ 0 & -0.5 & 1 \end{bmatrix}\right)$$

	$T = 50$	$T = 100$	$T = 200$
Bayes	0.42	0.34	0.30
ML	0.44	0.36	0.33

Table 6.1: MSE for Bayesian vs. ML Estimation

Note that the first observed series y_t^1 is just the first factor x_t^1 and that this factor is independent of the other two factors.

My objective in this section will be to estimate the "true" data $y_t^{true} = y_t - \varepsilon_t$ or simply $y_t^{true} = Hx_t$ based on the observed series y_t. Note that this true series is never observed. In addition to the noise ε_t I drop observations from the first three series. Thus for these series I have five observations followed by five missing values, then five observations and so on. Simulations proceed by

1. Simulating data according to equations (6.1) and (6.2)

2. Dropping blocks of five observations from the first three series

3. Estimating the model by simulation as outlined in chapter 1 and by maximum likelihood as outlined in chapter 5

4. Calculating the mean squared error $1/T \sum(\hat{y}_t - y_t^{true})^2$ for all observed series and averaging over the 10 series for 1000 repetitions

Table 6.1 presents the results when I estimate a model with three factors and two lags. As is evident from the table Bayesian estimation yields slightly better results in terms of mean squared error. However, this comes at the cost of longer run time as we are simulating entire distributions, not simply estimating a few parameters. Note, however, that Bayesian estimation may be faster for models with large numbers of lags, such as mixed frequency models, as inverting the covariance matrix for predicted factors $P_{t+1|t}$ becomes increasingly burdensome as p gets large.

Chapter 7

An Introduction to Mixed Frequency Models

Because state space models are so apt and handling missing data they are particularly well suited to mixed frequency data sets in which, for example, a quarterly variable will not be observed for two out of three months. The software accompanying this book does not include routines for mixed frequency data as each case tends to be fairly specific and components of routines will need to be tailored to the application. Instead, I provide a brief introduction to simply outline how one might incorporate mixed frequency data into the models discussed thus far.

Suppose first that our model is in log levels and, as a concrete example, that frequencies are either monthly or quarterly. Denote y_t^q the log of a quarterly observation in month t and y_t^m the log of a monthly observation in month t. Then

$$e^{y_t^m} = e^{y_t^m} + e^{y_{t-1}^m} + e^{y_{t-2}^m} \tag{7.1}$$

The difficulty lies in the fact that equation (1) is linear in the log variables while equation (7.1) is not; to overcome this issue simply take a linear approximation of (7.1) yielding

$$y_t^m = \frac{1}{3}(y_t^m + y_{t-1}^m + y_{t-2}^m) \tag{7.2}$$

Plugging equation (1) into the above yields the linear state space structure:

$$y_t^q = \frac{1}{3}Hx_t + \frac{1}{3}Hx_{t-1} + \frac{1}{3}Hx_{t-2} + \varepsilon_t \tag{7.3}$$

Note that this requires that the model include at least three lags of factors, although one need not estimate coefficients on factors with more than one lag in the transition equation.

To put the model into log differences we begin with equation (7.2) and note that what we observe, Δy_t^q, is

$$\begin{aligned} y_t^q - y_{t-3}^q &= \frac{1}{3}(y_t^m - y_{t-3}^m) + \frac{1}{3}(y_{t-1}^m - y_{t-4}^m) + \frac{1}{3}(y_{t-2}^m - y_{t-5}^m) \\ &= \frac{1}{3}\Delta y_t^w + \frac{2}{3}\Delta y_{t-1}^w + \Delta y_{t-2}^w + \frac{2}{3}\Delta y_{t-3}^w + \frac{1}{3}\Delta y_{t-4}^w \end{aligned} \tag{7.4}$$

This is the result presented in Mariano and Murasawa (2003). Unlike the levels case, we now need to include at least four lags of the factors.

Estimates of H and R now must take into account the structure of the data in (7.2) for level data or (7.4) for differenced data. A simple way to proceed is to define a new helper matrix J_q for low frequency data. Maintaining the monthly/quarterly structure and the three lag model by way of example, the helper matrix for quarterly data, assuming we are dealing with data in levels, will be

$$J_q = \begin{bmatrix} \frac{1}{3}I_m & \frac{1}{3}I_m & \frac{1}{3}I_m \end{bmatrix} \tag{7.5}$$

or, for quarterly variables in first differences

$$J_q = \begin{bmatrix} \frac{1}{3}I_m & \frac{2}{3}I_m & I_m & \frac{2}{3}I_m & \frac{1}{3}I_m \end{bmatrix} \tag{7.6}$$

The difficulty with a more general structure, monthly/daily data for example, is two fold. First, the number of high frequency periods (days) in a low frequency period (months) gets large. For differenced data our helper matrix for monthly variables, assuming 31 days in the month, becomes the $m \times 61$ matrix

$$J_q = \begin{bmatrix} \frac{1}{31}I_m & \frac{2}{31}I_m & \dots & \frac{2}{31}I_m & \frac{1}{31}I_m \end{bmatrix} \tag{7.7}$$

Thus the vector of factors z_t must include at least 61 lags. Second, the number of days in the month is not constant. Nor is the number of days in February constant from one year to the next. Thus our helper matrix J_q must change depending on the number of days in the current month. Though conceptually simple, these facts can become computationally difficult. To illustrate the point, our daily/monthly data requires 61 lags in z_t. If our model includes a modest three factors, than z_t has 183 elements, a huge number for a literature in which there or typically only a few latent factors.

The structure of the transition equation is an additional consideration when the number of high frequency periods (days) in a low frequency period (months) is large. The frequency for factors is the highest frequency in the model — in our daily/monthly example factors are thus daily. However, assuming we are interested in nowcasting or forecasting a low frequency (monthly) variable we would like to include, in our monthly-daily example, at least a month of lags in the transition equation. This raises the problem of overparameterization: with three factors and 30 lags we will be estimating 90 parameters in the transition equation excluding the covariances in Q. Our solution is to again introduce a helper matrix which, for the transition equation, we call J_B. We then specify the number of lags at each frequency in the model. For a daily/weekly/monthly model we we then have p_d daily lags, p_w weekly lags, and p_m monthly lags (in the transition equation we use the term month to refer to 30 days regardless of the actual number of days in the month). For a model with two factors and $p_d = 1$, $p_w = 1$, and $p_m = 1$ our transition equation equation can be described by

$$B = \begin{bmatrix} b_{11}^d & b_{12}^d & b_{11}^w & b_{12}^w & b_{11}^m & b_{12}^m \\ b_{21}^d & b_{22}^d & b_{21}^w & b_{22}^w & b_{21}^m & b_{22}^m \end{bmatrix}$$

$$J_B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ \frac{1}{7} & 0 & \frac{1}{7} & 0 & \frac{1}{7} & 0 & \frac{1}{7} & \dots \\ 0 & \frac{1}{7} & 0 & \frac{1}{7} & 0 & \frac{1}{7} & 0 & \dots \\ \frac{1}{30} & 0 & \frac{1}{30} & 0 & \frac{1}{30} & 0 & \frac{1}{30} & \dots \\ 0 & \frac{1}{30} & 0 & \frac{1}{30} & 0 & \frac{1}{30} & 0 & \dots \end{bmatrix}$$

where b_{21}^w refers to the multiplier of factor 1 on factor 2 at a weekly frequency, and finally

$$X_{t+1} = BJ_BZ_t + e_t$$

Equipped with the (sparse) helper matrices described above our model fits into the smoothing and filtering algorithms described in chapter 3. The only complication is that $\tilde{H}_t = HJ_t$ changes from one month to the next as the number of days in a month changes. This does not present any theoretical difficulties but requires careful coding when writing estimation routines.

References

Doz, C., Giannone, D., and Reichlin, L., 2012. A Quasi-Maximum Likelihood Approach for Large, Approximate Dynamic Factor Models. *Review of Economics and Statistics*, 94(4):1014–1024.

Durbin, J. and Koopman, S., April 2001. An efficient and simple simulation smoother for state space time series analysis. Computing in Economics and Finance 2001 52, Society for Computational Economics.

Durbin, J. and Koopman, S. J., 2012. *Time Series Analysis by State Space Methods.* Oxford University Press.

Hamilton, J., 1994. *Time Series Analysis.* Princeton University Press.

Kalman, R., 01 1960. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of basic Engineering*, 82:35–45.

Koop, G., Poirier, D., and Tobias, J., 2007. *Bayesian Econometric Methods.* Cambridge University Press.

Lutkepohl, H., 2007. *New Introduction to Multiple Time Series Analysis.* Springer.

Mariano, R. S. and Murasawa, Y., 2003. A new coincident index of business cycles based on monthly and quarterly series. *Journal of Applied Econometrics*, 18(4):427–443.

Robert, C. P. and Casella, G., 2010. *Monte Carlo Statistical Methods.* Springer Publishing Company, Incorporated.

Sarkka, S., 2013. *Bayesian Filtering and Smoothing.* Cambridge University Press.

Stock, J. H. and Watson, M. W., 2011. Dynamic Factor Models. In Clements, M. P. and Hendry, D. F., editors, *The Oxford Handbook of Economic Forecasting.* Oxford, Oxford Handbooks.

Stock, J. H. and Watson, M. W., 2016. Factor Models and Structural Vector Autoregressions in Macroeconomics. In Taylor, J. B. and Uhlig, H., editors, *Handbook of Macroeconomics*, volume 2A, pages 415–525. North-Holland.

Watson, M. W. and Engle, R. F., 1983. Alternative Algorithms for the Estimation of Dynamic Factor, Mimic and Varying Coefficient Regression Models. *Journal of Econometrics*, 23(3):385–400.

Index