Notes on Bayesian Macroeconometrics

Joshua C. C. Chan
Economics Discipline Group
University of Technology Sydney

Version 1.4
June 2017

Contents

Chapter 1

Overview of Bayesian Econometrics

Bayesian methods are based on a few elementary rules in probability theory. At the core is Bayes' theorem, which tells us how our subjective beliefs about the parameters should be updated in light of new information. In this introductory chapter we give an overview of Bayesian theory and computation.

1.1 Bayesian Basics

The fundamental organizing principle in Bayesian econometrics is Bayes' theorem. It forms the unifying principle on how Bayesians estimate model parameters, conduct inference, compare models, etc.

Bayes' theorem states that for events A and B, the conditional probability of A given B is:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A)\mathbb{P}(B \mid A)}{\mathbb{P}(B)},$$

where $\mathbb{P}(A)$ and $\mathbb{P}(B)$ are the marginal probabilities for events A and B, respectively. This expression tells us how our view about event A should change in light of information in event B.

To apply Bayes' theorem to estimation and inference, we first introduce some notations. Suppose we have a model that is characterized by the **likelihood function** $p(\mathbf{y} \mid \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector of model parameters. Intuitively, the likelihood function specifies how the observed data are generated from the model given a particular set of parameters.

Now, suppose we have obtained an observed sample $\mathbf{y} = (y_1, \ldots, y_T)'$, and we would like to learn about $\boldsymbol{\theta}$. How should we proceed? The goal of Bayesian methods is to obtain the **posterior distribution** $p(\boldsymbol{\theta} \,|\, \mathbf{y})$ that summaries all the information about the parameter vector $\boldsymbol{\theta}$ given the data.

Applying Bayes' theorem, the posterior distribution can be computed as

$$p(\boldsymbol{\theta} \,|\, \mathbf{y}) = \frac{p(\boldsymbol{\theta})p(\mathbf{y} \,|\, \boldsymbol{\theta})}{p(\mathbf{y})} \propto p(\boldsymbol{\theta})p(\mathbf{y} \,|\, \boldsymbol{\theta}),$$

where $p(\boldsymbol{\theta})$ is the **prior distribution** and

$$p(\mathbf{y}) = \int p(\boldsymbol{\theta})p(\mathbf{y} \,|\, \boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}$$

is the **marginal likelihood** that plays a crucial role in Bayesian model comparison—we will discuss this quantity in more detail in later chapters. Bayes' theorem says that knowledge of $\boldsymbol{\theta}$ comes from two sources: the prior distribution and an observed sample $y_1, \ldots, y_T$ summarized by the likelihood.

The prior distribution $p(\boldsymbol{\theta})$ incorporates our subjective beliefs about $\boldsymbol{\theta}$ before we look at the data. The need to specify a prior distribution is sometimes seen as a weakness of the Bayesian approach. However, the prior allows the researcher to formally incorporate any nondata information. As such, it can be used to impose "structure" into the analysis. Moreover, prior distributions can often be interpreted as a form of regularization from the frequentist perspective.

The posterior distribution $p(\boldsymbol{\theta} \,|\, \mathbf{y})$ characterizes all relevant information about $\boldsymbol{\theta}$ given the data. For example, if we wish to obtain a point estimate of $\boldsymbol{\theta}$, we might compute the posterior mean $\mathbb{E}(\boldsymbol{\theta} \,|\, \mathbf{y})$. To characterize the uncertainty about $\boldsymbol{\theta}$, we might report the posterior standard deviations of the parameters. For instance, for the ith element of $\boldsymbol{\theta}$, we can compute $\sqrt{\mathrm{Var}(\theta_i \,|\, \mathbf{y})}$.

In principle, these quantities can be computed given the posterior distribution $p(\boldsymbol{\theta} \,|\, \mathbf{y})$. In practice, however, they are often not available analytically. In those cases we would require simulation to approximate those quantities of interest.

To outline the main idea, suppose we obtain R *independent* draws from $p(\boldsymbol{\theta} \,|\, \mathbf{y})$, say, $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(R)}$. If we assume the first moment exists, i.e., $\mathbb{E}(\boldsymbol{\theta} \,|\, \mathbf{y}) < \infty$, then by the weak law of large numbers the sample mean $\widehat{\boldsymbol{\theta}} = R^{-1} \sum_{r=1}^{R} \boldsymbol{\theta}^{(r)}$ converges in probability to $\mathbb{E}(\boldsymbol{\theta} \,|\, \mathbf{y})$ as R tends to infinity. Since we can control the simulation size, we could approximate the posterior mean arbitrarily well—if we are patient enough. Similarly, other moments or quantiles can be estimated using the sample analogs.

Hence, estimation and inference become essentially a computational problem of obtaining draws from the posterior distribution. In general, sampling from arbitrary

distribution is a difficult problem. Fortunately, there is now a large family of algorithms generally called **Markov chain Monte Carlo** (MCMC) methods to sample from complex distributions.

The basic idea behind these algorithms is to construct a Markov chain so that its limiting distribution is the target distribution—in our case the target is the posterior distribution. By construction, samples from the MCMC algorithms are autocorrelated. Fortunately, similar convergence theorems—called ergodic theorems—hold for these correlated samples. Under some weak regularity conditions, we can use draws from these MCMC algorithms to estimate any functions of the parameters arbitrary well, provided that the population analogs exist.

In the next sections we will give a few simple examples to illustrate these points.

1.2 Normal Model with Known Variance

To illustrate the basic mechanics of Bayesian analysis, we start with a toy example. Suppose we take N independent measurements $y_1, \ldots, y_N$ of an unknown quantity μ, where the magnitude of measurement error is known. In addition, from a small pilot study μ is estimated to be about μ_0.

Our goal is to obtain the posterior distribution $p(\mu \,|\, \mathbf{y})$ given the sample $\mathbf{y} = (y_1, \ldots, y_N)'$. To that end, we need two ingredients: a likelihood function and a prior for the parameter μ.

One simple model for this measurement problem is the normal model:

$$(y_n \,|\, \mu) \sim \mathcal{N}(\mu, \sigma^2), \quad n = 1, \ldots, N, \tag{1.1}$$

where the variance σ^2 is assumed to be known. Then, the model (1.1) defines the likelihood function $p(\mathbf{y} \,|\, \mu)$.

Since the scale of the pilot study is small, there is substantial uncertainty around the estimate. A reasonable prior would be

$$\mu \sim \mathcal{N}(\mu_0, \sigma_0^2), \tag{1.2}$$

where both μ_0 and σ_0^2 are known.

All the relevant information about μ is summarized by the posterior distribution, which can be obtained by Bayes' theorem:

$$p(\mu \,|\, \mathbf{y}) = \frac{p(\mu) p(\mathbf{y} \,|\, \mu)}{p(\mathbf{y})}.$$

It turns out that $p(\mu \,|\, \mathbf{y})$ is a Gaussian distribution.

Below we will prove this claim. The derivations are instructive, but they may look messy at the beginning. However, it is important to work through this example as all later models build on the insights obtained from this one.

In the first step, we write out the likelihood function $p(\mu \,|\, \mathbf{y})$. Recall that we say a random variable X follows a **normal** or **Gaussian distribution**, and we write $X \sim \mathcal{N}(a, b^2)$, if its density is given by

$$f(x; a, b^2) = (2\pi b^2)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2b^2}(x-a)^2}.$$

It follows from (1.1) that the likelihood function is a product of N normal densities:

$$\begin{aligned} p(\mu \,|\, \mathbf{y}) &= \prod_{n=1}^{N} (2\pi\sigma^2)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2\sigma^2}(y_n-\mu)^2} \\ &= (2\pi\sigma^2)^{-\frac{N}{2}} \mathrm{e}^{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n-\mu)^2}. \end{aligned} \tag{1.3}$$

Similarly, the prior density $p(\mu)$ is given by

$$p(\mu) = (2\pi\sigma_0^2)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2}. \tag{1.4}$$

Next, we combine the likelihood (1.3) and the prior (1.4) to obtain the posterior distribution. To that end, note that the variable in $p(\mu \,|\, \mathbf{y})$ is μ, and we can ignore any constants that do not involve μ. Now, by Bayes' theorem, we have

$$\begin{aligned} p(\mu \,|\, \mathbf{y}) &\propto p(\mu) p(\mathbf{y} \,|\, \mu) \\ &\propto \mathrm{e}^{-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2} \mathrm{e}^{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n-\mu)^2} \\ &\propto \exp\left[-\frac{1}{2}\left(\frac{\mu^2 - 2\mu\mu_0}{\sigma_0^2} + \frac{N\mu^2 - 2\mu\sum_{n=1}^{N} y_n}{\sigma^2}\right)\right] \\ &\propto \exp\left[-\frac{1}{2}\left(\left(\frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}\right)\mu^2 - 2\mu\left(\frac{\mu_0}{\sigma_0^2} + \frac{N\bar{y}}{\sigma^2}\right)\right)\right], \end{aligned} \tag{1.5}$$

where $\bar{y} = N^{-1}\sum_{n=1}^{N} y_n$ is the sample mean. Since the exponent is quadratic in μ, $(\mu \,|\, \mathbf{y})$ is Gaussian, and we next determine the mean and variance of the distribution.

In this example, the posterior distribution is in the same family as the prior—both are Gaussian. In this case, the prior is called a **conjugate prior** for the likelihood function.

Now, suppose $(\mu \,|\, \mathbf{y}) \sim \mathcal{N}(\widehat{\mu}, D_\mu)$ for some mean $\widehat{\mu}$ and variance D_μ. Using the definition of the Gaussian density, we can rewrite the posterior distribution as

$$\begin{aligned} p(\mu \,|\, \mathbf{y}) &= (2\pi D_\mu)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2D_\mu}(\mu-\widehat{\mu})^2} \\ &\propto \mathrm{e}^{-\frac{1}{2}\left(\frac{1}{D_\mu}\mu^2 - 2\mu\frac{\widehat{\mu}}{D_\mu}\right)}. \end{aligned}$$

Next, compare this expression with (1.5). Since the two expressions are identical for any $\mu \in \mathbb{R}$, the coefficients on μ^2 must be the same, i.e.,

$$D_\mu = \left(\frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}\right)^{-1}.$$

Similarly, the coefficients on μ must be the same as well:

$$\begin{aligned}\frac{\widehat{\mu}}{D_\mu} &= \frac{\mu_0}{\sigma_0^2} + \frac{N\bar{y}}{\sigma^2} \\ \widehat{\mu} &= D_\mu\left(\frac{\mu_0}{\sigma_0^2} + \frac{N\bar{y}}{\sigma^2}\right).\end{aligned}$$

Therefore, we can rewrite the posterior mean as

$$\begin{aligned}\widehat{\mu} &= \left(\frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}\right)^{-1}\left(\frac{\mu_0}{\sigma_0^2} + \frac{N\bar{y}}{\sigma^2}\right) \\ &= \frac{\frac{1}{\sigma_0^2}}{\frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}}\mu_0 + \frac{\frac{N}{\sigma^2}}{\frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}}\bar{y}.\end{aligned}$$

Hence, the posterior mean is a weighted average of the prior mean μ_0 and the sample mean $\bar{y}$, where the weights are, respectively, the inverse of the prior variance σ_0^2 and the inverse of the sample mean variance σ^2/N. Consequently, for a given prior variance σ_0^2, the more observations we have, the larger the weight we give to the sample mean. Conversely, for a given sample size N and measurement variance σ^2, the smaller the prior variance—i.e., the more certain we are about the parameter before looking at the data— the more weight we give to the prior mean.

This is a general feature of Bayesian analysis: posterior results are influenced by both the data information summarized by the likelihood and the our subject beliefs summarized by the prior distribution.

As a numerical example, suppose $\mu_0 = 10, \bar{y} = 20, \sigma_0^2 = \sigma^2 = 1$. In other words, the prior and the data disagree on the location of μ, and the prior variance is the same as the variance of one observation.

We consider two cases: $N = 1$ and $N = 10$. The corresponding posterior distributions of μ are plotted in Figure 1.2.

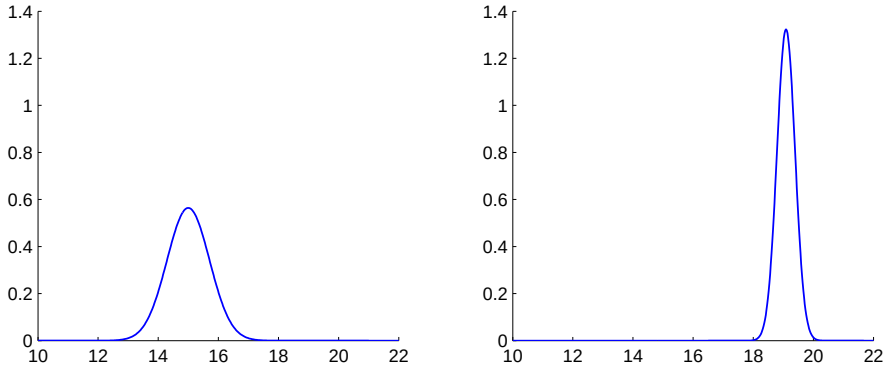

Figure 1.1: Posterior densities of μ for $N = 1$ (left panel) and $N = 10$ (right panel).

In the first case when $N = 1$, the posterior mean of μ is 15—the exact middle of the prior mean and sample mean. When $N = 10$, however, the posterior mean becomes much closer to the sample mean, as data information dominates. In addition, the posterior density in the latter case is also less dispersed—as more data information comes in, the uncertainty around μ becomes smaller.

Since $(\mu \,|\, \mathbf{y}) \sim \mathcal{N}(\widehat{\mu}, D_\mu)$, we can easily compute various quantities of interest, such as $\sqrt{\text{Var}(\mu \,|\, \mathbf{y})}$, $\mathbb{P}(\mu > 0 \,|\, \mathbf{y})$, and a 95% credible set for μ.

But suppose we wish to calculate the posterior mean of some function g of μ, which may not be available analytically. More precisely, consider

$$\mathbb{E}(g(\mu) \,|\, \mathbf{y}) = \int g(\mu) p(\mu \,|\, \mathbf{y}) \mathrm{d}\mu.$$

In general, this integration cannot be solved analytically. However, we can estimate this quantity using **Monte Carlo integration**. Specifically, we generate R draws $\mu^{(1)}, \ldots, \mu^{(R)}$ from $p(\mu \,|\, \mathbf{y})$, and compute

$$\widehat{g} = \frac{1}{R} \sum_{r=1}^{R} g(\mu^{(r)}).$$

By the weak law of large numbers, $\widehat{g}$ converges weakly in probability to $\mathbb{E}(g(\mu) \,|\, \mathbf{y})$ as R tends to infinity. Since we control the simulation size R, we can in principle estimate $\mathbb{E}(g(\mu) \,|\, \mathbf{y})$ arbitrarily well.

As an example, suppose $g(x) = \log|x|$ and $(\mu \,|\, \mathbf{y}) \sim \mathcal{N}(\widehat{\mu}, D_\mu)$, where $\widehat{\mu} = 19.09$ and $D_\mu = 0.09$. The following MATLAB script `norm_1para.m` implements the Monte Carlo integration with $R = 10000$ draws.

```
R = 10000;
```

```
mu_hat = 19.09; Dmu = .09;
mu = mu_hat + sqrt(Dmu)*randn(R,1);
g_hat = mean(log(abs(mu)));
```

The command `randn(R,1)` generates a $R \times 1$ vector of standard Gaussian random variables. Recall that if $Z \sim \mathcal{N}(0,1)$, then

$$Y = a + bZ$$

follows the $\mathcal{N}(a, b^2)$ distribution. Hence, in the third line we obtain R draws from the normal distribution with mean `mu_hat` and variance `Dmu`. Then, in the fourth line we take the absolute value of the draws, followed by taking logs. Finally, we compute the mean.

1.3 Normal Model with Unknown Variance

Now, we extend the previous one-parameter model to allow the variance of the measurement error σ^2 to be unknown. The model is

$$(y_n \,|\, \mu, \sigma^2) \sim \mathcal{N}(\mu, \sigma^2), \quad n = 1, \ldots, N, \tag{1.6}$$

where both μ and σ^2 are now unknown. We assume the same prior for $\mu : \mathcal{N}(\mu_0, \sigma_0^2)$. As for σ^2, which takes only positive values, a convenient prior is the inverse-gamma prior. It turns out that this prior is conjugate for the likelihood of the model in (1.6).

A random variable X is said to have an **inverse-gamma distribution** with shape parameter $\alpha > 0$ and scale parameter $\beta > 0$ if its density is given by

$$f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-(\alpha+1)} \mathrm{e}^{-\beta/x}. \tag{1.7}$$

We write $X \sim \mathcal{IG}(\alpha, \beta)$. This is the parameterization we use throughout the book. There are other common parameterizations for the inverse-gamma distribution—when comparing derivations and results across books and papers, it is important to first determine the parameterization used.

For $X \sim \mathcal{IG}(\alpha, \beta)$, its mean and variance are given by

$$\mathbb{E}X = \frac{\beta}{\alpha - 1}$$

for $\alpha > 1$, and

$$\mathrm{Var}(X) = \frac{\beta^2}{(\alpha-1)^2(\alpha-2)}$$

for $\alpha > 2$.

Given the likelihood and priors, we can now derive the joint posterior distribution of (μ, σ^2). Again, by Bayes' theorem, the joint posterior density is given by

$$\begin{aligned} p(\mu, \sigma^2 \,|\, \mathbf{y}) &\propto p(\mu, \sigma^2, \mathbf{y}) \\ &\propto p(\mu)p(\sigma^2)p(\mathbf{y} \,|\, \mu, \sigma^2) \\ &\propto \mathrm{e}^{-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2}(\sigma^2)^{-(\nu_0+1)}\mathrm{e}^{-\frac{S_0}{\sigma^2}}\prod_{n=1}^{N}(\sigma^2)^{-\frac{1}{2}}\mathrm{e}^{-\frac{1}{2\sigma^2}(y_n-\mu)^2}. \end{aligned} \tag{1.8}$$

Even though we have an explicit expression for the joint posterior density, it is not obvious how we can compute analytically various quantities of interest, such as, $\mathbb{E}(\mu \,|\, \mathbf{y})$, the posterior mean of μ or $\mathrm{Var}(\sigma^2 \,|\, \mathbf{y})$, the posterior variance of σ^2. One way forward is to use Monte Carlo simulation to approximate those quantities.

For example, to approximate $\mathrm{Var}(\sigma^2 \,|\, \mathbf{y})$, we first obtain draws from the posterior distribution $(\mu, \sigma^2 \,|\, \mathbf{y})$, say, $(\mu^{(1)}, \sigma^{2(1)}), \ldots, (\mu^{(R)}, \sigma^{2(R)})$. Then, we compute

$$\frac{1}{R}\sum_{r=1}^{R}(\sigma^{2(r)} - \bar{\sigma}^2)^2,$$

where $\bar{\sigma}^2$ is the mean of $\sigma^{2(1)}, \ldots, \sigma^{2(R)}$.

Now the problem becomes: How do we sample from the posterior distribution? This brings us to Markov chain Monte Carlo (MCMC) methods, which are a broad class of algorithms for sampling from arbitrary probability distributions. This is achieved by constructing a Markov chain such that its limiting distribution is the desired distribution. Below we discuss one such method, called **Gibbs sampling**.

Specifically, suppose we wish to sample from the target distribution $p(\boldsymbol{\Theta}) = p(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$. A Gibbs sampler constructs a Markov chain $\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \ldots$ using the full conditional distributions $p(\boldsymbol{\theta}_i \,|\, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}_{i+1}, \ldots, \boldsymbol{\theta}_n)$ as the transition kernels. Under certain regularity conditions, the limiting distribution of the Markov chain thus constructed is the target distribution.

Operationally, we start from an initial state $\boldsymbol{\Theta}^{(0)}$. Then, we repeat the following steps from $r = 1$ to R:

1. Given the current state $\boldsymbol{\Theta} = \boldsymbol{\Theta}^{(r)}$, generate $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_n)$ as follows:
   (a) Draw $\mathbf{Y}_1 \sim p(\mathbf{y}_1 \,|\, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_n)$.
   (b) Draw $\mathbf{Y}_i \sim p(\mathbf{y}_i \,|\, \mathbf{Y}_1, \ldots, \mathbf{Y}_{i-1}, \boldsymbol{\theta}_{i+1}, \ldots, \boldsymbol{\theta}_n)$, $i = 2, \ldots, n-1$.
   (c) Draw $\mathbf{Y}_n \sim p(\mathbf{y}_n \,|\, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n-1})$.

2. Set $\boldsymbol{\Theta}^{(r+1)} = \mathbf{Y}$.

It is important to note that the Markov chain $\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \ldots$ does not converge to a fixed vector of constants. Rather, it is the distribution of $\boldsymbol{\Theta}^{(r)}$ that converges to the target distribution.

In practice, one typically discards the first R_0 draws to eliminate the effect of the initial state $\boldsymbol{\Theta}^{(0)}$. The discarded draws are often refereed to as the "burn-in". There are a number of convergence diagnostics to test if the Markov chain has converged to the limiting distribution. One popular test is the Geweke's convergence diagnostics described in Geweke (1992).

Now, after this discussion of Gibbs sampling, we return to the estimation of the two-parameter normal model. To construct a Gibbs sampler to draw from the posterior distribution $p(\mu, \sigma^2 \,|\, \mathbf{y})$ in (1.8), we need to derive two conditional distributions: $p(\mu \,|\, \mathbf{y}, \sigma^2)$ and $p(\sigma^2 \,|\, \mathbf{y}, \mu)$.

To derive the first conditional distribution, note that given σ^2, this is the same normal model with known variance discussed in last section. Thus, using the same derivation before, we have

$$\begin{aligned} p(\mu \,|\, \mathbf{y}, \sigma^2) &\propto p(\mu) p(\mathbf{y} \,|\, \mu, \sigma^2) \\ &\propto \mathrm{e}^{\left[-\frac{1}{2}\left(\left(\frac{1}{\sigma_0^2}+\frac{N}{\sigma^2}\right)\mu^2 - 2\mu\left(\frac{\mu_0}{\sigma_0^2}+\frac{N\bar{y}}{\sigma^2}\right)\right)\right]}. \end{aligned}$$

Hence, $(\mu \,|\, \mathbf{y}, \sigma^2) \sim \mathcal{N}(\widehat{\mu}, D_\mu)$, where

$$D_\mu = \left(\frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}\right)^{-1}, \quad \widehat{\mu} = D_\mu \left(\frac{\mu_0}{\sigma_0^2} + \frac{N\bar{y}}{\sigma^2}\right).$$

Next, we derive the conditional distribution of σ^2:

$$\begin{aligned} p(\sigma^2 \,|\, \mathbf{y}, \mu) &\propto p(\sigma^2) p(\mathbf{y} \,|\, \mu, \sigma^2) \\ &\propto (\sigma^2)^{-(\nu_0+1)} \mathrm{e}^{-\frac{S_0}{\sigma^2}} (\sigma^2)^{-N/2} \mathrm{e}^{-\frac{1}{2\sigma^2}\sum_{n=1}^N (y_n-\mu)^2} \\ &\propto (\sigma^2)^{-(\nu_0+N/2+1)} \mathrm{e}^{-\frac{S_0+\sum_{n=1}^N (y_n-\mu)^2/2}{\sigma^2}}. \end{aligned}$$

Is this a known distribution? It turns out that this is an inverse-gamma distribution. Comparing this expression with the generic inverse-gamma distribution given in (1.7), we have

$$(\sigma^2 \,|\, \mathbf{y}, \mu) \sim \mathcal{IG}\left(\nu_0 + \frac{N}{2}, S_0 + \frac{1}{2}\sum_{n=1}^N (y_n - \mu)^2\right).$$

To summarize, the Gibbs sampler for the two-parameter model is given as below. Pick some initial values $\mu^{(0)} = a_0$ and $\sigma^{2(0)} = b_0 > 0$. Then, repeat the following steps from $r = 1$ to R:

1. Draw $\mu^{(r)} \sim p(\mu \,|\, \mathbf{y}, \sigma^{2(r-1)})$.
2. Draw $\sigma^{2(r)} \sim p(\sigma^2 \,|\, \mathbf{y}, \mu^{(r)})$.

After discarding the first R_0 draws as burn-in, we can use the draws $\mu^{(R_0+1)}, \ldots, \mu^{(R)}$ and $\sigma^{2(R_0+1)}, \ldots, \sigma^{2(R)}$ to compute various quantities of interest. For example, we can use

$$\frac{1}{R - R_0} \sum_{r=R_0+1}^{R} \mu^{(r)}$$

as an estimate of $\mathbb{E}(\mu \,|\, \mathbf{y})$.

As an illustration, the following MATLAB script `norm_2para.m` first generates a dataset of $N = 100$ observations from the two-parameter normal model with $\mu = 3$ and $\sigma^2 = 0.1$. Then, it implements a Gibbs sampler to sequentially to sample from the two conditional distributions: $p(\mu \,|\, \mathbf{y}, \sigma^2)$ and $p(\sigma^2 \,|\, \mathbf{y}, \mu)$.

```
% norm_2para.m
nsim = 10000; burnin = 1000;
N = 100; mu = 3; sig2 = .1;
y = mu + sqrt(sig2)*randn(N,1);
    % prior
mu0 = 0; sig20 = 100;
nu0 = 3; S0 = .5;
    % initialize the Markov chain
mu = 0; sig2 = 1;
store_theta = zeros(nsim,2);

for isim = 1:nsim + burnin
        % sample mu
    Dmu = 1/(1/sig20 + N/sig2);
    mu_hat = Dmu*(mu0/sig20 + sum(y)/sig2);
    mu = mu_hat + sqrt(Dmu)*randn;

        % sample sig2
    sig2 = 1/gamrnd(nu0+N/2,1/(S0+sum((y-mu).^2)/2));

    if isim > burnin
        isave = isim - burnin;
```

```
            % store the parameters
        store_theta(isave,:) = [mu sig2];
    end

end
theta_hat = mean(store_theta)';
```

Sampling from the normal distribution $(\mu \,|\, \mathbf{y}, \sigma^2)$ is the same as before. Sampling from the inverse-gamma distribution $(\sigma^2 \,|\, \mathbf{y}, \mu)$ is done by taking the inverse of a gamma draw—using the built-in function `gamrnd`. Also note that the parameterization of the inverse-gamma distribution in MATLAB is different from the one we use here. Consequently, a minor adjustment is needed.

Using 10,000 posterior draws after a burn-in of 1,000, the posterior means of μ and σ^2 are estimated to be 3.01 and 0.12, respectively.

Chapter 2

Normal Linear Regression

The workhorse model in econometrics is the normal linear regression model. Virtually all other more flexible models are built upon this foundation. It is therefore vital to fully understand how one estimates this model. In this chapter we will provide the details in deriving the likelihood and the posterior sampler.

2.1 Linear Regression in Matrix Notation

To start, suppose we have data on a dependent variable y_t for $t = 1, \ldots, T$. Then, consider the following linear regression model:

$$y_t = \beta_1 + x_{2,t}\beta_2 + \cdots + x_{k,t}\beta_k + \varepsilon_t, \tag{2.1}$$

where $\varepsilon_1, \ldots, \varepsilon_T$ are assumed to be iid $\mathcal{N}(0, \sigma^2)$, $1, x_{2,t}, \ldots, x_{k,t}$ are the k regressors and $\beta_1, \ldots, \beta_k$ are the associated regression coefficients.

To derive the likelihood, it is more convenient to write (2.1) in matrix notation. In particular, we stack the observations over $t = 1, \ldots, T$ so that each row represents the observation at time t. Let $\mathbf{y} = (y_1, \ldots, y_T)'$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)'$. Then, rewrite the whole system of T equations in (2.1) as:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{pmatrix} = \begin{pmatrix} 1 & x_{2,1} & \cdots & x_{k,1} \\ 1 & x_{2,2} & \cdots & x_{k,2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{2,T} & \cdots & x_{k,T} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_T \end{pmatrix}$$

Or more succinctly,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}. \tag{2.2}$$

Since we assume that $\varepsilon_1, \ldots, \varepsilon_T$ are iid $\mathcal{N}(0, \sigma^2)$, $\boldsymbol{\varepsilon}$ has a multivariate normal distribution with mean vector $\mathbf{0}_T$ and covariance matrix $\sigma^2\mathbf{I}_T$, where $\mathbf{0}_T$ is a $T \times 1$ vector of zeros and $\mathbf{I}_T$ is the $T \times T$ identity matrix. That is,

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I}_T). \tag{2.3}$$

2.2 Derivation of Likelihood Function

Recall that the likelihood function is defined as the joint density of the data given the parameters. In our case, the likelihood function is the joint density of $\mathbf{y}$ given $\boldsymbol{\beta}$ and σ^2. To derive the likelihood of the normal linear regression model, we use two useful results. First, an *affine transformation*—i.e., a linear transformation followed by a translation—of a normal random vector is also a normal random vector. Now, $\mathbf{y}$ is an affine transformation of $\boldsymbol{\varepsilon}$, which is assumed to have a multivariate normal distribution given in (2.3). Thus, $\mathbf{y}$ also has a normal distribution. Since a normal distribution is uniquely determined by its mean vector and covariance matrix, it suffices to compute the mean and covariance matrix of $\mathbf{y}$.

This brings us to the next useful result: suppose $\mathbf{u}$ has a mean vector $\boldsymbol{\mu}_\mathbf{u}$ and covariance matrix $\boldsymbol{\Sigma}_\mathbf{u}$. Let $\mathbf{v} = \mathbf{A}\mathbf{u} + \mathbf{c}$ for constant matrices $\mathbf{A}$ and $\mathbf{c}$. Then the mean vector and covariance matrix of $\mathbf{v}$ are given by

$$\mathbb{E}\,\mathbf{v} = \mathbf{A}\boldsymbol{\mu}_\mathbf{u} + \mathbf{c}, \qquad \mathrm{Cov}(\mathbf{u}) = \mathbf{A}\boldsymbol{\Sigma}_\mathbf{u}\mathbf{A}'.$$

Using this result, it is easy to see that given $\boldsymbol{\beta}$ and σ^2,

$$\mathbb{E}\,\mathbf{y} = \mathbf{X}\boldsymbol{\beta}, \qquad \mathrm{Cov}(\mathbf{y}) = \sigma^2\mathbf{I}_T.$$

Putting it all together, we have

$$(\mathbf{y} \,|\, \boldsymbol{\beta}, \sigma^2) \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_T),$$

and the likelihood function is given by:

$$\begin{aligned} p(\mathbf{y} \,|\, \boldsymbol{\beta}, \sigma^2) &= |2\pi\sigma^2\mathbf{I}_T|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\sigma^2\mathbf{I}_T)^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})} \\ &= (2\pi\sigma^2)^{-\frac{T}{2}} \mathrm{e}^{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}, \end{aligned} \tag{2.4}$$

where $|\cdot|$ denotes the determinant. Also note that the second equality follows from the result that for an $n \times n$ matrix $\mathbf{A}$ and scalar c, $|c\mathbf{A}| = c^n|\mathbf{A}|$.

2.3 Independent Priors

The model parameters are $\boldsymbol{\beta}$ and σ^2. Here we consider a convenient prior that assumes prior independence between $\boldsymbol{\beta}$ and σ^2, i.e., $p(\boldsymbol{\beta}, \sigma^2) = p(\boldsymbol{\beta})p(\sigma^2)$. In particular, we consider

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \mathbf{V}_{\boldsymbol{\beta}}), \qquad \sigma^2 \sim \mathcal{IG}(\nu_0, S_0)$$

with prior densities

$$p(\boldsymbol{\beta}) = (2\pi)^{-\frac{k}{2}} |\mathbf{V}_{\boldsymbol{\beta}}|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)' \mathbf{V}_{\boldsymbol{\beta}}^{-1} (\boldsymbol{\beta}-\boldsymbol{\beta}_0)}, \tag{2.5}$$

$$p(\sigma^2) = \frac{S_0^{\nu_0}}{\Gamma(\nu_0)} (\sigma^2)^{-(\nu_0+1)} \mathrm{e}^{-\frac{S_0}{\sigma^2}}. \tag{2.6}$$

2.4 Gibbs Sampler

Now, we derive a Gibbs sampler for the normal linear regression with likelihood given in (2.4) and priors given in (2.5)–(2.6). Specifically, we need to derive the two conditional densities $p(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta})$ and $p(\boldsymbol{\beta} \,|\, \mathbf{y}, \sigma^2)$.

First, using (2.4) and (2.6), we show that the conditional density $p(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta})$ is inverse-gamma:

$$\begin{aligned} p(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta}) &\propto p(\mathbf{y} \,|\, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta}) p(\sigma^2) \\ &\propto (\sigma^2)^{-\frac{T}{2}} \mathrm{e}^{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})} \times (\sigma^2)^{-(\nu_0+1)} \mathrm{e}^{-\frac{S_0}{\sigma^2}} \\ &= (\sigma^2)^{-\left(\frac{T}{2}+\nu_0+1\right)} \mathrm{e}^{-\frac{1}{\sigma^2}\left(S_0 + \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right)}. \end{aligned}$$

We recognize this as the kernel of an inverse-gamma density. In fact, we have

$$(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta}) \sim \mathcal{IG}\left(\nu_0 + \frac{T}{2}, S_0 + \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right).$$

Next, we derive the conditional density $p(\boldsymbol{\beta} \,|\, \mathbf{y}, \sigma^2)$. To that end, first note that the likelihood in (2.4) involves the quadratic term $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$, which can be expanded as

$$\begin{aligned} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) &= \mathbf{y}'\mathbf{y} - \mathbf{y}'\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta}'\mathbf{X}'\mathbf{y} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \\ &= \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} - 2\boldsymbol{\beta}'\mathbf{X}'\mathbf{y} + \mathbf{y}'\mathbf{y}, \end{aligned}$$

where we have used the fact that $\mathbf{y}'\mathbf{X}\boldsymbol{\beta}$ is a scalar, and therefore it is equal to its transpose:

$$\mathbf{y}'\mathbf{X}\boldsymbol{\beta} = (\boldsymbol{\beta}'\mathbf{X}'\mathbf{y})' = \boldsymbol{\beta}'\mathbf{X}'\mathbf{y}.$$

Similarly, the quadratic term in the prior (2.5) can be expanded as

$$(\boldsymbol{\beta}-\boldsymbol{\beta}_0)'\mathbf{V}_{\boldsymbol{\beta}}^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)=\boldsymbol{\beta}'\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}-2\boldsymbol{\beta}'\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_0+\boldsymbol{\beta}_0'\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_0.$$

Finally, the conditional density $p(\boldsymbol{\beta}\,|\,\mathbf{y},\sigma^2)$ is given by

$$\begin{aligned}
p(\boldsymbol{\beta}\,|\,\mathbf{y},\sigma^2) &\propto p(\mathbf{y}\,|\,\boldsymbol{\beta},\sigma^2)p(\boldsymbol{\beta})p(\sigma^2)\\
&\propto \mathrm{e}^{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}\times \mathrm{e}^{-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)'\mathbf{V}_{\boldsymbol{\beta}}^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)}\\
&\propto \mathrm{e}^{-\frac{1}{2}(\boldsymbol{\beta}'\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}-2\boldsymbol{\beta}'\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_0)}\mathrm{e}^{-\frac{1}{2\sigma^2}(\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}-2\boldsymbol{\beta}'\mathbf{X}'\mathbf{y})}\\
&= \mathrm{e}^{-\frac{1}{2}\left[\boldsymbol{\beta}'\left(\mathbf{V}_{\boldsymbol{\beta}}^{-1}+\frac{1}{\sigma^2}\mathbf{X}'\mathbf{X}\right)\boldsymbol{\beta}-2\boldsymbol{\beta}'\left(\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_0+\frac{1}{\sigma^2}\mathbf{X}'\mathbf{y}\right)\right]}. \qquad (2.7)
\end{aligned}$$

Since the exponent is quadratic in $\boldsymbol{\beta}$, $p(\boldsymbol{\beta}\,|\,\mathbf{y},\sigma^2)$ is a multivariate normal density, say,

$$(\boldsymbol{\beta}\,|\,\mathbf{y},\sigma^2)\sim\mathcal{N}(\widehat{\boldsymbol{\beta}},\mathbf{D}_{\boldsymbol{\beta}})$$

for some mean vector $\widehat{\boldsymbol{\beta}}$ and covariance matrix $\mathbf{D}_{\boldsymbol{\beta}}$. Next, we derive explicit expressions for $\widehat{\boldsymbol{\beta}}$ and $\mathbf{D}_{\boldsymbol{\beta}}$.

The kernel of $\mathcal{N}(\widehat{\boldsymbol{\beta}},\mathbf{D}_{\boldsymbol{\beta}})$ is simply

$$\mathrm{e}^{-\frac{1}{2}\left(\boldsymbol{\beta}'\mathbf{D}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}-2\boldsymbol{\beta}'\mathbf{D}_{\boldsymbol{\beta}}^{-1}\widehat{\boldsymbol{\beta}}\right)}.$$

Comparing this kernel with the expression in (2.7), we conclude that

$$\mathbf{D}_{\boldsymbol{\beta}}=\left(\mathbf{V}_{\boldsymbol{\beta}}^{-1}+\frac{1}{\sigma^2}\mathbf{X}'\mathbf{X}\right)^{-1},\quad \widehat{\boldsymbol{\beta}}=\mathbf{D}_{\boldsymbol{\beta}}\left(\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_0+\frac{1}{\sigma^2}\mathbf{X}'\mathbf{y}\right).$$

Even though one can use the built-in function `mvnrnd` in MATLAB to sample from $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$, it is instructive to see how it can be done by simply transforming independent standard normal random variables.

Algorithm 2.1. (Sampling from Normal Distribution).

To generate R independent draws from $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ of dimension n, carry out the following steps:

1. Compute the lower Cholesky factorization $\boldsymbol{\Sigma}=\mathbf{B}\mathbf{B}'$.
2. Generate $\mathbf{Z}=(Z_1,\ldots,Z_n)'$ by drawing $Z_1,\ldots,Z_n\sim\mathcal{N}(0,1)$.
3. Return $\mathbf{U}=\boldsymbol{\mu}+\mathbf{B}\mathbf{Z}$.
4. Repeat Steps 2 and 3 independently R times.

Finally, we summarize the the Gibbs sampler for the linear regression model as follows:

Algorithm 2.2. (Gibbs Sampler for the Linear Regression Model).

Pick some initial values $\boldsymbol{\beta}^{(0)} = \mathbf{a}_0$ and $\sigma^{2(0)} = b_0 > 0$. Then, repeat the following steps from $r = 1$ to R:

1. Draw $\sigma^{2(r)} \sim p(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta}^{(r-1)})$ (inverse-gamma).
2. Draw $\boldsymbol{\beta}^{(r)} \sim p(\boldsymbol{\beta} \,|\, \mathbf{y}, \sigma^{2(r)})$ (multivariate normal).

As an example, the following MATLAB script `linreg.m` first generates a sample of $T = 500$ observations from a normal linear regression model. It then implements the Gibbs sampler in Algorithm 2.2, where the sampler is initialized using the least squares estimate. The posterior means of the model parameters are stored in the variable `theta_hat` and the corresponding 95% credible intervals are stored in `theta_CI`.

```
% linreg.m
nsim = 10000; burnin = 1000;

    % generate the data
T = 500; % sample size
beta = [1 5]';  sig2 = .5;
X = [ones(T,1) 1+randn(T,1)];
y = X*beta + sqrt(sig2)*randn(T,1);

    % prior
beta0 = [0 0]'; iVbeta0 = eye(2)/100;
nu0 = 3;  S0 = 1*(nu0-1);

    % initialize the Markov chain
beta = (X'*X)\(X'*y);
sig2 = sum((y-X*beta).^2)/T;
store_theta = zeros(nsim,3);

for isim = 1:nsim + burnin
        % sample beta
    Dbeta = (iVbeta0 + X'*X/sig2)\speye(2);
    beta_hat = Dbeta*(iVbeta0*beta0 + X'*y/sig2);
    C = chol(Dbeta,'lower');
    beta = beta_hat + C*randn(2,1);
```

```
        % sample sig2
    e = y-X*beta;
    sig2 = 1/gamrnd(nu0 + T/2,1/(S0 + e'*e/2));

        % store the parameters
    if isim > burnin
        isave = isim - burnin;
        store_theta(isave,:) = [beta' sig2];
    end
end
theta_hat = mean(store_theta);
theta_CI = quantile(store_theta,[.025 .975]);
```

Chapter 3

Linear Regression with General Covariance Matrix

Last chapter we discussed the estimation of the standard normal linear regression with independent errors using the Gibbs sampler. We now consider a few extensions of this standard model, including regressions with non-Gaussian and serially correlated errors. To estimate these more flexible variants, we will introduce a few sampling techniques and numerical methods that will be drawn on heavily in later chapters.

3.1 General Framework

In the last chapter, we considered a linear regression of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_T)$. In particular, the elements of $\boldsymbol{\varepsilon}$ are independent and have the same variance σ^2. Here we relax this modeling assumption and instead consider $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$, where $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ is a generic $T \times T$ covariance matrix that depends on the parameter vector $\boldsymbol{\theta}$. For example, in the standard linear regression, we have simply $\boldsymbol{\Sigma}_{\boldsymbol{\theta}} = \sigma^2\mathbf{I}_T$ with $\boldsymbol{\theta} = \sigma^2$. For now we leave $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ unspecified; we will consider a few examples in coming sections.

It follows from the distributional assumption of $\boldsymbol{\varepsilon}$ that the joint distribution of the data is $(\mathbf{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$ with the likelihood function:

$$\begin{aligned} p(\mathbf{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\theta}) &= |2\pi\boldsymbol{\Sigma}_{\boldsymbol{\theta}}|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})} \\ &= (2\pi)^{-\frac{T}{2}} |\boldsymbol{\Sigma}_{\boldsymbol{\theta}}|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}, \end{aligned} \tag{3.1}$$

where we have used the result that for an $n \times n$ matrix $\mathbf{A}$ and scalar c, $|c\mathbf{A}| = c^n|\mathbf{A}|$.

We assume the independent priors $p(\boldsymbol{\beta}, \boldsymbol{\theta}) = p(\boldsymbol{\beta})p(\boldsymbol{\theta})$ with $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \mathbf{V}_{\boldsymbol{\beta}})$ as before. We will specify the prior $p(\boldsymbol{\theta})$ later on when the structure of the model is determined.

To estimate the model using MCMC methods, we iteratively sample from the conditional densities $p(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\theta})$ and $p(\boldsymbol{\theta} \,|\, \mathbf{y}, \boldsymbol{\beta})$. Next, we derive the conditional density $p(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\theta})$.

Given the likelihood (3.1) and the prior for $\boldsymbol{\beta}$, one can use a similar derivation as in (2.7) to show that

$$(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\theta}) \sim \mathcal{N}(\widehat{\boldsymbol{\beta}}, \mathbf{D}_{\boldsymbol{\beta}}), \tag{3.2}$$

where

$$\mathbf{D}_{\boldsymbol{\beta}} = \left(\mathbf{V}_{\boldsymbol{\beta}}^{-1} + \mathbf{X}'\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\mathbf{X}\right)^{-1}, \quad \widehat{\boldsymbol{\beta}} = \mathbf{D}_{\boldsymbol{\beta}}\left(\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_0 + \mathbf{X}'\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\mathbf{y}\right).$$

Note that the above expressions involve the inverse of the $T \times T$ covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$. Inverting this matrix in general is very time-consuming. Fortunately, for most common models either $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ or $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}$ is a **band matrix**—i.e., a sparse matrix where the nonzero elements are arranged along the main diagonal. Exploiting this structure can drastically reduce the computational costs.

In the following sections, we look at a variety of examples for $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$.

3.2 Linear Regression with t Errors

The first example we consider is a linear regression with Student's t errors. Compared to normal distributions, t distributions have heavier tails. As such, they are more robust against misspecification. A large literature has demonstrated that models with heavier tails than those of normal distributions generally fit financial and macroeconomic data better.

We say that a random variable X follows a **Student's t distribution** if its density function is given by

$$f(x) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\nu\pi\sigma^2}\Gamma(\nu/2)}\left(1 + \frac{(x-\mu)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}. \tag{3.3}$$

We denote the distribution as $\mathcal{T}_\nu(\mu, \sigma^2)$.

If we assume that the errors $\{\varepsilon_t\}$ are iid $\mathcal{T}_\nu(\mu, \sigma^2)$, we can in principle derive the likelihood as before. However, the posterior distribution of $\boldsymbol{\beta}$ would no longer be normal, and estimation becomes more difficult.

3.2.1 Latent Variable Representation of t Distribution

Below we introduce a computational strategy to facilitate estimation. In a nutshell, we introduce a vector of *latent variables* such that given these latent variables, standard estimation methods can be used. In our setting, instead of working with the t distribution directly, we write it as a scale mixture of normal distributions. Given the augmented scale mixtures, the errors $\{\varepsilon_t\}$ then follow normal distributions with different variances. Consequently, standard estimation methods involving normal likelihood can then be used. This strategy is first implemented in Geweke (1993).

More specifically, suppose $(X \,|\, \lambda) \sim \mathcal{N}(\mu, \lambda\sigma^2)$, where λ is a latent variable that scales the variance of X. If we assume that λ has an inverse-gamma distribution, particularly, $\lambda \sim \mathcal{IG}(\nu/2, \nu/2)$, then the marginal distribution of X—unconditionally of λ—is $\mathcal{T}_\nu(\mu, \sigma^2)$.

To see this, note that the joint density of X and λ is given by

$$\begin{aligned} f(x,\lambda) &= f(x \,|\, \lambda) f(\lambda) \\ &= (2\pi\lambda\sigma^2)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2\lambda\sigma^2}(x-\mu)^2} \times \frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \lambda^{-\left(\frac{\nu}{2}+1\right)} \mathrm{e}^{-\frac{\nu}{2\lambda}} \\ &= (2\pi\sigma^2)^{-\frac{1}{2}} \frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \lambda^{-\left(\frac{\nu+1}{2}+1\right)} \mathrm{e}^{-\frac{\nu}{2\lambda}\left(1+\frac{(x-\mu)^2}{\nu\sigma^2}\right)}. \end{aligned}$$

We next integrate this joint density with respect to λ to obtain the marginal density of X. To that end, we use the following result:

$$\int_0^\infty z^{-(\alpha+1)} \mathrm{e}^{-\frac{\beta}{z}} \mathrm{d}z = \beta^{-\alpha}\Gamma(\alpha), \tag{3.4}$$

which follows from the fact that the density of an inverse-gamma random variable $Z \sim \mathcal{IG}(\alpha, \beta)$ integrates to one, i.e.,

$$\int_0^\infty \frac{\beta^\alpha}{\Gamma(\alpha)} z^{-(\alpha+1)} \mathrm{e}^{-\frac{\beta}{z}} \mathrm{d}z = 1.$$

Hence, the marginal density of X is given by:

$$\begin{aligned} f(x) &= \int_0^\infty f(x,\lambda) \mathrm{d}\lambda \\ &= (2\pi\sigma^2)^{-\frac{1}{2}} \frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \int_0^\infty \lambda^{-\left(\frac{\nu+1}{2}+1\right)} \mathrm{e}^{-\frac{\nu}{2\lambda}\left(1+\frac{(x-\mu)^2}{\nu\sigma^2}\right)} \mathrm{d}\lambda \\ &= (2\pi\sigma^2)^{-\frac{1}{2}} \frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \left(1+\frac{(x-\mu)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}} \left(\frac{\nu}{2}\right)^{-\frac{\nu+1}{2}} \Gamma\left(\frac{\nu+1}{2}\right) \\ &= \frac{\Gamma((\nu+1)/2)}{\sqrt{\nu\pi\sigma^2}\Gamma(\nu/2)} \left(1+\frac{(x-\mu)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}, \end{aligned}$$

which is the density of $\mathcal{T}_\nu(\mu, \sigma^2)$ given in (3.3). In the above derivation, the third equality follows from (3.4) with $\alpha = (\nu+1)/2$ and $\beta = \nu(1 + \frac{(x-\mu)^2}{\nu\sigma^2})/2$.

Hence, we have shown that if $X \sim \mathcal{T}_\nu(\mu, \sigma^2)$, it has the latent variable representation $(X \,|\, \lambda) \sim \mathcal{N}(\mu, \lambda\sigma^2)$, where $\lambda \sim \mathcal{IG}(\nu/2, \nu/2)$. This is an instant of a general computational strategy called **data augmentation**, in which latent variables are introduced to facilitate computations. We will see more examples in coming chapters.

3.2.2 Estimation with Known ν

Using the latent variable representation discussed above, a linear regression with Student's t errors can be written as:

$$y_t = \mathbf{x}_t'\boldsymbol{\beta} + \varepsilon_t,$$

where $\mathbf{x}_t = (1, x_{2,t}, \ldots, x_{k,t})'$ is the vector of covariates, $(\varepsilon_t \,|\, \lambda_t) \sim \mathcal{N}(0, \sigma^2\lambda_t)$ and $\lambda_t \sim \mathcal{IG}(\nu/2, \nu/2)$. Assume for now that ν is known and consider the following independent, conjugate priors

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \mathbf{V}_{\boldsymbol{\beta}}), \quad \sigma^2 \sim \mathcal{IG}(\nu_0, S_0).$$

Compared to the normal linear regression, we have an extra block of latent variables $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_T)'$ in addition to the parameters $\boldsymbol{\beta}$ and σ^2.

The Gibbs sampler consists of sequentially drawing from the following three conditional densities: $p(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\lambda}, \sigma^2)$, $p(\boldsymbol{\lambda} \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2)$ and $p(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\lambda})$.

The first step of drawing from $p(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\lambda}, \sigma^2)$ is easy, because it falls within the framework in Section 3.1. To see that, rewrite the model as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_T, \sigma^2\boldsymbol{\Lambda})$ with $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_T)$. Therefore, we can use the result in (3.2) with $\boldsymbol{\Sigma}_{\boldsymbol{\theta}} = \sigma^2\boldsymbol{\Lambda}$ to obtain:

$$(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\lambda}, \sigma^2) \sim \mathcal{N}(\widehat{\boldsymbol{\beta}}, \mathbf{D}_{\boldsymbol{\beta}}), \tag{3.5}$$

where

$$\mathbf{D}_{\boldsymbol{\beta}} = \left(\mathbf{V}_{\boldsymbol{\beta}}^{-1} + \frac{1}{\sigma^2}\mathbf{X}'\boldsymbol{\Lambda}^{-1}\mathbf{X}\right)^{-1}, \quad \widehat{\boldsymbol{\beta}} = \mathbf{D}_{\boldsymbol{\beta}}\left(\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_0 + \frac{1}{\sigma^2}\mathbf{X}'\boldsymbol{\Lambda}^{-1}\mathbf{y}\right).$$

Note that $\boldsymbol{\Lambda}^{-1} = \text{diag}(\lambda_1^{-1}, \ldots, \lambda_T^{-1})$ is a diagonal matrix that can easily be constructed in MATLAB (more discussion on the computation below).

Next, to sample from $p(\boldsymbol{\lambda} \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2)$, note that it is a T-dimensional density. But fortunately, $\lambda_1, \ldots, \lambda_T$ are conditionally independent given the data and $(\boldsymbol{\beta}, \sigma^2)$, i.e., $p(\boldsymbol{\lambda} \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2)$ is a product of T univariate densities. Moreover, each of these is an inverse-gamma density. To show this, recall that the prior distribution of each λ_t is given by $\lambda_t \sim \mathcal{I}G(\nu/2, \nu/2)$. Then, we have

$$\begin{aligned} p(\boldsymbol{\lambda} \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2) &\propto p(\mathbf{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\lambda}, \sigma^2) p(\boldsymbol{\lambda} \,|\, \nu) p(\boldsymbol{\beta}) p(\sigma^2) \\ &\propto p(\mathbf{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\lambda}, \sigma^2) p(\boldsymbol{\lambda} \,|\, \nu) \\ &\propto \prod_{t=1}^{T} \left[(2\pi\lambda_t\sigma^2)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2\lambda_t\sigma^2}(y_t - \mathbf{x}_t'\boldsymbol{\beta})^2} \times (\lambda_t)^{-\left(\frac{\nu}{2}+1\right)} \mathrm{e}^{-\frac{\nu}{2\lambda_t}} \right] \\ &\propto \prod_{t=1}^{T} \left[(\lambda_t)^{-\left(\frac{\nu+1}{2}+1\right)} \mathrm{e}^{-\frac{1}{2\lambda_t}\left(\nu + \frac{(y_t - \mathbf{x}_t'\boldsymbol{\beta})^2}{\sigma^2}\right)} \right], \end{aligned}$$

which is a product of inverse-gamma kernels. Therefore, we conclude

$$(\lambda_t \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2) \sim \mathcal{I}G\left(\frac{\nu+1}{2}, \frac{1}{2}\left(\nu + \frac{(y_t - \mathbf{x}_t'\boldsymbol{\beta})^2}{\sigma^2} \right) \right).$$

Lastly, we leave it as an exercise to show that

$$(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\lambda}) \sim \mathcal{I}G\left(\nu_0 + \frac{T}{2}, S_0 + \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'\boldsymbol{\Lambda}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right).$$

We summarize the Gibbs sampler for the linear regression with t errors (and known ν) as follows:

Algorithm 3.1. Gibbs Sampler for the Linear Regression with t errors (ν known).

Pick some initial values $\boldsymbol{\beta}^{(0)} = \mathbf{a}_0$ and $\sigma^{2(0)} = b_0 > 0$. Then, repeat the following steps from $r = 1$ to R:

1. Draw $\lambda^{(r)} \sim p(\lambda_t \,|\, \mathbf{y}, \boldsymbol{\beta}^{2(r-1)}, \sigma^{2(r-1)})$ for $t = 1, \ldots, T$ (inverse-gamma).
2. Draw $\boldsymbol{\beta}^{(r)} \sim p(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\lambda}^{(r)}, \sigma^{2(r-1)})$ (multivariate normal).
3. Draw $\sigma^{2(r)} \sim p(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\lambda}^{(r)}, \boldsymbol{\beta}^{(r)})$ (inverse-gamma).

3.2.3 Estimation with Unknown ν

In practice, of course, the degree of freedom parameter ν would not be known. In this section we allow ν to be unknown and estimate it from the data. To do so, all

we need to do is to add an extra block to Algorithm 3.1 to sample ν given other parameters and latent variables. Since the other blocks are implemented given ν, they stay exactly the same as before. This modular nature of MCMC algorithms is one of its strength—extensions can often be implemented by slightly changing existing code.

To complete the model specification, we need to specify a prior for ν. Here we consider the uniform prior on the interval $(2, \bar{\nu})$. Assuming $\nu > 2$ ensures the variance of the error term exists. And we choose the upper bound $\bar{\nu}$ sufficiently large, e.g., $\bar{\nu} = 50$, so that the t distribution can in principle closely resemble the normal distribution.

With this uniform prior, the conditional density $p(\nu \,|\, \mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\beta}, \sigma^2)$ is given by:

$$\begin{aligned} p(\nu \,|\, \mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\beta}, \sigma^2) &\propto p(\mathbf{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\lambda}, \sigma^2) p(\boldsymbol{\lambda} \,|\, \nu) p(\boldsymbol{\beta}) p(\sigma^2) p(\nu) \\ &\propto p(\boldsymbol{\lambda} \,|\, \nu) p(\nu) \\ &\propto \prod_{t=1}^{T} \frac{(\nu/2)^{\frac{\nu}{2}}}{\Gamma(\nu/2)} \lambda_t^{-\left(\frac{\nu}{2}+1\right)} \mathrm{e}^{-\frac{\nu}{2\lambda_t}} \\ &= \frac{(\nu/2)^{\frac{T\nu}{2}}}{\Gamma(\nu/2)^T} \left(\prod_{t=1}^{T} \lambda_t \right)^{-\left(\frac{\nu}{2}+1\right)} \mathrm{e}^{-\frac{\nu}{2} \sum_{t=1}^{T} \lambda_t^{-1}} \end{aligned} \tag{3.6}$$

for $2 < \nu < \bar{\nu}$, and 0 otherwise. Note that conditional on $\boldsymbol{\lambda}$, ν does not depend on other parameters or the data, i.e., $p(\nu \,|\, \mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\beta}, \sigma^2) = p(\nu \,|\, \boldsymbol{\lambda})$. This conditional density of ν is nonstandard. To implement the Gibbs sampler, we somehow need to find a way to sample from this nonstandard density.

To that end, we introduce the **inverse-transform method**, which can be used, in principle, to simulate a random variable X from any cumulative distribution function (cdf) F:

Algorithm 3.2. (Inverse-Transform Method).

1. Generate U from $\mathcal{U}(0, 1)$.
2. Return $X = F^{-1}(U)$, where F^{-1} is the inverse function of F.

It is easy to check that the random variable $X = F^{-1}(U)$ has the cdf F. Specifically, the cdf of X evaluated at x is defined to be

$$\mathbb{P}(X \leqslant x) = \mathbb{P}(F^{-1}(U) \leqslant x) = \mathbb{P}(U \leqslant F(x)) = \int_0^{F(x)} 1 \,\mathrm{d}u = F(x),$$

which is the same as F evaluated at x.

In practice, however, the inverse-transform method is difficult to implement, since the inverse cdf is often not available analytically. Here we introduce a simulation method called the Griddy-Gibbs sampler that is useful for approximating sampling from univariate distributions with bounded support. The Griddy-Gibbs sampler is essentially a discretized version of the inverse-transform method and it only requires the evaluation of the density (up to a normalizing constant).

The idea is to construct an approximation of the cdf of X on a fine grid. Given this discretized cdf, we can then implement the inverse-transform method for a discrete random variable.

Algorithm 3.3. (Griddy-Gibbs Sampler).

Suppose we wish to sample X with density f and bounded support on (a, b).

1. Construct a grid with grid points $x_1, \ldots, x_n$, where $x_1 = a$ and $x_n = b$.
2. Compute $F_i = \sum_{j=1}^{i} f(x_j)$.
3. Generate U from $\mathcal{U}(0, 1)$.
4. Find the smallest positive integer k such that $F_k \geqslant U$ and return $X = x_k$.

The following MATLAB function `sample_nu_GG.m` implements the Griddy-Gibbs sampler to sample from the conditional density $p(\nu \,|\, \boldsymbol{\lambda})$. This script takes three inputs: `lam`, the vector of latent variables $\boldsymbol{\lambda}$; `nu_ub`, the prior upper bound $\overline{\nu}$; and `n_grid`, the number of grid points. In the implementation we evaluate the log-density rather than the density itself to achieve better numerical accuracy. In addition, note that we construct a random grid so that the grid points are different across MCMC iterations. Otherwise, with the same set of grid points across iterations, the exact values of ν would be drawn multiple times.

```
function [nu f_nu] = sample_nu_GG(lam,nu_ub,n_grid)
  T = size(lam,1);
  sum1 = sum(log(lam));
  sum2 = sum(1./lam);
  f_nu = @(x) T*(x/2.*log(x/2)-gammaln(x/2)) ...
    - (x/2+1)*sum1 - x/2*sum2;
  nu_grid = linspace(2+rand/100,nu_ub-rand/100,n_grid)';
  lp_nu = f_nu(nu_grid);           % log-density of nu
  p_nu = exp(lp_nu - max(lp_nu));  % density of nu (unnormalized)
  p_nu = p_nu/sum(p_nu);   % density of nu (normalized)
  cdf_nu = cumsum(p_nu);   % cdf of nu
  nu = nu_grid(find(rand<cdf_nu, 1));
```

```
end
```

If the grid is fine enough, the Griddy-Gibbs sampler typically provides a good approximation. The main drawback, however, is that it is difficult to generalize to higher dimensions—the number of grid points needs to increase exponentially to maintain the same level of accuracy.

Next, we discuss an alternative way to handle situations where at least one of the Gibbs steps requires drawing from a nonstandrd distribution. Suppose we wish to generate a sample from the density f. Similar to the Gibbs sampler, the **Metropolis-Hastings algorithm** constructs a Markov chain $\{\mathbf{X}_t, t = 0, 1, \ldots\}$ in such a way that its limiting density is f. Suppose the current state of the Markov chain is in state $\mathbf{x}$ at time t. A transition of the Markov chain from state $\mathbf{x}$ is carried out in two stages. First a *proposal* state $\mathbf{Y}$ is drawn from a transition density $q(\cdot \,|\, \mathbf{x})$. This state is *accepted* as the new state, with probability $\alpha(\mathbf{x}, \mathbf{Y})$, or *rejected* otherwise. In the latter case the chain remains in state $\mathbf{x}$.

By choosing the **acceptance probability** to satisfy the detailed balance equations, the Markov chain thus constructed would have the desired limiting density f. The explicit acceptance probability of each type of Metropolis-Hastings algorithms would be given below. Here we would not go into the details of Markov chain theory. We refer the interested readers to Chib and Greenberg (1995) for more details.

One variant of the Metropolis-Hastings algorithms is the so-called **independence-chain sampler** obtained by choosing the proposal transition density $q(\mathbf{y} \,|\, \mathbf{x})$ to be independent of $\mathbf{x}$; that is, $q(\mathbf{y} \,|\, \mathbf{x}) = g(\mathbf{y})$ for some pdf $g(\mathbf{y})$. Thus, starting from a previous state $\mathbf{X}$ a candidate state $\mathbf{Y}$ is generated from $g(\mathbf{y})$ and accepted with probability

$$\alpha(\mathbf{X}, \mathbf{Y}) = \min\left\{\frac{f(\mathbf{Y})g(\mathbf{X})}{f(\mathbf{X})g(\mathbf{Y})}, 1\right\}.$$

Here it is important that the proposal density g is close to the target f. One popular strategy of obtaining such a proposal density is to approximate f using a normal distribution. In particular, the mean is chosen to be the mode of f, or equivalently, the mode of $\log f$. The precision matrix is then set to be the negative Hessian of $\log f$ evaluated at the mode.

It remains to find the mode of $\log f$. A necessary condition to be the maximum is that $\mathbf{S}(\mathbf{x}) = \partial \log f / \partial \mathbf{x} = \mathbf{0}$. A well-known root-finding algorithm is the **Newton–Raphson method**. This is an iterative procedure where, starting from a guess $\mathbf{x}$, a "better" guess is obtained by approximating the score via a linear function. More precisely, suppose that $\mathbf{x}$ is our initial guess for $\widehat{\mathbf{x}}$ (the root of $\mathbf{S}$). If $\widehat{\mathbf{x}}$ is reasonably

close to $\mathbf{x}$, a first-order Taylor approximation of $\mathbf{S}$ around $\mathbf{x}$ gives

$$\mathbf{S}(\widehat{\mathbf{x}}) \approx \mathbf{S}(\mathbf{x}) + \mathbf{H}(\mathbf{x})(\widehat{\mathbf{x}} - \mathbf{x}),$$

where $\mathbf{H}$ is the Hessian of $\log f$; that is, the matrix of second-order partial derivatives of $\log f$. Since $\mathbf{S}(\widehat{\mathbf{x}}) = \mathbf{0}$ by definition, we have

$$\widehat{\mathbf{x}} \approx \mathbf{x} - \mathbf{H}^{-1}(\mathbf{x})\,\mathbf{S}(\mathbf{x})\ .$$

This suggests the following Newton–Raphson recursion for finding successively better guesses $\mathbf{x}_1, \mathbf{x}_2, \ldots$ converging to $\widehat{\mathbf{x}}$:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \mathbf{H}^{-1}(\mathbf{x}_t)\,\mathbf{S}(\mathbf{x}_t). \tag{3.7}$$

The sequence of successive values is guaranteed to converge to the actual root, provided the function is smooth enough (e.g., has continuous second-order derivatives) and the initial guess is close enough to the root.

We summarize the Newton-Raphson method as follows:

Algorithm 3.4. (Newton-Raphson Method).

Set the tolerance level $\varepsilon > 0$.

1. Pick some initial guess $\mathbf{x}_0$.
2. For $t = 1, 2, \ldots$, apply the Newton–Raphson recursion (3.7), until some stopping criterion, e.g., $|\mathbf{S}(\mathbf{x}_t)| < \varepsilon$, is met.

As an example, we compute the maximum of $\log p(\nu \,|\, \boldsymbol{\lambda})$ using the Newton-Raphson method. It follows from (3.6) that the log-density is given by

$$\log p(\nu \,|\, \boldsymbol{\lambda}) = \frac{T\nu}{2}\log(\nu/2) - T\log\Gamma(\nu/2) - \Big(\frac{\nu}{2}+1\Big)\sum_{t=1}^{T}\lambda_t - \frac{\nu}{2}\sum_{t=1}^{T}\lambda_t^{-1} + c,$$

where c is a constant not dependent of ν. It is easy to check that the first and second derivatives of the log-density with respect to ν are given by

$$\begin{aligned}
\frac{\mathrm{d}\log p(\nu \,|\, \boldsymbol{\lambda})}{\mathrm{d}\nu} &= \frac{T}{2}\log(\nu/2) + \frac{T}{2} - \frac{T}{2}\Psi(\nu/2) - \frac{1}{2}\sum_{t=1}^{T}\log\lambda_t - \frac{1}{2}\sum_{t=1}^{T}\lambda_t^{-1} \\
\frac{\mathrm{d}^2\log p(\nu \,|\, \boldsymbol{\lambda})}{\mathrm{d}\nu^2} &= \frac{T}{2\nu} - \frac{T}{4}\Psi'(\nu/2),
\end{aligned}$$

where $\Psi(x) = \mathrm{d}\log\Gamma(x)/\mathrm{d}x$ and $\Psi'(x) = \mathrm{d}\Psi(x)/\mathrm{d}x$ are respectively the digamma and trigamma functions.

The following MATLAB script `newton_raphson_ex.m` implements the Newton-Raphson method for maximizing $\log p(\nu \mid \boldsymbol{\lambda})$. In particular, it generates a sample of inverse-gamma random variables and stores them in the variable `lam` as $\boldsymbol{\lambda}$. It then goes through the Newton-Raphson recursion in (3.7) until the first derivative of $\log p(\nu \mid \boldsymbol{\lambda})$ is less than 10^{-5}. Note that the mode and the Hessian evaluated at the mode are stored as, respectively, `nut` and `H_nu`.

```
% newton_raphson_ex.m
T = 100;
nu = 5;
lam = 1./gamrnd(nu/2,2/nu,T,1);

T = size(lam,1);
sum1 = sum(log(lam));
sum2 = sum(1./lam);
df_nu = @(x) T/2*(log(x/2) + 1 - psi(x/2)) - .5*(sum1+sum2);
d2f_nu = @(x) T/(2*x) - T/4*psi(1,x/2);
S_nu = 1;
nut = 10;
while abs(S_nu) > 10^(-5)   % stopping criteria
    S_nu = df_nu(nut);
    H_nu = d2f_nu(nut);
    nut = nut - H_nu\S_nu;
end
```

3.2.4 Empirical Example: Fitting Inflation Using an AR(2) with t Errors

We illustrate the estimation methods using an empirical example that involves US PCE inflation. Specifically, let PCE_t denote the PCE index at time t. Then, the annualized growth rate $y_t = 400 \log(\text{PCE}_t/\text{PCE}_{t-1})$ is used as the data. The sample period is from 1959Q1 to 2015Q4.

We consider the following AR(2) model with t errors:

$$y_t = \beta_1 + y_{t-1}\beta_2 + y_{t-2}\beta_3 + \varepsilon_t,$$

where $(\varepsilon_t \mid \lambda_t) \sim \mathcal{N}(0, \lambda_t \sigma^2)$ with $\lambda_t \sim \mathcal{I}G(\nu/2, \nu/2)$.

The MATLAB script `linreg_t.m` is given below. Note that the $T \times T$ inverse matrix $\boldsymbol{\Lambda}^{-1} = (\lambda_1^{-1}, \ldots, \lambda_T^{-1})$ is constructed as a sparse matrix using the line

```
iLam = sparse(1:T,1:T,1./lam);
```

```
% linreg_t.m
nsim = 20000; burnin = 1000;

  % load data
data_raw = load('USPCE_2015Q4.csv');
data = 400*log(data_raw(2:end)./data_raw(1:end-1));
y0 = data(1:2); % [y_{-1}, y_0]
y = data(3:end);
T = size(y,1);
X = [ones(T,1) [y0(2);y(1:end-1)] [y0(1);y0(2);y(1:end-2)]];

  % prior
beta0 = zeros(3,1); iVbeta = speye(3)/100;
rho0 = 0; iVrho = 1;
nu0 = 3; S0 = 1*(nu0 - 1);
nu_ub = 50;

  % initialize the Markov chain
nu = 5;
beta = (X'*X)\(X'*y);
sig2 = sum((y-X*beta).^2)/T;
lam = 1./gamrnd(nu/2,2/nu,T,1);
iLam = sparse(1:T,1:T,1./lam);
store_theta = zeros(nsim,5);  % [beta' sig2 nu]
count_nu = 0;

for isim = 1:nsim + burnin
        % sample beta
    Dbeta = (iVbeta + X'*iLam*X/sig2)\speye(3);
    beta_hat = Dbeta*(iVbeta*beta0 + X'*iLam*y/sig2);
    C = chol(Dbeta,'lower');
    beta = beta_hat + C*randn(3,1);

        % sample lam
    e = y - X*beta;
    lam = 1./gamrnd((nu+1)/2,2./(nu+e.^2/sig2));
    iLam = sparse(1:T,1:T,1./lam);

        % sample sig2
    sig2 = 1/gamrnd(nu0+T/2,1/(S0 + e'*iLam*e/2));

        % sample nu
    [nu,flag] = sample_nu_MH(lam,nu,nu_ub);
```

```
    count_nu = count_nu + flag;

        % store the parameters
    if isim > burnin
        isave = isim - burnin;
        store_theta(isave,:) = [beta' sig2 nu];
    end
end
```

Using a sample of 20000 draws, the posterior means of $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)'$ and ν are estimated to be $(1.99, 0.32, 0.37)'$ and 5.18. Figure 3.1 depicts the histogram of the 20000 posterior draws of ν. As is apparently from the graph, most of the mass of the density is below 10. This indicates that the tails of the error distribution are substantially heavily than those of the normal.

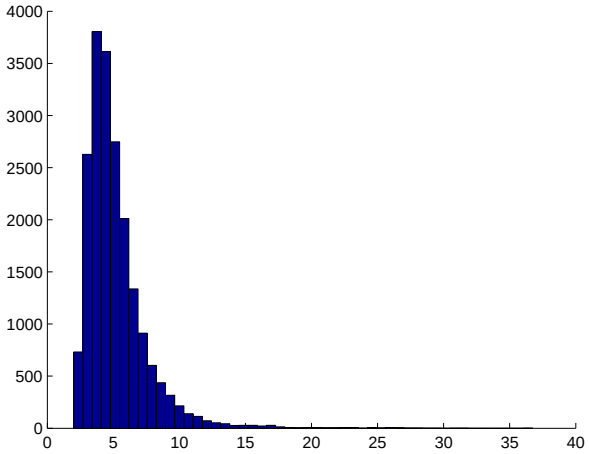

Figure 3.1: Histogram of the posterior draws of ν.

3.3 Linear Regression Model with Moving Average Errors

So far we have assumed that the regression errors are serially independent. In this section we consider a simple model for serially correlated errors. In particular, consider again the linear regression model:

$$y_t = \mathbf{x}_t'\boldsymbol{\beta} + \varepsilon_t,$$

where the errors $\{\varepsilon_t\}$ have a MA(1) structure:

$$\varepsilon_t = u_t + \psi u_{t-1}. \tag{3.8}$$

We set $|\psi| < 1$ for identification purpose, and assume that $u_0 = 0$ and $u_1, \ldots, u_T$ are iid $\mathcal{N}(0, \sigma^2)$. Obviously, if $\psi = 0$, it reduces to the standard linear regression with independent errors. Hence, it would be of interest to test if $\psi = 0$ given the data.

The above model also fits into the general framework described in Section 3.1. To make use of the results derived there, it suffices to derive the distribution of $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_T)'$. Following Chan (2013), we first rewrite (3.8) as

$$\boldsymbol{\varepsilon} = \mathbf{H}_\psi \mathbf{u}, \tag{3.9}$$

where $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_T)$, and

$$\mathbf{H}_\psi = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \psi & 1 & 0 & \cdots & 0 \\ 0 & \psi & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & \cdots & \psi & 1 \end{pmatrix}.$$

We note a few points about computations that are important for later chapters. First, $\mathbf{H}_\psi$ is a $T \times T$ lower triangular matrix with ones on the main diagonal. Consequently, the determinant of $\mathbf{H}_\psi$ is one. Since its determinant is always nonzero, it is invertible. Second, $\mathbf{H}_\psi$ is a **band matrix**—its only nonzero elements are on the main diagonal and the one below the main diagonal. We will exploit this band structure to speed up computation. The inverse $\mathbf{H}_\psi^{-1}$, however, is full. Even though we might write $\mathbf{H}_\psi^{-1}$ in the following derivations, this quantity is not meant to be computed.

It follows from (3.9) that $\boldsymbol{\varepsilon}$ is a linear transformation of a normal vector. Hence, $\boldsymbol{\varepsilon}$ also has a multivariate normal distribution. Next, we compute its mean vector and covariance matrix. It is easy to see that the mean vector is zero: $\mathbb{E}\,\boldsymbol{\varepsilon} = \mathbf{H}_\psi \mathbb{E}\,\mathbf{u} = \mathbf{0}_T$. For the covariance matrix, we have

$$\mathrm{Cov}(\boldsymbol{\varepsilon}) = \mathbf{H}_\psi \mathrm{Cov}(\mathbf{u}) \mathbf{H}_\psi' = \sigma^2 \mathbf{H}_\psi \mathbf{H}_\psi'.$$

Therefore, the covariance matrix is also a band matrix. Finally, we have

$$(\mathbf{y} \,|\, \boldsymbol{\beta}, \sigma^2, \psi) \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{H}_\psi \mathbf{H}_\psi'). \tag{3.10}$$

The linear regression with MA(1) errors therefore falls within the framework in Section 3.1 with $\boldsymbol{\Sigma}_{\boldsymbol{\theta}} = \sigma^2 \mathbf{H}_\psi \mathbf{H}_\psi'$. We assume the same independent priors for $\boldsymbol{\beta}$ and σ^2 as before. For ψ, we assume a uniform prior on the interval $(-1, 1)$: $\psi \sim \mathcal{U}(-1, 1)$.

The Gibbs sampler consists of sequentially drawing from the following three conditional distributions: $p(\boldsymbol{\beta} \,|\, \mathbf{y}, \psi, \sigma^2)$, $p(\psi \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2)$ and $p(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta}, \psi)$.

The first step of drawing from $p(\boldsymbol{\beta} \,|\, \mathbf{y}, \psi, \sigma^2)$ is easy, as we can directly apply the results in Section 3.1 to obtain:

$$(\boldsymbol{\beta} \,|\, \mathbf{y}, \psi, \sigma^2) \sim \mathcal{N}(\widehat{\boldsymbol{\beta}}, \mathbf{D}_{\boldsymbol{\beta}}), \tag{3.11}$$

where

$$\mathbf{D}_{\boldsymbol{\beta}} = \left(\mathbf{V}_{\boldsymbol{\beta}}^{-1} + \frac{1}{\sigma^2} \mathbf{X}' (\mathbf{H}_\psi \mathbf{H}_\psi')^{-1} \mathbf{X} \right)^{-1}, \quad \widehat{\boldsymbol{\beta}} = \mathbf{D}_{\boldsymbol{\beta}} \left(\mathbf{V}_{\boldsymbol{\beta}}^{-1} \boldsymbol{\beta}_0 + \frac{1}{\sigma^2} \mathbf{X}' (\mathbf{H}_\psi \mathbf{H}_\psi')^{-1} \mathbf{y} \right).$$

As mentioned, the inverse $\mathbf{H}_\psi^{-1}$ is a full matrix, and we will avoid computing it explicitly. Instead, we obtain $\widetilde{\mathbf{X}} = \mathbf{H}_\psi^{-1} \mathbf{X}$ by solving the band linear systems

$$\mathbf{H}_\psi \mathbf{Z} = \mathbf{X}$$

for $\mathbf{Z}$. It is easy to see that the unique solution is $\mathbf{Z} = \mathbf{H}_\psi^{-1} \mathbf{X}$. Solving the linear systems is fast as $\mathbf{H}_\psi$ is banded. This can be done in MATLAB using \, the backslash operator. We then obtain $\widetilde{\mathbf{y}} = \mathbf{H}_\psi^{-1} \mathbf{y}$ similarly. Once we have both $\widetilde{\mathbf{y}}$ and $\widetilde{\mathbf{X}}$, we can quickly compute $\mathbf{D}_{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\beta}}$ using:

$$\mathbf{D}_{\boldsymbol{\beta}} = \left(\mathbf{V}_{\boldsymbol{\beta}}^{-1} + \frac{1}{\sigma^2} \widetilde{\mathbf{X}}' \widetilde{\mathbf{X}} \right)^{-1}, \quad \widehat{\boldsymbol{\beta}} = \mathbf{D}_{\boldsymbol{\beta}} \left(\mathbf{V}_{\boldsymbol{\beta}}^{-1} \boldsymbol{\beta}_0 + \frac{1}{\sigma^2} \widetilde{\mathbf{X}}' \widetilde{\mathbf{y}} \right).$$

Alternatively, the conditional distribution of $\boldsymbol{\beta}$ can be derived by first "whitening" the errors. In particular, consider left-multiplying the regression $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ by $\mathbf{H}_\psi^{-1}$. Then, we have

$$\widetilde{\mathbf{y}} = \widetilde{\mathbf{X}} \boldsymbol{\beta} + \mathbf{H}_\psi^{-1} \boldsymbol{\varepsilon} = \widetilde{\mathbf{X}} \boldsymbol{\beta} + \mathbf{u},$$

where $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_T)$. Since the transformed observations are now serially independent, we can apply the standard linear regression results derived in the previous chapter to arrive at the same conditional distribution of $\boldsymbol{\beta}$.

This approach of "whitening" the errors is first considered in Chib and Greenberg (1994), although their implementation is different: instead of left-multiplying the matrix $\mathbf{H}_\psi^{-1}$, they compute the elements of $\widetilde{\mathbf{y}}$ and $\widetilde{\mathbf{X}}$ through a system of equations. The former approach is typically faster as it vectorizes the operations and avoids any for-loops. Using band matrix algorithms to speed up computations for autoregressive models is first proposed in Chan and Jeliazkov (2009) in the context of linear Gaussian state space models. Chan (2013) later adapts the algorithms to fit moving average models.

Next, note that given the uniform prior $\mathcal{U}(-1,1)$ for ψ, the conditional distribution $p(\psi \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2)$ is given by

$$p(\psi \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2) \propto p(\mathbf{y} \,|\, \boldsymbol{\beta}, \sigma^2, \psi) \mathbb{1}(|\psi| < 1),$$

where $\mathbb{1}(\cdot)$ is the indicator function and $p(\mathbf{y} \,|\, \boldsymbol{\beta}, \sigma^2, \psi)$ is the likelihood. This is not a standard distribution, but we can obtain a draw of ψ using a Metropolis-Hastings step. Specifically, we use the Gaussian proposal $\mathcal{N}(\widehat{\psi}, D_\psi)$, where $\widehat{\psi}$ is the mode of $p(\psi \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2)$ and D_ψ is the inverse of the negative Hessian evaluated at the mode.

To obtain the proposal density, we first describe an efficient way to evaluate the likelihood function of the MA(1) model. Traditionally, this is done by writing the model as a linear state space model (see Chapter 6) and evaluating the likelihood using the Kalman filter. Below we use a direct method in Chan (2013) that is based on fast band matrix algorithms.

Specifically, noting that the determinant of $\mathbf{H}_\psi$ is one, it follows from (3.10) that the likelihood has the following explicit expression:

$$\begin{aligned} p(\mathbf{y} \,|\, \boldsymbol{\beta}, \sigma^2, \psi) &= |2\pi\sigma^2 \mathbf{H}_\psi \mathbf{H}_\psi'|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\sigma^2 \mathbf{H}_\psi \mathbf{H}_\psi')^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})} \\ &= (2\pi\sigma^2)^{-\frac{T}{2}} \mathrm{e}^{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{H}_\psi \mathbf{H}_\psi')^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}. \end{aligned}$$

This expression can be evaluated quickly—the only difficulty is to compute the quadratic term in the exponent.

As before, we compute $\mathbf{u} = \mathbf{H}_\psi^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ by solving the band system

$$\mathbf{H}_\psi \mathbf{z} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$$

for $\mathbf{z}$. Then, the quadratic term is simply $\mathbf{u}'\mathbf{u}$. Hence, we have a quick way to evaluate the likelihood of the MA(1) model.

One complication of obtaining the mode $\widehat{\psi}$ is that we do not have analytical expressions for the first two derivatives of the target $\log p(\mathbf{y} \,|\, \boldsymbol{\beta}, \sigma^2, \psi)$—so the Newton-Raphson method cannot be applied. However, we can obtain the mode numerically. In particular, since ψ is bounded on the interval $(-1, 1)$, we can use the MATLAB built-in function `fminbnd` to find the minimizer of the *negative* log-density, which is the same as maximizer of the log-density $\widehat{\psi}$. Once we have the mode, we can use finite difference methods to compute the second derivative of the log-density evaluated at the mode to obtain D_ψ. This is illustrated in the empirical application in the next section.

Lastly, we leave it as an exercise to show that

$$(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta}, \psi) \sim \mathcal{IG}\left(\nu_0 + \frac{T}{2}, S_0 + \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{H}_\psi \mathbf{H}_\psi')^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right).$$

3.3.1 Empirical Example: Fitting Inflation Using an AR(2) with MA(1) Errors

In this section we fit the same inflation data using an AR(2) model with MA(1) errors. We organize the code into one main program and two supporting routines.

The first supporting routine is a function called `llike_MA1.m` that evaluates the log-likelihood. The $T \times T$ matrix $\mathbf{H}_\psi$ is constructed as a sparse matrix using the line

```
Hpsi = speye(T) + sparse(2:T,1:(T-1),psi*ones(1,T-1),T,T);
```

Then, we compute $\mathbf{u} = \mathbf{H}_\psi^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ by solving the relevant band system using the backslash operator. The MATLAB is given below.

```
function ell = llike_MA1(psi,e,sig2)
    T = length(e);
    Hpsi = speye(T) + sparse(2:T,1:(T-1),psi*ones(1,T-1),T,T);
    u = Hpsi\e;
    ell = -T/2*log(2*pi*sig2) -.5*(u'*u)/sig2;
end
```

The second supporting routine is a function called `sample_psi.m` that samples ψ using a Metropolis-Hastings step. In particular, we define the function `fpsi` as the negative log-likelihood and we find its minimizer using `fminbnd`. Then, we compute the negative Hessian numerically using finite difference methods. Once we have obtain $\widehat{\psi}$ and D_ψ, we implement the Metropolis-Hastings step as described in Section 3.2.3.

```
function [psi, flag] = sample_psi(psi,e,sig2)
    fpsi = @(x) -llike_MA1(x,e,sig2);
    psi_hat = fminbnd(fpsi,-1,1);  % the mode
        % compute the negative Hessian
    Del = .01;  % step size
    f_1 = llike_MA1(psi_hat+Del,e,sig2);
    f_0 = llike_MA1(psi_hat,e,sig2);
    f_n1 = llike_MA1(psi_hat-Del,e,sig2);
    d2f_psi = (f_n1 - 2*f_0 + f_1)/Del^2;
    Dpsi = -1/d2f_psi;
    if Dpsi < 0
        Dpsi = .25;
```

```
    end
    psic = psi_hat + chol(Dpsi,'lower')*randn;
    if abs(psic) < .99
        lg = @(x) -.5*(x-psi_hat)'*(Dpsi\(x-psi_hat));
        alpMH = llike_MA1(psic,e,sig2) - llike_MA1(psi,e,sig2) ...
            + lg(psi) - lg(psic);
    else
        alpMH = -inf;
    end
    flag = alpMH > log(rand);
    if flag
        psi = psic;
    end
end
```

In the main MATLAB script `linreg_ma1.m`, after loading the data as before, we implement the posterior sampler as follows:

```
% linreg_ma1.m

% prior
beta0 = zeros(3,1); iVbeta = speye(3)/100;
nu0 = 3; S0 = 1*(nu0 - 1);

% initialize the Markov chain
psi = 0;
beta = (X'*X)\(X'*y);
sig2 = sum((y-X*beta).^2)/T;
Hpsi = speye(T) + sparse(2:T,1:(T-1),psi*ones(1,T-1),T,T);
store_theta = zeros(nsim,5);  % [beta' sig2 psi]
count_psi = 0;

for isim = 1:nsim + burnin
        % sample beta
    X_tilde = Hpsi\X; y_tilde = Hpsi\y;
    Dbeta = (iVbeta + X_tilde'*X_tilde/sig2)\speye(3);
    beta_hat = Dbeta*(iVbeta*beta0 + X_tilde'*y_tilde/sig2);
    beta = beta_hat + chol(Dbeta,'lower')*randn(3,1);

        % sample psi
    e = y - X*beta;
    [psi, flag] = sample_psi(psi,e,sig2);
```

```
    Hpsi = speye(T) + sparse(2:T,1:(T-1),psi*ones(1,T-1),T,T);
    count_psi = count_psi + flag;

        % sample sig2
    tmp = Hpsi\e;
    sig2 = 1/gamrnd(nu0+T/2,1/(S0 + tmp'*tmp/2));

    if (mod(isim, 2000) == 0)
        disp([num2str(isim) ' loops... ']);
    end

        % store the parameters
    if isim > burnin
        isave = isim - burnin;
        store_theta(isave,:) = [beta' sig2 psi];
    end
end
theta_hat = mean(store_theta)
theta_CI = quantile(store_theta,[.025 .975])
```

Using a sample of 50000 draws, the posterior means of $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)'$ and ψ are estimated to be $(0.39, 0.93, 0.01)'$ and -0.67. Figure 3.2 depicts the histogram of the 50000 posterior draws of ψ. The mode of the density is about -0.7 and most of the mass is below 0. In fact, the estimate of $\mathbb{P}(\psi < 0 \,|\, \mathbf{y})$ is about 0.999, showing that allowing the error process to have an MA(1) structure is empirically relevant. In Section 4.1.1 we will formally compare the MA(1) model with the standard linear regression with independent errors.

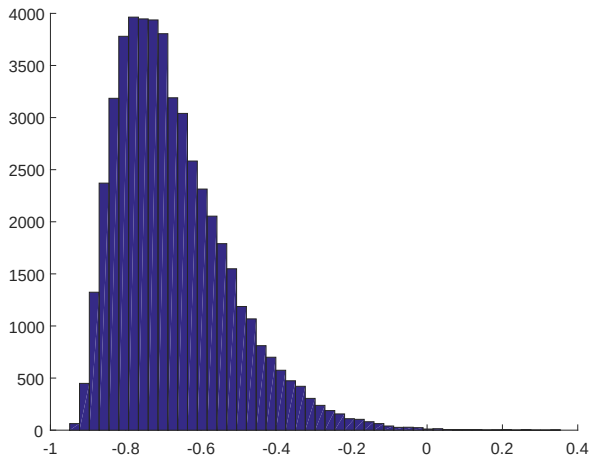

Figure 3.2: Histogram of the posterior draws of ψ.

Chapter 4

Bayesian Model Comparison

In most applications there are many competing models one can entertain to fit the same data. In addition, hypothesis testing can also be framed as a model comparison exercise. Hence, model comparison arises naturally in applied work. In this chapter we discuss how Bayesians test hypothesis and compare nonnested models. We will start with the gold standard of using the marginal likelihood as a model comparison criterion. We then consider a few other alternatives that are often used in applied work.

4.1 Marginal Likelihood

To set the stage, suppose we wish to compare a set of models $\{M_1, \ldots, M_K\}$. Each model M_k is formally defined by a likelihood function $p(\mathbf{y} \mid \boldsymbol{\theta}_k, M_k)$ and a prior distribution $p(\boldsymbol{\theta}_k \mid M_k)$, where $\boldsymbol{\theta}_k$ is a model-specific parameter vector. Note that we explicitly include the model label M_k to distinguish different likelihoods for different models.

The **marginal likelihood** or **marginal data density** under model M_k is defined as

$$p(\mathbf{y} \mid M_k) = \int p(\mathbf{y} \mid \boldsymbol{\theta}_k, M_k) p(\boldsymbol{\theta}_k \mid M_k) \mathrm{d}\boldsymbol{\theta}_k. \tag{4.1}$$

In other words, the marginal likelihood is the normalizing constant of the posterior distribution.

The marginal likelihood can be interpreted as a density forecast of the data under model M_k evaluated at the actual observed data $\mathbf{y}$. Hence, if the observed data are likely under the model, the corresponding marginal likelihood would be "large" and vice versa.

To see that, write the data as $\mathbf{y} = (y_1, \ldots, y_T)'$ and let $\mathbf{y}_{1:t} = (y_1, \ldots, y_t)'$ denote all the data up to time t. Then, we can factor the marginal likelihood as follows:

$$p(\mathbf{y} \,|\, M_k) = p(y_1 \,|\, M_k) \prod_{t=1}^{T-1} p(y_{t+1} \,|\, \mathbf{y}_{1:t}, M_k), \tag{4.2}$$

where $p(y_{t+1} \,|\, \mathbf{y}_{1:t}, M_k)$ is the **predictive likelihood**, which is simply a one-step-ahead density forecast for y_{t+1} evaluated at the actual observation.

Given two models M_i and M_j, if the marginal likelihood of M_i is larger than that of M_j, then the observed data are more likely under model M_i compared to model M_j. This is therefore viewed as evidence in favor of model M_i. The weight of evidence can be quantified by the **posterior odds ratio** between the two models, which can be written as:

$$\frac{\mathbb{P}(M_i \,|\, \mathbf{y})}{\mathbb{P}(M_j \,|\, \mathbf{y})} = \underbrace{\frac{\mathbb{P}(M_i)}{\mathbb{P}(M_j)}}_{\text{prior odds ratio}} \times \underbrace{\frac{p(\mathbf{y} \,|\, M_i)}{p(\mathbf{y} \,|\, M_j)}}_{\text{Bayes factor}},$$

where $\mathbb{P}(M_i)/\mathbb{P}(M_j)$ is the prior odds ratio. The ratio of the marginal likelihoods $p(\mathbf{y} \,|\, M_i)/p(\mathbf{y} \,|\, M_j)$ is called the **Bayes factor** in favor of model M_i against M_j.

If both models are equally probable *a priori*—i.e., the prior odds ratio is one—the posterior odds ratio between the two models is then equal to the Bayes factor. In this case, if, say, the Bayes factor in favor of model M_i is 100, then model M_i is 100 times more likely than model M_j given the data.

One can show that if data are generated from model M_1, the Bayes factor would on average pick the correct model over some distinct model, say, M_2. To see that, we compute the expected log Bayes factor in favor of model M_1 with respect to the distribution $p(\mathbf{y} \,|\, M_1)$:

$$\mathbb{E}_{\mathbf{y}} \left[\log \frac{p(\mathbf{y} \,|\, M_1)}{p(\mathbf{y} \,|\, M_2)} \right] = \int p(\mathbf{y} \,|\, M_1) \, \log \frac{p(\mathbf{y} \,|\, M_1)}{p(\mathbf{y} \,|\, M_2)} \mathrm{d}\mathbf{y}.$$

This expression is the Kullback-Leibler divergence from $p(\mathbf{y} \,|\, M_1)$ to $p(\mathbf{y} \,|\, M_2)$, and it is strictly positive unless $p(\mathbf{y} \,|\, M_1) = p(\mathbf{y} \,|\, M_2)$, in which case it is zero. We will discuss the Kullback-Leibler divergence in more detail in Section 4.1.4.

Since the expected log Bayes factor in favor of model M_1 is positive, the marginal likelihood of M_1 is larger than that of M_2 on average. In addition, the Bayes factor is a consistent model selection criterion—i.e., it will asymptotically select the candidate model having the correct structure with probability one.

For Bayesians it is natural to use the Bayes factor as a model comparison criterion. However, except for simple models, the computation of the marginal likelihood is difficult—the integral in (4.1) is often high-dimensional and cannot be obtained analytically. But when comparing *nested* models—i.e., when one model is a restricted

version of the other model—the Bayes factor has a simple expression that can often be easily estimated using posterior output.

4.1.1 Savage–Dicky Density Ratio

For nested models, the corresponding Bayes factor can be written as a ratio of two densities. To set the stage, let M_U denote the *unrestricted* model with model parameters partitioned into two subsets: $\boldsymbol{\theta} = (\boldsymbol{\psi}, \boldsymbol{\omega})$. Next, let M_R denote the *restricted* version of M_U with free parameters $\boldsymbol{\psi}$, and $\boldsymbol{\omega} = \boldsymbol{\omega}_0$ for some constant vector $\boldsymbol{\omega}_0$.

Let $p(\boldsymbol{\psi}, \boldsymbol{\omega} \,|\, M_U)$ represent the prior distribution under the unrestricted model. For simplicity, we assume that $\boldsymbol{\psi}$ and $\boldsymbol{\omega}$ are independent *a priori*, i.e., $p(\boldsymbol{\psi}, \boldsymbol{\omega} \,|\, M_U) = p(\boldsymbol{\psi} \,|\, M_U)p(\boldsymbol{\omega} \,|\, M_U)$.[1] Hence, the induced prior of $\boldsymbol{\psi}$ under the restricted model M_R is the marginal distribution $p(\boldsymbol{\psi} \,|\, M_R) = \int p(\boldsymbol{\psi}, \boldsymbol{\omega} \,|\, M_U)\mathrm{d}\boldsymbol{\omega} = p(\boldsymbol{\psi} \,|\, M_U)$.

Then, the Bayes factor is equivalent to the ratio of prior and posterior densities under M_U evaluated at $\boldsymbol{\omega} = \boldsymbol{\omega}_0$. More precisely, the Bayes factor in favor of the unrestricted model M_U can be written as

$$\mathrm{BF}_{UR} = \frac{p(\boldsymbol{\omega} = \boldsymbol{\omega}_0 \,|\, M_U)}{p(\boldsymbol{\omega} = \boldsymbol{\omega}_0 \,|\, \mathbf{y}, M_U)}.$$

This is called the **Savage–Dickey density ratio**. The proof of this equality can be found in, for example, Verdinelli and Wasserman (1995).

Intuitively, if $\boldsymbol{\omega} = \boldsymbol{\omega}_0$ is more likely under the prior relative to the posterior—in that case the numerator $p(\boldsymbol{\omega} = \boldsymbol{\omega}_0 \,|\, M_U)$ is larger than the denominator $p(\boldsymbol{\omega} = \boldsymbol{\omega}_0 \,|\, \mathbf{y}, M_U)$—then it is viewed as evidence in favor of the unrestricted model. This approach has been used to compute the Bayes factor in many empirical applications, such as Koop and Potter (1999), Deborah and Strachan (2009) and Koop, Leon-Gonzalez and Strachan (2010).

Using this ratio of densities, we can avoid the typically difficult task of computing the marginal likelihood. The prior distribution is usually chosen as some convenient distribution, and we can evaluate the numerator $p(\boldsymbol{\omega} = \boldsymbol{\omega}_0 \,|\, M_U)$ analytically. The denominator can often be estimated using the Monte Carlo average

$$\frac{1}{R}\sum_{r=1}^{R} p(\boldsymbol{\omega} = \boldsymbol{\omega}_0 \,|\, \mathbf{y}, \boldsymbol{\psi}^{(r)}, M_U), \tag{4.3}$$

[1] This is a sufficient but not a necessary condition for the Savage–Dickey density ratio to hold. We adopt this stronger condition to avoid interpreting a conditional distribution given a measure zero set.

where $\boldsymbol{\psi}^{(1)}, \ldots, \boldsymbol{\psi}^{(R)}$ are posterior draws from the unrestricted model M_U.

As an example, we revisit the linear regression model with MA(1) errors in Section 3.3.1. Recall that if the MA(1) coefficient ψ is zero, then the MA(1) model reduces to the standard regression with independent errors. Hence, the Bayes factor in favor of the MA(1) model can be written as the density ratio $p(\psi = 0)/p(\psi = 0 \,|\, \mathbf{y})$.

Under the uniform prior $\mathcal{U}(-1, 1)$, $p(\psi = 0) = 0.5$. The denominator $p(\psi = 0 \,|\, \mathbf{y})$ can be estimated using the Monte Carlo average in (4.3). One complication is that the conditional density $p(\psi \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2)$ is nonstandard. However, it is bounded on the interval $(-1, 1)$, and we can evaluate the density on a grid. In particular, the following MATLAB function constructs the marginal density $p(\psi \,|\, \mathbf{y})$ given the posterior draws of $\boldsymbol{\beta}$ and σ^2.

```
n_grid = 400; % number of grid points
psi_grid = linspace(-.99,.99,n_grid)';  % build a grid for psi
psi_grid = sort([psi_grid;0]);  % insert 0
n_grid = size(psi_grid,1);
idx_0 = find(psi_grid==0); % index for 0
lp_psi = zeros(n_grid,1);  % log posterior density
store_p_psi = zeros(n_grid,1);
for isim = 1:nsim
    beta = store_theta(isim,1:3)';  % load stored posterior draws
    sig2 = store_theta(isim,4);
    for igrid = 1:n_grid
        psi = psi_grid(igrid);
        lp_psi(igrid) = llike_MA1(psi,y-X*beta,sig2);
    end
    p_psi = exp(lp_psi-max(lp_psi));  % exponentiate the log-density
    p_psi = p_psi/(sum(p_psi)*(psi_grid(2)-psi_grid(1))); % normalize
    store_p_psi = store_p_psi + p_psi;
end
p_psi_hat = store_p_psi/nsim;
BF_UR = 0.5/p_psi_hat(idx_0);
```

In the code, we first construct a uniform grid on $(-0.99, 0.99)$. Note that we insert the value 0 to ensure that it is one of the grid points. Next, for each set of posterior draws of $\boldsymbol{\beta}$ and σ^2, the inner loop evaluates $\log p(\psi \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2)$—up to a normalizing constant—on each of the grid point (the function `llike_MA1.m` is given in Section 3.3.1). Then, we normalize the density so that the area under the curve is one. Finally, we average these conditional densities of ψ over the posterior draws to obtain the marginal density $p(\psi \,|\, \mathbf{y})$. The posterior density of ψ is reported in Figure 4.1.

Compared to the prior, the posterior has little mass around zero. In fact, the Bayes factor in favor of the MA(1) model is about 40. That is, assuming that both models are equally probable *a priori*, the MA(1) model becomes 40 times more likely given the data compared to the standard linear regression with independent errors.

Before we end this section, we provide a word of warning about the Monte Carlo estimator in (4.3). Even though the estimator is simulation consistent, its numerical accuracy using a finite simulation size is not guaranteed. It is therefore important to be aware of the conditions under which it is likely to yield unreliable results, and to understand what conclusion we can draw in those cases.

In particular, the Monte Carlo estimator depends on the full conditional density of $\boldsymbol{\omega}$ evaluated at $\boldsymbol{\omega}_0$. This estimator is likely to be unstable if the density has virtually no mass at $\boldsymbol{\omega}_0$. In that case, the Savage–Dickey density ratio would tend to be large, but the exact value is unlikely to be accurately estimated. It would help to corroborate the conclusion by visually inspecting the prior and posterior densities of $\boldsymbol{\omega}$, as done in Figure 4.1. If the posterior density has little mass around $\boldsymbol{\omega}_0$ relative to the prior, this can be viewed as evidence against the restricted model.

This discussion applies to our example. Even though the estimated Bayes factor of 40 is likely to be inaccurate (e.g., another run might give an estimate of 45), we are reasonably confident that there is strong evidence against the restricted model.

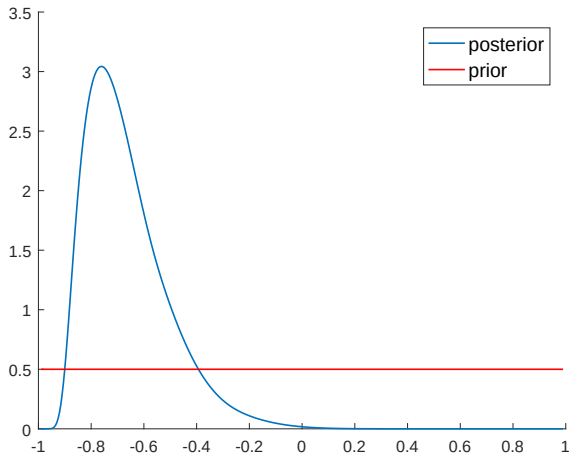

Figure 4.1: Prior and posterior densities of the MA(1) coefficient ψ.

Below we discuss a few common computational methods to estimate the marginal likelihood $p(\mathbf{y} \mid M_k)$ for a generic model M_k. Some methods are more convenient or more accurate for some models but not for others. In addition, it is often a good idea to compute the marginal likelihood using different methods to double check the

results.

4.1.2 Modified Harmonic Mean

One popular method for estimating the marginal likelihood is the **modified harmonic mean** of Gelfand and Dey (1994). Recent applications in macroeconomics include Liu, Waggoner and Zha (2011) and Bianchi (2013). For notational convenience, we will drop the model index M_k, and write the marginal likelihood, likelihood and prior simply as $p(\mathbf{y})$, $p(\mathbf{y} \,|\, \boldsymbol{\theta})$ and $p(\boldsymbol{\theta})$, respectively.

Let f denote a probability density function whose support is contained in the support of the posterior distribution. The modified harmonic mean is based on the following identity:

$$\mathbb{E}\left[\frac{f(\boldsymbol{\theta})}{p(\boldsymbol{\theta})p(\mathbf{y} \,|\, \boldsymbol{\theta})} \,\middle|\, \mathbf{y}\right] = \int \frac{f(\boldsymbol{\theta})}{p(\boldsymbol{\theta})p(\mathbf{y} \,|\, \boldsymbol{\theta})} \frac{p(\boldsymbol{\theta})p(\mathbf{y} \,|\, \boldsymbol{\theta})}{p(\mathbf{y})} \,\mathrm{d}\boldsymbol{\theta} = p(\mathbf{y})^{-1}, \tag{4.4}$$

where the expectation is taken with respect to the posterior distribution $p(\boldsymbol{\theta} \,|\, \mathbf{y}) = p(\boldsymbol{\theta})p(\mathbf{y} \,|\, \boldsymbol{\theta})/p(\mathbf{y})$. Therefore, one can estimate $p(\mathbf{y})$ using the following estimator:

$$\widehat{p(\mathbf{y})}_{\mathrm{GD}} = \left[\frac{1}{R}\sum_{r=1}^{R} \frac{f(\boldsymbol{\theta}^{(r)})}{p(\boldsymbol{\theta}^{(r)})p(\mathbf{y} \,|\, \boldsymbol{\theta}^{(r)})}\right]^{-1}, \tag{4.5}$$

where $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(R)}$ are posterior draws. Note that this estimator is simulation consistent in the sense that it converges to $p(\mathbf{y})$ in probability as R tends to infinity, but it is not unbiased—i.e., $\mathbb{E}\,\widehat{p(\mathbf{y})}_{\mathrm{GD}} \neq p(\mathbf{y})$ in general.

Geweke (1999) shows that if the function f has tails thinner than those of the posterior distribution, then the modified harmonic mean in (4.5) has a finite variance. Since the posterior distribution is asymptotically normal under certain regularity conditions, Geweke (1999) recommends using a normal approximation of the posterior distribution with tail truncations. More precisely, let $\widehat{\boldsymbol{\theta}}$ and $\mathbf{Q}_{\boldsymbol{\theta}}$ denote the posterior mean and posterior covariance matrix of $\boldsymbol{\theta}$, respectively. Consider choosing f to be the $\mathcal{N}(\widehat{\boldsymbol{\theta}}, \mathbf{Q}_{\boldsymbol{\theta}})$ density truncated within the region

$$\{\boldsymbol{\theta} \in \mathbb{R}^m : (\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})' \mathbf{Q}_{\boldsymbol{\theta}}^{-1} (\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}) < \chi^2_{\alpha,m}\},$$

where m is the dimension of $\boldsymbol{\theta}$ and $\chi^2_{\alpha,m}$ is the $(1-\alpha)$ quantile of the χ^2_m distribution.

The main advantage of the modified harmonic mean is that it is easy to implement—the programming effort is minimal and only posterior draws are required. While this estimator is typically accurate for low dimensional problems (e.g., less than a dozen parameters), it might be numerically unstable in high-dimensional problems.

As an empirical example, we revisit the AR(2) model with t errors in Section 3.2.4 for modeling PCE inflation. Recall that the posterior mean of the degree of freedom parameter ν of the t distribution is about 5, and most of the mass of the distribution is below 10. This indicates that the tails of the error distribution are much heavier than those of the Gaussian. Here we formally compare the t model with the Gaussian model.

The parameters of the AR(2) model with t errors are $\boldsymbol{\theta} = (\beta_1, \beta_2, \beta_3, \sigma^2, \nu)'$. We first obtain 20000 posterior draws of $\boldsymbol{\theta}$ using the posterior sampler in Section 3.2.4, which are stored in the variable `store_theta`. Then, we compute the marginal likelihood estimator using these posterior draws as specified in (4.5).

To construct the estimator, we need to evaluate the prior distributions: $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \mathbf{V}_{\boldsymbol{\beta}}), \sigma^2 \sim \mathcal{IG}(\nu_0, S_0)$ and $\nu \sim \mathcal{U}(2, \bar{\nu})$. To that end, we define the functions `lmvnpdf.m` and `ligampdf.m` that return the log densities of the normal and inverse-gamma distributions, respectively.

```
function lden = lmvnpdf(x,mu,Sig)
    n = length(mu);
    CSig = chol(Sig,'lower');
    e = CSig\(x-mu);
    lden = - n/2*log(2*pi) - sum(log(diag(CSig))) - .5*(e'*e);
end
```

```
function lden = ligampdf(x,a,b)
    lden = a.*log(b) - gammaln(a) - (a+1).*log(x) - b./x;
end
```

We then construct the function `prior` that evaluates the joint prior distribution of $\boldsymbol{\theta}$.

```
prior = @(b,s,n) lmvnpdf(b,beta0,iVbeta\speye(3)) ...
    + ligampdf(s,nu0,S0) + log(1/(nu_ub-2));
```

Finally, the following MATLAB function `linreg_t_GD.m` implements the modified harmonic mean for the t model.

```
function ml = linreg_t_GD(store_theta,y,X,prior)
    [nsim, m] = size(store_theta);
    T = size(X,1);
```

```
    theta_hat = mean(store_theta)';
    Qtheta = cov(store_theta);
    alp = .05;     % significance level for truncation
    chi2q = chi2inv(1-alp,m);
        % normalizing constant for f
    const_f = log(1/(1-alp)) - m/2*log(2*pi) -.5*log(det(Qtheta));
    store_w = - inf(nsim,1);
    for isim = 1:nsim
        theta = store_theta(isim,:)';
        beta = theta(1:m-2);
        sig2 = theta(m-1);
        nu = theta(m);
        if (theta-theta_hat)'*(Qtheta\(theta-theta_hat)) < chi2q
            llike = T*(gammaln((nu+1)/2) - gammaln(nu/2)...
                - .5*log(nu*pi*sig2)) - (nu+1)/2 ...
                * sum(log(1 + (y-X*beta).^2/(sig2*nu)));
            f = @(th) const_f - .5*(th-theta_hat)'*(Qtheta\(th-theta_hat));
            store_w(isim) = f(theta) - (llike + prior(beta,sig2,nu));
        end
    end
    maxllike = max(store_w);
    ml = log(mean(exp(store_w-maxllike))) + maxllike;
    ml = -ml;
end
```

In the implementation we set $\alpha = 0.05$. For numerical stability, we mostly compute the values in log scale. Using 20000 posterior draws, the marginal likelihood for the t model is estimated to be -554. We also compute the marginal likelihood for the AR(2) model with Gaussian errors. The estimate is -565.5. Therefore, the Bayes factor in favor of the t model is $\exp(11.5) \approx 98716$, which shows overwhelming evidence for the t model. This conclusion is inline with the estimation results with small values of ν.

4.1.3 Chib's Method

An alternative way to compute the marginal likelihood, which is due to Chib (1995), is based on the observation that the marginal likelihood is the normalizing constant of the posterior distribution. Or equivalently,

$$p(\mathbf{y}) = \frac{p(\mathbf{y} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \,|\, \mathbf{y})}.$$

Hence, a natural estimator for $p(\mathbf{y})$—when written in log scale—is the quantity

$$\log \widehat{p(\mathbf{y})}_{\text{Chib}} = \log p(\mathbf{y} \,|\, \boldsymbol{\theta}^*) + \log p(\boldsymbol{\theta}^*) - \log \widehat{p(\boldsymbol{\theta}^* \,|\, \mathbf{y})}, \tag{4.6}$$

where $\boldsymbol{\theta}^*$ is any point in the support of the posterior distribution. For reasons we will explain below, it is useful to choose $\boldsymbol{\theta}^*$ to be some "high density" point such as the posterior mean or mode.

Often we can evaluate analytically both the likelihood function and the prior distribution, and the only unknown quantity is the posterior ordinate $p(\boldsymbol{\theta}^* \,|\, \mathbf{y})$. As before, we will estimate this quantity by Monte Carlo methods. In particular, if all the full conditional distributions are known, then $p(\boldsymbol{\theta}^* \,|\, \mathbf{y})$ can be estimated by sampling draws from a series of suitably designed Gibbs samplers, the so-called *reduced runs*.

To give a concrete example, suppose we can partition the parameters $\boldsymbol{\beta}$ into three blocks $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$ and $\boldsymbol{\beta}_3$ with full conditional distributions $p(\boldsymbol{\theta}_1 \,|\, \mathbf{y}, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$, $p(\boldsymbol{\theta}_2 \,|\, \mathbf{y}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_3)$ and $p(\boldsymbol{\theta}_3 \,|\, \mathbf{y}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$. We emphasize that these conditional distributions are fully known, and they can be evaluated exactly.

We first factor $p(\boldsymbol{\theta}^* \,|\, \mathbf{y})$ as:

$$\begin{aligned}\log p(\boldsymbol{\theta}^* \,|\, \mathbf{y}) &= \log p(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \boldsymbol{\theta}_3^* \,|\, \mathbf{y}) \\ &= \log p(\boldsymbol{\theta}_1^* \,|\, \mathbf{y}) + \log p(\boldsymbol{\theta}_2^* \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*) + \log p(\boldsymbol{\theta}_3^* \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*).\end{aligned}$$

And we obtain each of the three terms separately. The last term $p(\boldsymbol{\theta}_3^* \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*)$ is known, as we assume all the conditional distributions are fully known. The remaining two terms can be estimated using Monte Carlo methods. In particular, the first term $p(\boldsymbol{\theta}_1^* \,|\, \mathbf{y})$ can be estimated using draws from the main Gibbs run, whereas $p(\boldsymbol{\theta}_2^* \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*)$ can be estimated using draws from a reduced run. In what follows, we provide the computational details.

To estimate $p(\boldsymbol{\theta}_1^* \,|\, \mathbf{y})$, first note that

$$\begin{aligned}p(\boldsymbol{\theta}_1^* \,|\, \mathbf{y}) &= \int\int p(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3 \,|\, \mathbf{y}) \mathrm{d}\boldsymbol{\theta}_2 \mathrm{d}\boldsymbol{\theta}_3 \\ &= \int\int p(\boldsymbol{\theta}_1^* \,|\, \mathbf{y}, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3) p(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3 \,|\, \mathbf{y}) \mathrm{d}\boldsymbol{\theta}_2 \mathrm{d}\boldsymbol{\theta}_3.\end{aligned}$$

Hence, the quantity $p(\boldsymbol{\theta}_1^* \,|\, \mathbf{y})$ can be estimated by the Monte Carlo average

$$\widehat{p(\boldsymbol{\theta}_1^* \,|\, \mathbf{y})} = \frac{1}{R} \sum_{r=1}^{R} p\left(\boldsymbol{\theta}_1^* \,|\, \mathbf{y}, \boldsymbol{\theta}_2^{(r)}, \boldsymbol{\theta}_3^{(r)}\right),$$

where $(\boldsymbol{\theta}_2^{(1)}, \boldsymbol{\theta}_3^{(1)}), \ldots, (\boldsymbol{\theta}_2^{(R)}, \boldsymbol{\theta}_3^{(R)})$ are sampled from the marginal posterior distribution $p(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3 \,|\, \mathbf{y})$. That is, these draws are obtained from the main Gibbs run.

As mentioned above, even though in principle $\boldsymbol{\theta}_1^*$ can be set to be any value so long as it is in the support of the posterior distribution, in practice $\boldsymbol{\theta}_1^*$ is often chosen to be some "high density" point. The reason is clear from the above Monte Carlo estimator: choosing $\boldsymbol{\theta}_1^*$ near the mean or mode would ensure that the value of $p(\boldsymbol{\theta}_1^* \,|\, \mathbf{y}, \boldsymbol{\theta}_2^{(r)}, \boldsymbol{\theta}_3^{(r)})$ is sufficiently large for most pairs of $(\boldsymbol{\theta}_2^{(r)}, \boldsymbol{\theta}_3^{(r)})$. In turns the estimator would be more numerically stable.

Similarly, note that

$$\begin{aligned} p(\boldsymbol{\theta}_2^* \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*) &= \int p(\boldsymbol{\theta}_2^*, \boldsymbol{\theta}_3 \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*) \mathrm{d}\boldsymbol{\theta}_3 \\ &= \int p(\boldsymbol{\theta}_2^* \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*, \boldsymbol{\theta}_3) p(\boldsymbol{\theta}_3 \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*) \mathrm{d}\boldsymbol{\theta}_3. \end{aligned}$$

Hence, $p(\boldsymbol{\theta}_2^* \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*)$ can be estimated by

$$\widehat{p(\boldsymbol{\theta}_2^* \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*)} = \frac{1}{R} \sum_{r=1}^{R} p\left(\boldsymbol{\theta}_2^* \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*, \boldsymbol{\theta}_3^{(r)}\right),$$

where $\boldsymbol{\theta}_3^{(1)}, \ldots, \boldsymbol{\theta}_3^{(R)}$ are sampled from $p(\boldsymbol{\theta}_3 \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*)$. These draws can be obtained using a *reduced run*.

More specifically, we initialize $\boldsymbol{\theta}_2^{(0)} = \mathbf{a}_0$ and $\boldsymbol{\theta}_3^{(0)} = \mathbf{b}_0$. Then, we repeat the following steps from $r = 1$ to R:

1. Draw $\boldsymbol{\theta}_2^{(r)} \sim p\left(\boldsymbol{\theta}_2 \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*, \boldsymbol{\theta}_3^{(r-1)}\right)$.
2. Draw $\boldsymbol{\theta}_3^{(r)} \sim p\left(\boldsymbol{\theta}_3 \,|\, \mathbf{y}, \boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^{(r)}\right)$.

If there are more blocks in the Gibbs samplers, more reduced runs are needed. These additional samples obviously require more programming effort and computational time, but they also reduce simulation error and contribute to a higher accuracy of the resulting estimator.

The main limitation of the Chib's method is that it requires complete knowledge of all conditional distributions in the Gibbs sampler. To overcome this limitation, Chib and Jeliazkov (2001) extend the basic approach to tackle cases in which some conditional distributions are nonstandard, and Metropolis-Hastings steps are required. However, implementation of these extensions are considerably more involved.

To illustrate the Chib's method, we compute the marginal likelihood of the AR(2) model with MA(1) errors in Section 3.3.1 for modeling PCE inflation. The parameters of the model are $\boldsymbol{\theta} = (\beta_1, \beta_2, \beta_3, \sigma^2, \psi)'$, and these parameters are drawn in three

blocks: $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)'$, σ^2 and ψ. The full conditional distributions of the first two blocks are known—they are Gaussian and inverse-gamma—but the conditional distribution of ψ is nonstandard. However, as described in Section 4.1.1, we can approximate the distribution on a grid.

We first obtain 50000 posterior draws of $\boldsymbol{\theta}$ using the posterior sampler in Section 3.3.1, which are stored in the variable `store_theta`. Then, we implement the Chib's method in the function `linreg_ma1_chib.m`. For ease of exposition, we break the whole function into several parts.

In the first part, we set $\boldsymbol{\theta}^*$ to be the posterior mean—in the code it is called `theta_hat`. Then, we evaluate the likelihood function and the prior distribution at $\boldsymbol{\theta}^*$. The functions `lmvnpdf.m` and `ligampdf.m` return the log densities of the normal and inverse-gamma distributions, respectively, and are given in Section 4.1.2.

```
function ml = linreg_ma1_chib(store_theta,y,X,beta0,iVbeta,nu0,S0)
nsim_re = 5000; % size of reduced runs
[nsim,m] = size(store_theta);
T = size(y,1);
theta_hat = mean(store_theta)';
beta_s = theta_hat(1:m-2);
sig2_s = theta_hat(m-1);
psi_s = theta_hat(m);

    % evaluate the log likelihood at beta_s,sig2_s,psi_s
llike = llike_MA1(psi_s,y-X*beta_s,sig2_s);

    % define the prior
prior = @(b,s,p) lmvnpdf(b,beta0,iVbeta\speye(3)) ...
    + ligampdf(s,nu0,S0) + log(1/2);
```

In the second part, we estimate $p(\boldsymbol{\beta}^* \,|\, \mathbf{y})$ using draws from the main Gibbs run. Specifically, recall that $(\boldsymbol{\beta} \,|\, \mathbf{y}, \sigma^2, \psi)$ is Gaussian, and we can evaluate the log density at $\boldsymbol{\beta}^*$ using the function `lmvnpdf`.

```
    % evaluate the posterior of beta at beta_s
store_lpbeta = zeros(nsim,1);
for isim = 1:nsim
    sig2 = store_theta(isim,m-1);
    psi = store_theta(isim,m);
    Hpsi = speye(T) + sparse(2:T,1:(T-1),psi*ones(1,T-1),T,T);
    X_tilde = Hpsi\X; y_tilde = Hpsi\y;
```

```
    Dbeta = (iVbeta + X_tilde'*X_tilde/sig2)\speye(3);
    beta_hat = Dbeta*(iVbeta*beta0 + X_tilde'*y_tilde/sig2);
    store_lpbeta(isim) = lmvnpdf(beta_s,beta_hat,Dbeta);
end
lpbeta = log(mean(exp(store_lpbeta)));
```

In the third part, we estimate $p(\sigma^{2*} \,|\, \mathbf{y}, \boldsymbol{\beta}^*)$ using draws from a reduced run. Since $p(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta}, \psi)$ is an inverse-gamma distribution, we can evaluate this density using the function `ligampdf`.

```
    % evaluate the posterior of sig2 at sig2_s using a reduced run
beta = beta_s; % fix beta at the posterior mean
e = y - X*beta;
sig2 = sig2_s;
store_lpsig2 = zeros(nsim_re,1);
for isim = 1:nsim_re
        % sample psi
    psi = sample_psi(psi,e,sig2);
    Hpsi = speye(T) + sparse(2:T,1:(T-1),psi*ones(1,T-1),T,T);
        % sample sig2
    tmp = Hpsi\e;
    sig2 = 1/gamrnd(nu0+T/2,1/(S0 + tmp'*tmp/2));

    store_lpsig2(isim,:) = ligampdf(sig2_s,nu0+T/2,S0 + tmp'*tmp/2);
end
lpsig2 = log(mean(exp(store_lpsig2)));
```

In the last part, we estimate $p(\psi^* \,|\, \mathbf{y}, \boldsymbol{\beta}^*, \sigma^{2*})$. This is done by evaluating the density on a fine grid as described in Section 4.1.1. Finally, given all the ingredients, we estimate the log marginal likelihood using the Chib's method in (4.6).

```
    % evaluate the posterior of psi at psi_s
n_grid = 799; % number of grid points
psi_grid = linspace(-.99,.99,n_grid)';  % build a grid for psi
psi_grid = sort([psi_grid;psi_s]);  % insert psi_s
n_grid = size(psi_grid,1);
idx_psi = (psi_grid==psi_s); % index for psi_s
lp_psi = zeros(n_grid,1);  % log posterior density
for igrid = 1:n_grid
    psi = psi_grid(igrid);
    lp_psi(igrid) = llike_MA1(psi,y-X*beta_s,sig2_s);
```

```
end
p_psi = exp(lp_psi-max(lp_psi));  % exponentiate the log-density
p_psi = p_psi/(sum(p_psi)*(psi_grid(2)-psi_grid(1))); % normalize
lppsi = log(p_psi(idx_psi));

ml = llike + prior(beta_s,sig2_s,psi_s) - (lpbeta + lpsig2 + lppsi);
end
```

Using 50000 posterior draws and 5000 draws from the reduced run, the log marginal likelihood is estimated to be -561.6. In Section 4.1.2 we compute the log marginal likelihood of the AR(2) model with independent, Gaussian errors using the modified harmonic mean, and the estimate is -565.5. This implies a Bayes factor—in favor of the MA(1) model—of about 49. Compare this with the Bayes factor of 40 obtained using the Savage–Dickey density ratio in Section 4.1.1. As mentioned there, that estimate is likely to be inaccurate. But in either case, there is strong evidence to support modeling the errors as an MA(1) process compared to assuming them to be independent.

4.1.4 Cross-Entropy Method

Both the modified harmonic mean and the Chib's method compute the marginal likelihood using MCMC draws. By construction these draws are autocorrelated and they are typically costly to obtain. In this section we discuss an **importance sampling** estimator based on *independent* draws from convenient distributions.

The basic idea is to bias the sampling distribution in such a way that more "important values" are generated in the simulation. The sample is then weighted to correct for the use of a different distribution to give an unbiased estimator. In our context of estimating the marginal likelihood, consider the following importance sampling estimator:

$$\widehat{p(\mathbf{y})}_{\text{IS}} = \frac{1}{R}\sum_{r=1}^{R} \frac{p(\mathbf{y} \mid \boldsymbol{\theta}^{(r)})p(\boldsymbol{\theta}^{(r)})}{g(\boldsymbol{\theta}^{(r)})}, \tag{4.7}$$

where $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(R)}$ are independent draws obtained from the **importance density** $g(\cdot)$ that dominates $p(\mathbf{y} \mid \cdot)p(\cdot)$—i.e., $g(\mathbf{x}) = 0 \Rightarrow p(\mathbf{y} \mid \mathbf{x})p(\mathbf{x}) = 0$.

The estimator in (4.7) is unbiased and simulation consistent for any such g. But its performance depends critically on the choice of the importance density. Below we describe a variant of the classic **cross-entropy method** to construct g optimally. The original cross-entropy method was developed for rare-event simulation by Rubinstein (1997, 1999) using a multilevel procedure to obtain the optimal importance sampling density (see also Rubinstein and Kroese, 2004, for a book-length

treatment). Chan and Kroese (2012) demonstrate that this optimal importance sampling density can be obtained more accurately in one step using MCMC methods. We follow Chan and Eisenstat (2015) to use this new variant for marginal likelihood estimation.

The cross-entropy method is based on a few simple observations. First, for marginal likelihood estimation there exists an importance density that gives a zero-variance estimator. In particular, if we use the posterior distribution as the importance density, i.e., $g(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \,|\, \mathbf{y}) = p(\mathbf{y} \,|\, \boldsymbol{\theta})p(\boldsymbol{\theta})/p(\mathbf{y})$, then the associated importance sampling estimator (4.7) has zero variance:

$$\widehat{p(\mathbf{y})}_{\text{IS}} = \frac{1}{R}\sum_{r=1}^{R} \frac{p(\mathbf{y} \,|\, \boldsymbol{\theta}^{(r)})p(\boldsymbol{\theta}^{(r)})}{g(\boldsymbol{\theta}^{(r)})} = \frac{1}{R}\sum_{r=1}^{R} \frac{p(\boldsymbol{\theta}^{(r)})p(\mathbf{y} \,|\, \boldsymbol{\theta}^{(r)})}{p(\boldsymbol{\theta}^{(r)})p(\mathbf{y} \,|\, \boldsymbol{\theta}_r)/p(\mathbf{y})} = p(\mathbf{y}),$$

and we need to produce only $R = 1$ sample. We denote this zero-variance importance density as g^*.

Although in principle g^* gives the best possible estimator for $p(\mathbf{y})$, it cannot be used in practice as its normalizing constant—the marginal likelihood—is the very unknown quantity we wish to estimate. However, this suggests a practical approach to obtain an optimal importance density. Intuitively, if we choose an importance density g that is "close enough" to g^* so that both behave similarly, the resulting importance sampling estimator should have reasonable accuracy. Hence, our goal is to locate a convenient density that is in a well-defined sense "close" to g^*.

To that end, consider a parametric family $\mathcal{F} = \{f(\boldsymbol{\theta}; \mathbf{v})\}$ indexed by the parameter vector $\mathbf{v}$ within which we locate the optimal importance density. We find the density $f(\boldsymbol{\theta}; \mathbf{v}^*) \in \mathcal{F}$ such that it is the "closest" to g^*. One convenient measure of closeness between densities is the **Kullback-Leibler divergence** or the **cross-entropy distance**. Specifically, let h_1 and h_2 be two probability density functions. Then, the cross-entropy distance from h_1 to h_2 is defined as:

$$\mathcal{D}(h_1, h_2) = \int h_1(\mathbf{x}) \log \frac{h_1(\mathbf{x})}{h_2(\mathbf{x})} \mathrm{d}\mathbf{x}.$$

Given this measure, we next locate the density $f(\cdot; \mathbf{v}) \in \mathcal{F}$ such that $\mathcal{D}(g^*, f(\cdot; \mathbf{v}))$ is minimized:

$$\begin{aligned}\mathbf{v}^*_{\text{ce}} &= \operatorname*{argmin}_{\mathbf{v}} \mathcal{D}(g^*, f(\cdot; \mathbf{v})) \\ &= \operatorname*{argmin}_{\mathbf{v}} \left(\int g^*(\boldsymbol{\theta}) \log g^*(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} - p(\mathbf{y})^{-1} \int p(\mathbf{y} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta}) \log f(\boldsymbol{\theta}; \mathbf{v}) \mathrm{d}\boldsymbol{\theta} \right),\end{aligned}$$

where we used the fact that $g^*(\boldsymbol{\theta}) = p(\mathbf{y} \,|\, \boldsymbol{\theta})p(\boldsymbol{\theta})/p(\mathbf{y})$. Since the first term does not depend on $\mathbf{v}$, solving the CE minimization problem is equivalent to finding

$$\mathbf{v}^*_{\text{ce}} = \operatorname*{argmax}_{\mathbf{v}} \int p(\mathbf{y} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta}) \log f(\boldsymbol{\theta}; \mathbf{v}) \mathrm{d}\boldsymbol{\theta}.$$

In practice, this optimization problem is often difficult to solve analytically. Instead, we consider its stochastic counterpart:

$$\widehat{\mathbf{v}}_{\text{ce}}^* = \underset{\mathbf{v}}{\text{argmax}} \frac{1}{M} \sum_{m=1}^{M} \log f(\boldsymbol{\theta}_m; \mathbf{v}), \tag{4.8}$$

where $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(M)}$ are posterior draws. In other words, $\widehat{\mathbf{v}}_{\text{ce}}^*$ is exactly the maximum likelihood estimate for $\mathbf{v}$ if we treat $f(\boldsymbol{\theta}; \mathbf{v})$ as the likelihood function with parameter vector $\mathbf{v}$ and $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(M)}$ an observed sample. Since finding the maximum likelihood estimator is a standard problem, solving (4.8) is typically easy. In particular, analytical solutions to (4.8) can be found explicitly for the exponential family (e.g., Rubinstein and Kroese, 2004, p. 70).

In practice, the parametric family $\mathcal{F}$ is often chosen so that each member $f(\boldsymbol{\theta}; \mathbf{v})$ is a product of densities, e.g., $f(\boldsymbol{\theta}; \mathbf{v}) = f(\boldsymbol{\theta}_1; \mathbf{v}_1) \times \cdots \times f(\boldsymbol{\theta}_B; \mathbf{v}_B)$, where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_B)$ and $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_B)$. In that case, one can reduce the possibly high-dimensional maximization problem (4.8) into B low-dimensional problems, which can then be readily solved.

Once the optimal density is located, we set $g(\cdot) = f(\cdot; \widehat{\mathbf{v}}_{\text{ce}}^*)$ and use it to construct the importance sampling estimator in (4.7). The main advantage of this importance sampling approach is that it is fast and easy to implement. In addition, since it is based on independent draws, the numerical standard error of the estimator is readily available.

To illustrate the cross-entropy method, we revisit the AR(2) model with t errors in Section 3.2.4. The parameters are $\boldsymbol{\theta} = (\boldsymbol{\beta}', \sigma^2, \nu)'$. We use 20000 posterior draws of $\boldsymbol{\theta}$—stored in the variable `store_theta`—to construct the optimal importance density.

To that end, we consider the parametric family

$$\mathcal{F} = \{f_{\mathcal{N}}(\boldsymbol{\beta}; \mathbf{b}, \mathbf{B}) f_{\mathcal{IG}}(\sigma^2; \gamma_1, \gamma_2) f_{\mathcal{IG}}(\nu; \alpha_1, \alpha_2)\}.$$

The optimal CE parameters $\widehat{\mathbf{b}}^*$ and $\widehat{\mathbf{B}}^*$ are simply the sample mean and covariance matrix of the posterior draws of $\boldsymbol{\beta}$. The optimal CE parameters for the densities of σ^2 and ν can be obtained numerically by using the built-in function `gamfit` (see the code below).

Once these CE parameters are obtained we compute the importance sampling estimator in (4.7) using draws from normal and inverse-gamma distributions. Finally, we define the function `prior` to evaluate the joint prior distribution of $\boldsymbol{\theta}$.

```
prior = @(b,s,n) lmvnpdf(b,beta0,iVbeta\speye(3)) ...
    + ligampdf(s,nu0,S0) + log(1/(nu_ub-2)) - 10^(100)*(n<2||n>nu_ub);
```

The functions `lmvnpdf` and `ligampdf` return the log densities of the normal and inverse-gamma distributions, respectively, and are given in Section 4.1.2. Note that in the code we have added the penalty term -10^{100} when $\nu < 2$ or $\nu > \overline{\nu}$ to ensure $\nu \in (2, \overline{\nu})$. This is necessary as the importance density generates ν from an inverse-gamma distribution, which has support on the positive real line.

The main MATLAB script to implement the cross-entropy method is given below. In the code, R is the number of importance sampling draws. It is reset so that it is a multiple of 20—we divide the whole sample into 20 batches to obtain 20 different estimates in order to compute the numerical standard error.

```
function [ml,ml_std] = linreg_t_CE(store_theta,y,X,prior,R)
R = 20*ceil(R/20);   % make R divisible by 20
m = size(store_theta,2);
T = size(y,1);
    % obtain parameters for the IS density
b_hat = mean(store_theta(:,1:m-2))';
B_hat = cov(store_theta(:,1:m-2))';
tmp = gamfit(1./store_theta(:,m-1));
gam1_hat = tmp(1); gam2_hat = 1./tmp(2);
tmp = gamfit(1./store_theta(:,m));
alp1_hat = tmp(1); alp2_hat = 1./tmp(2);

    % obtain IS draws from the optimal density
theta_IS = zeros(R,m);
theta_IS(:,1:m-2) = repmat(b_hat',R,1) + (chol(B_hat,'lower')*randn(m-2,R))';
theta_IS(:,m-1) = 1./gamrnd(gam1_hat,1./gam2_hat,R,1);
theta_IS(:,m) = 1./gamrnd(alp1_hat,1./alp2_hat,R,1);

    % construct the IS density
g_IS = @(b,s,n) lmvnpdf(b,b_hat,B_hat) + ligampdf(s,gam1_hat,gam2_hat) ...
    + ligampdf(n,alp1_hat,alp2_hat);
store_w = zeros(R,1);

for isim  = 1:R
    theta = theta_IS(isim,:)';
    beta = theta(1:m-2);
    sig2 = theta(m-1);
    nu = theta(m);
    llike = T*(gammaln((nu+1)/2) - gammaln(nu/2) - .5*log(nu*pi*sig2)) ...
        - (nu+1)/2*sum(log(1 + (y-X*beta).^2/(sig2*nu)));
    store_w(isim) = llike + prior(beta,sig2,nu) - g_IS(beta,sig2,nu);
end
```

```
shortw = reshape(store_w,R/20,20); % divide the draws into 20 batches
maxw = max(shortw); % find the max for normalization

bigml = log(mean(exp(shortw-repmat(maxw,R/20,1)),1)) + maxw;
ml = mean(bigml);
ml_std = std(bigml)/sqrt(20);
end
```

Using a sample of $R = 10000$ importance sampling draws, the marginal likelihood for the t model is estimated to be -554.1 with a numerical standard error of 0.012. This is very close to the modified harmonic mean estimate of -554.

4.1.5 Computational Pitfalls

In this section we describe a few common computational problems when computing the marginal likelihood. These problems arise from the high-dimensional Monte Carlo integration and are not specific to marginal likelihood estimation. As a general principle, one should reduce the dimension of the Monte Carlo integration by analytically integrate out any random variables whenever possible.

To illustrate this point, consider again the linear regression with t errors in Section 3.2. Recall that the model can be represented in two different ways. In the first representation, the likelihood $p(\mathbf{y} \,|\, \boldsymbol{\beta}, \sigma^2, \nu)$—more precisely the **observed-data likelihood** or **integrated likelihood**—is a product of t densities and does not depend on the latent variables $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_T)'$.

The second representation is the latent variable representation, in which the **complete-data likelihood** $p(\mathbf{y}, \boldsymbol{\lambda} \,|\, \boldsymbol{\beta}, \sigma^2, \nu)$ is factored as:

$$p(\mathbf{y}, \boldsymbol{\lambda} \,|\, \boldsymbol{\beta}, \sigma^2, \nu) = p(\mathbf{y} \,|\, \boldsymbol{\lambda}, \boldsymbol{\beta}, \sigma^2, \nu) p(\boldsymbol{\lambda} \,|\, \nu),$$

where the **conditional likelihood** $p(\mathbf{y} \,|\, \boldsymbol{\lambda}, \boldsymbol{\beta}, \sigma^2, \nu)$ is a product of Gaussian densities given the latent variables $\boldsymbol{\lambda}$, and $p(\boldsymbol{\lambda} \,|\, \nu)$ is a product of inverse-gamma densities.

The two representations are related via the equality:

$$p(\mathbf{y} \,|\, \boldsymbol{\beta}, \sigma^2, \nu) = \int p(\mathbf{y} \,|\, \boldsymbol{\lambda}, \boldsymbol{\beta}, \sigma^2, \nu) p(\boldsymbol{\lambda} \,|\, \nu) \mathrm{d}\boldsymbol{\lambda}.$$

That is, we get back the integrated likelihood if we integrate out the latent variables $\boldsymbol{\lambda}$ with respect to $p(\boldsymbol{\lambda} \,|\, \nu)$.

Given these two representations, we can estimate the marginal likelihood using Chib's method in two different ways. The first is based on the integrated likeli-

hood:

$$\log p(\mathbf{y}) = \log p(\mathbf{y} \mid \boldsymbol{\beta}^*, \sigma^{2*}, \nu^*) + \log p(\boldsymbol{\beta}^*, \sigma^{2*}, \nu^*) - \log p(\boldsymbol{\beta}^*, \sigma^{*2}, \nu^* \mid \mathbf{y}). \quad (4.9)$$

The second is based on the complete-data likelihood:

$$\log p(\mathbf{y}) = \log p(\mathbf{y}, \boldsymbol{\lambda}^* \mid \boldsymbol{\beta}^*, \sigma^{2*}, \nu^*) + \log p(\boldsymbol{\beta}^*, \sigma^{2*}, \nu^*) - \log p(\boldsymbol{\lambda}^*, \boldsymbol{\beta}^*, \sigma^{2*}, \nu^* \mid \mathbf{y}). \quad (4.10)$$

Even though both are valid identities, we expect that the second approach would give a much larger variance due to the variability of the high-dimensional vector $\boldsymbol{\lambda}$. In fact, the Chib's method based on (4.10) is extremely unreliable and tends to give a substantial upward bias, as first pointed out by Frühwirth-Schnatter and Wagner (2008).

Using the PCE data, the log marginal likelihood estimate based on (4.9) is -553.6, which is close to the modified harmonic mean estimate of -554. However, if we instead compute the marginal likelihood using (4.10), the estimate becomes -386.8. These results are consistent with the findings in Frühwirth-Schnatter and Wagner (2008), who also recommend against using the complete-data likelihood in estimating the marginal likelihood.

As previously mentioned, this numerical problem arises from the high-dimensional Monte Carlo integration and is not specific to Chib's method. For example, Chan and Grant (2015) find that the marginal likelihood estimates calculated using the modified harmonic mean of the conditional likelihood can also have a substantial bias. In one of their examples, they show that the log marginal likelihood of an unobserved components model should be -591.94, but the modified harmonic mean estimate is -494.6. Even when the simulation size is increased to ten millions, the finite sample bias is still substantial—the estimate is -502.70 and is far from the correct value.

4.2 Information Criteria

The marginal likelihood is conceptually simple and has a natural interpretation. However, one potential drawback is that it is relatively sensitive to the prior distribution. This can be seen from the factorization in (4.2). Specifically, the predictive likelihood $p(y_1 \mid M_k)$ depends entirely on the prior distribution and not on the data. In addition, the component $p(y_{t+1} \mid \mathbf{y}_{1:t}, M_k)$ is likely to be heavily influenced by the prior distribution when t is small.

In this section we discuss alternative Bayesian model selection criteria that are relatively insensitive to the priors.

4.2.1 Bayesian Information Criterion

The **Bayesian information criterion** (BIC) is a popular model selection criterion based on the trade-off between model fit and model complexity. It is first introduced in Schwarz (1978), and is therefore sometimes called the **Schwarz information criterion**. The BIC for model M_k with likelihood function $p(\mathbf{y} \mid \boldsymbol{\theta}_k, M_k)$ is defined as

$$\text{BIC}_k = -2 \log p(\mathbf{y} \mid \widehat{\boldsymbol{\theta}}_k, M_k) + p_k \log T,$$

where $\widehat{\boldsymbol{\theta}}_k$ is the maximum likelihood estimator, p_k is the number of parameters and T the sample size. In practice, instead of computing the maximum likelihood estimator, one typically evaluates the likelihood over the posterior draws to find the maximum value. This often provides a good approximation.

Given a set of competing models, the preferred model is the one with the minimum BIC value. As is obvious from the definition, the BIC does not not only reward goodness-of-fit as measured by the maximized likelihood value, but it also includes a penalty term that is an increasing function of the number of parameters. Consequently, the BIC discourages over-fitting.

The difference of the BICs between two models is asymptotically equivalent to the log Bayes factor. More precisely, for models M_i and M_j, it can be shown that (see, e.g., Kass and Raftery, 1995)

$$\frac{(\text{BIC}_i - \text{BIC}_j) - \log \text{BF}_{ij}}{\log \text{BF}_{ij}} \to 0$$

as T tends to infinity. Consequently, the BIC is also a consistent model selection criterion like the Bayes factor.

As an illustration, we compare the AR(2) model with t errors in Section 3.2.4 with the standard version with Gaussian errors. Using the marginal likelihood as a model comparison criterion, we find overwhelming evidence in favor of the t model. Here we consider the BIC instead.

We can easily compute the BIC for the t model by modifying the posterior sampler in Section 3.2.4. In particular, we define the variable `store_llike = zeros(nsim,1);` outside the loop. Then, after the burn-in period, we store the log-likelihood value:

```
llike = T*(gammaln((nu+1)/2) -gammaln(nu/2) -.5*log(nu*pi*sig2))....
    -(nu+1)/2*sum(log(1 + (y-X*beta).^2/(sig2*nu)));
store_llike(isave,:) = llike;
```

Finally, once the posterior sampler is completed, we compute the BIC, noting that the number of parameters for the t model is 5.

```
max_llike = max(store_llike);
BIC = -2*max_llike + 5*log(T);
```

Using 20000 posterior draws, the BIC for the t model is estimated to be 1092. For the AR(2) with Gaussian errors, where the number of parameters is 4, the BIC is 1113. Hence, the BIC favors the t model as well.

4.2.2 Deviance Information Criterion

Another popular information criterion for model comparison is the deviance information criterion (DIC) introduced by Spiegelhalter et al. (2002). It is based on the **deviance**, which is defined as

$$D(\boldsymbol{\theta}) = -2 \log p(\mathbf{y} \,|\, \boldsymbol{\theta}) + 2 \log h(\mathbf{y}),$$

where $p(\mathbf{y} \,|\, \boldsymbol{\theta})$ is the likelihood function and $h(\mathbf{y})$ is some fully specified standardizing term that is a function of the data alone. The deviance may be interpreted as the residual information in data conditional on $\boldsymbol{\theta}$, and can therefore be viewed as a measure of "surprise". For the purpose of model comparison, the function $h(\mathbf{y})$ is often set to be unity for all models. We follow this convention in what follows.

Next, we define the **effective number of parameters** p_D of the parametric model as

$$p_D = \overline{D(\boldsymbol{\theta})} - D(\widetilde{\boldsymbol{\theta}}),$$

where

$$\overline{D(\boldsymbol{\theta})} = -2\mathbb{E}_{\boldsymbol{\theta}}[\log p(\mathbf{y} \,|\, \boldsymbol{\theta}) \,|\, \mathbf{y}]$$

is the posterior mean deviance and $\widetilde{\boldsymbol{\theta}}$ is an estimate of $\boldsymbol{\theta}$, which is often taken as the posterior mean. Heuristically, the effective number of parameters measures the reduction in surprise or uncertainty due to estimation. The larger the reduction, the more complex the model is.

The posterior mean deviance can be estimated by averaging $-2 \log p(\mathbf{y} \,|\, \boldsymbol{\theta})$ over the posterior draws of $\boldsymbol{\theta}$. Since the deivance is in log scale, the estimation is typically easy and numerically stable.

Then, the **deviance information criterion** is defined as a trade-off between model fit and model complexity. Specifically, it is the sum of the posterior mean deviance, which can be used as a Bayesian measure of model fit, and the effective number of parameters that measures model complexity:

$$\text{DIC} = \overline{D(\boldsymbol{\theta})} + p_D.$$

Given a set of competing models for the data, the preferred model is the one with the minimum DIC value.

It is clear from the above definition that the DIC depends on the prior only via its effect on the posterior distribution. In situations where the likelihood information dominates, one would expect that the DIC is insensitive to different prior distributions.

As an numerical example, we use the DIC to compare the AR(2) model with MA(1) errors in Section 3.3.1 with a standard AR(2) with independent errors. To compute the DIC for the MA(1) model, we only need minor modifications of the posterior sampler in Section 3.3.1. Specifically, we define the variable `store_dev = zeros(nsim,1);` outside the loop. Then, after the burn-in period, we store the deviance value:

```
store_dev(isave) = -2*llike_MA1(psi,y-X*beta,sig2);
```

Then, once the posterior sampler is completed, we compute the DIC as below.

```
beta_tilde = theta_hat(1:3)';
sig2_tilde = theta_hat(4);
psi_tilde = theta_hat(5);
pD = mean(store_dev) + 2*llike_MA1(psi_tilde,y-X*beta_tilde,sig2_tilde);
DIC = mean(store_dev) + pD;
```

Using 50000 posterior draws, the DIC for the MA(1) model is estimated to be 1083.5. The estimated effective number of parameters is 4.2, which is slightly less than the actual number of parameters of 5. For the AR(2) with independent errors, the DIC is 1097.6 and the effective number of parameters is 4. Hence, the DIC favors the MA(1) model as well, which is inline with the model comparison results using the marginal likelihood.

4.2.3 Variants Based on Conditional Likelihood

For latent variable models, such as the linear regression with t errors in Section 3.2, Celeux et al. (2006) point out that there are numerous alternative definitions of the DIC depending on different concepts of the likelihood.

For example, suppose we augment the model $p(\mathbf{y} \mid \boldsymbol{\theta})$ with a vector of latent variables $\mathbf{z}$ with density $p(\mathbf{z} \mid \boldsymbol{\theta})$ such that

$$p(\mathbf{y} \mid \boldsymbol{\theta}) = \int p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{z}) p(\mathbf{z} \mid \boldsymbol{\theta}) \mathrm{d}\mathbf{z} = \int p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) \mathrm{d}\mathbf{z},$$

where $p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{z})$ is the conditional likelihood and $p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta})$ is the complete-data likelihood. Then, one can define the DIC using the conditional likelihood $p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{z})$

$$\text{DIC}_{\text{con}} = -4\mathbb{E}_{\boldsymbol{\theta},\mathbf{Z}}[\log p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{Z}) \mid \mathbf{y}] + 2 \log p(\mathbf{y} \mid \widetilde{\mathbf{z}}, \widetilde{\boldsymbol{\theta}}), \tag{4.11}$$

where $\widetilde{\mathbf{z}}$ and $\widetilde{\boldsymbol{\theta}}$ are the posterior means.

One advantage of this variant is that $p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{z})$ is typically available in closed-form and is easy to evaluate. However, a few recent papers have warned against using DIC_{con} as a model comparison criterion on both theoretical and practical grounds. Li, Zeng and Yu (2012) argue that the conditional likelihood of the augmented data is nonregular and hence invalidates the standard asymptotic arguments that are needed to justify the original DIC. On practical grounds, Millar (2009) and Chan and Grant (2016) provide Monte Carlo evidence that this DIC variant almost always favors the most complex models. Hence, we recommend avoid using the DIC based on the conditional likelihood.

4.3 Further Reading

Marginal likelihood estimation has generated a vast literature and we have only covered a few popular approaches. We refer interested readers to the articles Han and Carlin (2001), Friel and Pettitt (2008) and Ardia et al. (2012) for a more comprehensive review.

In addition to the marginal likelihood and information criteria, there are a variety of new Bayesian approaches for model comparison and hypothesis testing, such as those developed in Li and Yu (2012), Li, Zeng and Yu (2012) and Li, Zeng and Yu (2014).

Chapter 5

Mixture Models

So far we have focused on the linear regression model with Gaussian innovations. One main advantage of the standard regression is that estimation is easy—a simple Gibbs sampler can be used to estimate the model. In this chapter we consider a few extensions with more flexible error distributions. We will use data augmentation to facilitate estimation, so that the basic structure of the posterior sampler is preserved.

5.1 Scale Mixture of Normals

A number of distributions can be written as a scale mixture of normals. In fact, we saw one such example in Chapter 3—we represented a t distribution as a scale mixture of normals. In particular, we introduced a latent variable that scales the variance of the error distribution that is assumed to be normal.

More precisely, consider the linear regression

$$y_t = \mathbf{x}_t'\boldsymbol{\beta} + \varepsilon_t,$$

where y_t is the dependent variable and $\mathbf{x}_t$ is a vector of covariates. Conditional on a latent variable λ_t, the error term ε_t has a Gaussian distribution: $(\varepsilon_t \,|\, \lambda_t) \sim \mathcal{N}(0, \sigma^2\lambda_t)$.

We showed in Chapter 3 that if we assume $\lambda_t \sim \mathcal{IG}(\nu/2, \nu/2)$, then the marginal distribution of ε_t unconditional on λ_t is a t distribution with degree of freedom parameter ν. In this section we consider another example. In particular, suppose that the scale-mixing variable λ_t follows an exponential distribution $\mathsf{Exp}(1/2)$ with density function

$$f(x) = \frac{1}{2}\mathrm{e}^{-\frac{1}{2}x}, \qquad x > 0.$$

Then, marginally ε_t follows a **double-exponential distribution** with scale parameter $\sigma > 0$ and density function

$$f(\varepsilon) = \frac{1}{2\sigma} \mathrm{e}^{-\frac{|\varepsilon|}{\sigma}}.$$

Before we prove this claim, we first compare the density function of the double-exponential distribution with those of the standard normal and t distributions in Figure 5.1. More specifically, all three distributions have the same zero mean and unit variance, and the degree of freedom parameter for the t distribution is $\nu = 3$.

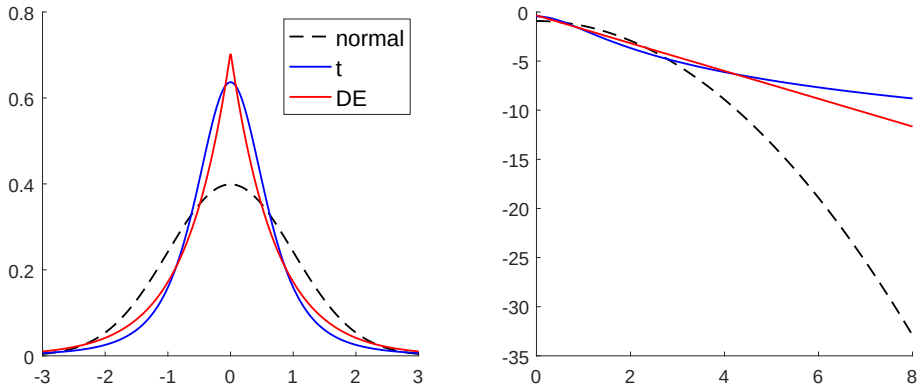

Figure 5.1: The density functions (left panel) and the log densities (right panel) of the standard normal distribution, the t distribution with $\nu = 3$ and the double-exponential distribution.

As the left panel in Figure 5.1 shows, compared to the standard normal distribution, the double-exponential distribution has more mass around zero. In addition, its tails are decaying at a rate that is in between those of the standard normal distribution and the t distribution: slower compared to the former but faster relative to the latter. This, of course, reflects the fact that the tails of the double-exponential go to zero at the rate $\exp(-c_1|x|)$ compared to $\exp(-c_2x^2)$ for normal and $|x|^{-c_3}$ for t, where c_1, c_2 and c_3 are positive constants.

Next, we show that if $(\varepsilon \,|\, \lambda) \sim \mathcal{N}(0, \sigma^2\lambda)$ with $\lambda \sim \mathsf{Exp}(1/2)$, then marginally of λ, ε follows the double-exponential function with scale parameter σ.

First, it is easy to check that the joint density of ε and λ is given by

$$f(\varepsilon, \lambda) = \frac{1}{2}(2\pi\sigma^2)^{-\frac{1}{2}} \lambda^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}\left(\lambda + \frac{\varepsilon^2}{\sigma^2}\lambda^{-1}\right)}.$$

To obtain the marginal density of ε, we integrate out λ:

$$\begin{aligned} f(\varepsilon) &= \int_0^\infty f(\varepsilon, \lambda) \mathrm{d}\lambda \\ &= \frac{1}{2}(2\pi\sigma^2)^{-\frac{1}{2}} \int_0^\infty \lambda^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}\left(\lambda + \frac{\varepsilon^2}{\sigma^2}\lambda^{-1}\right)} \mathrm{d}\lambda. \end{aligned}$$

To compute this integral, we make a change of variable and let $\gamma = \lambda^{\frac{1}{2}}$. The corresponding Jacobian of transformation is therefore 2γ. Then,

$$\begin{aligned} f(\varepsilon) &= \frac{1}{2}(2\pi\sigma^2)^{-\frac{1}{2}} \int_0^\infty 2\gamma \times \gamma^{-1} \mathrm{e}^{-\frac{1}{2}\left(\gamma^2 + \frac{\varepsilon^2}{\sigma^2}\gamma^{-2}\right)} \mathrm{d}\gamma \\ &= (2\pi\sigma^2)^{-\frac{1}{2}} \left(\frac{\pi}{2}\right)^{\frac{1}{2}} \mathrm{e}^{-|\frac{\varepsilon}{\sigma}|} \\ &= \frac{1}{2\sigma} \mathrm{e}^{-\frac{|\varepsilon|}{\sigma}}. \end{aligned}$$

In the above computation we used the following result:

$$\int_0^\infty \mathrm{e}^{-\frac{1}{2}(a^2u^2 + b^2u^{-2})} \mathrm{d}u = \left(\frac{\pi}{2a^2}\right)^{\frac{1}{2}} \mathrm{e}^{-|ab|}.$$

In particular, in our computation $a = 1$ and $b = \varepsilon/\sigma$. Since the marginal density function of ε is the same as that of the double-exponential distribution, we have proved our claim.

5.1.1 Estimation

Next, we construct a Gibbs sampler to estimate the linear regression model with double-exponential errors. For convenience, we reproduce the model below:

$$y_t = \mathbf{x}_t'\boldsymbol{\beta} + \varepsilon_t,$$

where $(\varepsilon_t \,|\, \lambda_t) \sim \mathcal{N}(0, \lambda_t\sigma^2)$ and $\lambda_t \sim \mathsf{Exp}(1/2)$. We assume the independent priors $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \mathbf{V}_{\boldsymbol{\beta}})$ and $\sigma^2 \sim \mathcal{IG}(\nu_0, S_0)$ as before.

Similar to the approach discussed in Chapter 3, we construct a Gibbs sampler that sequentially draws from $(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\lambda}, \sigma^2)$, $(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\lambda})$, and $(\boldsymbol{\lambda} \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2)$. In particular, the first two conditional distributions are exactly the same as before

$$\begin{aligned} &(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\lambda}, \sigma^2) \sim \mathcal{N}(\widehat{\boldsymbol{\beta}}, \mathbf{D}_{\boldsymbol{\beta}}), \\ &(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\lambda}) \sim \mathsf{IG}\left(\nu_0 + \frac{T}{2}, S_0 + \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'\boldsymbol{\Lambda}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right), \end{aligned}$$

where $\mathbf{D}_{\boldsymbol{\beta}} = (\mathbf{V}_{\boldsymbol{\beta}}^{-1} + \mathbf{X}'\boldsymbol{\Lambda}^{-1}\mathbf{X}/\sigma^2)^{-1}$, $\widehat{\boldsymbol{\beta}} = \mathbf{D}_{\boldsymbol{\beta}}(\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_0 + \mathbf{X}'\boldsymbol{\Lambda}^{-1}\mathbf{y}/\sigma^2)$, and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_T)$.

Next, to derive the full conditional distribution of $\boldsymbol{\lambda}$, first note that $\lambda_1, \ldots, \lambda_T$ are conditionally independent given the data and other parameters. In particular, we have

$$\begin{aligned} p(\lambda_t \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2) &\propto (2\pi\sigma^2\lambda_t)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2\sigma^2\lambda_i}(y_t - \mathbf{x}_t'\boldsymbol{\beta})^2} \times \frac{1}{2}\mathrm{e}^{-\frac{1}{2}\lambda_t} \\ &\propto \lambda_t^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}\left(\lambda_t + \frac{(y_t - \mathbf{x}_t'\boldsymbol{\beta})^2}{\sigma^2}\lambda_t^{-1}\right)}. \end{aligned} \tag{5.1}$$

This does not appear to be a standard distribution. However, it turns out that the inverse λ_t^{-1} follows the inverse Gaussian distribution.

We say that the random variable X follows an **inverse Gaussian distribution** if its kernel function is given by

$$f(x) \propto x^{-\frac{3}{2}} \mathrm{e}^{-\frac{\psi(x-\mu)^2}{2x\mu^2}}, \qquad x > 0.$$

We denote the distribution as $\mathsf{invGauss}(\psi, \mu)$.

Suppose $X \sim \mathsf{invGauss}(\psi, \mu)$. Below we derive the kernel of the inverse $Z = 1/X$. Since the Jacobian of transformation is $J = z^{-2}$, we have

$$\begin{aligned} f(z) &\propto z^{-2} \times z^{\frac{3}{2}} \mathrm{e}^{-\frac{\psi(z^{-1}-\mu)^2}{2z^{-1}\mu^2}} \\ &\propto z^{-2} z^{\frac{3}{2}} \mathrm{e}^{-\frac{\psi}{2\mu^2}(z^{-1} + \mu^2 z)} \\ &= z^{-\frac{1}{2}} \mathrm{e}^{-\frac{\psi}{2}(z + \mu^{-2}z^{-1})}. \end{aligned}$$

Comparing this kernel with the quantity in (5.1), we conclude that

$$(\lambda_i^{-1} \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma^2) \sim \mathsf{invGauss}\left(1, \frac{\sigma}{|y_i - \mathbf{x}_i'\boldsymbol{\beta}|}\right).$$

To implement the Gibbs sampler, we need an efficient way to sample from the inverse Gaussian distribution. We will provide a proof here, but a draw from the $\mathsf{invGauss}(\psi, \mu)$ distribution can be obtained as follows: first, obtain a draw from the chi-squared distribution with degree of freedom parameter 1: $\nu_0 \sim \chi_1^2$. Then, compute

$$\begin{aligned} X_1 &= \mu + \frac{\mu^2\nu_0}{2\psi} - \frac{\mu}{2\psi}\sqrt{4\mu\psi\nu_0 + \mu^2\nu_0^2} \\ X_2 &= \frac{\mu^2}{X_1}. \end{aligned}$$

Set X equal to X_1 with probability $\mu/(\mu + X_1)$ and equal to X_2 with probability $X_1/(\mu + X_1)$. We refer the readers to Koop et al. (2007, p. 260) for more details.

The following MATLAB script `igaurnd.m` implements this algorithm to sample from the $\mathsf{invGauss}(\psi, \mu)$ distribution.

```
function x = igaurnd(psi,mu,n)
nu0 = randn(n,1).^2;
x1 = mu + mu.^2.*nu0./(2*psi) ...
    - mu./(2*psi).*sqrt(4*mu.*psi.*nu0 + mu.^2.*nu0.^2);
x2 = mu.^2./x1;
p = mu./(mu+x1);
U = p>rand(n,1);
x = U.*x1 + (1-U).*x2;
end
```

5.1.2 Empirical Example: Fitting Inflation Using an AR(2) with Double Exponential Errors

As an empirical illustration, we revisit the empirical application in Section 3.2.4. There we fitted the US PCE inflation using a linear regression model with t errors. The degree of freedom parameter of the t distribution is estimated to be about 5, showing that the tails of the error distribution are much heavier than those of the normal.

In addition, we conducted a Bayesian model comparison exercise in Section 4.1.2 in which we compared this t model with the standard normal regression with Gaussian errors. Consistent with the estimation results, the data favor the t model: the marginal likelihood of the former is estimated to be -554, whereas the estimate for the latter is -565.5.

Here we consider the following AR(2) model with double-exponential errors:

$$y_t = \beta_1 + y_{t-1}\beta_2 + y_{t-2}\beta_3 + \varepsilon_t,$$

where $(\varepsilon_t \mid \lambda_t) \sim \mathcal{N}(0, \lambda_t \sigma^2)$ with $\lambda_t \sim \mathsf{Exp}(1/2)$.

We implement the 3-block Gibbs sampler as described in the previous section. As before, we construct the $T \times T$ inverse matrix $\mathbf{\Lambda}^{-1} = (\lambda_1^{-1}, \ldots, \lambda_T^{-1})$ as a sparse matrix using the line

```
iLam = sparse(1:T,1:T,1./lam);
```

The main MATLAB script `linreg_de.m` is given below.

```
% linreg_de.m
nsim = 20000; burnin = 1000;

    % load data
data_raw = load('USPCE_2015Q4.csv');
data = 400*log(data_raw(2:end)./data_raw(1:end-1));
y0 = data(1:2); % [y_{-1}, y_0]
y = data(3:end);
T = size(y,1);
X = [ones(T,1) [y0(1); y(1:end-1)] [y0; y(1:end-2)]];

    % prior
beta0 = zeros(3,1); iVbeta = speye(3)/100;
nu0 = 3; S0 = 1*(nu0 - 1);

    % initialize the Markov chain
beta = (X'*X)\(X'*y);
sig2 = sum((y-X*beta).^2)/T;
lam = gamrnd(1,2,T,1);
iLam = sparse(1:T,1:T,1./lam);

store_theta = zeros(nsim,4);  % [beta' sig2]

for isim = 1:nsim + burnin
        % sample beta
    Dbeta = (iVbeta + X'*iLam*X/sig2)\speye(3);
    beta_hat = Dbeta*(iVbeta*beta0 + X'*iLam*y/sig2);
    C = chol(Dbeta,'lower');
    beta = beta_hat + C*randn(3,1);

        % sample sig2
    e = y - X*beta;
    sig2 = 1/gamrnd(nu0+T/2,1/(S0 + e'*iLam*e/2));

        % sample lam
    mu = sqrt(sig2)./abs(y-X*beta);
    lam = 1./igaurnd(ones(T,1),mu);
    iLam = sparse(1:T,1:T,1./lam);

        % store the parameters
    if isim > burnin
```

```
        isave = isim - burnin;
        store_theta(isave,:) = [beta' sig2];
    end
end
```

Using a sample of 20000 draws, the posterior means of $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)'$ are estimated to be $(1.72, 0.34, 0.38)'$. These estimates are similar to those under the t model. We also compute the marginal likelihood using the cross-entropy method (see Section 4.1.4) . The estimate is -551.5, compared to the estimate of -554 under the t model. The result therefore suggests that the tails of the error distribution are heavier than those of the normal distribution but thinner than those of t.

5.2 Finite Mixture of Normals

Last section discussed an example of a scale mixture of normals. Another type of mixtures is the **finite mixture of normal distributions**. To keep the discussion concrete, consider a standard linear regression

$$y_t = \mathbf{x}_t'\boldsymbol{\beta} + \varepsilon_t,$$

where $(\varepsilon_t \,|\, \sigma^2) \sim \mathcal{N}(0, \sigma^2)$. We can think of the observation y_t coming from the normal distribution $(y_t \,|\, \boldsymbol{\beta}, \sigma^2) \sim \mathcal{N}(\mathbf{x}_t'\boldsymbol{\beta}, \sigma^2)$.

It is therefore natural to consider the extension to a mixture of two normal components:

$$(y_t \,|\, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q) \sim q\,\mathcal{N}(\mathbf{x}_t'\boldsymbol{\beta}_1, \sigma_1^2) + (1-q)\mathcal{N}(\mathbf{x}_t'\boldsymbol{\beta}_2, \sigma_2^2),$$

where $q \in (0,1)$ is the mixture probability. In words, it means that y_t is drawn from the first component $\mathcal{N}(\mathbf{x}_t'\boldsymbol{\beta}_1, \sigma_1^2)$ with probability q; it is from the second component $\mathcal{N}(\mathbf{x}_t'\boldsymbol{\beta}_2, \sigma_2^2)$ with probability $1-q$.

This type of finite normal mixtures are often useful for modeling time series that behaves differently across different regimes. For example, the level and dynamic of inflation during expansion might be different from those in recession. By having two—or more—components, the model can handle structural breaks or time instabilities in the time series.

The density function of y_t under the 2-component mixture is given by

$$(y_t \,|\, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q) = q(2\pi\sigma_1^2)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2\sigma_1^2}(\mathbf{y}_t - \mathbf{x}_t'\boldsymbol{\beta}_1)^2} + (1-q)(2\pi\sigma_2^2)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2\sigma_2^2}(\mathbf{y}_t - \mathbf{x}_t'\boldsymbol{\beta}_2)^2}. \tag{5.2}$$

For estimation, this mixture density is more difficult to handle. The reason is that the usual normal and inverse-gamma priors for the regression coefficients and the

variance, respectively, are no longer conjugate. To circumvent this problem, we again use the data augmentation approach.

More specifically, we introduce a discrete random variable $S_t \in \{1, 2\}$ with $\mathbb{P}(S_t = 1 \,|\, q) = q$, which is often called the **component label** or **component indicator**. Then, given this component label, the dependent variable is drawn from the associated distribution:

$$(y_t \,|\, S_t, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2) \sim \mathcal{N}(\mathbf{x}_t'\boldsymbol{\beta}_{S_t}, \sigma_{S_t}^2).$$

Note that above we use the shorthand notation $\boldsymbol{\beta}_{S_t}$ to denote $\boldsymbol{\beta}_1$ if $S_t = 1$ and $\boldsymbol{\beta}_2$ if $S_t = 2$. Similarly for $\sigma_{S_t}^2$.

Hence, given the component labels $S_1, \ldots, S_T$, we can simply partition the dataset into two subsets: the observations with the labels $S_t = 1$ and those with the label $S_t = 2$. Then, the regime-specific parameters $\boldsymbol{\beta}_i$ and $\sigma_i^2, i = 1, 2$, can be estimated using the relevant observations.

To show that this latent variable representation gives the same model, we simply integrate out the component label S_t:

$$\begin{aligned} p(y_t \,|\, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q) &= \sum_{i=1}^{2} p(y_t \,|\, S_t = i, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2)\mathbb{P}(S_t = i \,|\, q) \\ &= (2\pi\sigma_1^2)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2\sigma_1^2}(\mathbf{y}_t - \mathbf{x}_t'\boldsymbol{\beta}_1)^2} q + (2\pi\sigma_2^2)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2\sigma_2^2}(\mathbf{y}_t - \mathbf{x}_t'\boldsymbol{\beta}_2)^2} (1 - q). \end{aligned}$$

Hence, this latent variable representation therefore implies the same observed-data likelihood in (5.2).

5.2.1 Estimation

To complete the model specification, we assume the usual independent priors for the regression coefficients and variances: $\boldsymbol{\beta}_i \sim \mathcal{N}(\boldsymbol{\beta}_0, \mathbf{V}_{\boldsymbol{\beta}})$ and $\sigma_i^2, \sim \mathcal{IG}(\nu_0, S_0)$ for $i = 1, 2$. For q, we assume a **beta prior**: $q \sim \mathcal{B}(a_0, b_0)$ with density function

$$p(q) = \frac{\Gamma(a_0 + b_0)}{\Gamma(a_0)\Gamma(b_0)} q^{a_0 - 1}(1 - q)^{b_0 - 1},$$

where $\Gamma(\cdot)$ is the gamma function. Finally, the prior for $\mathbf{S} = (S_1, \ldots, S_T)'$ is given above—i.e., the component labels are mutually independent with success probability $\mathbb{P}(S_t = 1 \,|\, q) = q$.

To estimate the 2-component normal mixture model, we consider the following 4-block Gibbs sampler:

1. Draw $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2) \sim p(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \,|\, \mathbf{y}, \mathbf{S}, \sigma_1^2, \sigma_2^2, q)$.
2. Draw $(\sigma_1^2, \sigma_2^2) \sim p(\sigma_1^2, \sigma_2^2 \,|\, \mathbf{y}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \mathbf{S}, q)$.
3. Draw $\mathbf{S} \sim p(\mathbf{S} \,|\, \mathbf{y}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q)$.
4. Draw $q \sim p(q \,|\, \mathbf{y}, \mathbf{S}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2)$.

To implement Step 1, first observe that only those observations with component labels $S_t = 1$ are informative about $\boldsymbol{\beta}_1$; similarly for $\boldsymbol{\beta}_2$. In fact, we can split the dataset into two samples depending on whether $S_t = 1$ or $S_t = 2$. Then, we sample $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ separately. To that end, let $\mathbf{y}_1$ denote the stacked observations y_t corresponding to the sample with $S_t = 1$, and let $\mathbf{X}_1$ denote the associated regressors. Then, we have

$$(\boldsymbol{\beta}_1 \,|\, \mathbf{y}, \mathbf{S}, \sigma_1^2) \sim \mathcal{N}(\widehat{\boldsymbol{\beta}}_1, \mathbf{D}_{\boldsymbol{\beta}_1}),$$

where

$$\mathbf{D}_{\boldsymbol{\beta}_1} = \left(\mathbf{V}_{\boldsymbol{\beta}}^{-1} + \frac{1}{\sigma_1^2}\mathbf{X}_1'\mathbf{X}_1\right)^{-1}, \quad \widehat{\boldsymbol{\beta}}_1 = \mathbf{D}_{\boldsymbol{\beta}_1}\left(\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_0 + \frac{1}{\sigma_1^2}\mathbf{X}_1'\mathbf{y}_1\right).$$

Similarly, we can sample $\boldsymbol{\beta}_2$ from its conditional distribution

$$(\boldsymbol{\beta}_2 \,|\, \mathbf{y}, \mathbf{S}, \sigma_2^2) \sim \mathcal{N}(\widehat{\boldsymbol{\beta}}_2, \mathbf{D}_{\boldsymbol{\beta}_2}),$$

where $\widehat{\boldsymbol{\beta}}_2$ and $\mathbf{D}_{\boldsymbol{\beta}_2}$ are computed using the subsample corresponding to $S_t = 2$.

As in Step 1, we next sample σ_1^2 and σ_2^2 using the corresponding subsamples. More specifically, we have

$$(\sigma_1^2 \,|\, \mathbf{y}, \mathbf{S}, \boldsymbol{\beta}_1) \sim \mathcal{IG}\left(\nu_0 + \frac{T_1}{2}, S_0 + \frac{1}{2}(\mathbf{y}_1 - \mathbf{X}_1\boldsymbol{\beta}_1)'(\mathbf{y}_1 - \mathbf{X}_1\boldsymbol{\beta}_1)\right),$$

where T_1 is the number of observations with $S_t = 1$. We sample σ_2^2 similarly.

To sample $\mathbf{S} = (S_1, \ldots, S_T)'$, first note that the component indicators are conditionally independent given the data and the parameters. Specifically, the joint conditional mass function of $\mathbf{S}$ can be written as a product of T univariate mass functions:

$$p(\mathbf{S} \,|\, \mathbf{y}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q) \propto \prod_{t=1}^{T} p(y_t \,|\, S_t, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2) p(S_t \,|\, q).$$

Therefore, we can sample each S_t separately.

Next, each component indicator S_t is a Bernoulli random variable that takes value in $\{1, 2\}$, and it suffices to compute its success probability $\mathbb{P}(S_t = 1 \,|\, y_t, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q)$. To that end, note that for $i = 1, 2$, we have

$$\begin{aligned}\mathbb{P}(S_t = i \,|\, y_t, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q) &= c_t p(y_t \,|\, S_t = i, \boldsymbol{\beta}_i, \sigma_i^2)\mathbb{P}(S_t = i \,|\, q) \\ &= c_t \phi(y_t; \mathbf{x}_t'\boldsymbol{\beta}_i, \sigma_i^2)\mathbb{P}(S_t = i \,|\, q),\end{aligned}$$

where c_t is the normalizing constant and $\phi(a; \mu, \sigma^2)$ is the Gaussian density with mean μ and variance σ^2. Next, we use the fact that the probabilities sum to one to calculate c_t:

$$\begin{aligned}1 &= c_t\mathbb{P}(S_t = 1 \,|\, y_t, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q) + c_t\mathbb{P}(S_t = 2 \,|\, y_t, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q) \\ c_t &= \frac{1}{\phi(y_t; \mathbf{x}_t'\boldsymbol{\beta}_1, \sigma_1^2)q + \phi(y_t; \mathbf{x}_t'\boldsymbol{\beta}_2, \sigma_2^2)(1-q)}.\end{aligned}$$

Finally, putting all these derivations together, we conclude that

$$\mathbb{P}(S_t = 1 \,|\, y_t, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q) = \frac{\phi(y_t; \mathbf{x}_t'\boldsymbol{\beta}_1, \sigma_1^2)q}{\phi(y_t; \mathbf{x}_t'\boldsymbol{\beta}_1, \sigma_1^2)q + \phi(y_t; \mathbf{x}_t'\boldsymbol{\beta}_2, \sigma_2^2)(1-q)}$$

and $\mathbb{P}(S_t = 2 \,|\, y_t, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q) = 1 - \mathbb{P}(S_t = 1 \,|\, y_t, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q)$.

To simulate S_t with success probability $q_t = \mathbb{P}(S_t = 1 \,|\, y_t, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2, q)$, we first obtain a uniform draw $Z \sim \mathcal{U}(0, 1)$. If $Z < q_t$, we set $S_t = 1$; otherwise we set $S_t = 2$. To see that S_t has the desired mass function, it suffices to compute the probability that it is 1—its probability is given by

$$\mathbb{P}(Z < q_t) = \int_0^{q_t} 1 \mathrm{d}x = q_t,$$

where we used the fact the density function of $\mathcal{U}(0, 1)$ is unity.

Finally, to implement Step 4, we derive the full conditional distribution of q below. To that end, observe that q only appears in its prior and the prior probabilities $\mathbb{P}(S_t = 1 \,|\, q) = q$ and $\mathbb{P}(S_t = 2 \,|\, q) = 1 - q$. The conditional mass function of S_t can be written succinctly as

$$p(S_t \,|\, q) = q^{\mathbb{1}(S_t=1)}(1-q)^{1-\mathbb{1}(S_t=1)},$$

where $\mathbb{1}(\cdot)$ is the indicator function that takes the value 1 if the argument is true and 0 otherwise. Then, we have

$$\begin{aligned}p(q \,|\, \mathbf{y}, \mathbf{S}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2) &\propto p(q) \times \prod_{t=1}^{T} p(S_t \,|\, q) \\ &\propto q^{a_0-1}(1-q)^{b_0-1} \times \prod_{t=1}^{T} q^{\mathbb{1}(S_t=1)}(1-q)^{1-\mathbb{1}(S_t=1)} \\ &= q^{a_0+T_1-1}(1-q)^{b_0+T_2-1},\end{aligned}$$

where $T_1 = \sum_{t=1}^{T} \mathbb{1}(S_t = 1)$ and $T_2 = T - T_1$. Hence, we update q by counting the occurrences of each component.

The above density function is in fact the kernel of the $\mathcal{B}(a_0+T_1, b_0+T_2)$ distribution. A draw from a beta distribution is implemented in most standard packages. In MATLAB this is done using the command `betarnd`.

5.2.2 Empirical Example: Fitting Inflation Using an AR(2) with a 2-Component Normal Mixture

For illustration, we reconsider the empirical application in Section 5.1.2 that involves fitting the US PCE inflation using an AR(2) model with double exponential errors. Here the model is an AR(2) regression with a 2-component normal mixture. More specifically, consider the following model:

$$y_t = \beta_{S_t,1} + y_{t-1}\beta_{S_t,2} + y_{t-2}\beta_{S_t,3} + \varepsilon_t,$$

where $S_t \in \{1, 2\}$ is the component indicator and $(\varepsilon_t \,|\, S_t) \sim \mathcal{N}(0, \sigma^2_{S_t})$. In other words, both the error variance and the regression coefficients can differ across components.

The following MATLAB script `linreg_fin_mix.m` implements the 4-block Gibbs sampler described in the previous section. Note that when drawing the AR coefficients, we impose the condition that the implied AR process is stationary. For an AR(2) process, this amounts to imposing three inequalities: $\beta_{i,2} + \beta_{i,3} < 1$, $\beta_{i,3} - \beta_{i,2} < 1$, and $\beta_{i,3} > -1$, $i = 1, 2$. Also note that when we sample $\mathbf{S}$, we vectorize the operations, instead of using a for-loop to draw each S_t one at a time.

```
% linreg_fin_mix.m
nsim = 50000; burnin = 1000;
    % load data
data_raw = load('USPCE_2015Q4.csv');
data = 400*log(data_raw(2:end)./data_raw(1:end-1));
y0 = data(1:2); % [y_{-1}, y_0]
y = data(3:end);
T = size(y,1);
    % prior
beta0 = zeros(3,1); iVbeta = speye(3)/100;
nu0 = 3; S0 = 1*(nu0 - 1);
a0 = 2; b0 = 2;
    % constuct a few things
store_theta = zeros(nsim,9); %[beta,sig2,q]
```

```
store_S = zeros(T,2);
like = zeros(T,2);
X = [ones(T,1) [y0(2);y(1:end-1)] [y0(1);y0(2);y(1:end-2)]];
    % initialize the Markov chain
S = [ones(floor(T/2),1);2*ones(T-floor(T/2),1)];
beta = zeros(6,1);
sig2 = zeros(2,1);
for i = 1:2
    idx = (S == i); Ti = sum(idx); Xi = X(idx,:); yi = y(idx,:);
    betai = (Xi'*Xi)\(Xi'*yi);
    beta((i-1)*3+1:i*3) = betai;
    e = yi - Xi*betai;
    sig2(i) = e'*e/Ti;
end
q = .5;

for isim = 1:nsim + burnin
    for i = 1:2
            % extract data for state S_t == i
        idx = (S == i);
        Ti = sum(idx);
        Xi = X(idx,:);
        yi = y(idx,:);
            % sample beta
        Dbeta = (iVbeta + Xi'*Xi/sig2(i))\speye(3);
        beta_hat = Dbeta*(iVbeta*beta0 + Xi'*yi/sig2(i));
        C = chol(Dbeta,'lower');
        is_sty = false;
        while ~is_sty
            betai = beta_hat + C*randn(3,1);
                % check if stationary
            if betai(2)+betai(3)<1 && betai(3)-betai(2)<1 && betai(3)>-1
                is_sty = true;
            end
        end
        beta((i-1)*3+1:i*3) = betai;

            % sample sig2
        e = yi - Xi*betai;
        sig2(i) = 1/gamrnd(nu0+Ti/2,1/(S0 + e'*e/2));
    end

        % sample S
```

```
    for i = 1:2
        betai = beta((i-1)*3+1:i*3);
        sig2i = sig2(i);
        like(:,i) = normpdf(y,X*betai,sqrt(sig2i)*ones(T,1));
    end
    joint_den = [q*like(:,1) (1-q)*like(:,2)];
    prob = joint_den./repmat(sum(joint_den,2),1,2);
    S = 2 - (prob(:,1) > rand(T,1));

        % sample q
    T1 = sum(S == 1);
    q = betarnd(a0+T1,b0+T-T1);

        % store the parameters
    if isim > burnin
        isave = isim - burnin;
        store_theta(isave,:) = [beta' sig2' q];
        for j=1:2
            store_S(:,j) = store_S(:,j) + (S == j);
        end
    end
end
theta_hat = mean(store_theta)';
S_hat = store_S/nsim;
```

Using a sample of 50000 posterior draws, we estimate the posterior probabilities that $S_t = 1$ for $t = 1, \ldots, T$. The results are plotted in Figure 5.2. It appears that the first component corresponds to the periods of deep recessions—late 1940s, mid-1970s and the Great Recession. The posterior mean of q is estimated to be 0.17, indicating that the first component includes about 17% of the observations.

The posterior estimates of the first set of AR coefficients $\boldsymbol{\beta}_1 = (\beta_{1,1}, \beta_{1,2}, \beta_{1,3})'$ are $(-4.79, 1.19, -0.54)$. These estimates suggest that the average inflation under the first component is negative and is highly persistent. By contrast, the estimates of $\boldsymbol{\beta}_2 = (\beta_{2,1}, \beta_{2,2}, \beta_{2,3})'$ are $(1.93, 0.31, 0.39)$, suggesting a positive average inflation and a moderately persistent inflation process. The estimates of the variances σ_1^2 and σ_2^2 are 4.46 and 4.73, respectively. These results suggest that the variances are not substantially different across the two components.

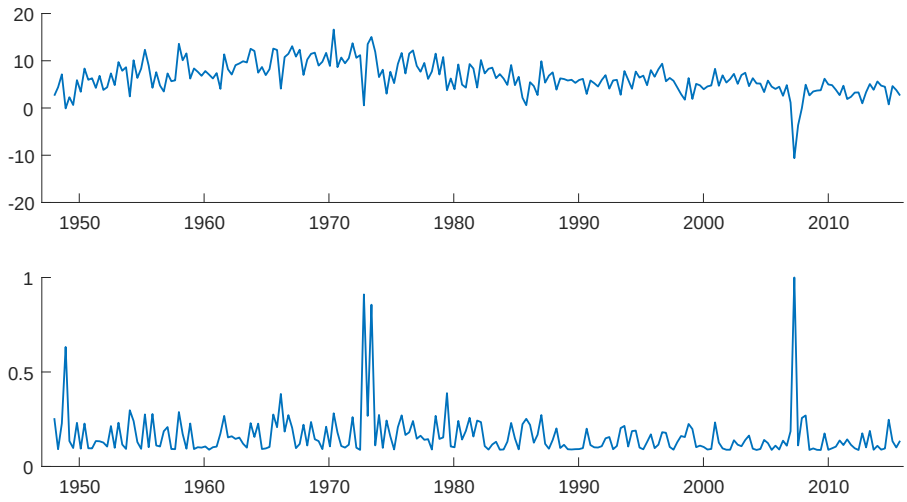

Figure 5.2: PCE inflation (top panel) and the posterior probabilities $\mathbb{P}(S_t = 1 \,|\, \mathbf{y})$ (bottom panel).

Chapter 6

Unobserved Components Models

In this chapter we discuss a class of models generally known as **unobserved components models**, which have become increasingly popular for modeling macroeconomic time-series. In particular, unobserved components models are often used to decompose time-series into trend and cycle components. For example, inflation or output gaps can be obtained by fitting an unobserved components model. These models fall within a broader class of models called **state space model**—in particular, they are *linear* state space models. In the coming chapters we will see more examples of state space models.

6.1 Local Level Model

The simplest unobserved components model is the following local level model:

$$y_t = \tau_t + \varepsilon_t, \tag{6.1}$$

where $\{\varepsilon_t\}$ are assumed to be iid $\mathcal{N}(0, \sigma^2)$. The *local level* or the *trend* component τ_t is modeled as a random walk:

$$\tau_t = \tau_{t-1} + \eta_t, \tag{6.2}$$

where $\{\eta_t\}$ are iid $\mathcal{N}(0, \omega^2)$ and τ_0 is treated as an unknown parameter.

In other words, the local level model decomposes y_t into two components: the non-stationary trend component τ_t and the transitory cycle component ε_t. The random walk assumption in (6.2) captures the idea that the trend evolves smoothly over time, and that consecutive terms are "close". More precisely, the conditional distribution of τ_t given τ_{t-1} and ω^2 is $\mathcal{N}(\tau_{t-1}, \omega^2)$. In particular, it implies that τ_t centers around τ_{t-1}, and ω^2 controls probabilistically how close these two quantities are—ω^2

therefore controls how smooth the overall trend is. In the limit that ω^2 goes to 0, the local level model reduces to a linear regression with a constant intercept.

The state equation (6.2) implies that τ_t has a *stochastic* trend—i.e., the unobserved component τ_t can wander widely over time (can trend upward or downward), and the trending behavior is nondeterministic—as opposed to a *deterministic* trend, such as

$$\alpha_t = \alpha + \beta t.$$

In the latter case, the trend α_t is an exact function of time, whereas the stochastic trend τ_t involves a random error term η_t. To see that τ_t exhibits trending behavior and can wander widely, note that by recursive substitution, we have

$$\tau_t = \tau_0 + \sum_{s=1}^{t} \eta_s.$$

Hence, the conditional variance of τ_t given the initial τ_0 is

$$\text{Var}(\tau_t \,|\, \tau_0) = t\omega^2,$$

i.e., the variance of the stochastic trend τ_t is increasing linear with time, which implies that τ_t can wander over an increasing range of values as time increases.

As mentioned above, the local level model is an example of a state space model. More generally, a state space model consists of two modeling levels. In the first level, observations are related to the latent or unobserved variables called **states** according to the **observation** or **measurement equation**. In the second level, the evolution of the states is modeled via the **state** or **transition equation**.

In the local level model, the state is τ_t; (6.1) is the observation equation and (6.2) is the state equation. In addition, the system is an instance of a *linear Gaussian* state space model in which both equations are linear in the state τ_t and both the error terms are Gaussian.

6.1.1 Estimation

To estimate the model in (6.1) and (6.2), we construct a Gibbs sampler to sequentially sample the states $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_T)'$ and the three parameters: the initial state τ_0 and the two variances σ^2 and ω^2. The main challenge is the sampling of the states, which are of dimension T. The states from linear Gaussian models are traditionally sampled by Kalman filter based methods, such as Carter and Kohn (1994), Früwirth-Schnatter (1994), de Jong and Shephard (1995) and Durbin and Koopman (2002).

Here we introduce some more recent algorithms based on band matrix routines to sample the states. One main advantage of this new approach is the transparent derivation—all we need is standard linear regression results derived in previous chapters. In addition, these new algorithms are also more computationally efficient compared to Kalman filter based methods. The following exposition is based on the formulation in Chan and Jeliazkov (2009). See also the discussion on computational efficiency in McCausland, Miller and Pelletier (2011).

To complete the model specification, we assume the following independent priors for τ_0, σ^2 and ω^2:

$$\tau_0 \sim \mathcal{N}(a_0, b_0), \quad \sigma^2 \sim \mathcal{IG}(\nu_{\sigma^2}, S_{\sigma^2}), \quad \omega^2 \sim \mathcal{IG}(\nu_{\omega^2}, S_{\omega^2}).$$

In our setting the prior on the smoothness parameter ω^2 typically has a large impact on the posterior results. This is because ω^2 is inferred only from the states $\boldsymbol{\tau}$, which are not directly observed. In other words, there are two layers: the data $\mathbf{y}$ inform us about $\boldsymbol{\tau}$, which in turn reveal information about ω^2. Consequently, the prior on ω^2 plays a relatively larger role. Here we assume the conventional inverse-gamma prior on ω^2. In later sections we explore other forms of priors and investigate their impact on the posterior results.

We can use the following 4-block Gibbs sampler to simulate from the joint posterior $p(\boldsymbol{\tau}, \sigma^2, \omega^2, \tau_0 \,|\, \mathbf{y})$:

1. sample $(\boldsymbol{\tau} \,|\, \mathbf{y}, \sigma^2, \omega^2, \tau_0)$,
2. sample $(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\tau}, \omega^2, \tau_0)$,
3. sample $(\omega^2 \,|\, \mathbf{y}, \boldsymbol{\tau}, \sigma^2, \tau_0)$,
4. sample $(\tau_0 \,|\, \mathbf{y}, \boldsymbol{\tau}, \sigma^2, \omega^2)$.

To implement the first step, we first derive the conditional density of $\boldsymbol{\tau}$:

$$p(\boldsymbol{\tau} \,|\, \mathbf{y}, \sigma^2, \omega^2, \tau_0) \propto p(\mathbf{y} \,|\, \boldsymbol{\tau}, \sigma^2) p(\boldsymbol{\tau} \,|\, \omega^2, \tau_0).$$

The density $p(\mathbf{y} \,|\, \boldsymbol{\tau}, \sigma^2)$ is the **conditional likelihood** given the states $\boldsymbol{\tau}$, as defined by the observation equation (6.1). This is to distinguish from the **observed-data** or **integrated likelihood** defined as

$$p(\mathbf{y} \,|\, \sigma^2, \omega^2, \tau_0) = \int p(\mathbf{y} \,|\, \boldsymbol{\tau}, \sigma^2) p(\boldsymbol{\tau} \,|\, \omega^2, \tau_0) \mathrm{d}\boldsymbol{\tau}.$$

The second density in the integrand, $p(\boldsymbol{\tau} \,|\, \omega^2, \tau_0)$, is the prior density of $\boldsymbol{\tau}$ implied by (6.2). In what follows, we derive explicit expressions for both $p(\mathbf{y} \,|\, \boldsymbol{\tau}, \sigma^2)$ and

$p(\boldsymbol{\tau} \,|\, \omega^2, \tau_0)$, and show that the conditional density $p(\boldsymbol{\tau} \,|\, \mathbf{y}, \sigma^2, \omega^2, \tau_0)$ is in fact normal. Then, we discuss an efficient algorithm based on band matrix routines to draw from this high-dimensional density.

Let $\mathbf{y} = (y_1, \ldots, y_T)'$ and define $\boldsymbol{\varepsilon}$ similarly. First, rewrite the observation equation (6.1) in matrix notation:

$$\mathbf{y} = \boldsymbol{\tau} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_T)$, and the conditional likelihood is given by

$$p(\mathbf{y} \,|\, \boldsymbol{\tau}, \sigma^2) = (2\pi\sigma^2)^{-\frac{T}{2}} \mathrm{e}^{-\frac{1}{2\sigma^2}(\mathbf{y}-\boldsymbol{\tau})'(\mathbf{y}-\boldsymbol{\tau})}. \tag{6.3}$$

Hence, the local level model is just a linear regression with regression matrix $\mathbf{X} = \mathbf{I}_T$ and regression coefficient vector $\boldsymbol{\tau}$. If we can derive the prior of $\boldsymbol{\tau}$, we can use standard linear regression results to obtain the full conditional density of $\boldsymbol{\tau}$.

To that end, we rewrite the state equation (6.2) as:

$$\mathbf{H}\boldsymbol{\tau} = \widetilde{\boldsymbol{\alpha}}_\tau + \boldsymbol{\eta}, \tag{6.4}$$

where $\widetilde{\boldsymbol{\alpha}}_\tau = (\tau_0, 0, \ldots, 0)'$, $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \omega^2 \mathbf{I}_T)$ and

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \end{pmatrix}.$$

Note that $|\mathbf{H}| = 1 \neq 0$, and therefore $\mathbf{H}$ is invertible. We claim that $\mathbf{H}^{-1}\widetilde{\boldsymbol{\alpha}}_\tau = \tau_0 \mathbf{1}_T$, where $\mathbf{1}_T$ is a $T \times 1$ column of ones. This can be seen by solving the triangular system $\mathbf{H}\mathbf{z} = \widetilde{\boldsymbol{\alpha}}_\tau$ for $\mathbf{z} = (z_1, \ldots, z_T)'$ by forward substitution. Specifically, the first equation of the system reads $z_1 = \tau_0$. The second equation is $-z_1 + z_2 = 0$, and solving it for z_2 we get $z_2 = \tau_0$. Indeed, the t-th equation is $-z_{t-1} + z_t = 0$, and therefore $z_t = z_{t-1}$. We conclude that $z_1 = z_2 = \cdots = z_T = \tau_0$. Since $\mathbf{z} = \mathbf{H}^{-1}\widetilde{\boldsymbol{\alpha}}_\tau$ by construction, we have proved the claim.

The above computation has shown that if we only want the quantity $\mathbf{A}^{-1}\mathbf{b}$, it is unnecessary to compute the inverse $\mathbf{A}^{-1}$. Instead, we can simply solve $\mathbf{A}\mathbf{z} = \mathbf{b}$ for $\mathbf{z}$. Moreover, if $\mathbf{A}$ is a band lower triangular matrix, the number of operations in solving the system is linear in the dimension of the system. Hence, it can be done very quickly even when the dimension is large.

Now, it follows from (6.4) that

$$(\boldsymbol{\tau} \,|\, \omega^2, \tau_0) \sim \mathcal{N}(\tau_0 \mathbf{1}_T, \omega^2 (\mathbf{H}'\mathbf{H})^{-1}),$$

with density

$$p(\boldsymbol{\tau} \,|\, \omega^2, \tau_0) = (2\pi\omega^2)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2\omega^2}(\boldsymbol{\tau}-\tau_0\mathbf{1}_T)'\mathbf{H}'\mathbf{H}(\boldsymbol{\tau}-\tau_0\mathbf{1}_T)}. \tag{6.5}$$

Hence, using a similar derivation in Section 3.1, we can combine (6.3) and (6.5) to get

$$(\boldsymbol{\tau} \,|\, \mathbf{y}, \sigma^2, \omega^2, \tau_0) \sim \mathcal{N}(\widehat{\boldsymbol{\tau}}, \mathbf{K}_{\boldsymbol{\tau}}^{-1}),$$

where

$$\mathbf{K}_{\boldsymbol{\tau}} = \frac{1}{\omega^2}\mathbf{H}'\mathbf{H} + \frac{1}{\sigma^2}\mathbf{I}_T, \quad \widehat{\boldsymbol{\tau}} = \mathbf{K}_{\boldsymbol{\tau}}^{-1}\left(\frac{\tau_0}{\omega^2}\mathbf{H}'\mathbf{H}\mathbf{1}_T + \frac{1}{\sigma^2}\mathbf{y}\right).$$

The main difficulty of sampling $(\boldsymbol{\tau} \,|\, \mathbf{y}, \sigma^2, \omega^2, \tau_0)$ using Algorithm 2.1 is that the covariance matrix $\mathbf{K}_{\boldsymbol{\tau}}^{-1}$ is a full $T \times T$ matrix, and computing its Cholesky factor is time-consuming—it involves $\mathcal{O}(T^3)$ operations. Fortunately, the precision matrix $\mathbf{K}_{\boldsymbol{\tau}}$ is a band matrix, and computing its Cholesky factor involves only $\mathcal{O}(T)$ operations. Hence, it would be useful to have a method to draw from a normal distribution using only the precision matrix. This is described in the following algorithm.

Algorithm 6.1. (Sampling from Normal Distribution Given Precision).

To generate R independent draws from $\mathcal{N}(\boldsymbol{\mu}, \mathbf{K}^{-1})$ of dimension n, carry out the following steps:

1. Compute the lower Cholesky factorization $\mathbf{K} = \mathbf{B}\mathbf{B}'$.
2. Generate $\mathbf{Z} = (Z_1, \ldots, Z_n)'$ by drawing $Z_1, \ldots, Z_n \sim \mathcal{N}(0, 1)$.
3. Return $\mathbf{U} = \boldsymbol{\mu} + (\mathbf{B}')^{-1}\mathbf{Z}$.
4. Repeat Steps 2 and 3 independently R times.

To check that $\mathbf{U} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}^{-1})$, we first note that $\mathbf{U}$ is an affine transformation of the normal random vector $\mathbf{Z}$, so it has a normal distribution. It is easy to check that $\mathbb{E}\mathbf{U} = \boldsymbol{\mu}$. The covariance matrix of $\mathbf{U}$ is

$$\mathrm{Cov}(\mathbf{U}) = (\mathbf{B}')^{-1}\mathrm{Cov}(\mathbf{Z})((\mathbf{B}')^{-1})' = (\mathbf{B}')^{-1}(\mathbf{B})^{-1} = (\mathbf{B}\mathbf{B}')^{-1} = \mathbf{K}^{-1}.$$

Hence, $\mathbf{U}$ has the desired distribution.

We make a few comments on the computations. As mentioned previously, to compute $\widehat{\boldsymbol{\tau}}$, one needs not compute the inverse $\mathbf{K}_{\boldsymbol{\tau}}^{-1}$. Instead, we solve the linear system

$$\mathbf{K}_{\boldsymbol{\tau}}\mathbf{z} = \frac{\tau_0}{\omega^2}\mathbf{H}'\mathbf{H}\mathbf{1}_T + \frac{1}{\sigma^2}\mathbf{y}$$

for $\mathbf{z}$. Since $\mathbf{K}_{\boldsymbol{\tau}}$ is a band matrix, its Cholesky factor is a band triangular matrix, Consequently, solving the above system involves only $\mathcal{O}(T)$ operations. We can similarly calculate $\boldsymbol{\mu} + (\mathbf{B}')^{-1}\mathbf{Z}$ without computing the inverse of $\mathbf{B}'$.

This type of precision-based algorithms using band matrix routines for drawing high-dimensional normal random vector is considered in Rue (2001) in the context of Gaussian Markov random fields. Chan and Jeliazkov (2009) and McCausland, Miller and Pelletier (2011) later derive similar algorithms for linear Gaussian state space models. We will call these algorithms **precision samplers**.

Step 2 and Step 3 can be easily implemented, as both $(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\tau}, \omega^2, \tau_0)$ and $(\omega^2 \,|\, \mathbf{y}, \boldsymbol{\tau}, \sigma^2, \tau_0)$ are inverse-gamma random variables. In particular, one can show that

$$(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\tau}, \omega^2, \tau_0) \sim \mathcal{IG}\left(\nu_{\sigma^2} + \frac{T}{2}, S_{\sigma^2} + \frac{1}{2}(\mathbf{y} - \boldsymbol{\tau})'(\mathbf{y} - \boldsymbol{\tau})\right)$$
$$(\omega^2 \,|\, \mathbf{y}, \boldsymbol{\tau}, \sigma^2, \tau_0) \sim \mathcal{IG}\left(\nu_{\omega^2} + \frac{T}{2}, S_{\omega^2} + \frac{1}{2}(\boldsymbol{\tau} - \tau_0 \mathbf{1}_T)'\mathbf{H}'\mathbf{H}(\boldsymbol{\tau} - \tau_0 \mathbf{1}_T)\right).$$

Finally, recall that τ_0 only appears in the first state equation

$$\tau_1 = \tau_0 + \eta_1,$$

where $\eta_1 \sim \mathcal{N}(0, \omega^2)$. Given the normal prior $\tau_0 \sim \mathcal{N}(a_0, b_0)$, we can again use standard linear regression results to get

$$(\tau_0 \,|\, \mathbf{y}, \boldsymbol{\tau}, \sigma^2, \omega^2) \sim \mathcal{N}(\widehat{\tau}_0, K_{\tau_0}^{-1}),$$

where

$$K_{\tau_0} = \frac{1}{b_0} + \frac{1}{\omega^2}, \quad \widehat{\tau}_0 = K_{\tau_0}^{-1}\left(\frac{a_0}{b_0} + \frac{\tau_1}{\omega^2}\right).$$

We summarize the the Gibbs sampler for the local level model as follows:

Algorithm 6.2. (Gibbs Sampler for the Local Level Model).

Pick some initial values $\sigma^{2(0)} = s_0 > 0$, $\omega^{2(0)} = o_0 > 0$ and $\tau_0 = t_0$. Then, repeat the following steps from $r = 1$ to R:

1. Draw $\boldsymbol{\tau}^{(r)} \sim p(\boldsymbol{\tau} \,|\, \mathbf{y}, \sigma^{2(r-1)}, \omega^{2(r-1)}, \tau_0^{(r-1)})$ (multivariate normal).
2. Draw $\sigma^{2(r)} \sim p(\sigma^2 \,|\, \mathbf{y}, \boldsymbol{\tau}^{(r)}, \omega^{2(r-1), \tau_0^{(r-1)}})$ (inverse-gamma).
3. Draw $\omega^{2(r)} \sim p(\omega^2 \,|\, \mathbf{y}, \boldsymbol{\tau}^{(r)}, \sigma^{2(r)}, \tau_0^{(r-1)})$ (inverse-gamma).
4. Draw $\tau_0^{(r)} \sim p(\tau_0 \,|\, \mathbf{y}, \boldsymbol{\tau}^{(r)}, \sigma^{2(r)}, \omega^{2(r)})$ (normal).

We next illustrate the above algorithm using an application that estimates the trend inflation.

6.1.2 Empirical Example: Estimating Trend Inflation

In this empirical example we decompose the US PCE inflation into its trend and cyclical components. Specifically, we fit the PCE data from 1959Q1 to 2015Q4 using the unobserved components model in (6.1)–(6.2).

As mentioned above, ω^2 controls the smoothness of the trend component, and here we elicit the hyperparameters to reflect the desired smoothness. In particular, we set $\nu_{\omega^2} = 3$ and $S_{\omega^2} = 2 \times 0.25^2$, so that the prior mean of ω^2 is 0.25^2. Hence, with high probability, the difference between consecutive trend terms is less than 0.5.

After loading the PCE dataset `USPCE_2015Q4.csv` as in Section 3.2.4, we implement Algorithm 6.2 using the following MATLAB script `UC.m`. Note that in MATLAB the backslash operator \ is used to solve linear system. For example, to solve $\mathbf{Az} = \mathbf{b}$ for $\mathbf{z}$, we can use the command

```
z = A \ b
```

```
% UC.m
    % prior
a0 = 5; b0 = 100;
nu_sig0 = 3; S_sig0 = 1*(nu_sig0-1);
nu_omega0 = 3; S_omega0 = .25^2*(nu_omega0-1);

    % initialize the Markov chain
sig2 = 1; omega2 = .1; tau0 = 5;

    % compute a few things outside the loop
H = speye(T) - sparse(2:T,1:(T-1),ones(1,T-1),T,T);
HH = H'*H;
HHiota = HH*ones(T,1);

for isim = 1:nsim+burnin
        % sample tau
    Ktau = HH/omega2 + speye(T)/sig2;
    Ctau = chol(Ktau,'lower');
    tau_hat = Ktau\(tau0/omega2*HHiota + y/sig2);
    tau = tau_hat + Ctau'\randn(T,1);

        % sample sig2
    sig2 = 1/gamrnd(nu_sig0 + T/2,1/(S_sig0 + (y-tau)'*(y-tau)/2));

        % sample omega2
```

```
    omega2 = 1/gamrnd(nu_omega0 + T/2, ...
        1/(S_omega0 + (tau-tau0)'*HH*(tau-tau0)/2));

        % sample tau0
    Ktau0 = 1/b0 + 1/omega2;
    tau0_hat = Ktau0\(a0/b0 + tau(1)/omega2);
    tau0 = tau0_hat + sqrt(Ktau0)'\randn;

    if isim>burnin
        i = isim-burnin;
        store_tau(i,:) = tau';
        store_theta(i,:) = [sig2 omega2 tau0];
    end
end
```

Figure 6.1 reports the posterior mean of τ_t based on a sample of 20000 posterior draws after a burn-in period of 1000. It is evident that the trend inflation is able to capture the low-frequency movements in the PCE inflation. In particular, it gradually increases from about 5% in early 1960s to over 10% in late 1970s. Then it exhibits a steady decline throughout the Great Moderation to between 2% and 3.5% after the Great Recession.

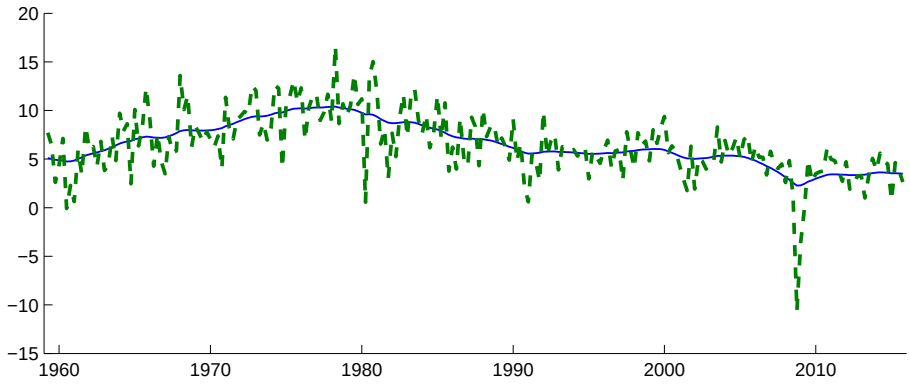

Figure 6.1: Estimated posterior mean of τ_t (solid line) and the PCE inflation (dotted line).

6.2 Application: Estimating Output Gap

We now look at a more substantive application of using an unobserved components model to measure the overall slack in the economy. One popular metric for this

purpose is the so-called **output gap**, defined as the deviation of the actual output of an economy from its potential or trend output. To estimate the output gap, we need a model for the trend output, and unobserved models provide a natural framework for this task.

There is a large literature of using unobserved components models to estimate the output gap. Prominent examples include Watson (1986), Clark (1987) and Morley, Nelson and Zivot (2003). Here we consider the unobserved components model considered in Grant and Chan (2017*b*), which nests the popular Hodrick-Prescott (HP) filter in Hodrick and Prescott (1980, 1997).

The trend-cycle decomposition of aggregate output is motivated by the idea that it can be usefully viewed as the sum of two separate components: a nonstationary component that represents the trend and a transitory deviation from the trend. Consider the following decomposition of the log real GDP y_t:

$$y_t = \tau_t + c_t, \tag{6.6}$$

where τ_t is the trend and c_t is the stationary, cyclical component.

The cyclical component c_t—which is our measure of the output gap—is modeled as a zero mean stationary AR(p) process:

$$c_t = \phi_1 c_{t-1} + \cdots + \phi_p c_{t-p} + u_t^c, \tag{6.7}$$

where $u_t^c \sim \mathcal{N}(0, \sigma_c^2)$ and the initial conditions are assumed to be zero: $c_0 = \cdots = c_{-1+p} = 0$.

Since the trend output is expected to be growing over time, one popular specification for the nonstationary trend τ_t is a random walk with drift:

$$\tau_t = \mu + \tau_{t-1} + \widetilde{u}_t^\tau.$$

Here the drift μ can be interpreted as the average growth rate of trend output. Since μ is a constant, this trend specification implies that the trend output is growing on average at a constant rate, which is not supported by the data. For instance, using US real GDP from 1947Q1 to 1998Q2, Perron and Wada (2009) find a break in trend output growth at 1973Q1. Using more recent data, Luo and Startz (2014) and Grant and Chan (2017*a*) find a break at 2006Q1 and 2007Q1, respectively.

Here we consider an alternative trend specification that has a time-varying growth rate:

$$\Delta\tau_t = \Delta\tau_{t-1} + u_t^\tau, \tag{6.8}$$

where Δ is the first difference operator such that $\Delta x_t = x_t - x_{t-1}$, $u_t^\tau \sim \mathcal{N}(0, \sigma_\tau^2)$ and τ_0 and τ_{-1} are treated as unknown parameters. Since $\Delta\tau_t$ can be interpreted as the trend growth rate at time t, this trend specification implies that the trend growth follows a random walk. As such, by construction this unobserved components model incorporates a stochastic trend growth rate.

6.2.1 Estimation

In this section we discuss the estimation of the unobserved components model defined by (6.6), (6.7) and (6.8). We follow Morley, Nelson and Zivot (2003) and set $p = 2$. The model parameters are $\boldsymbol{\phi} = (\phi_1, \phi_2)', \sigma_c^2, \sigma_\tau^2$ and $\boldsymbol{\gamma} = (\tau_0, \tau_{-1})'$. We assume the following independent priors:

$$\boldsymbol{\phi} \sim \mathcal{N}(\boldsymbol{\phi}_0, \mathbf{V}_{\boldsymbol{\phi}}) 1(\boldsymbol{\phi} \in \mathbf{R}), \quad \boldsymbol{\gamma} \sim \mathcal{N}(\boldsymbol{\gamma}_0, \mathbf{V}_{\boldsymbol{\gamma}}), \quad \sigma_c^2 \sim \mathcal{IG}(\nu_{\sigma_c^2}, S_{\sigma_c^2}), \quad \sigma_\tau^2 \sim \mathcal{U}(0, b_{\sigma_\tau^2}),$$

where R is the stationarity region. Here we consider a uniform prior for σ_τ^2 instead of the conventional inverse-gamma prior.

We can use the following 5-block posterior sampler to simulate from the joint posterior $p(\boldsymbol{\tau}, \boldsymbol{\phi}, \sigma^2, \omega^2, \boldsymbol{\gamma} \,|\, \mathbf{y})$:

1. sample $(\boldsymbol{\tau} \,|\, \mathbf{y}, \boldsymbol{\phi}, \sigma_c^2, \sigma_\tau^2, \boldsymbol{\gamma})$,
2. sample $(\boldsymbol{\phi} \,|\, \mathbf{y}, \boldsymbol{\tau}, \sigma_c^2, \sigma_\tau^2, \boldsymbol{\gamma})$,
3. sample $(\sigma_c^2 \,|\, \mathbf{y}, \boldsymbol{\tau}, \boldsymbol{\phi}, \sigma_\tau^2, \boldsymbol{\gamma})$,
4. sample $(\sigma_\tau^2 \,|\, \mathbf{y}, \boldsymbol{\tau}, \boldsymbol{\phi}, \sigma_c^2, \boldsymbol{\gamma})$,
5. sample $(\boldsymbol{\gamma} \,|\, \mathbf{y}, \boldsymbol{\tau}, \boldsymbol{\phi}, \sigma_c^2, \sigma_\tau^2)$.

Stack $\mathbf{y} = (y_1, \ldots, y_T)'$, and similarly define $\mathbf{c}$, $\mathbf{u}^c$ and $\mathbf{u}^\tau$. Then, rewrite the system (6.6), (6.7) and (6.8) in matrix notation:

$$\begin{aligned} \mathbf{y} &= \boldsymbol{\tau} + \mathbf{c}, \\ \mathbf{H}_{\boldsymbol{\phi}} \mathbf{c} &= \mathbf{u}^c, \\ \mathbf{H}_2 \boldsymbol{\tau} &= \widetilde{\boldsymbol{\alpha}}_\tau + \mathbf{u}^\tau, \end{aligned}$$

where $\widetilde{\boldsymbol{\alpha}}_\tau = (\tau_0 + \Delta\tau_0, -\tau_0, 0, \ldots, 0)'$ and

$$\mathbf{H}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{pmatrix}, \quad \mathbf{H}_{\boldsymbol{\phi}} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -\phi_1 & 1 & 0 & 0 & \cdots & 0 \\ -\phi_2 & -\phi_1 & 1 & 0 & \cdots & 0 \\ 0 & -\phi_2 & -\phi_1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\phi_2 & -\phi_1 & 1 \end{pmatrix}.$$

Both $\mathbf{H}_2$ and $\mathbf{H}_{\boldsymbol{\phi}}$ are band matrices with unit determinant, and therefore they are invertible. Hence, given the parameters, it follows that

$$\begin{aligned} (\mathbf{y} \,|\, \boldsymbol{\tau}, \boldsymbol{\phi}, \sigma_c^2) &\sim \mathcal{N}(\boldsymbol{\tau}, \sigma_c^2 (\mathbf{H}_{\boldsymbol{\phi}}' \mathbf{H}_{\boldsymbol{\phi}})^{-1}) \\ (\boldsymbol{\tau} \,|\, \boldsymbol{\gamma}, \sigma_\tau^2) &\sim \mathcal{N}(\boldsymbol{\alpha}_\tau, \sigma_\tau^2 (\mathbf{H}_2' \mathbf{H}_2)^{-1}), \end{aligned}$$

where $\boldsymbol{\alpha}_\tau = \mathbf{H}_2^{-1}\widetilde{\boldsymbol{\alpha}}_\tau = (\tau_0 + \Delta\tau_0, \tau_0 + 2\Delta\tau_0, \ldots, \tau_0 + T\Delta\tau_0)'$. Then, by standard linear regression results, we have

$$(\boldsymbol{\tau} \,|\, \mathbf{y}, \boldsymbol{\phi}, \sigma_c^2, \sigma_\tau^2, \boldsymbol{\gamma}) \sim \mathcal{N}(\widehat{\boldsymbol{\tau}}, \mathbf{K}_{\boldsymbol{\tau}}^{-1}), \tag{6.9}$$

where

$$\mathbf{K}_{\boldsymbol{\tau}} = \frac{1}{\sigma_\tau^2}\mathbf{H}_2'\mathbf{H}_2 + \frac{1}{\sigma_c^2}\mathbf{H}_{\boldsymbol{\phi}}'\mathbf{H}_{\boldsymbol{\phi}}, \quad \widehat{\boldsymbol{\tau}} = \mathbf{K}_{\boldsymbol{\tau}}^{-1}\left(\frac{1}{\sigma_\tau^2}\mathbf{H}_2'\mathbf{H}_2\boldsymbol{\alpha}_\tau + \frac{1}{\sigma_c^2}\mathbf{H}_{\boldsymbol{\phi}}'\mathbf{H}_{\boldsymbol{\phi}}\mathbf{y}\right).$$

Since $\mathbf{K}_{\boldsymbol{\tau}}$ is a band precision matrix, we can sample $\boldsymbol{\tau}$ efficiently using the precision sampler as described in Algorithm 6.1.

To sample $\boldsymbol{\phi}$ in Step 2, we write (6.7) as

$$\mathbf{c} = \mathbf{X}_{\boldsymbol{\phi}}\boldsymbol{\phi} + \mathbf{u}^c,$$

where $\mathbf{X}_{\boldsymbol{\phi}}$ is a $T \times 2$ matrix consisting of lagged values of c_t:

$$\mathbf{X}_{\boldsymbol{\phi}} = \begin{pmatrix} c_0 & c_{-1} \\ c_1 & c_0 \\ \vdots & \vdots \\ c_{T-1} & c_{T-2} \end{pmatrix}.$$

Again, using standard regression results, we obtain

$$(\boldsymbol{\phi} \,|\, \mathbf{y}, \boldsymbol{\tau}, \sigma_c^2, \sigma_\tau^2, \boldsymbol{\gamma}) \sim \mathcal{N}(\widehat{\boldsymbol{\phi}}, \mathbf{K}_{\boldsymbol{\phi}}^{-1})1(\boldsymbol{\phi} \in R),$$

where

$$\mathbf{K}_{\boldsymbol{\phi}} = \mathbf{V}_{\boldsymbol{\phi}}^{-1} + \frac{1}{\sigma_c^2}\mathbf{X}_{\boldsymbol{\phi}}'\mathbf{X}_{\boldsymbol{\phi}}, \quad \widehat{\boldsymbol{\phi}} = \mathbf{K}_{\boldsymbol{\phi}}^{-1}\left(\mathbf{V}_{\boldsymbol{\phi}}^{-1}\boldsymbol{\phi}_0 + \frac{1}{\sigma_c^2}\mathbf{X}_{\boldsymbol{\phi}}'\mathbf{c}\right).$$

A draw from this truncated normal distribution can be obtained by the acceptance-rejection method, i.e., we keep sampling from $\mathcal{N}(\widehat{\boldsymbol{\phi}}, \mathbf{K}_{\boldsymbol{\phi}}^{-1})$ until $\boldsymbol{\phi} \in R$.

Step 3 can be easily implemented as the full conditional distribution of σ_c^2 is inverse-gamma:

$$(\sigma_c^2 \,|\, \mathbf{y}, \boldsymbol{\tau}, \boldsymbol{\phi}, \sigma_\tau^2, \boldsymbol{\gamma}) \sim \mathcal{IG}\left(\nu_{\sigma_c^2} + \frac{T}{2}, S_{\sigma_c^2} + \frac{1}{2}(\mathbf{c} - \mathbf{X}_{\boldsymbol{\phi}}\boldsymbol{\phi})'(\mathbf{c} - \mathbf{X}_{\boldsymbol{\phi}}\boldsymbol{\phi})\right).$$

Next, to implement Step 4, notice that the conditional density of σ_τ^2 is given by

$$p(\sigma_\tau^2 \,|\, \mathbf{y}, \boldsymbol{\tau}, \boldsymbol{\phi}, \sigma_c^2, \boldsymbol{\gamma}) \propto (\sigma_\tau^2)^{-\frac{T}{2}} \mathrm{e}^{-\frac{1}{2\sigma_\tau^2}\sum_{t=1}^T (\Delta\tau_t - \Delta\tau_{t-1})^2} \mathbb{1}(0 < \sigma_\tau^2 < b_{\sigma_\tau^2}),$$

which is not a standard density. However, we can sample from it using a Griddy-Gibbs step as described in Algorithm 3.3.

Lastly, to sample $\boldsymbol{\gamma} = (\tau_0, \tau_{-1})'$, first note that

$$\boldsymbol{\alpha}_\tau = \begin{pmatrix} \tau_0 + \Delta\tau_0 \\ \tau_0 + 2\Delta\tau_0 \\ \cdots \\ \tau_0 + T\Delta\tau_0 \end{pmatrix} = \mathbf{X}_{\boldsymbol{\gamma}}\boldsymbol{\gamma},$$

where

$$\mathbf{X}_{\boldsymbol{\gamma}} = \begin{pmatrix} 2 & -1 \\ 3 & -2 \\ \vdots & \vdots \\ T+1 & -T \end{pmatrix}.$$

Hence, we have

$$\boldsymbol{\tau} = \mathbf{X}_{\boldsymbol{\gamma}}\boldsymbol{\gamma} + \mathbf{H}_2^{-1}\mathbf{u}^\tau$$

with $\mathbf{H}_2^{-1}\mathbf{u}^\tau \sim \mathcal{N}(\mathbf{0}, \sigma_\tau^2(\mathbf{H}_2'\mathbf{H}_2)^{-1})$. Then, by standard regression results, we have

$$(\boldsymbol{\gamma} \,|\, \mathbf{y}, \boldsymbol{\tau}, \boldsymbol{\phi}, \sigma_c^2, \sigma_\tau^2) \sim \mathcal{N}(\widehat{\boldsymbol{\gamma}}, \mathbf{K}_\gamma^{-1}),$$

where

$$\mathbf{K}_\gamma = \mathbf{V}_\gamma^{-1} + \frac{1}{\sigma_\tau^2}\mathbf{X}_\gamma'\mathbf{H}_2'\mathbf{H}_2\mathbf{X}_\gamma, \quad \widehat{\boldsymbol{\gamma}} = \mathbf{K}_\gamma^{-1}\left(\mathbf{V}_\gamma^{-1}\boldsymbol{\gamma}_0 + \frac{1}{\sigma_\tau^2}\mathbf{X}_\gamma'\mathbf{H}_2'\mathbf{H}_2\boldsymbol{\tau}\right).$$

6.2.2 Empirical Results

In this section we report US output gap estimates obtained from the unobserved components model discussed in the previous section. The dataset consists of US real GDP from 1947Q1 to 2015Q4. We assume relatively large prior variances with $\mathbf{V}_{\boldsymbol{\phi}} = \mathbf{I}_2$ and $\mathbf{V}_{\boldsymbol{\gamma}} = 100\mathbf{I}_2$. For the prior means, we set $\boldsymbol{\phi}_0 = (1.3, -0.7)'$ and $\boldsymbol{\gamma}_0 = (750, 750)'$. In particular, these values imply that the AR(2) process of the transitory component has two complex roots. Furthermore, we assume the uppper bound of the uniform prior on σ_τ^2 to be 0.01.

The following MATLAB script `UC_output_gap.m` implements the Gibbs sampler discussed in last section.

```
% UC_output_gap.m
nsim = 10000;  burnin = 1000;
data_raw = load('USGDP_2015Q4.csv');
data = 100*log(data_raw);
y = data;  T = length(y);
```

```
    % prior
a0 = [750;750]; B0 = 100*eye(2);
phi0 = [1.3 -.7]'; iVphi = speye(2);
nu_sigc2 = 3; S_sigc2 = 1*(nu_sigc2-1);
sigtau2_ub = .01;

    % initialize for storeage
store_theta = zeros(nsim,6); % [phi, sigc2, sigtau2, tau0]
store_tau = zeros(nsim,T);
store_mu = zeros(nsim,T);    % annualized trend growth

    % initialize the Markov chain
phi = [1.34 -.7]';
tau0 = [y(1) y(1)]'; % [tau_{0}, tau_{-1}]
sigc2 = .5;  sigtau2 = .001;

    % construct a few things
H2 = speye(T) - 2*sparse(2:T,1:(T-1),ones(1,T-1),T,T) ...
    + sparse(3:T,1:(T-2),ones(1,T-2),T,T);
H2H2 = H2'*H2;
Hphi = speye(T) - phi(1)*sparse(2:T,1:(T-1),ones(1,T-1),T,T) + ...
    - phi(2)*sparse(3:T,1:(T-2),ones(1,T-2),T,T);
Xtau0 = [(2:T+1)' -(1:T)'];
n_grid = 500;  count_phi = 0;

for isim = 1:nsim+burnin
        % sample tau
    alp_tau = H2\[2*tau0(1)-tau0(2);-tau0(1);sparse(T-2,1)];
    Ktau = H2H2/sigtau2 + Hphi'*Hphi/sigc2;
    tau_hat = Ktau\(H2H2*alp_tau/sigtau2 + Hphi'*Hphi*y/sigc2);
    tau = tau_hat + chol(Ktau,'lower')'\randn(T,1);

        % sample phi
    c = y-tau;
    Xphi = [[0;c(1:T-1)] [0;0;c(1:T-2)]];
    Kphi = iVphi + Xphi'*Xphi/sigc2;
    phi_hat = Kphi\(iVphi*phi0 + Xphi'*c/sigc2);
    phic = phi_hat + chol(Kphi,'lower')'\randn(2,1);
    if sum(phic) < .99 && phic(2) - phic(1) < .99 && phic(2) > -.99
        phi = phic;
        Hphi = speye(T) - phi(1)*sparse(2:T,1:(T-1),ones(1,T-1),T,T)...
            - phi(2)*sparse(3:T,1:(T-2),ones(1,T-2),T,T);
        count_phi = count_phi + 1;
```

```
    end

        % sample sigc2
    sigc2 = 1/gamrnd(nu_sigc2 + T/2,1/(S_sigc2 ...
        + (c-Xphi*phi)'*(c-Xphi*phi)/2));

        % sample sigtau2
    del_tau = [tau0(1);tau(1:T)] - [tau0(2);tau0(1);tau(1:T-1)];
    f_tau = @(x) -T/2*log(x) ...
        - sum((del_tau(2:T) - del_tau(1:T-1)).^2)./(2*x);
    sigtau2_grid = linspace(rand/1000,sigtau2_ub-rand/1000,n_grid);
    lp_sigtau2 = f_tau(sigtau2_grid);
    p_sigtau2 = exp(lp_sigtau2-max(lp_sigtau2));
    p_sigtau2 = p_sigtau2/sum(p_sigtau2);
    cdf_sigtau2 = cumsum(p_sigtau2);
    sigtau2 = sigtau2_grid(find(rand<cdf_sigtau2, 1 ));

        % sample tau0
    Ktau0 = B0\speye(2) + Xtau0'*H2H2*Xtau0/sigtau2;
    tau0_hat = Ktau0\(B0\a0 + Xtau0'*H2H2*tau/sigtau2);
    tau0 = tau0_hat + chol(Ktau0,'lower')'\randn(2,1);

    if isim > burnin
        i = isim-burnin;
        store_tau(i,:) = tau';
        store_theta(i,:) = [phi' sigc2 sigtau2 tau0'];
        store_mu(i,:) = 4*(tau-[tau0(1);tau(1:end-1)])';
    end
end
```

Figure 6.2 reports the output gap estimates $\{c_t\}$ over the whole sample. The negative estimates generally coincide with the NBER recession dates, and they are especially large in magnitude during the recessions in the 1940s, 1960s and 1980s. In contrast, the trough of the Great Recession is only is about -3.1%. Moreover, the output gap has essentially closed at the end of the sample.

Next, we present the estimated annualized trend output growth rate—the posterior means of $4\Delta\tau_t = 4(\tau_t - \tau_{t-1})$—in Figure 6.3. It is clear that there is substantial time variation in trend growth over the past six decades. Specifically, the annualized trend growth rate fluctuates between 3.5% and 4% from the beginning of the sample until 1970. It then begins a steady decline and reaches about 3% in mid-1970s. The estimated trend growth rate remains stable at about 3% from the mid-1970s until 2000, when it starts another gradual decline to about 1.6% in the middle of the

Great Recession.

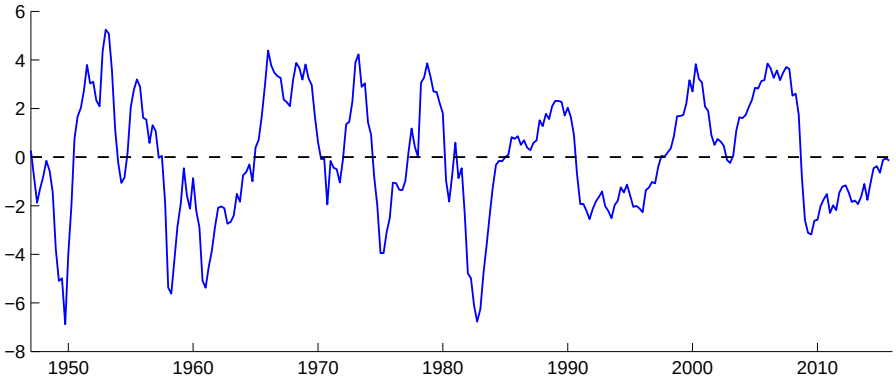

Figure 6.2: The output gap estimates.

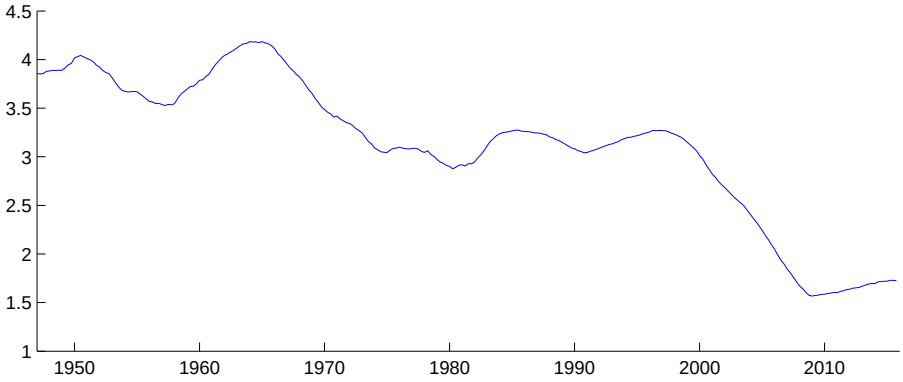

Figure 6.3: Estimates of the annualized growth in trend output $4\Delta\tau_t$.

6.3 Noncentered Parameterization

In this section we extend the local level model in Section 6.1 to allow for a time-varying drift. Specifically, consider the following dynamic linear trend model:

$$y_t = \tau_t + \varepsilon_t, \tag{6.10}$$

where $\{\varepsilon_t\}$ are assumed to be iid $\mathcal{N}(0, \sigma_y^2)$ and the trend component τ_t is modeled as a random walk with a time-varying drift:

$$\tau_t = \tau_{t-1} + \alpha_t + u_t^\tau, \tag{6.11}$$

$$\alpha_t = \alpha_{t-1} + u_t^\alpha, \tag{6.12}$$

where $\{u_t^\tau\}$ and $\{u_t^\alpha\}$ are iid $\mathcal{N}(0, \sigma_\tau^2)$ and $\mathcal{N}(0, \sigma_\alpha^2)$, respectively, and the initial conditions τ_0 and α_0 are treated as unknown parameters. Since the drift α_t can be interpreted as the slope of the trend and it is time-varying, the above model is sometimes called the **local slope model**.

It is often of interest to test if the drift α_t is time-varying or constant. Formally, a constant drift model is a restricted version of the above model with $\sigma_\alpha^2 = 0$, under which $\alpha_1 = \cdots = \alpha_T = \alpha_0$. If we can ignore the technical difficulty that 0 is at the boundary of the parameter space of σ_α^2, the Bayes factor in favor of the unrestricted model against the restricted version with $\sigma_\alpha^2 = 0$ can be obtained using the Savage–Dickey density ratio (see Section 4.1.1)

$$\text{BF}_{\text{UR}} = \frac{p(\sigma_\alpha^2 = 0)}{p(\sigma_\alpha^2 = 0 \,|\, \mathbf{y})},$$

where the numerator is the marginal prior density of σ_α^2 evaluated at 0 and the denominator is the marginal posterior evaluated at 0. Intuitively, if σ_α^2 is more unlikely to be zero under the posterior density relative to the prior density, then it is viewed as evidence in favor of the time-varying model. Hence, to compute the relevant Bayes factor, one only needs to evaluate two univariate densities at a point, which is often easy to do.

However, this easy approach cannot be directly applied in our setting due to two related issues. First, as mentioned, the value 0 is at the boundary of the parameter space of σ_α^2—therefore the Savage-Dickey density ratio approach is not applicable. Second, the conventional inverse-gamma prior for σ_α^2 has zero density at zero.

Here we introduce the noncentered parameterization of Frühwirth-Schnatter and Wagner (2010) to overcome these technical difficulties. Instead of the error variance σ_α^2, we work with the standard derivation σ_α that is *defined* to have its support on the whole real line. Using the noncentered parameterization to perform specification tests for models with time-varying parameters is considered in Chan (2016), and we closely follow the discussion there.

Now, consider the following unobserved components model in the noncentered parameterization:

$$y_t = \tau_0 + \sigma_\tau \widetilde{\tau}_t + t\alpha_0 + \sigma_\alpha \sum_{s=1}^{t} \widetilde{\alpha}_s + \varepsilon_t, \tag{6.13}$$

$$\widetilde{\tau}_t = \widetilde{\tau}_{t-1} + \widetilde{u}_t^\tau, \tag{6.14}$$

$$\widetilde{\alpha}_t = \widetilde{\alpha}_{t-1} + \widetilde{u}_t^\alpha, \tag{6.15}$$

where $\{\widetilde{u}_t^\tau\}$ and $\{\widetilde{u}_t^\alpha\}$ are mutually independent $\mathcal{N}(0, 1)$ random variables and $\widetilde{\tau}_0 = \widetilde{\alpha}_0 = 0$. To see that it is a reparameterization of the model in (6.10), (6.11) and

(6.12), let

$$\alpha_t = \alpha_0 + \sigma_\alpha \widetilde{\alpha}_t \tag{6.16}$$

$$\tau_t = \tau_0 + \sigma_\tau \widetilde{\tau}_t + t\alpha_0 + \sigma_\alpha \sum_{s=1}^{t} \widetilde{\alpha}_s. \tag{6.17}$$

Then, it is clear that $y_t = \tau_t + \varepsilon_t$, and

$$\begin{aligned} \alpha_t - \alpha_{t-1} &= \sigma_\alpha(\widetilde{\alpha}_t - \widetilde{\alpha}_{t-1}) = \sigma_\alpha \widetilde{u}_t^\alpha \\ \tau_t - \tau_{t-1} &= \alpha_0 + \sigma_\alpha \widetilde{\alpha}_t + \sigma_\tau(\widetilde{\tau}_t - \widetilde{\tau}_{t-1}) \\ &= \alpha_t + \sigma_\tau \widetilde{u}_t^\tau. \end{aligned}$$

Hence, it becomes the original model in (6.10)–(6.12).

Frühwirth-Schnatter and Wagner (2010) consider a normal prior centered at 0 for σ_α: $\sigma_\alpha \sim \mathcal{N}(0, V_{\sigma_\alpha})$. This normal prior on the standard deviation σ_α has two main advantages over the usual inverse-gamma prior on the variance σ_α^2.

First, by a change of variable (see, e.g., Kroese and Chan, 2014, Section 3.5), we can show that the implied prior for σ_α^2 is $\mathcal{G}(\frac{1}{2}, \frac{1}{2V_{\sigma_\alpha}})$, where $\mathcal{G}(a, b)$ denotes the Gamma distribution with mean a/b. Compared to the conventional inverse-gamma prior, this gamma prior has more mass concentrated around small values. Hence, it provides shrinkage—*a priori* it favors the more parsimonious constant-coefficient model. The second advantage is that it is a conjugate prior for σ_α under the noncentered parameterization—it therefore facilitates computation.

The parameter σ_α has support on the real line, but its sign is not identified—its normal prior is symmetric around 0 and changing both the signs of σ_α and $\widetilde{\alpha}_t$ does not alter the likelihood value. Consequently, the posterior distribution of σ_α can be bimodal, and care needs to be taken to sample from the posterior distribution.

6.3.1 Estimation

In this section we discuss the estimation of the local slope model in the noncentered parameterization in (6.13), (6.14) and (6.15). The model parameters are $\boldsymbol{\beta} = (\tau_0, \alpha_0, \sigma_\tau, \sigma_\alpha)'$ and σ_y^2. For the Savage-Dickey density ratio identity to hold, the priors under the restricted and unrestricted models need to satisfy a certain condition. Here we assume a sufficient condition that the restricted and unrestricted parameters are independent *a priori*. In particular, we assume the following independent priors:

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \mathbf{V}_{\boldsymbol{\beta}}), \quad \sigma_y^2 \sim \mathcal{IG}(\nu_{\sigma_y^2}, S_{\sigma_y^2}),$$

where the third and fourth elements of $\boldsymbol{\beta}_0$ are zero and $\mathbf{V}_{\boldsymbol{\beta}}$ is a diagonal matrix. Then, the Gibbs sampler consists of the following steps:

1. sample $(\widetilde{\boldsymbol{\tau}}, \widetilde{\boldsymbol{\alpha}} \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma_y^2)$,
2. sample $(\boldsymbol{\beta} \,|\, \mathbf{y}, \widetilde{\boldsymbol{\tau}}, \widetilde{\boldsymbol{\alpha}}, \sigma_y^2)$,
3. randomly permute the signs of $(\widetilde{\boldsymbol{\tau}}, \widetilde{\boldsymbol{\alpha}})$ and $(\sigma_\tau, \sigma_\alpha)$,
4. sample $(\sigma_y^2 \,|\, \mathbf{y}, \widetilde{\boldsymbol{\tau}}, \widetilde{\boldsymbol{\alpha}}, \boldsymbol{\beta})$.

As mentioned before, the signs of σ_τ and σ_α are not identified, and the corresponding posterior distributions can be bimodal. Step 3 improves the mixing of the Markov chain by randomly permuting the signs of $(\widetilde{\boldsymbol{\tau}}, \widetilde{\boldsymbol{\alpha}})$ and $(\sigma_\tau, \sigma_\alpha)$.

One complication of implementing the first step is that $\widetilde{\alpha}_t$ enters the observation equation (6.13) as the sum $\sum_{s=1}^t \widetilde{\alpha}_s$. To simplify computations, let $\widetilde{A}_t = \sum_{s=1}^t \widetilde{\alpha}_s$. Next, we derive the prior of $\widetilde{\mathbf{A}} = (\widetilde{A}_1, \dots, \widetilde{A}_T)'$ implied by (6.15).

First recall that (6.15) implies $\mathbf{H}\widetilde{\boldsymbol{\alpha}} = \widetilde{\mathbf{u}}^\alpha$, where $\widetilde{\mathbf{u}}^\alpha \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_T)$ and $\mathbf{H}$ is the first difference matrix as before. Since $\widetilde{A}_1 = \widetilde{\alpha}_1$ and $\widetilde{A}_t - \widetilde{A}_{t-1} = \widetilde{\alpha}_t$, we have $\mathbf{H}\widetilde{\mathbf{A}} = \widetilde{\boldsymbol{\alpha}}$. Consequently, $\widetilde{\mathbf{A}} = \mathbf{H}^{-1}\widetilde{\boldsymbol{\alpha}} = \mathbf{H}^{-2}\widetilde{\mathbf{u}}^\alpha$, and therefore

$$\widetilde{\mathbf{A}} \sim \mathcal{N}(\mathbf{0}, (\mathbf{H}_2'\mathbf{H}_2)^{-1}),$$

where we have used the fact that $\mathbf{H}^2 = \mathbf{H}_2$.

Now, to sample $\widetilde{\boldsymbol{\tau}}$ and $\widetilde{\mathbf{A}}$ jointly, we write (6.13) in terms of $\boldsymbol{\gamma} = (\widetilde{\boldsymbol{\tau}}', \widetilde{\mathbf{A}}')'$:

$$\mathbf{y} = \tau_0 \mathbf{1}_T + \alpha_0 \mathbf{1}_{1:T} + \mathbf{X}_{\boldsymbol{\gamma}}\boldsymbol{\gamma} + \boldsymbol{\varepsilon},$$

where $\mathbf{1}_{1:T} = (1, 2, 3, \dots, T)'$ and $\mathbf{X}_{\boldsymbol{\gamma}} = (\sigma_\tau \mathbf{I}_T, \sigma_\alpha \mathbf{I}_T)$. Note that $\mathbf{X}_{\boldsymbol{\gamma}}$ is a sparse but not a band matrix. Hence, computations involving $\mathbf{X}_{\boldsymbol{\gamma}}$, though still fast, would not be as fast as those involving band matrices.

Next we derive the prior on $\boldsymbol{\gamma}$. Note that (6.14) implies $\widetilde{\boldsymbol{\tau}} \sim \mathcal{N}(\mathbf{0}, (\mathbf{H}'\mathbf{H})^{-1})$. Combining the prior on $\widetilde{\mathbf{A}}$ and using the assumption that $\widetilde{\boldsymbol{\tau}}$ and $\widetilde{\mathbf{A}}$ are independent, we have

$$\boldsymbol{\gamma} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{\boldsymbol{\gamma}}^{-1}),$$

where $\mathbf{P}_{\boldsymbol{\gamma}} = \operatorname{diag}(\mathbf{H}'\mathbf{H}, \mathbf{H}_2'\mathbf{H}_2)$. It then follows from standard linear regression result that

$$(\boldsymbol{\gamma} \,|\, \mathbf{y}, \boldsymbol{\beta}, \sigma_y^2) \sim \mathcal{N}(\widehat{\boldsymbol{\gamma}}, \mathbf{K}_{\boldsymbol{\gamma}}^{-1}),$$

where

$$\mathbf{K}_{\boldsymbol{\gamma}} = \mathbf{P}_{\boldsymbol{\gamma}} + \frac{1}{\sigma_y^2}\mathbf{X}_{\boldsymbol{\gamma}}'\mathbf{X}_{\boldsymbol{\gamma}}, \quad \widehat{\boldsymbol{\gamma}} = \mathbf{K}_{\boldsymbol{\gamma}}^{-1}\left(\frac{1}{\sigma_y^2}\mathbf{X}_{\boldsymbol{\gamma}}'(\mathbf{y} - \tau_0\mathbf{1}_T - \alpha_0\mathbf{1}_{1:T})\right).$$

As noted above, $\mathbf{X}_{\boldsymbol{\gamma}}$ is a sparse but not a band matrix. Same as $\mathbf{P}_{\boldsymbol{\gamma}}$. Regardless, we can sample $\boldsymbol{\gamma}$ using the precision sampler in Algorithm 6.1 as before. The other steps are standard and we will leave their derivations as an exercise. Finally, given the posterior draws of $\widetilde{\boldsymbol{\tau}}$ and $\widetilde{\boldsymbol{\alpha}}$ we can recover $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}$ using equations (6.16) and (6.17).

6.3.2 Simulated Example

In this section we provide a simulated example to illustrate the estimation methods discussed in the previous section. In particular, we generate a dataset of $T = 250$ observations with $\sigma_\alpha = 0.1$ and $\sigma_\tau = 0$. We run the following MATLAB script `UC_noncen_sim.m` to obtain 20000 posterior draws for analysis.

```
% UC_noncen_sim.m
nsim = 20000; burnin = 1000;

% generate data
T = 250;
randn('seed',123456); rand('seed',789012);
alp0 = .1; tau0 = 3; sigy2 = .5;
sigalp = .1; sigtau = 0;
H = speye(T) - sparse(2:T,1:(T-1),ones(1,T-1),T,T);
HH = H'*H;
alp_tilde = chol(HH,'lower')'\randn(T,1);
tau_tilde = chol(HH,'lower')'\randn(T,1);
alp = alp0 + sigalp*alp_tilde;
tau = tau0 + sigtau*tau_tilde + (1:T)'*alp0 ...
    + sigalp*(H\alp_tilde);
y = tau + sqrt(sigy2)*randn(T,1);
truealp = alp; truetau = tau;

    % initialize for storage
store_tau = zeros(nsim,T);
store_alp = zeros(nsim,T);
store_theta = zeros(nsim,5);

    % prior
beta0 = zeros(4,1); iVbeta = diag([1 1 .1 .1]);
nu_sig0 = 3; S_sig0 = 1*(nu_sig0-1);

% compute a few things outside the loop
HH = H'*H;
H2 = H*H;
H2H2 = H2'*H2;
Pgam = [HH sparse(T,T); sparse(T,T) H2H2];

    % initialize the Markov chain
sigy2 = var(y);
alp0 = 0; tau0 = 0;
```

```
sigtau = .1; sigalp = .1;

for isim = 1:nsim+burnin
        % sample alp and tau
    Xgam = [sigtau*speye(T) sigalp*speye(T)];
    Kgam = Pgam + Xgam'*Xgam/sigy2;
    gam_hat = Kgam\(1/sigy2*Xgam'*(y-tau0-(1:T)'*alp0));
    gam = gam_hat + chol(Kgam,'lower')'\randn(2*T,1);

        % sample beta
    X = [ones(T,1) (1:T)' gam(1:T) gam(T+1:end)];
    beta = sample_beta(y,X,sigy2,beta0,iVbeta);
    tau0 = beta(1); alp0 = beta(2); sigtau = beta(3); sigalp = beta(4);

        % permutate the signs of gam and (sigtau, sigalp)
    U = -1 + 2*(rand>0.5);
    gam = U*gam;
    sigalp = U*sigalp;
    sigtau = U*sigtau;

        % compute tau and alp
    tau_tilde = gam(1:T);
    A_tilde = gam(T+1:end);
    alp_tilde = H*A_tilde;
    alp = alp0 + sigalp*alp_tilde;
    tau = tau0 + sigtau*tau_tilde + (1:T)'*alp0 ...
    + sigalp*(H\alp_tilde);

        % sample sig2
    sigy2 = 1/gamrnd(nu_sig0 + T/2,1/(S_sig0 + (y-tau)'*(y-tau)/2));

    if isim>burnin
        isave = isim-burnin;
        store_tau(isave,:) = tau';
        store_alp(isave,:) = alp';
        store_theta(isave,:) = [beta' sigy2];
    end
end
```

Figure 6.4 illustrates the sparsity pattern of the precision matrix $\mathbf{K}_\gamma$: the zero elements are white while the nonzeros ones are blue. As is evident from the figure, the precision matrix $\mathbf{K}_\gamma$ has mostly zero elements: the 500×500 matrix has only 2492 elements. However, this matrix is not banded.

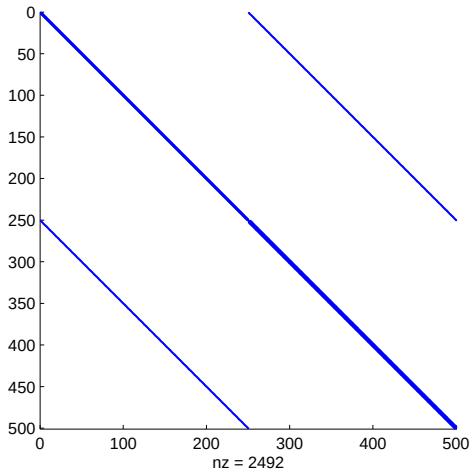

Figure 6.4: The sparsity pattern of the precision matrix $\mathbf{K}_{\boldsymbol{\gamma}}$.

Figure 6.5 plots the histograms of the posterior draws of σ_τ and σ_α. The posterior density of σ_α is clearly bimodal, where the two modes are 0.1 and -0.1, compared to the true value of 0.1. The posterior density of σ_τ is bimodal as well, but it has much more mass around zero. In fact, the estimated Savage-Dickey density ratio $p(\sigma_\tau = 0)/p(\sigma_\tau = 0 \,|\, \mathbf{y})$ is 0.079, implying a Bayes factor of 12.66 in favor of the restricted model with $\sigma_\tau = 0$.

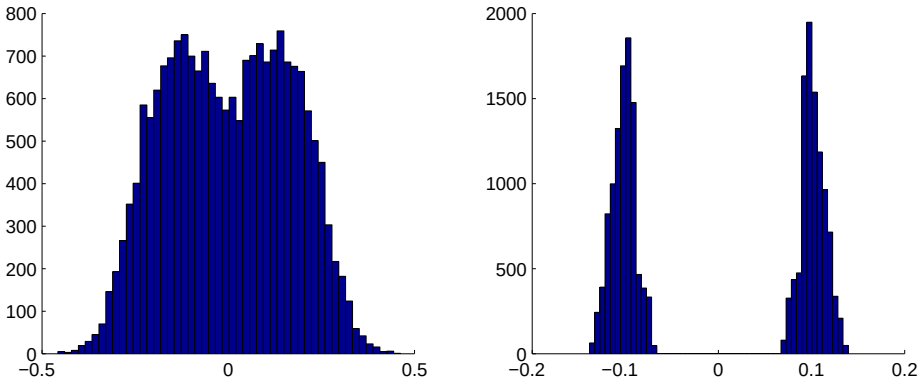

Figure 6.5: Histograms of the posterior draws of σ_τ (left panel) and σ_α (right panel).

Chapter 7

Stochastic Volatility Models

In this chapter we study a class of *nonlinear* state space models called **stochastic volatility models** in which the variance is time-varying. Unlike Generalized Autoregressive Conditional Heteroscedastic (GARCH) models where the the variance process is a deterministic function of past observations and parameters, in stochastic volatility model the variance process is, well, stochastic.

In this and later chapters we will study a variety of stochastic volatility models and illustrate the methods using financial and macroeconomic data. We will start with a plain vanilla stochastic volatility model that is an important building block of more sophisticated models.

7.1 Basic Stochastic Volatility Model

In macroeconomic and financial time series, it is often observed that large changes in observations tend to be followed by large changes, while small changes are followed by small changes—the phenomenon that is referred to as *volatility clustering*. For example, during financial crises movements in financial asset returns tend be large (of either sign), whereas in "normal periods" the same asset returns might exhibit little time variation.

Figure 7.1 depicts the AUD/USD daily returns from January 2005 to December 2012. For a long stretch of time from early 2005 to mid-2007, the daily returns mostly fluctuate between $\pm 2\%$. However, during the Great Recession of 2007-2008, the volatility of the daily returns increases dramatically—often reaching as high as $\pm 4\%$, sometimes even larger.

Models that assume a constant variance, by definition, cannot accommodate time-

varying volatility, and therefore cannot model volatility clustering that is a prominent feature in a wide range of macroeconomic and financial data. Below we focus our discussion on modeling the variance of the time series, and we assume for the moment that the observations have a zero mean; a constant mean or a suitable conditional mean process such as an AR(p) component can be added later on, as is done in the empirical application in Section 7.2.

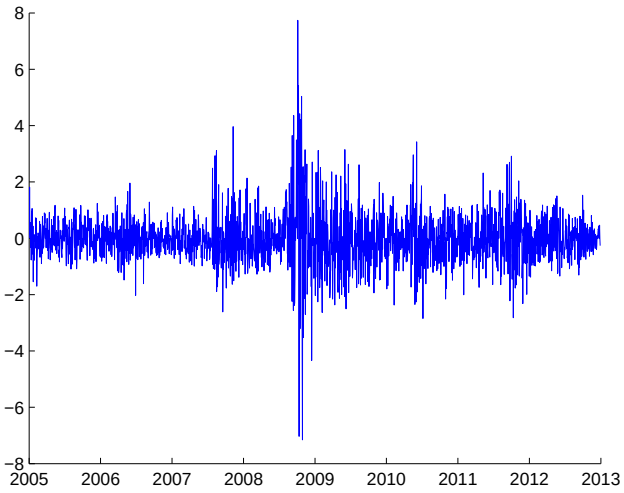

Figure 7.1: AUD/USD daily returns from January 2005 to December 2012.

Under the **stochastic volatility model**, the observation at time t is given by

$$y_t = \mathrm{e}^{\frac{1}{2}h_t}\varepsilon_t, \tag{7.1}$$

where $\varepsilon_t \sim \mathcal{N}(0,1)$. It follows that the conditional variance of y_t given h_t is $\mathrm{Var}(y_t \,|\, h_t) = \mathrm{e}^{h_t}$—hence, the conditinoal variance is time-varying. The state h_t is often called the **log-volatility**. Here we assume the states $\{h_t\}$ evolve according to a random walk process

$$h_t = h_{t-1} + u_t, \tag{7.2}$$

where $u_t \sim \mathcal{N}(0, \sigma_h^2)$ and is independent of ε_t at all leads and lags. The initial condition h_0 is treated as an unknown parameter. Instead of the random walk process considered in (7.2), in many finance applications a stationary AR(1) process is used instead.

This stochastic volatility model is an example of a *nonlinear* state space model where the observation equation (7.1) is nonlinear in the state. The major challenge of estimating this nonlinear model is that the joint conditional density of the states $\mathbf{h} = (h_1, \ldots, h_T)'$ given the model parameters and the data is high-dimensional

and nonstandard—in contrast to a linear Gaussian state space model where the conditional density of the states is Gaussian. Consequently, Bayesian estimation using MCMC becomes more difficult. In next section we introduce an efficient algorithm to sample $\mathbf{h}$ jointly.

7.1.1 Auxiliary Mixture Sampler

In this section we discuss the so-called **auxiliary mixture sampler** of Kim, Shephard and Chib (1998) for estimating the stochastic volatility model in (7.1)–(7.2). The idea is to approximate the nonlinear stochastic volatility model using a mixture of linear Gaussian models, where estimation of the latter models is standard.

Specifically, we first transform the observation y_t so that the resulting observation equation becomes *linear* in the log-volatility h_t. More precisely, we square both sides of (7.1) and take the logarithm:

$$y_t^* = h_t + \varepsilon_t^*, \tag{7.3}$$

where $y_t^* = \log y_t^2$ and $\varepsilon_t^* = \log \varepsilon_t^2$. In practice, we often set $y_t^* = \log(y_t^2 + c)$ for some small constant c, say, $c = 10^{-4}$, to avoid numerical problems when y_t is close to zero.

Then, (7.2) and (7.3) define a linear state space model in h_t. Note, however, that the error ε_t^* no longer has a Gaussian distribution—in fact, it follows a log-χ_1^2 distribution—and the machinery for fitting linear Gaussian state space models cannot be directly applied.

To tackle this difficulty, the second step of the auxiliary mixture sampling approach is to obtain an appropriate Gaussian mixture that well approximates the density function of ε_t^*, denoted as $f(\varepsilon_t^*)$. More precisely, consider the following n-component Gaussian mixture:

$$f(\varepsilon_t^*) \approx \sum_{i=1}^{n} p_i \, \varphi(\varepsilon_t^* \,;\, \mu_i, \sigma_i^2), \tag{7.4}$$

where $\varphi(x \,;\, \mu, \sigma^2)$ denotes the Gaussian density with mean μ and variance σ^2 and p_i is the probability of the i-th mixture component.

We can equivalently write the mixture density in (7.4) in terms of an auxiliary random variable $s_t \in \{1, \ldots, n\}$ that serves as the mixture component indicator (hence, the name of the approach):

$$(\varepsilon_t^* \,|\, s_t = i) \;\sim\; \mathcal{N}(\mu_i, \sigma_i^2), \tag{7.5}$$

$$\mathbb{P}(s_t = i) \;=\; p_i. \tag{7.6}$$

Using this representation, we have a linear Gaussian model conditional on the component indicators $\mathbf{s} = (s_1, \ldots, s_T)'$ and the simulation techniques for estimating such a model can be applied.

The only missing piece is a suitable Gaussian mixture. By matching the moments of the log-χ_1^2 distribution, Kim, Shepherd and Chib (1998) propose a seven-component Gaussian mixture

$$f(x) = \sum_{i=1}^{7} p_i \, \varphi(x \,;\, \mu_i - 1.2704, \sigma_i^2),$$

where the values of the parameters are given in Table 7.1. We emphasize that these values are fixed and do not depend on any unknown parameters. Hence, this approximation does not require any additional computation time in the estimation.

Table 7.1: A seven-component Gaussian mixture for approximating the log-χ_1^2 distribution.

component	p_i	μ_i	σ_i^2
1	0.00730	−10.12999	5.79596
2	0.10556	−3.97281	2.61369
3	0.00002	−8.56686	5.17950
4	0.04395	2.77786	0.16735
5	0.34001	0.61942	0.64009
6	0.24566	1.79518	0.34023
7	0.25750	−1.08819	1.26261

In summary, using the Gaussian mixture approximation in (7.5) and (7.6), the model (7.2) and (7.3) is now conditionally linear Gaussian given the component indicators $\mathbf{s} = (s_1, \ldots, s_T)'$.

To complete the model specification, we assume independent prior distributions for σ_h^2 and h_0:

$$\sigma_h^2 \sim \mathcal{IG}(\nu_h, S_h), \quad h_0 \sim \mathcal{N}(a_0, b_0).$$

Then, posterior draws from $p(\mathbf{h}, \mathbf{s}, \sigma_h^2, h_0 \,|\, \mathbf{y})$ can be obtained via a Gibbs sampler that cycles through

1. sample $(\mathbf{s} \,|\, \mathbf{y}, \mathbf{h}, \sigma_h^2, h_0)$,
2. sample $(\mathbf{h} \,|\, \mathbf{y}, \mathbf{s}, \sigma_h^2, h_0)$,
3. sample $(\sigma_h^2 \,|\, \mathbf{y}, \mathbf{h}, \mathbf{s}, h_0)$,
4. sample $(h_0 \,|\, \mathbf{y}, \mathbf{h}, \mathbf{s}, \sigma_h^2)$.

The posterior draws obtained are technically from an approximate model, but one can reweight these draws using importance sampling weights to get the exact posterior moments under the original model. However, this step is often skipped in practice as reweighting makes little difference; see also the discussion in Kim, Shepherd and Chib (1998).

To implement Step 1, note that given $\mathbf{y}$ and $\mathbf{h}$, the errors $\boldsymbol{\varepsilon}^* = (\varepsilon_1^*, \ldots, \varepsilon_T^*)'$ are known. It then follows from (7.5) and (7.6) that

$$p(\mathbf{s} \,|\, \mathbf{y}, \mathbf{h}, \sigma_h^2, h_0) = \prod_{t=1}^{T} p(s_t \,|\, y_t, h_t, \sigma_h^2, h_0),$$

i.e., the component indicators $\{s_t\}$ are conditional independent given $\mathbf{y}$, $\mathbf{h}$ and other parameters. Therefore, we can sample each s_t independently.

Since s_t is a discrete random variable that follows a seven-point distribution, it can be easily sampled using the inverse-transform method, provided we can compute $\mathbb{P}(s_t = i \,|\, y_t, h_t, \sigma_h^2, h_0)$ for $i = 1, \ldots, 7$. By Bayes' theorem, we have

$$\mathbb{P}(s_t = i \,|\, y_t, h_t, \sigma_h^2, h_0) = \frac{1}{c_t} p_i \, \varphi(y_t^* \,;\, h_t + \mu_i - 1.2704, \sigma_i^2),$$

where $c_t = \sum_{j=1}^{7} p_j \, \varphi(y_t^* \,;\, h_t + \mu_j - 1.2704, \sigma_j^2)$ is a normalization constant.

To implement Step 2, recall that we have a conditionally linear Gaussian model given the component indicators $\mathbf{s}$. We first rewrite the observation equation (7.3) in matrix notation:

$$\mathbf{y}^* = \mathbf{h} + \boldsymbol{\varepsilon}^*,$$

where $(\boldsymbol{\varepsilon}^* \,|\, \mathbf{s}) \sim \mathcal{N}(\mathbf{d}_\mathbf{s}, \boldsymbol{\Sigma}_\mathbf{s})$ with $\mathbf{d}_\mathbf{s} = (\mu_{s_1} - 1.2704, \ldots, \mu_{s_T} - 1.2704)'$, $\boldsymbol{\Sigma}_\mathbf{s} = \text{diag}(\sigma_{s_1}^2, \ldots, \sigma_{s_T}^2)$, and the fixed parameters $\mu_1, \ldots, \mu_7$ and $\sigma_1^2, \ldots, \sigma_7^2$ are given in Table 7.1. By a simple change of variable, we have

$$(\mathbf{y}^* \,|\, \mathbf{s}, \mathbf{h}) \sim \mathcal{N}(\mathbf{h} + \mathbf{d}_\mathbf{s}, \boldsymbol{\Sigma}_\mathbf{s}).$$

Next, rewrite the state equation (7.2) in matrix form:

$$\mathbf{H}\mathbf{h} = \widetilde{\boldsymbol{\alpha}}_h + \mathbf{u},$$

where $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \sigma_h^2 \mathbf{I}_T)$, $\widetilde{\boldsymbol{\alpha}}_h = (h_0, 0, \ldots, 0)'$ and $\mathbf{H}$ is the usual first difference matrix. Noting that $\mathbf{H}^{-1}$ exists and $\mathbf{H}^{-1}\widetilde{\boldsymbol{\alpha}}_h = h_0 \mathbf{1}_T$, we have

$$(\mathbf{h} \,|\, \sigma_h^2, h_0) \sim \mathcal{N}(h_0 \mathbf{1}_T, \sigma_h^2 (\mathbf{H}'\mathbf{H})^{-1}).$$

Then, it follows from standard regression results that

$$(\mathbf{h} \,|\, \mathbf{y}, \mathbf{s}, \sigma_h^2, h_0) \sim \mathcal{N}(\widehat{\mathbf{h}}, \mathbf{K}_\mathbf{h}^{-1}),$$

where

$$\mathbf{K_h} = \frac{1}{\sigma_h^2}\mathbf{H}'\mathbf{H} + \mathbf{\Sigma}_\mathbf{s}^{-1}, \quad \widehat{\mathbf{h}} = \mathbf{K}_\mathbf{h}^{-1}\left(\frac{1}{\sigma_h^2}\mathbf{H}'\mathbf{H}h_0\mathbf{1}_T + \Sigma_\mathbf{s}^{-1}(\mathbf{y}^* - \mathbf{d_s})\right).$$

Since $\mathbf{K_h}$ is again a band matrix, we can sample $\mathbf{h}$ efficiently using the precision sampler as described in Algorithm 6.1 as before.

Steps 3 and 4 can be done easily, as both conditional distributions are standard. In particular,

$$(\sigma_h^2 \,|\, \mathbf{y}, \mathbf{h}, h_0) \sim \mathcal{I}G\left(\nu_h + \frac{T}{2}, S_h + \frac{1}{2}\sum_{t=1}^{T}(\tau_t - \tau_{t-1})^2\right)$$

and

$$(h_0 \,|\, \mathbf{y}, \mathbf{h}, \sigma_h^2) \sim \mathcal{N}(\widehat{h}_0, K_{h_0}^{-1}),$$

where

$$K_{h_0} = \frac{1}{b_0} + \frac{1}{\sigma_h^2}, \quad \widehat{h}_0 = K_{h_0}^{-1}\left(\frac{a_0}{b_0} + \frac{h_1}{\sigma_h^2}\right).$$

The following MATLAB function `SVRW.m` implements Steps 1 and 2 described above to sample $\mathbf{s}$ and $\mathbf{h}$ from their respective full conditional distributions. Note that in the code we vectorize the operations in sampling $\mathbf{s}$. In particular, instead of using a for-loop to sample each s_t at a time, we first construct a $T \times 7$ matrix of probabilities, where each row stores the probabilities of s_t belonging to each of the 7 components. Then $\mathbf{s}$ is sampled jointly.

```
% SVRW.m
function h = SVRW(ystar,h,h0,sigh2)
T = length(h);
    % normal mixture
pj = [0.0073 .10556 .00002 .04395 .34001 .24566 .2575];
mj = [-10.12999 -3.97281 -8.56686 2.77786 .61942 1.79518 -1.08819]...
    - 1.2704;  % caution: means are adjusted!
sigj2 = [5.79596 2.61369 5.17950 .16735 .64009 .34023 1.26261];
sigj = sqrt(sigj2);

    % sample S from a 7-point distrete distribution
temprand = rand(T,1);
q = repmat(pj,T,1).*normpdf(repmat(ystar,1,7),...
    repmat(h,1,7)+repmat(mj,T,1),repmat(sigj,T,1));
q = q./repmat(sum(q,2),1,7);
S = 7 - sum(repmat(temprand,1,7)<cumsum(q,2),2) + 1;

    % sample h
```

```
H = speye(T) - sparse(2:T,1:(T-1),ones(1,T-1),T,T);
HH = H'*H;
d_s = mj(S)';
iSig_s = sparse(1:T,1:T,1./sigj2(S));
Kh = HH/sigh2 + iSig_s;
h_hat = Kh\(h0/sigh2*HH*ones(T,1) + iSig_s*(ystar-d_s));
h = h_hat + chol(Kh,'lower')'\randn(T,1);
end
```

7.1.2 Empirical Example: Modeling AUD/USD Returns

We illustrate the estimation of the stochastic volatility model using the data depicted in Figure 7.1: AUD/USD daily returns from 02 January 2005 to 31 December 2012. Specifically, we allow for a constant conditional mean in the observation equation

$$y_t = \mu + \mathrm{e}^{\frac{1}{2}h_t}\varepsilon_t,$$

where $\varepsilon_t \sim \mathcal{N}(0,1)$. The state equation is given in (7.2). We assume the same independent priors for σ_h^2 and h_0 as before. For μ, we consider $\mu \sim \mathcal{N}(\mu_0, V_\mu)$.

To estimate the model with the constant conditional mean μ, we simply need an extra block to sample from the conditional density $p(\mu \,|\, \mathbf{y}, \mathbf{h})$, and modify the Gibbs sampler discussed in Section 7.1.1 by replacing y_t with $y_t - \mu$. In particular, one can check that

$$(\mu \,|\, \mathbf{y}, \mathbf{h}) \sim \mathcal{N}(\widehat{\mu}, K_\mu^{-1}),$$

where $K_\mu^{-1} = V_\mu^{-1} + \sum_{t=1}^T \mathrm{e}^{-h_t}$ and $\widehat{\mu} = K_\mu^{-1}(\mu_0/V_\mu + \sum_{t=1}^T \mathrm{e}^{-h_t} y_t)$.

The following MATLAB script `SV_AUD_eg.m` first loads the dataset `AUDUSD.csv` and implements the Gibbs sampler outlined above. In particular, it calls the function `SVRW.m` to sample $\mathbf{s}$ and $\mathbf{h}$ from their respective full conditional densities.

```
% SV_AUD_eg.m
nsim = 20000; burnin = 1000;
load 'AUDUSD.csv';
y = AUDUSD; T = length(y);

    % prior
mu0 = 0; Vmu = 100;
a0 = 0; b0 = 100;
nu_h = 3; S_h = .2*(nu_h-1);

    % initialize the Markov chain
```

```
sigh2 = .05; mu = mean(y);
h0 = log(var(y)); h = h0*ones(T,1);
H = speye(T) - sparse(2:T,1:(T-1),ones(1,T-1),T,T);
HH = H'*H;

    % initialize for storage
store_theta = zeros(nsim,3); % [mu h0 sigh2]
store_h = zeros(nsim,T);

for isim = 1:nsim + burnin
        % sample mu
    Kmu = 1/Vmu + sum(1./exp(h));
    mu_hat = Kmu\(mu0/Vmu + sum(y./exp(h)));
    mu = mu_hat + sqrt(Kmu)'\randn;

        % sample h
    ystar = log((y-mu).^2 + .0001);
    h = SVRW(ystar,h,h0,sigh2);

        % sample sigh2
    sigh2 = 1/gamrnd(nu_h + T/2, 1/(S_h + (h-h0)'*HH*(h-h0)/2));

        % sample h0
    Kh0 = 1/b0 + 1/sigh2;
    h0_hat = Kh0\(a0/b0 + h(1)/sigh2);
    h0 = h0_hat + sqrt(Kh0)'\randn;
end
```

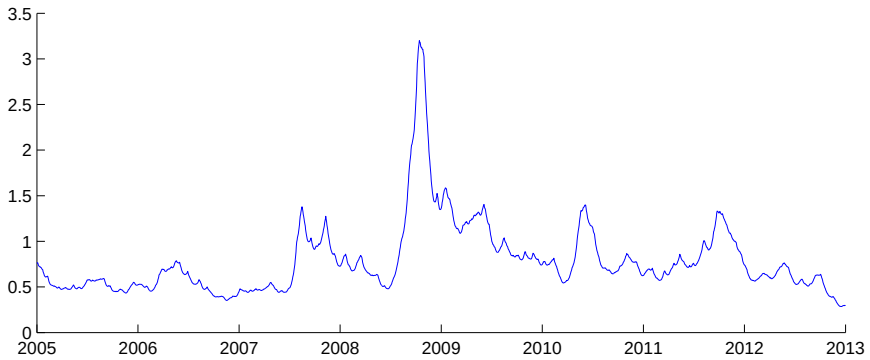

Figure 7.2: Posterior means of the time-varying standard deviation $\{e^{h_t/2}\}$ for the AUD/USD daily returns data.

We obtain 20000 posterior draws after a burn-in period of 1000. Figure 7.2 depicts the posterior means of the time-varying standard deviation. As the figure shows, there is substantial time variation in the volatility. In particular, between 2005 and early 2007, the estimated standard deviation mostly fluctuates around 0.5%. It increases to about 1% in mid-2007 and peaks at 3% during the Great Recession. Although it goes down substantially after 2009, it is still much higher than the pre-crisis level.

7.2 Application: Modeling Inflation Using UC-SV Model

In this application we revisit the unobserved components model considered in Stock and Watson (2007) for modeling inflation. Specifically, they decompose the inflation series into a trend and a transitory component, where each component has an independent stochastic volatility process. A large and growing literature has shown that allowing for stochastic volatility substantially improves inflation forecasts (see, e.g., Clark, 2011; Koop and Korobilis, 2012; Chan, 2013). Here we evaluate the in-sample fit and formally test if both stochastic volatility processes are needed for modeling trend inflation in the US.

To test whether there is time variation in the stochastic volatility processes, we reparameterize the UC-SV model of Stock and Watson (2007) using the noncentered parameterization discussed in Section 6.3. That is, instead of the error variance in the state equation, we work with the standard derivation that is defined to have support on the whole real line. Consequently, we can use the Savage-Dickey density ratio to compare the unrestricted stochastic volatility model against the restricted version in which the standard deviation is zero and the variance is constant.

7.2.1 UC-SV Model in Noncentered Parameterization

Consider first the UC-SV model of Stock and Watson (2007) in the original parameterization:

$$\begin{aligned} y_t &= \tau_t + \varepsilon_t^y, \\ \tau_t &= \tau_{t-1} + \varepsilon_t^\tau, \\ h_t &= h_{t-1} + \varepsilon_t^h, \\ g_t &= g_{t-1} + \varepsilon_t^g, \end{aligned}$$

where $\varepsilon_t^y \sim \mathcal{N}(0, \mathrm{e}^{h_t})$, $\varepsilon_t^\tau \sim \mathcal{N}(0, \mathrm{e}^{g_t})$, $\varepsilon_t^h \sim \mathcal{N}(0, \omega_h^2)$, $\varepsilon_t^g \sim \mathcal{N}(0, \omega_g^2)$, and the initial conditions τ_0, h_0 and g_0 are treated as unknown parameters.

In this model the inflation series y_t is decomposed into the trend inflation τ_t and the inflation gap ε_t^y, where both components have time-varying volatility. By comparing the variances of the trend and the inflation gap, we can assess the relative importance of each component in driving the variance of the inflation over time.

Now, we rewrite the above UC-SV model in the noncentered parameterization discussed in Section 6.3 as follows:

$$\begin{aligned} y_t &= \tau_t + \mathrm{e}^{\frac{1}{2}(h_0+\omega_h \widetilde{h}_t)}\widetilde{\varepsilon}_t^y, && (7.7)\\ \tau_t &= \tau_{t-1} + \mathrm{e}^{\frac{1}{2}(g_0+\omega_g \widetilde{g}_t)}\widetilde{\varepsilon}_t^\tau, && (7.8)\\ \widetilde{h}_t &= \widetilde{h}_{t-1} + \widetilde{\varepsilon}_t^h, && (7.9)\\ \widetilde{g}_t &= \widetilde{g}_{t-1} + \widetilde{\varepsilon}_t^g, \end{aligned}$$

where $\widetilde{h}_0 = \widetilde{g}_0 = 0$, $\widetilde{\varepsilon}_t^y$, $\widetilde{\varepsilon}_t^\tau$, $\widetilde{\varepsilon}_t^h$ and $\widetilde{\varepsilon}_t^g$ are iid $\mathcal{N}(0,1)$.

Hence, using this setup, we can turn off the stochastic volatility in the trend and the transitory component by assuming that $\omega_g = 0$ and $\omega_h = 0$ respectively. In their analysis, Stock and Watson (2007) fix the parameters $\omega_h^2 = \omega_g^2 = 0.2$. In a recent paper, Moura and Turatti (2014) estimate the two parameters while maintaining the assumption that $\omega_h^2 = \omega_g^2 = \omega^2$. Using inflation in the G7 countries, they find that the estimates of ω^2 are statistically different from the calibrated value of 0.2 for a few countries. Here we estimate both ω_h^2 and ω_g^2 and allow them to be different.

7.2.2 Estimation

Next we outline the estimation of the UC-SV model in the noncentered parameterization. The model parameters are ω_g, ω_h, τ_0, h_0 and g_0; and there are three types of states: $\boldsymbol{\tau} = (\tau_1, \dots, \tau_T)'$, $\widetilde{\mathbf{h}} = (\widetilde{h}_1, \dots, \widetilde{h}_T)'$ and $\widetilde{\mathbf{g}} = (\widetilde{g}_1, \dots, \widetilde{g}_T)'$.

As in Section 6.3, we assume normal priors for ω_g and ω_h: $\omega_g \sim \mathcal{N}(0, V_{\omega_g})$ and $\omega_h \sim \mathcal{N}(0, V_{\omega_h})$. We set $V_{\omega_h} = V_{\omega_g} = 0.2$ so that the implied prior means are $\mathbb{E}\,\omega_h^2 = \mathbb{E}\,\omega_g^2 = 0.2$, which are the same as the calibrated value in Stock and Watson (2007). We further assume that $h_0 \sim \mathcal{N}(a_{0,h}, b_{0,h})$, $g_0 \sim \mathcal{N}(a_{0,g}, b_{0,g})$ and $\tau_0 \sim \mathcal{N}(a_{0,\tau}, b_{0,\tau})$.

Here we highlight two key steps in the Gibbs sampler: sampling $\boldsymbol{\tau}$ and $\widetilde{\mathbf{h}}$ from their full conditional distributions. Recall that $\mathbf{h} = h_0\mathbf{1}_T + \omega_h\widetilde{\mathbf{h}}$ and $\mathbf{g} = g_0\mathbf{1}_T + \omega_g\widetilde{\mathbf{g}}$. Then, it follows from (7.7) that

$$(\mathbf{y} \mid \boldsymbol{\tau}, \widetilde{\mathbf{h}}, h_0, \omega_h) \sim \mathcal{N}(\boldsymbol{\tau}, \boldsymbol{\Omega}_{\mathbf{h}}),$$

where $\boldsymbol{\Omega}_{\mathbf{h}} = \mathrm{diag}(\mathrm{e}^{h_1}, \dots, \mathrm{e}^{h_T})$. Similarly, (7.8) implies that

$$(\boldsymbol{\tau} \mid \tau_0, \widetilde{\mathbf{g}}, g_0, \omega_g) \sim \mathcal{N}(\tau_0\mathbf{1}_T, (\mathbf{H}'\boldsymbol{\Omega}_{\mathbf{g}}^{-1}\mathbf{H})^{-1}),$$

where $\mathbf{\Omega_g} = \text{diag}(\text{e}^{g_1}, \ldots, \text{e}^{g_T})$ and $\mathbf{H}$ is the usual first difference matrix. Then by standard linear regression results, we have

$$(\boldsymbol{\tau} \,|\, \mathbf{y}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}}, \omega_h, \omega_g, \tau_0, h_0, g_0) \sim \mathcal{N}(\widehat{\boldsymbol{\tau}}, \mathbf{K}_{\boldsymbol{\tau}}^{-1}),$$

where

$$\mathbf{K}_{\boldsymbol{\tau}} = \mathbf{H}'\mathbf{\Omega}_{\mathbf{g}}^{-1}\mathbf{H} + \mathbf{\Omega}_{\mathbf{h}}^{-1}, \quad \widehat{\boldsymbol{\tau}} = \mathbf{K}_{\boldsymbol{\tau}}^{-1}(\mathbf{H}'\mathbf{\Omega}_{\mathbf{g}}^{-1}\mathbf{H}\tau_0\mathbf{1}_T + \mathbf{\Omega}_{\mathbf{h}}^{-1}\mathbf{y}).$$

We then use the precision sampler in Algorithm 6.1 to sample $\boldsymbol{\tau}$ as before.

To sample $\widetilde{\mathbf{h}}$, we use the auxiliary mixture sampler discussed in Section 7.1.1. Specifically, let $y_t^* = \log(y_t - \tau_t)^2$ and $\widetilde{\varepsilon}_t^{y*} = \log(\widetilde{\varepsilon}_t^{y})^2$ for $t = 1, \ldots, T$. Then, it follows from (7.7) that

$$\mathbf{y}^* = h_0\mathbf{1}_T + \omega_h\widetilde{\mathbf{h}} + \widetilde{\boldsymbol{\varepsilon}}^{y*},$$

where $\mathbf{y}^* = (y_1^*, \ldots, y_T^*)'$ and $\widetilde{\boldsymbol{\varepsilon}}^{y*} = (\widetilde{\varepsilon}_1^{y*}, \ldots, \widetilde{\varepsilon}_T^{y*})'$. We then augment the model with the component indicators $\mathbf{s} = (s_1, \ldots, s_T)'$ such that given these indicators, $(\widetilde{\boldsymbol{\varepsilon}}^{y*} \,|\, \mathbf{s}) \sim \mathcal{N}(\mathbf{d_s}, \mathbf{\Omega_s})$ with $\mathbf{d_s}$ and $\mathbf{\Omega_s}$ obtained from the Gaussian mixture approximation of the log χ_1^2 distribution. Using a similar argument as before, we can show that the component indicators are conditionally independent and we can sample each of them sequentially from a 7-point distribution.

Next, given $\mathbf{s}$ and other parameters, we sample $\widetilde{\mathbf{h}}$ from the normal distribution

$$(\widetilde{\mathbf{h}} \,|\, \mathbf{y}, \boldsymbol{\tau}, \omega_h, h_0) \sim \mathcal{N}(\widehat{\widetilde{\mathbf{h}}}, \mathbf{K}_{\widetilde{\mathbf{h}}}^{-1}),$$

where

$$\mathbf{K}_{\widetilde{\mathbf{h}}} = \mathbf{H}'\mathbf{H} + \omega_h^2\mathbf{\Omega}_{\mathbf{s}}^{-1}, \quad \widehat{\widetilde{\mathbf{h}}} = \mathbf{K}_{\boldsymbol{\tau}}^{-1}(\omega_h\mathbf{\Omega}_{\mathbf{s}}^{-1}(\mathbf{y}^* - h_0\mathbf{1}_T - \mathbf{d_s})).$$

Again we use the precision sampler described in Algorithm 6.1 to sample from this normal distribution as before.

7.2.3 Empirical Results

We fit the UC-SV model with quarterly US CPI inflation from 1947Q1 to 2015Q4, where the indices are transformed to annualized growth rates. Figure 7.3 reports the posterior means of the trend inflation based on a sample of 50000 posterior draws after a burn-in period of 1000.

Prior to the Great Moderation, the estimated trend inflation traces the actual inflation very closely. However, since the mid-1980s, the volatility of the trend inflation has decreased substantially. In addition, the trend inflation itself has exhibited a steady decline from about 5% to less than 2% at the end of the sample.

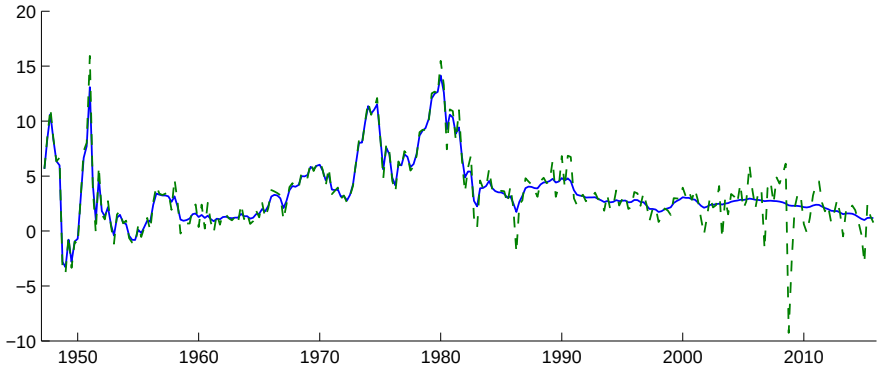

Figure 7.3: Posterior estimates of $\{\tau_t\}$ (solid line) and the CPI inflation (dotted line).

Next, we report in Figure 7.4 the estimated time-varying standard deviations of the transitory and the trend components. Consistent with the trend inflation depicted above, the variance of trend inflation is high on average before the Great Moderation—in fact, it is higher than the variance of the transitory component throughout the 1970s and early 1980s. Since 1980s, however, it has declined substantially and has remained low and stable till the end of the sample.

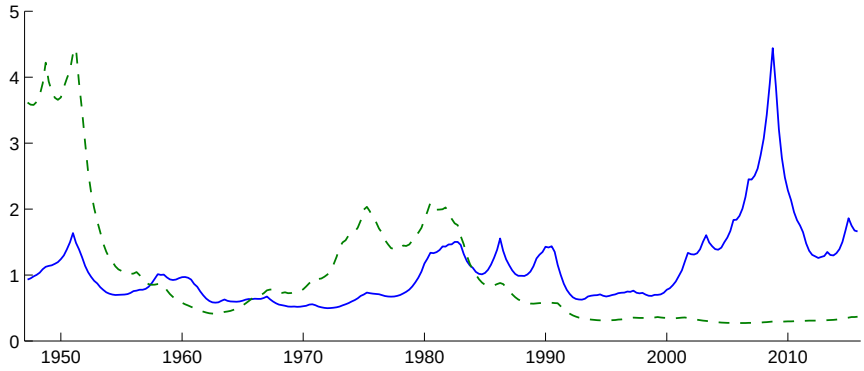

Figure 7.4: Posterior estimates of the time-varying standard deviations of the transitory component $\{\mathrm{e}^{h_t/2}\}$ (solid line) and the trend component $\{\mathrm{e}^{g_t/2}\}$ (dotted line).

The volatility estimates presented above suggest that both variances are time-varying. This is confirmed by the posterior densities of the standard deviations ω_h and ω_g. Specifically, Figure 7.5 plots the histograms of the posterior draws of ω_h and ω_g. Both are clearly bimodal and they have virtually no mass at zero. In fact, the log Bayes factor against the restricted model with $\omega_h = 0$ compared to the unrestricted UC-SV model is 94; the corresponding value against the restricted

model with $\omega_g = 0$ is 63. In both cases there is overwhelming evidence in support for allowing time-varying volatility.

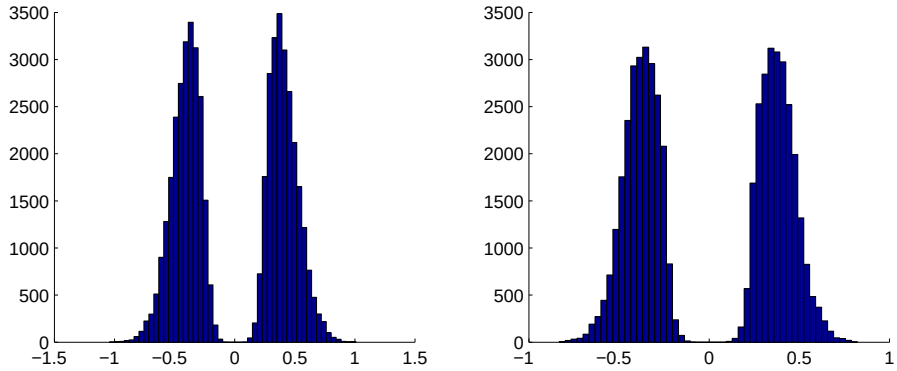

Figure 7.5: Histograms of the posterior draws of ω_h (left panel) and ω_g (right panel).

7.3 Stochastic Volatility in Mean Model

In this section we study a stochastic volatility model in which the stochastic volatility has a direct impact on the variable of interest. Originally developed by Koopman and Hol Uspensky (2002) as an alternative of the ARCH in mean model of Engle, Lilien and Robins (1987), the **stochastic volatility in mean model** allows the time-varying volatility enters the conditional mean as a covariate.

The stochastic volatility in mean model is widely used in empirical finance to study the relationship between assets returns and their volatility. More recently, it has been used to model macroeconomic data, as in Berument, Yalcin and Yildirim (2009) for investigating the inflation-inflation uncertainty relationship and Mumtaz and Zanetti (2013) for examining the impact of monetary shock volatility.

Specifically, consider the following model

$$y_t = \alpha e^{h_t} + \mathbf{x}_t'\boldsymbol{\beta} + \varepsilon_t^y, \qquad \varepsilon_t^y \sim \mathcal{N}(0, e^{h_t}), \tag{7.10}$$

$$h_t = h_{t-1} + \varepsilon_t^h, \qquad \varepsilon_t^h \sim \mathcal{N}(0, \sigma_h^2), \tag{7.11}$$

where $\mathbf{x}_t$ is a $k \times 1$ vector of covariates, $\boldsymbol{\beta}$ is the associated $k \times 1$ vector of coefficients, and the errors ε_t^y and ε_t^h are mutually and serially uncorrelated. As before, the log-volatility is assumed to follow a random walk process. Alternatively, it is also common to consider a stationary AR(1) process.

This model generalizes the basic stochastic volatility model by allowing for covariates. In particular, the contemporaneous variance e^{h_t} is included as one of the

covariates. The associated parameter α captures the impact of volatility, and is often the key parameter of interest.

Estimation of the stochastic volatility in mean model is more difficult. This is because now the log-volatility h_t also enters the conditional mean equation, and the auxiliary mixture sampler discussed in Section 7.1.1 cannot be applied to this setting. Consequently, we need to come up with an alternative way to sample the high-dimensional, nonstandard distribution of $\mathbf{h}$.

7.3.1 Estimation

In the original stochastic volatility in mean paper, Koopman and Hol Uspensky (2002) develop a simulated maximum likelihood estimator based on the Kalman filter to fit the model. Specifically, a Gaussian density is constructed to approximate the full conditional density of $\mathbf{h}$. This approximate density is then used to evaluate the integrated likelihood—obtained by integrating out h_t—via importance sampling.

Independent draws from this approximate density are obtained using Kalman filter-based algorithms such as those in Carter and Kohn (1994) and Durbin and Koopman (2002). Since there are only a few parameters in the a constant-coefficient stochastic volatility in mean model, one can maximize the likelihood numerically to obtain the maximum likelihood estimates.

Here we adopt a Bayesian approach and consider an efficient MCMC algorithm to simulate from the joint posterior distribution. The core of the algorithm is a Gaussian approximation of the full conditional density of $\mathbf{h}$. This density is the same as that used in Koopman and Hol Uspensky (2002) for the simulated maximum likelihood estimator.

However, the Gaussian approximation is constructed differently—instead of using the Kalman filter, we obtain the approximate density using the Newton-Raphson method based on band matrix algorithms. The latter approach is computationally more efficient as it exploits the special structure of the problem: the Hessian of the log-conditional density of $\mathbf{h}$ is a band matrix. Recent papers using band matrix algorithms to obtain approximate density include Rue, Martino and Chopin (2009) for nonlinear Markov random fields; McCausland (2012), Chan, Koop and Potter (2013, 2016) and Djegnéné and McCausland (2014) for nonlinear state space models.

More generally, the Bayesian approach is more flexible as it can be easily generalized to models with additional features, such as time-varying coefficients in Chan (2017). Due to the modular nature of MCMC algorithms, we can simulate each type of states one at a time, which reduces the dimension of the problem and makes estimation much easier.

In contrast, simulated maximum likelihood estimation is more difficult to be generalized to models with multiple nonlinear states. For example, in stochastic volatility models with time-varying coefficients, likelihood evaluation would involve "integrating out" simultaneously both the time-varying coefficients and the stochastic volatility by Monte Carlo methods. This likelihood evaluation step is in general computationally intenstive.

We now discuss a posterior sampler to fit the stochastic volatility in mean model in (7.10)–(7.11). Let $\boldsymbol{\gamma} = (\alpha, \boldsymbol{\beta}')'$ denote the regression coefficients. We complete the model specification by assuming the following independent priors:

$$\boldsymbol{\gamma} \sim \mathcal{N}(\boldsymbol{\gamma}_0, \mathbf{V}_\gamma), \quad \sigma_h^2 \sim \mathcal{IG}(\nu_h, S_h), \quad h_0 \sim \mathcal{N}(a_0, b_0).$$

Then, posterior draws can be obtained by sequentially sampling from:

1. $p(\mathbf{h} \,|\, \mathbf{y}, \boldsymbol{\gamma}, \sigma_h^2, h_0)$,
2. $p(\boldsymbol{\gamma} \,|\, \mathbf{y}, \mathbf{h}, \sigma_h^2, h_0)$,
3. $p(\sigma_h^2 \,|\, \mathbf{y}, \mathbf{h}, \boldsymbol{\gamma}, h_0)$,
4. $p(h_0 \,|\, \mathbf{y}, \mathbf{h}, \boldsymbol{\gamma}, \sigma_h^2)$.

Steps 2–4 are standard and here we focus on Step 1. We first discuss an efficient way to obtain a Gaussian approximation of $p(\mathbf{h} \,|\, \mathbf{y}, \boldsymbol{\gamma}, \sigma_h^2, h_0)$ using the Newton-Raphson method in Algorithm 3.4. To that end, we need to obtain the gradient and Hessian of the log-density $\log p(\mathbf{h} \,|\, \mathbf{y}, \boldsymbol{\gamma}, \sigma_h^2, h_0)$.

By Bayes' Theorem, we have

$$p(\mathbf{h} \,|\, \mathbf{y}, \boldsymbol{\gamma}, \sigma_h^2, h_0) \propto p(\mathbf{y} \,|\, \boldsymbol{\gamma}, \mathbf{h}) p(\mathbf{h} \,|\, \sigma_h^2, h_0),$$

and we next derive explicit expressions for both densities on the right-hand side. As before, the prior density of $\mathbf{h}$ is Gaussian with log-density

$$\log p(\mathbf{h} \,|\, \sigma_h^2, h_0) = -\frac{1}{2\sigma_h^2}(\mathbf{h}'\mathbf{H}'\mathbf{H}\mathbf{h} - 2\mathbf{h}'\mathbf{H}'\mathbf{H}h_0\mathbf{1}_T) + c_1,$$

where c_1 is a constant independent of $\mathbf{h}$. Hence, the gradient and Hessian of $\log p(\mathbf{h} \,|\, \sigma_h^2, h_0)$ are given by

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{h}} \log p(\mathbf{h} \,|\, \sigma_h^2, h_0) &= -\frac{1}{\sigma_h^2}\mathbf{H}'\mathbf{H}(\mathbf{h} - h_0\mathbf{1}_T), \\
\frac{\partial^2}{\partial \mathbf{h} \partial \mathbf{h}'} \log p(\mathbf{h} \,|\, \sigma_h^2, h_0) &= -\frac{1}{\sigma_h^2}\mathbf{H}'\mathbf{H}.
\end{aligned}$$

Next, let $\mathbf{f}$ and $\mathbf{G}$ denote, respectively, the gradient and the *negative* Hessian of $\log p(\mathbf{y} \,|\, \mathbf{h}, \boldsymbol{\gamma})$:

$$\mathbf{f} = \frac{\partial}{\partial \mathbf{h}} \log p(\mathbf{y} \,|\, \mathbf{h}, \boldsymbol{\gamma}), \quad \mathbf{G} = -\frac{\partial^2}{\partial \mathbf{h} \partial \mathbf{h}'} \log p(\mathbf{y} \,|\, \mathbf{h}, \boldsymbol{\gamma}).$$

Note that the log conditional likelihood $\log p(\mathbf{y} \,|\, \mathbf{h}, \boldsymbol{\gamma})$ has the form $\log p(\mathbf{y} \,|\, \mathbf{h}, \boldsymbol{\gamma}) = \sum_{t=1}^{T} \log p(y_t \,|\, h_t, \boldsymbol{\gamma})$. Since h_t only enters the density $p(y_t \,|\, h_t, \boldsymbol{\gamma})$, the t-th element of $\mathbf{f}$ is simply $f_t = \partial \log p(y_t \,|\, h_t, \boldsymbol{\gamma}) / \partial h_t$ and the negative Hessian $\mathbf{G}$ is diagonal, where the t-th diagonal element is $G_t = -\partial^2 \log p(y_t \,|\, h_t, \boldsymbol{\gamma}) / \partial h_t^2$.

Putting all these together, we have therefore derived the gradient and Hessian of $\log p(\mathbf{h} \,|\, \mathbf{y}, \boldsymbol{\gamma}, \sigma_h^2, h_0)$:

$$\mathbf{S}(\mathbf{h}) = \frac{\partial}{\partial \mathbf{h}} \log p(\mathbf{h} \,|\, \mathbf{y}, \boldsymbol{\gamma}, \sigma_h^2, h_0) = -\frac{1}{\sigma_h^2} \mathbf{H}' \mathbf{H} (\mathbf{h} - h_0 \mathbf{1}_T) + \mathbf{f},$$
$$\mathbf{F}(\mathbf{h}) = \frac{\partial^2}{\partial \mathbf{h} \partial \mathbf{h}'} \log p(\mathbf{h} \,|\, \mathbf{y}, \boldsymbol{\gamma}, \sigma_h^2, h_0) = -\left(\frac{1}{\sigma_h^2} \mathbf{H}' \mathbf{H} + \mathbf{G} \right).$$

Hence, we can implement the Newton-Raphson method in Algorithm 3.4. In particular, initialize the algorithm with $\mathbf{h} = \mathbf{h}^{(1)}$ for some constant vector $\mathbf{h}^{(1)}$. For $t = 1, 2, \ldots,$ use $\mathbf{h} = \mathbf{h}^{(t)}$ in the evaluation of $\mathbf{S}(\mathbf{h})$ and $\mathbf{F}(\mathbf{h})$, and compute the Newton-Raphson recursion:

$$\mathbf{h}^{(t+1)} = \mathbf{h}^{(t)} - \mathbf{F}(\mathbf{h}^{(t)})^{-1} \, \mathbf{S}(\mathbf{h}^{(t)}).$$

Repeat this procedure until some convergence criterion is reached, e.g., when $\|\mathbf{h}^{(t+1)} - \mathbf{h}^{(t)}\| < \varepsilon$ for some prefixed tolerance level ε. We will show below that the Hessian for the stochastic volatility model is globally negative definite. Consequently, the convergence of the Newton-Raphson method is typically very fast.

Now, let $\widehat{\mathbf{h}}$ be the mode obtained from the Newton-Raphson method and let $\mathbf{K}_{\mathbf{h}}$ denote the negative Hessian evaluated at the mode, i.e., $\mathbf{K}_{\mathbf{h}} = \mathbf{H}' \mathbf{H} / \sigma_h^2 + \mathbf{G}$. Then, we use $\mathcal{N}(\widehat{\mathbf{h}}, \mathbf{K}_{\mathbf{h}}^{-1})$ as the proposal density in an independence-chain Metropolis-Hastings step. Since the precision matrix is again a band matrix, we can use the precision sampler in Algorithm 6.1 to obtain draws from $\mathcal{N}(\widehat{\mathbf{h}}, \mathbf{K}_{\mathbf{h}}^{-1})$ efficiently.

Finally, for the stochastic volatility in mean model, we have

$$\begin{aligned} \log p(y_t \,|\, h_t, \boldsymbol{\gamma}) = &-\frac{1}{2} \log(2\pi) - \frac{1}{2} h_t - \frac{1}{2} \mathrm{e}^{-h_t} (y_t - \alpha \mathrm{e}^{h_t} - \mathbf{x}_t' \boldsymbol{\beta})^2 \\ &- \frac{1}{2} \log(2\pi) - \frac{1}{2} h_t - \frac{1}{2} \left(\alpha^2 \mathrm{e}^{h_t} + \mathrm{e}^{-h_t} (y_t - \mathbf{x}_t' \boldsymbol{\beta})^2 - 2\alpha (y_t - \mathbf{x}_t' \boldsymbol{\beta}) \right). \end{aligned}$$

It is easy to check that

$$f_t = \frac{\partial}{\partial h_t} \log p(y_t \,|\, h_t, \boldsymbol{\gamma}) = -\frac{1}{2} - \frac{1}{2} \alpha^2 \mathrm{e}^{h_t} + \frac{1}{2} \mathrm{e}^{-h_t} (y_t - \mathbf{x}_t' \boldsymbol{\beta})^2,$$
$$G_t = -\frac{\partial^2}{\partial h_t^2} \log p(y_t \,|\, h_t, \boldsymbol{\gamma}) = \frac{1}{2} \alpha^2 \mathrm{e}^{h_t} + \frac{1}{2} \mathrm{e}^{-h_t} (y_t - \mathbf{x}_t' \boldsymbol{\beta})^2 > 0.$$

Since $\mathbf{G}$ is a diagonal matrix with positive diagonal elements, it is positive definite. Consequently, the Hessian $\mathbf{F}(\mathbf{h}) = -(\mathbf{H}'\mathbf{H}/\sigma_h^2 + \mathbf{G})$ is negative definite regardless of the value of $\mathbf{h}$.

The following MATLAB function `sample_SVM_h.m` implements the above independence-chain Metropolis-Hastings step to sample $\mathbf{h}$. The first part of the function uses the Newton-Raphson method to obtain, $\widehat{\mathbf{h}}$, the mode of the log conditional density of $\mathbf{h}$, and stores it as the variable `h_hat`. It also computes $\mathbf{K_h}$, the negative Hessian evaluated at the mode, and stores it as `Kh`.

The second part of the function takes $\widehat{\mathbf{h}}$ and $\mathbf{K_h}$ computed from the Newton-Raphson method, and gets a candidate draw from $\mathcal{N}(\widehat{\mathbf{h}}, \mathbf{K_h}^{-1})$ using the precision sampler in Algorithm 6.1. The candidate draw is then accepted with probability as computed in the Metropolis-Hastings step.

```
% sample_SVM_h.m
function [h flag] = sample_SVM_h(y,alp,mu,h,h0,sigh2)
T = size(h,1);
flag = 0;
H = speye(T) - sparse(2:T,1:(T-1),ones(1,T-1),T,T);
HH = H'*H;
s2 = (y-mu).^2;
e_h = 1; ht = h;
while e_h> 10^(-4);
    exp_ht = exp(ht);
    tmp1 = .5*alp^2*exp_ht;
    tmp2 = .5*s2./exp_ht;
    f = -.5 - tmp1 + tmp2;
    G = sparse(1:T,1:T,tmp1+tmp2);
    S = -1/sigh2*HH*(ht - h0) + f;
    Kh = HH/sigh2 + G;
    new_ht = ht + Kh\S;
    e_h = max(abs(new_ht-ht));
    ht = new_ht;
end
h_hat = ht;
exp_ht = exp(h_hat);
G = sparse(1:T,1:T,.5*alp^2*exp_ht+.5*s2./exp_ht);
Kh = HH/sigh2 + G;
lph = @(x) -.5*(x-h0)'*HH*(x-h0)/sigh2 ...
    -.5*sum(x) -.5*exp(-x)'*(y-mu-alp*exp(x)).^2;
lg = @(x) -.5*(x-h_hat)'*Kh*(x-h_hat);
hc = h_hat + chol(Kh,'lower')'\randn(T,1);
```

```
alp_MH = lph(hc) - lph(h) + lg(h) - lg(hc);
if exp(alp_MH) > rand
    h = hc;
    flag = 1;
end
end
```

7.3.2 Empirical Example: Modeling Excess Returns on S&P 500

In this section we illustrate the estimation of the stochastic volatility in mean model by replicating part of the results in Koopman and Hol Uspensky (2002). In particular, they study the relationship between stock index returns and their volatility for the UK, the US and Japan. They find evidence of a weak negative relationship between returns and contemporaneous volatility.

We revisit their application using daily S&P 500 index between 02 January 2013 to 31 December 2015. Returns are computed on a continuously compounded basis and expressed in percentages: $R_t = 100 \log(P_t/P_{t-1})$, where P_t is the S&P 500 index at time t. We use the effective federal funds rate, denoted as i_t, as a proxy for the risk free rate of return. Finally, the excess returns are defined as $y_t = R_t - i_t$. Figure 7.6 plots the excess returns data.

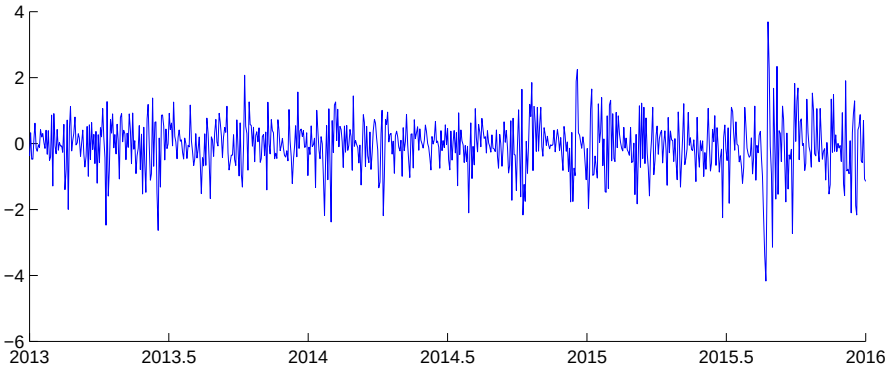

Figure 7.6: Excess returns for the S&P 500 between 02 January 2013 to 31 December 2015.

We implement the posterior sampler discussed in last section and obtain 20000 posterior draws after a burn-in period of 1000. We report in Figure 7.7 the posterior estimates of the time-varying standard deviation $\{e^{h_t/2}\}$. The estimates mostly

fluctuate between 0.5 and 1, except the brief episode at the second half of 2015 when the standard deviation peaks at about 2.

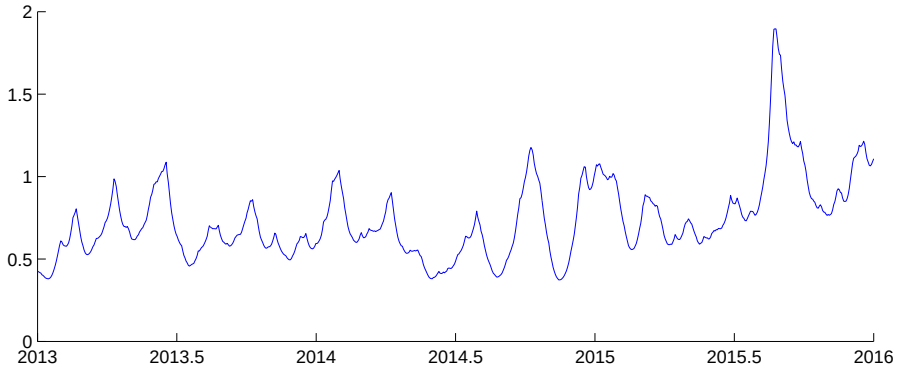

Figure 7.7: Posterior estimates of the time-varying standard deviation $\{e^{h_t/2}\}$.

The key parameter of interest is α, the regression coefficient on the variance e^{h_t}. Its posterior mean is -0.204 with a 95% credible interval $(-0.369, -0.048)$. Figure 7.8 depicts the histogram of the posterior draws of α, which has little mass at 0. Using data from 1975 to 1998, Koopman and Hol Uspensky (2002) report an estimate of -0.023 and an asymptotic 95% confidence interval $(-0.065, 0.019)$. Hence, our estimate has the same sign, but the absolute magnitude is much larger.

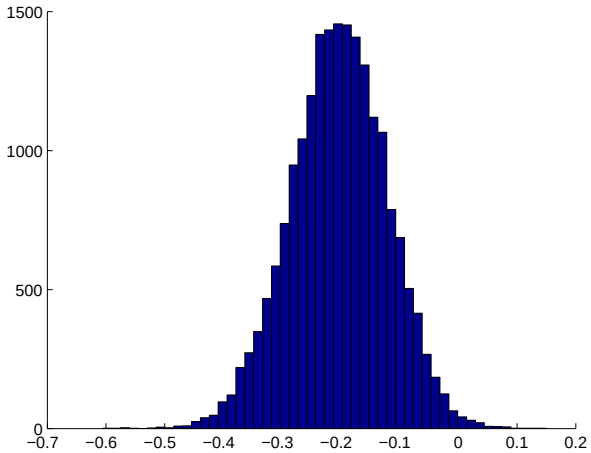

Figure 7.8: Histogram of the posterior draws of α.

Koopman and Hol Uspensky (2002) explain that the parameter α captures two opposing impacts of volatility on excess returns. On the one hand, the *expected*

component of volatility is *positively* correlated with excess returns, where risk-averse investors demand higher expected returns during more volatile periods. On the other hand, unexpected large shocks to the return process induce higher expected volatility in the future. If future cash flows are unchanged, the current stock index should fall. Hence, the unexpected component of volatility should be *negatively* correlated with excess returns. This is often referred to as the volatility feedback theory. At time t, h_{t+1} captures both the expected and unexpected components of volatility. Hence, a negative estimate of α indicates that the volatility feedback effect dominates.

Chapter 8

Vector Autoregressions

Vector autoregressions (VARs) have been widely used for macroeconomic forecasting and structural analysis since the seminal work of Sims (1980). In particular, VARs are often served as the benchmark for comparing forecast performance of new models and methods. VARs are also used to better understand the interactions between macroeconomic variables, often through the estimation of impulse response functions that characterize the effects of a variety of structural shocks on key economic variables.

Despite the empirical success of the standard constant-coefficient and homoscedastic VAR, there is a lot of recent work in extending these conventional VARs to models with time-varying regression coefficients and stochastic volatility. These extensions are motivated by the widely observed structural instabilities and time-varying volatility in a variety of macroeconomic time series.

In this chapter we will study a few of these more flexible VARs, including the time-varying parameter (TVP) VAR and VARs with stochastic volatility. An excellent review paper that covers many of the same topics is Koop and Korobilis (2010). We will begin with a basic VAR.

8.1 Basic Vector Autoregression

Suppose $\mathbf{y}_t = (y_{1t}, \ldots, y_{nt})'$ is a vector of dependent variables at time t. Consider the following VAR(p):

$$\mathbf{y}_t = \mathbf{b} + \mathbf{A}_1\mathbf{y}_{t-1} + \cdots + \mathbf{A}_p\mathbf{y}_{t-p} + \boldsymbol{\varepsilon}_t, \tag{8.1}$$

where $\mathbf{b}$ is an $n \times 1$ vector of intercepts, $\mathbf{A}_1, \ldots, \mathbf{A}_p$ are $n \times n$ coefficient matrices and $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. In other words, VAR($p$) is simply a multiple-equation regression

where the regressors are the lagged dependent variables.

To fix ideas, consider a simple example with $n = 2$ variables and $p = 1$ lag. Then, (8.1) can be written explicitly as:

$$\begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} + \begin{pmatrix} A_{1,11} & A_{1,12} \\ A_{1,21} & A_{1,22} \end{pmatrix} \begin{pmatrix} y_{1(t-1)} \\ y_{2(t-1)} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix},$$

where

$$\begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} \right).$$

The model in (8.1) runs from $t = 1, \ldots, T$, and it depends on the p initial conditions $\mathbf{y}_{-p+1}, \ldots, \mathbf{y}_0$. In principle these initial conditions can be modeled explicitly. Here all the analysis is done conditioned on these initial conditions. If the series is sufficiently long (e.g., $T > 50$), both approaches typically give essentially the same results.

8.1.1 Likelihood

To derive the likelihood for the VAR(p) in (8.1), we aim to write the system as the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

Then, we can simply apply the linear regression results to derive the likelihood.

Let's first work out our example with $n = 2$ variables and $p = 1$ lag. To that end, we stack the coefficients equation by equation, i.e., $\boldsymbol{\beta} = (b_1, A_{1,11}, A_{1,12}, b_2, A_{1,21}, A_{1,22})'$. Equivalent, we can write it using the vec operator that vectorizes a matrix by its columns: $\boldsymbol{\beta} = \text{vec}([\mathbf{b}, \mathbf{A}_1]')$. Given our definition of $\boldsymbol{\beta}$, we can easily work out the corresponding regression matrix $\mathbf{X}_t$:

$$\begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} 1 & y_{1(t-1)} & y_{2(t-1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & y_{1(t-1)} & y_{2(t-1)} \end{pmatrix} \begin{pmatrix} b_1 \\ A_{1,11} \\ A_{1,12} \\ b_2 \\ A_{1,21} \\ A_{1,22} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix}.$$

Or

$$\mathbf{y}_t = \left(\mathbf{I}_2 \otimes [1, \mathbf{y}_{t-1}']\right) \boldsymbol{\beta} + \boldsymbol{\varepsilon}_t,$$

where $\otimes$ is the Kronecker product.

More generally, we can write the VAR(p) as:

$$\mathbf{y}_t = \mathbf{X}_t\boldsymbol{\beta} + \boldsymbol{\varepsilon}_t,$$

where $\mathbf{X}_t = \mathbf{I}_n \otimes [1, \mathbf{y}'_{t-1}, \ldots, \mathbf{y}'_{t-p}]$ and $\boldsymbol{\beta} = \text{vec}([\mathbf{b}, \mathbf{A}_1, \cdots \mathbf{A}_p]')$. Then, stack the observations over $t = 1, \ldots, T$ to get

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_T \otimes \boldsymbol{\Sigma})$.

Now, since

$$(\mathbf{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\Sigma}) \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \mathbf{I}_T \otimes \boldsymbol{\Sigma}),$$

the likelihood function is given by:

$$\begin{aligned} p(\mathbf{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\Sigma}) &= |2\pi(\mathbf{I}_T \otimes \boldsymbol{\Sigma})|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{I}_T \otimes \boldsymbol{\Sigma})^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})} \\ &= (2\pi)^{-\frac{Tn}{2}} |\boldsymbol{\Sigma}|^{-\frac{T}{2}} \mathrm{e}^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{I}_T \otimes \boldsymbol{\Sigma}^{-1})(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}, \end{aligned} \tag{8.2}$$

where the second equality holds because $|\mathbf{I}_T \otimes \boldsymbol{\Sigma}| = |\boldsymbol{\Sigma}|^T$ and $(\mathbf{I}_T \otimes \boldsymbol{\Sigma})^{-1} = \mathbf{I}_T \otimes \boldsymbol{\Sigma}^{-1}$. Note that the likelihood can also be written as

$$p(\mathbf{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{Tn}{2}} |\boldsymbol{\Sigma}|^{-\frac{T}{2}} \mathrm{e}^{-\frac{1}{2}\sum_{t=1}^{T}(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta})'\boldsymbol{\Sigma}^{-1}(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta})}. \tag{8.3}$$

8.1.2 Independent Priors

Recall that in the normal linear regression model in Chapter 2, we assume independent normal and inverse-gamma priors for the coefficients $\boldsymbol{\beta}$ and the variance σ^2, respectively. Both are conjugate priors and the model can be easily estimated using the Gibbs sampler.

Here a similar result applies. But instead of an inverse-gamma prior, we need a multivariate generalization for the covariance matrix $\boldsymbol{\Sigma}$.

An $m \times m$ random matrix $\mathbf{Z}$ is said to have an **inverse-Wishart distribution** with shape parameter $\alpha > 0$ and scale matrix $\mathbf{W}$ if its density function is given by

$$f(\mathbf{Z}; \alpha, \mathbf{W}) = \frac{|\mathbf{W}|^{\alpha/2}}{2^{m\alpha/2}\Gamma_m(\alpha/2)} |\mathbf{Z}|^{-\frac{\alpha+m+1}{2}} \mathrm{e}^{-\frac{1}{2}\text{tr}(\mathbf{W}\mathbf{Z}^{-1})},$$

where Γ_m is the multivariate gamma function and $\text{tr}(\cdot)$ is the trace function. We write $\mathbf{Z} \sim \mathcal{I}W(\alpha, \mathbf{W})$. For $\alpha > m + 1$, $\mathbb{E}\mathbf{Z} = \mathbf{W}/(\alpha - m - 1)$.

For the VAR(p) with parameters $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$, we consider the independent priors:

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \mathbf{V}_{\boldsymbol{\beta}}), \quad \boldsymbol{\Sigma} \sim \mathcal{I}W(\nu_0, \mathbf{S}_0). \tag{8.4}$$

8.1.3 Gibbs Sampler

Now, we derive a Gibbs sampler for the VAR(p) with likelihood given in (8.2) and priors given in (8.4). Specifically, we derive the two conditional densities $p(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\Sigma})$ and $p(\boldsymbol{\Sigma} \,|\, \mathbf{y}, \boldsymbol{\beta})$.

The first step is easy, as standard linear regression results would apply. In fact, we have

$$(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\Sigma}) \sim \mathcal{N}(\widehat{\boldsymbol{\beta}}, \mathbf{K}_{\boldsymbol{\beta}}^{-1}),$$

where

$$\mathbf{K}_{\boldsymbol{\beta}} = \mathbf{V}_{\boldsymbol{\beta}}^{-1} + \mathbf{X}'(\mathbf{I}_T \otimes \boldsymbol{\Sigma}^{-1})\mathbf{X}, \quad \widehat{\boldsymbol{\beta}} = \mathbf{K}_{\boldsymbol{\beta}}^{-1}\left(\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_0 + \mathbf{X}'(\mathbf{I}_T \otimes \boldsymbol{\Sigma}^{-1})\mathbf{y}\right),$$

and we have used the result $(\mathbf{I}_T \otimes \boldsymbol{\Sigma})^{-1} = \mathbf{I}_T \otimes \boldsymbol{\Sigma}^{-1}$.

Next, we derive the conditional density $p(\boldsymbol{\Sigma} \,|\, \mathbf{y}, \boldsymbol{\beta})$. Recall that for conformable matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, we have

$$\operatorname{tr}(\mathbf{ABC}) = \operatorname{tr}(\mathbf{BCA}) = \operatorname{tr}(\mathbf{CAB}).$$

Now, combining the likelihood (8.3) and the prior (8.4), we obtain

$$\begin{aligned}
p(\boldsymbol{\Sigma} \,|\, \mathbf{y}, \boldsymbol{\beta}) &\propto p(\mathbf{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\Sigma}) p(\boldsymbol{\Sigma}) \\
&\propto |\boldsymbol{\Sigma}|^{-\frac{T}{2}} \mathrm{e}^{-\frac{1}{2}\sum_{t=1}^{T}(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta})'\boldsymbol{\Sigma}^{-1}(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta})} \times |\boldsymbol{\Sigma}|^{-\frac{\nu_0+n+1}{2}} \mathrm{e}^{-\frac{1}{2}\operatorname{tr}(\mathbf{S}_0\boldsymbol{\Sigma}^{-1})} \\
&\propto |\boldsymbol{\Sigma}|^{-\frac{\nu_0+n+T+1}{2}} \mathrm{e}^{-\frac{1}{2}\operatorname{tr}(\mathbf{S}_0\boldsymbol{\Sigma}^{-1})} \mathrm{e}^{-\frac{1}{2}\operatorname{tr}\left[\sum_{t=1}^{T}(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta})(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta})'\boldsymbol{\Sigma}^{-1}\right]} \\
&\propto |\boldsymbol{\Sigma}|^{-\frac{\nu_0+n+T+1}{2}} \mathrm{e}^{-\frac{1}{2}\operatorname{tr}\left[\left(\mathbf{S}_0 + \sum_{t=1}^{T}(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta})(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta})'\right)\boldsymbol{\Sigma}^{-1}\right]},
\end{aligned}$$

which is the kernel of an inverse-Wishart density. In fact, we have

$$(\boldsymbol{\Sigma} \,|\, \mathbf{y}, \boldsymbol{\beta}) \sim \mathcal{IW}\left(\nu_0 + T, \mathbf{S}_0 + \sum_{t=1}^{T}(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta})(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta})'\right).$$

We summarize the Gibbs sampler as follows

Algorithm 8.1. (Gibbs Sampler for the VAR(p)).

Pick some initial values $\boldsymbol{\beta}^{(0)} = \mathbf{c}_0$ and $\boldsymbol{\Sigma}^{(0)} = \mathbf{C}_0 > 0$. Then, repeat the following steps from $r = 1$ to R:

1. Draw $\boldsymbol{\beta}^{(r)} \sim p(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\Sigma}^{(r-1)})$ (multivariate normal).
2. Draw $\boldsymbol{\Sigma}^{(r)} \sim p(\boldsymbol{\Sigma} \,|\, \mathbf{y}, \boldsymbol{\beta}^{(r)})$ (inverse-Wishart).

8.1.4 Empirical Example: Small Model of the US Economy

In this empirical example we estimate a 3-variable VAR(2) using US quarterly data on CPI inflation rate, unemployment rate and Fed Funds rate from 1959Q1 to 2007Q4—the sample ends at 2007Q4 to avoid the periods when interest rate hits the zero lower bound. These three variables are commonly used in forecasting (e.g., Banbura, Giannone and Reichlin, 2010; Koop and Korobilis, 2010; Koop, 2013) and small DSGE models (e.g., An and Schorfheide, 2007).

Following Primiceri (2005), we order the interest rate last and treat it as the monetary policy instrument. The identified monetary policy shocks are interpreted as "non-systematic policy actions" that capture both policy mistakes and interest rate movements that are responses to variables other than inflation and unemployment.

We first implement the Gibbs sampler described in Algorithm 8.1. Then, given the posterior draws of $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$, we compute the impulse-response functions of the three variables to a 100-basis-point monetary policy shock.

To estimate the model, we use the MATLAB script `SURform2.m` below to construct the regression matrix $\mathbf{X}$. Recall that $\mathbf{X}_t = \mathbf{I}_n \otimes [1, \mathbf{y}'_{t-1}, \ldots, \mathbf{y}'_{t-p}]$ and we stack $\mathbf{X}_t$ over $t = 1, \ldots, T$ to obtain $\mathbf{X}$.

```
% SURform2.m
function X_out = SURform2(X, n)
    repX = kron(X,ones(n,1));
    [r,c] = size( X );
    idi = kron((1:r*n)',ones(c,1));
    idj = repmat((1:n*c)',r,1);
    X_out = sparse(idi,idj,reshape(repX',n*r*c,1));
end
```

Given the posterior draws of $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$, we then use the script `construct_IR.m` to compute the impulse-response functions of the three variables to a 100-basis-point monetary policy shock. More specifically, we consider two alternative paths: in one a 100-basis-point monetary policy shock hits the system, and in the other this shock is absent. We then let the two systems evolve according to the VAR(p) written as the regression

$$\mathbf{y}_t = \mathbf{X}_t\boldsymbol{\beta}_t + \mathbf{C}\widetilde{\boldsymbol{\varepsilon}}_t, \quad \widetilde{\boldsymbol{\varepsilon}}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_3),$$

for $t = 1, \ldots, n_{\text{hz}}$, where n_{hz} denotes the number of horizons and $\mathbf{C}$ is the Cholesky factor of $\boldsymbol{\Sigma}$. Each impulse-response function is then the difference between these two paths.

The MATLAB script `construct_IR.m` is given below:

```
function yIR = construct_IR(beta,Sig,n_hz,shock)
    n = size(Sig,1);
    p = (size(beta,1)/n-1)/n;
    CSig = chol(Sig,'lower');
    tmpZ1 = zeros(p,n); tmpZ = zeros(p,n);
    Yt1 = CSig*shock; Yt = zeros(n,1);
    yIR = zeros(n_hz,n); yIR(1,:) = Yt1';
    for t = 2:n_hz
            % update the regressors
        tmpZ = [Yt'; tmpZ(1:end-1,:)];
        tmpZ1 = [Yt1'; tmpZ1(1:end-1,:)];
            % evolution of variables if a shock hits
        e = CSig*randn(n,1);
        Z1 = reshape(tmpZ1',1,n*p);
        Xt1 = kron(speye(n), [1 Z1]);
        Yt1 = Xt1*beta + e;
            % evolution of variables if no shocks hit
        Z = reshape(tmpZ',1,n*p);
        Xt = kron(speye(n), [1 Z]);
        Yt = Xt*beta + e;
            % the IR is the difference of the two scenarios
        yIR(t,:) = (Yt1-Yt)';
    end
end
```

The main script `VAR.m` is given next. It first loads the dataset `data_Q.csv`, constructs the regression matrix $\mathbf{X}$ using the above function, and then implements the 2-block Gibbs sampler. Note that within the for-loop we compute the impulse-response functions for each set of posterior draws $(\boldsymbol{\beta}, \boldsymbol{\Sigma})$. Also notice that the shock variable `shock` is normalized so that the impulse responses are to 100 basis points rather than one standard deviation of the monetary policy shock.

```
% VAR.m
p = 2;          % if p > 4, need to change Y0 and Y below
nsim = 20000;  burnin = 1000;

    % load data
load 'data_Q.csv'; % 1959Q1 to 2013Q4
    % unemployment, inflation and interest rate; 1959Q1-2007Q4
data = data_Q(1:196,[10 2 3]);
Y0 = data(1:4,:);  % save the first 4 obs as the initial conditions
Y = data(5:end,:);
```

```
[T n] = size(Y);
y = reshape(Y',T*n,1);
k = n*p+1;          % # of coefficients in each equation
n_hz = 20;          % # of horizons for IRs
    % prior
nu0 = n+3; S0 = eye(n);
beta0 = zeros(n*k,1);
    % precision for coefficients = 1; for intercepts = 1/10
tmp = ones(k*n,1);  tmp(1:p*n+1:k*n) = 1/10;
iVbeta = sparse(1:k*n,1:k*n,tmp);
    % compute and define a few things
tmpY = [Y0(end-p+1:end,:); Y];
X_tilde = zeros(T,n*p);
for i=1:p
    X_tilde(:,(i-1)*n+1:i*n) = tmpY(p-i+1:end-i,:);
end
X_tilde = [ones(T,1) X_tilde];
X = SURform2(X_tilde,n);

    % initialize for storage
store_Sig = zeros(nsim,n,n);
store_beta = zeros(nsim,k*n);
store_yIR = zeros(n_hz,n);

    %% initialize the chain
beta = (X'*X)\(X'*y);
e = reshape(y - X*beta,n,T);
Sig = e*e'/T;  iSig = Sig\speye(n);

for isim = 1:nsim + burnin
        % sample beta
    XiSig = X'*kron(speye(T),iSig);
    XiSigX = XiSig*X;
    Kbeta = iVbeta + XiSigX;
    beta_hat = Kbeta\(iVbeta*beta0 + XiSig*y);
    beta =  beta_hat + chol(Kbeta,'lower')'\randn(n*k,1);

        % sample Sig
    e = reshape(y - X*beta,n,T);
    Sig = iwishrnd(S0 + e*e',nu0 + T);
    iSig = Sig\speye(n);

        % store the parameters
```

```
    if isim > burnin
        isave = isim - burnin;
        store_beta(isave,:) = beta';
        store_Sig(isave,:,:) = Sig;

            % compute impulse-responses
        CSig = chol(Sig,'lower');
                % 100 basis pts rather than 1 std. dev.
        shock = [0; 0; 1]/CSig(n,n);
        yIR = construct_IR(beta,Sig,n_hz,shock,X_T);
        store_yIR = store_yIR + yIR;
    end
end
yIR_hat = store_yIR/nsim;
```

The impulse-response functions of inflation, unemployment and interest rate to a 100-basis-point monetary policy shock are given in Figure 8.1.

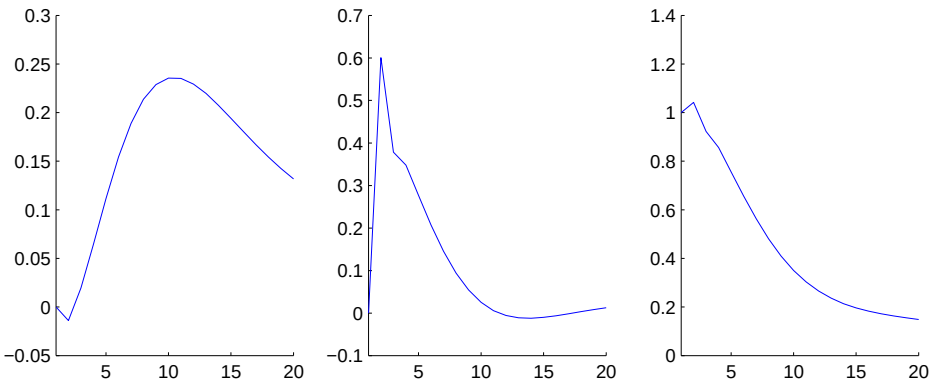

Figure 8.1: Impulse-response functions of unemployment (left panel), inflation (middle panel) and interest rate (right panel) to a 100-basis-point monetary policy shock.

By construction, the interest rate increases by 1% on impact, and it gradually goes back to zero over the next 20 quarters. The impact on unemployment is delayed and positive—the unemployment rate rises slowly and reaches a peak of about 0.25% after 3 years on impact.

In contrast, the inflation rate rises shortly on impact, and the effect is positive. The literature often refers to this empirical finding as the price puzzle—a contractionary monetary policy shock should dampen rather than increase inflation. Some argue that this reflects misspecification—some crucial variables are omitted, and both interest rate and inflation are responding to these omitted variables. Alternatively,

Primiceri (2005) argues that the monetary policy shocks identified in this context should be interpreted as "non-systematic policy actions" instead.

8.2 Time-Varying Parameter VAR

In this section we consider the **time-varying parameter vector autoregression** (TVP-VAR) in which the VAR coefficients gradually evolve over time. The time-varying parameter model permits the structure of the economy to change over time, while maintaining that such changes in dynamic behavior should occur smoothly. This framework allows us to study potential structural instability due to new technology, regulations and institutional changes.

As mentioned in Chapter 6, unobserved components models can be viewed as a regression with a time-varying parameter. This line of work can be traced back to the seminar paper of Harvey (1985). Canova (1993) considers a regression with time-varying coefficients and the paper by Cogley and Sargent (2001) appears to be the first VAR with time-varying parameters.

Consider again the VAR(p) but now with time-varying parameters:

$$\mathbf{y}_t = \mathbf{a}_t + \mathbf{A}_{1t}\mathbf{y}_{t-1} + \cdots + \mathbf{A}_{pt}\mathbf{y}_{t-p} + \boldsymbol{\varepsilon}_t, \tag{8.5}$$

where $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$.

As before, we can define $\mathbf{X}_t = \mathbf{I}_n \otimes [1, \mathbf{y}'_{t-1}, \ldots, \mathbf{y}'_{t-p}]$ and $\boldsymbol{\beta}_t = \text{vec}([\mathbf{a}_t,\ \mathbf{A}_{1t},\ \ldots,\ \mathbf{A}_{pt}]')$ and rewrite the above system as

$$\mathbf{y}_t = \mathbf{X}_t\boldsymbol{\beta}_t + \boldsymbol{\varepsilon}_t.$$

The time-varying parameters $\boldsymbol{\beta}_t$ are assumed to evolve as a random walk

$$\boldsymbol{\beta}_t = \boldsymbol{\beta}_{t-1} + \mathbf{u}_t, \tag{8.6}$$

where $\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and the initial conditions $\boldsymbol{\beta}_0$ are treated as parameters. Here we make the simplifying assumption that the covariance matrix $\mathbf{Q}$ is diagonal, i.e., $\mathbf{Q} = \text{diag}(q_1, \ldots, q_{kn})$. One can also consider the possibility of a block-diagonal matrix or even a full matrix.

To complete the model specification, consider independent priors for $\boldsymbol{\Sigma}$, $\boldsymbol{\beta}_0$ and the diagonal elements of $\mathbf{Q}$:

$$\boldsymbol{\Sigma} \sim \mathcal{I}W(\nu_0, \mathbf{S}_0), \quad \boldsymbol{\beta}_0 \sim \mathcal{N}(\mathbf{a}_0, \mathbf{B}_0), \quad q_i \sim \mathcal{I}G(\nu_{0,q_i}, S_{0,q_i}).$$

8.2.1 Estimation

In this section we describe a Gibbs sampler to estimate the TVP-VAR. The model parameters are $\boldsymbol{\beta}_0$, $\boldsymbol{\Sigma}$ and $\mathbf{Q}$, and the states are $\boldsymbol{\beta} = (\boldsymbol{\beta}_1', \ldots, \boldsymbol{\beta}_T')'$. We therefore consider a 4-block Gibbs sampler.

First, to sample $\boldsymbol{\beta}$, we rewrite the observation equation (8.5) as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_T \otimes \boldsymbol{\Sigma})$ and

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{X}_T \end{pmatrix}.$$

Hence, we have

$$(\mathbf{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\Sigma}) \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \mathbf{I}_T \otimes \boldsymbol{\Sigma}).$$

So we have reframed the TVP-VAR as a normal linear regression model. Next, we derive the prior of $\boldsymbol{\beta}$. To that end, rewrite the state equation (8.6) in matrix notation:

$$\mathbf{H}\boldsymbol{\beta} = \widetilde{\boldsymbol{\alpha}}_{\boldsymbol{\beta}} + \mathbf{u},$$

where $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_T \otimes \mathbf{Q})$, $\widetilde{\boldsymbol{\alpha}}_{\boldsymbol{\beta}} = (\boldsymbol{\beta}_0', \mathbf{0}, \ldots, \mathbf{0})'$ and

$$\mathbf{H} = \begin{pmatrix} \mathbf{I}_{nk} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{I}_{nk} & \mathbf{I}_{nk} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{nk} & \mathbf{I}_{nk} & \cdots & \mathbf{0} \\ \vdots & & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{I}_{nk} & \mathbf{I}_{nk} \end{pmatrix}.$$

Note that now $\mathbf{H}$ is of dimension $Tnk \times Tnk$, and is a multivariate generalization of the usual first difference matrix. Again $|\mathbf{H}| = 1$, and is therefore invertible. Using a similar argument as in Section 6.1.1, one can show that $\mathbf{H}^{-1}\widetilde{\boldsymbol{\alpha}}_{\boldsymbol{\beta}} = \mathbf{1}_T \otimes \boldsymbol{\beta}_0$. Therefore, the prior of $\boldsymbol{\beta}$ is given by

$$(\boldsymbol{\beta} \,|\, \boldsymbol{\beta}_0, \mathbf{Q}) \sim \mathcal{N}(\mathbf{1}_T \otimes \boldsymbol{\beta}_0, (\mathbf{H}'(\mathbf{I}_T \otimes \mathbf{Q}^{-1})\mathbf{H})^{-1}).$$

Finally, by standard linear regression results, we obtain

$$(\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\Sigma}, \boldsymbol{\beta}_0, \mathbf{Q}) \sim \mathcal{N}(\widehat{\boldsymbol{\beta}}, \mathbf{K}_{\boldsymbol{\beta}}^{-1}),$$

where

$$\begin{aligned} \mathbf{K}_{\boldsymbol{\beta}} &= \mathbf{H}'(\mathbf{I}_T \otimes \mathbf{Q}^{-1})\mathbf{H} + \mathbf{X}'(\mathbf{I}_T \otimes \boldsymbol{\Sigma}^{-1})\mathbf{X}, \\ \widehat{\boldsymbol{\beta}} &= \mathbf{K}_{\boldsymbol{\beta}}^{-1}\left(\mathbf{H}'(\mathbf{I}_T \otimes \mathbf{Q}^{-1})\mathbf{H}(\mathbf{1}_T \otimes \boldsymbol{\beta}_0) + \mathbf{X}'(\mathbf{I}_T \otimes \boldsymbol{\Sigma}^{-1})\mathbf{y}\right). \end{aligned}$$

Since $\mathbf{K}_{\boldsymbol{\beta}}$ is again a band matrix, we can use the precision sampler as described in Algorithm 6.1 to sample $\boldsymbol{\beta}$ efficiently.

Next, using a similar derivation as in Section 8.1.3, we can show that

$$(\boldsymbol{\Sigma} \,|\, \mathbf{y}, \boldsymbol{\beta}, \mathbf{Q}) \sim \mathcal{I}W\left(\nu_0 + T, \mathbf{S}_0 + \sum_{t=1}^{T} (\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta}_t)(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta}_t)'\right).$$

In addition, each of the diagonal elements of $\mathbf{Q} = \text{diag}(q_1, \ldots, q_k)$ has an inverse-gamma distribution:

$$(q_i \,|\, \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\beta}_0) \sim \mathcal{I}G\left(\nu_{0,q_i} + \frac{T}{2}, S_{0,q_i} + \frac{1}{2}\sum_{t=1}^{T} (\beta_{it} - \beta_{i(t-1)})^2\right).$$

Finally, since $\boldsymbol{\beta}_0$ only appears in the first state equation

$$\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 + \mathbf{u}_1,$$

where $\mathbf{u}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. Given the normal prior $\boldsymbol{\beta}_0 \sim \mathcal{N}(\mathbf{a}_0, \mathbf{B}_0)$, we can use standard linear regression results to get

$$(\boldsymbol{\beta}_0 \,|\, \mathbf{y}, \boldsymbol{\beta}, \mathbf{Q}) \sim \mathcal{N}(\widehat{\boldsymbol{\beta}}_0, \mathbf{K}_{\boldsymbol{\beta}_0}^{-1}),$$

where

$$\mathbf{K}_{\boldsymbol{\beta}_0} = \mathbf{B}_0^{-1} + \mathbf{Q}^{-1}, \quad \widehat{\boldsymbol{\beta}}_0 = \mathbf{K}_{\boldsymbol{\beta}_0}^{-1}\left(\mathbf{B}_0^{-1}\mathbf{a}_0 + \mathbf{Q}^{-1}\boldsymbol{\beta}_1\right).$$

We summarize the Gibbs sampler as follows:

Algorithm 8.2. (Gibbs Sampler for the TVP-VAR(p)).

Pick some initial values for $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$, $\mathbf{Q}^{(0)}$ and $\boldsymbol{\beta}_0^{(0)}$. Then, repeat the following steps from $r = 1$ to R:

1. Draw $\boldsymbol{\beta}^{(r)} \sim (\boldsymbol{\beta} \,|\, \mathbf{y}, \boldsymbol{\Sigma}^{(r-1)}, \mathbf{Q}^{(r-1)}, \boldsymbol{\beta}_0^{(r-1)})$ (multivariate normal).

2. Draw $\boldsymbol{\Sigma}^{(r)} \sim (\boldsymbol{\Sigma} \,|\, \mathbf{y}, \boldsymbol{\beta}^{(r)}, \mathbf{Q}^{(r-1)}, \boldsymbol{\beta}_0^{(r-1)})$ (inverse-Wishart).

3. Draw $\mathbf{Q}^{(r)} \sim (\mathbf{Q} \,|\, \mathbf{y}, \boldsymbol{\beta}^{(r)}, \boldsymbol{\Sigma}^{(r)}, \boldsymbol{\beta}_0^{(r-1)})$ (independent inverse-gammas).

4. Draw $\boldsymbol{\beta}_0^{(r)} \sim (\boldsymbol{\beta}_0 \,|\, \mathbf{y}, \boldsymbol{\beta}^{(r)}, \boldsymbol{\Sigma}^{(r)}, \mathbf{Q}^{(r)})$ (multivariate normal).

8.2.2 Empirical Example: Small Model of the US Economy Revisited

In the empirical example in Section 8.1.4, we consider a 3-variable VAR(2) of unemployment, inflation and interest rate, and use it to compute impulse-response functions to a 100-basis-point monetary policy shock. Given the sample spans from 1959Q1 to 2007Q4, one might wonder if these the responses to the monetary policy shocks have changed over time.

To address that question, here we revisit that example using a time-varying parameter VAR. In particular, we compare the impulse responses associated with the VAR coefficients in 1975 to those in 2005.

The following MATLAB script `VAR_TVP.m` implements the Gibbs sampler in Algorithm 8.2. In addition, it computes two sets of impulse responses by picking the relevant $\boldsymbol{\beta}_t$ and using the script `construct_IR.m`.

```
for isim = 1:nsim + burnin
        % sample beta
    HiQH = H'*sparse(1:T*n*k,1:T*n*k,repmat(1./Q,T,1))*H;
    XiSig = X'*kron(speye(T),iSig);
    Kbeta = HiQH + XiSig*X;
    beta_hat = Kbeta\(HiQH*kron(ones(T,1),beta0) + XiSig*y);
    beta =  beta_hat + chol(Kbeta,'lower')'\randn(T*n*k,1);

        % sample Sig
    e = reshape(y - X*beta,n,T);
    Sig = iwishrnd(S0 + e*e',nu0 + T);
    iSig = Sig\speye(n);

        % sample Q
    e = reshape(beta - [beta0;beta(1:end-n*k)],n*k,T);
    Q = 1./gamrnd(nu0q + T/2,1./(S0q + sum(e.^2,2)/2));

        % sample beta0
    Kbeta0 = iB0 + sparse(1:n*k,1:n*k,1./Q);
    beta0_hat = Kbeta0\(iB0*a0 + sparse(1:n*k,1:n*k,1./Q)*beta(1:n*k));
    beta0 =  beta0_hat + chol(Kbeta0,'lower')'\randn(n*k,1);

        % store the parameters
    if isim > burnin
        isave = isim - burnin;
        store_beta(isave,:) = beta';
```

```
        store_Sig(isave,:,:) = Sig;
        store_Q(isave,:) = Q';

           % compute impulse-responses
        CSig = chol(Sig,'lower');
               % 100 basis pts rather than 1 std. dev.
        shock = [0; 0; 1]/CSig(n,n);
        tmp_beta = reshape(beta,n*k,T);
        yIR_75 = construct_IR(tmp_beta(:,t0(1)),Sig,n_hz,shock);
        yIR_05 = construct_IR(tmp_beta(:,t0(2)),Sig,n_hz,shock);

        store_yIR_75 = store_yIR_75 + yIR_75;
        store_yIR_05 = store_yIR_05 + yIR_05;
        store_diff(isave,:,:) = yIR_05 - yIR_75;
    end
end
```

Figure 8.2 reports two sets of impulse-response functions to a 100-basis-point monetary policy shock. The first set is computed using the VAR coefficients at 1975Q1; the second set uses those at 2005Q1. As the figure shows, the impulse-response functions of both inflation and interest rate are very similar across the two time periods, whereas those of the unemployment seem to be more different. In particular, the response of unemployment to monetary policy shock is much more muted in 2005 compared to 1975.

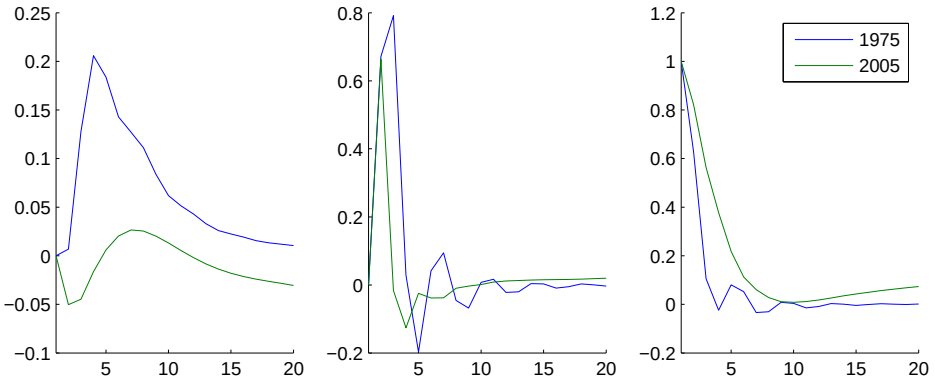

Figure 8.2: Impulse-response functions of unemployment (left panel), inflation (middle panel) and interest rate (right panel) to a 100-basis-point monetary policy shock.

We report in Figure 8.3 the mean differences between the two sets of impulse-response functions of the three variables, as well as the associated 90% credible

intervals. Despite some of the large absolute differences, parameter uncertainty is high and most of the credible intervals contain zero. There seems to be some evidence that the responses of unemployment are different across the two periods, but the evidence is not conclusive.

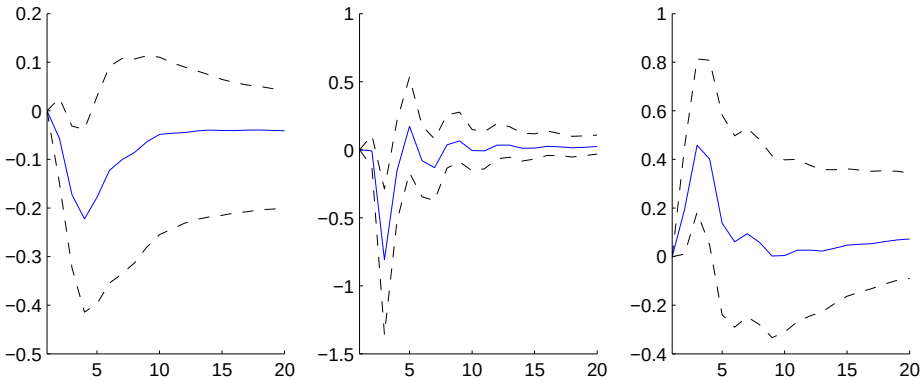

Figure 8.3: Differences between the impulse-response functions of unemployment (left panel), inflation (middle panel) and interest rate (right panel) at 2005Q1 and 1975Q1. The dotted lines are the 5% and 95% quantiles.

8.3 VAR with Stochastic Volatility

In this section we study how one can extend the standard VAR with homoscedastic errors to one with stochastic volatility. We start with a constant-coefficient VAR in (8.1) and write it as a linear regression in which the errors have a time-varying covariance matrix $\boldsymbol{\Sigma}_t$:

$$\mathbf{y}_t = \mathbf{X}_t\boldsymbol{\beta} + \boldsymbol{\varepsilon}_t, \quad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_t). \tag{8.7}$$

Ideally we want $\boldsymbol{\Sigma}_t$ to evolve smoothly, while at each time period $\boldsymbol{\Sigma}_t$ is a valid covariance matrix—i.e., it is symmetric and positive definite. There are a few approaches to model such an evolution, and here we follow the approach in the seminal paper by Cogley and Sargent (2005).

The idea is to model $\boldsymbol{\Sigma}_t$ as

$$\boldsymbol{\Sigma}_t^{-1} = \mathbf{L}'\mathbf{D}_t^{-1}\mathbf{L},$$

where $\mathbf{D}_t$ is a diagonal matrix and $\mathbf{L}$ is a lower triangular matrix with ones on the

main diagonal, i.e.,

$$\mathbf{D}_t = \begin{pmatrix} \mathrm{e}^{h_{1t}} & 0 & \cdots & 0 \\ 0 & \mathrm{e}^{h_{2t}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathrm{e}^{h_{nt}} \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_{21} & 1 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 1 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{n(n-1)} & 1 \end{pmatrix}.$$

By construction $\mathbf{\Sigma}_t$ is symmetric and positive definite for any values of $\mathbf{h}_t = (h_{1t}, \ldots, h_{nt})'$ and $\mathbf{a} = (a_{21}, a_{31}, a_{32}, \ldots, a_{n1}, \ldots, a_{n(n-1)})'$. Note that the dimension of $\mathbf{a}$ is $m = n(n-1)/2$. Then, each h_{it} is specified independently using a univariate stochastic volatility model as in Chapter 7. More precisely, each h_{it} evolves according to the following random walk

$$h_{it} = h_{i(t-1)} + u_{it}^h,$$

where $u_{it}^h \sim \mathcal{N}(0, \sigma_{h,i}^2)$ and h_{i0} is treated as an unknown parameter.

In contrast, the parameters $\mathbf{a}$ are restricted to be constant here. Primiceri (2005) considers an extension where these parameters are time-varying and modeled as random walks. It turns out all that requires is an extra block to sample these time-varying parameters from a linear Gaussian state space model.

By construction, $\mathbf{\Sigma}_t = \mathbf{L}^{-1}\mathbf{D}_t(\mathbf{L}^{-1})'$, and therefore we can express each element of $\mathbf{\Sigma}_t$ in terms of the elements of $\mathbf{D}_t$ and $\mathbf{L}^{-1} = (a^{ij})$. More precisely, we have

$$\begin{aligned} \sigma_{ii,t} &= \mathrm{e}^{h_{it}} + \sum_{k=1}^{i-1} \mathrm{e}^{h_{kt}}(a^{ik})^2, \quad i = 1, \ldots, n, \\ \sigma_{ij,t} &= a^{ij}\mathrm{e}^{h_{jt}} + \sum_{k=1}^{j-1} a^{ik}a^{jk}\mathrm{e}^{h_{kt}}, \quad 1 \leqslant j < i \leqslant n, \end{aligned}$$

where $\sigma_{ij,t}$ is the (i,j) element of $\mathbf{\Sigma}_t$. In particular, the log-volatility h_{1t} affects the variances of all the variables, whereas h_{nt} impacts only the last variable.

In addition, despite the assumption of a constant matrix $\mathbf{L}$, this setup also allows for some form of time-varying correlations among the innovations. This can be seen via a simple example. Using the formulas above, we have

$$\sigma_{11,t} = \mathrm{e}^{h_{1t}}, \quad \sigma_{22,t} = \mathrm{e}^{h_{2t}} + \mathrm{e}^{h_{1t}}(a^{21})^2, \quad \sigma_{12,t} = a^{21}\mathrm{e}^{h_{1t}}.$$

We have used the fact that $\mathbf{L}^{-1}$ is a lower triangular matrix with ones on the main diagonal, and therefore $a^{11} = 1$ and $a^{12} = 0$. Now, the (1,2) correlation coefficient is given by

$$\frac{\sigma_{12,t}}{\sqrt{\sigma_{11,t}\sigma_{22,t}}} = \frac{a^{21}}{\sqrt{\mathrm{e}^{h_{2t}-h_{1t}} + (a^{21})^2}}.$$

Hence, as long as h_{1t} and h_{2t} are not identical for all t, this correlation coefficient is time-varying.

To complete the model specification, consider independent priors for $\boldsymbol{\beta}$, $\mathbf{a}$, $\boldsymbol{\sigma}_h^2 = (\sigma_{h,1}^2, \dots, \sigma_{h,n}^2)'$ and $\mathbf{h}_0 = (h_{10}, \dots, h_{n0})'$:

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \mathbf{V}_{\boldsymbol{\beta}}), \quad \mathbf{a} \sim \mathcal{N}(\mathbf{a}_0, \mathbf{V}_{\mathbf{a}}), \quad \sigma_{h,i}^2 \sim \mathcal{IG}(\nu_{0,h_i}, S_{0,h_i}), \quad \mathbf{h}_0 \sim \mathcal{N}(\mathbf{b}_0, \mathbf{B}_0).$$

8.3.1 Estimation

To estimate the VAR-SV model, we describe below a Gibbs sampler that builds upon the auxiliary mixture sampler introduced in Chapter 7.

The model parameters are $\boldsymbol{\beta}$, $\mathbf{a}$, $\boldsymbol{\sigma}_h^2$ and $\mathbf{h}_0$, and the states are the log-volatility $\mathbf{h}_{i,1:T} = (h_{i1}, \dots, h_{iT})'$. Hence, we consider a 5-block Gibbs sampler. The two key steps are sampling of $\mathbf{a}$ and $\mathbf{h} = (\mathbf{h}_{1,1:T}', \dots, \mathbf{h}_{n,1:T}')'$, and we will describe them in detail below.

We begin with the sampling of $\mathbf{a}$, the lower triangular elements of $\mathbf{L}$. First observe that given $\mathbf{y}$ and $\boldsymbol{\beta}$, $\boldsymbol{\varepsilon} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$ is known. Then, we rewrite the model as a system of regressions in which ε_{it} is regressed on the negative values of $\varepsilon_{1t}, \dots, \varepsilon_{(i-1)t}$ for $i = 2, \dots, n$, and $a_{i1}, \dots, a_{i(i-1)}$ are the corresponding regression coefficients. If we can rewrite the model this way, then we can apply standard linear regression results to sample $\mathbf{a}$.

To that end, note that

$$\begin{aligned}
\mathbf{L}\boldsymbol{\varepsilon}_t &= \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_{21} & 1 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 1 & \cdots & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 1 \end{pmatrix} \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \\ \varepsilon_{3t} \\ \vdots \\ \varepsilon_{nt} \end{pmatrix} = \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} + a_{21}\varepsilon_{1t} \\ \varepsilon_{3t} + a_{31}\varepsilon_{1t} + a_{32}\varepsilon_{2t} \\ \vdots \\ \varepsilon_{nt} + \sum_{j=1}^{n-1} a_{nj}\varepsilon_{jt} \end{pmatrix} \\
&= \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \\ \varepsilon_{3t} \\ \vdots \\ \varepsilon_{nt} \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ -\varepsilon_{1t} & 0 & 0 & 0 & 0 & \cdots & & \vdots \\ 0 & -\varepsilon_{1t} & -\varepsilon_{2t} & 0 & 0 & \cdots & & 0 \\ \vdots & & & \ddots & \ddots & & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & -\varepsilon_{1t} & \cdots & -\varepsilon_{t(n-1)} \end{pmatrix} \begin{pmatrix} a_{21} \\ a_{31} \\ a_{32} \\ \vdots \\ a_{n(n-1)} \end{pmatrix}
\end{aligned}$$

Or more succinctly,

$$\mathbf{L}\boldsymbol{\varepsilon}_t = \boldsymbol{\varepsilon}_t - \mathbf{E}_t\mathbf{a}.$$

Noting that $|\boldsymbol{\Sigma}_t| = |\mathbf{D}_t|$, we can rewrite the likelihood implied by (8.7) as

$$\begin{aligned} p(\mathbf{y} \,|\, \boldsymbol{\beta}, \mathbf{a}, \mathbf{h}) &\propto \left(\prod_{t=1}^{T} |\mathbf{D}_t|^{-\frac{1}{2}}\right) \exp\left(-\frac{1}{2}\sum_{t=1}^{T} \boldsymbol{\varepsilon}_t'(\mathbf{L}'\mathbf{D}_t^{-1}\mathbf{L})\boldsymbol{\varepsilon}_t\right) \\ &= \left(\prod_{t=1}^{T} |\mathbf{D}_t|^{-\frac{1}{2}}\right) \exp\left(-\frac{1}{2}\sum_{t=1}^{T} (\mathbf{L}\boldsymbol{\varepsilon}_t)'\mathbf{D}_t^{-1}(\mathbf{L}\boldsymbol{\varepsilon}_t)\right) \\ &= \left(\prod_{t=1}^{T} |\mathbf{D}_t|^{-\frac{1}{2}}\right) \exp\left(-\frac{1}{2}\sum_{t=1}^{T} (\boldsymbol{\varepsilon}_t - \mathbf{E}_t\mathbf{a})'\mathbf{D}_t^{-1}(\boldsymbol{\varepsilon}_t - \mathbf{E}_t\mathbf{a})\right). \end{aligned}$$

In other words, the likelihood is the same as that implied by the regression

$$\boldsymbol{\varepsilon}_t = \mathbf{E}_t\mathbf{a} + \boldsymbol{\eta}_t, \tag{8.8}$$

where $\boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{D}_t)$. Therefore, stacking (8.8) over $t = 1, \ldots, T$, we have

$$\boldsymbol{\varepsilon} = \mathbf{E}\mathbf{a} + \boldsymbol{\eta},$$

where $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{D})$ with $\mathbf{D} = \text{diag}(\mathbf{D}_1, \ldots, \mathbf{D}_T)$. Given the prior $\mathbf{a} \sim \mathcal{N}(\mathbf{a}_0, \mathbf{V}_\mathbf{a})$, it then follows that

$$(\mathbf{a} \,|\, \mathbf{y}, \boldsymbol{\beta}, \mathbf{h}) \sim \mathcal{N}(\widehat{\mathbf{a}}, \mathbf{K}_\mathbf{a}^{-1}),$$

where

$$\mathbf{K}_\mathbf{a} = \mathbf{V}_\mathbf{a}^{-1} + \mathbf{E}'\mathbf{D}^{-1}\mathbf{E}, \quad \widehat{\mathbf{a}} = \mathbf{K}_\mathbf{a}^{-1}\left(\mathbf{V}_\mathbf{a}^{-1}\mathbf{a}_0 + \mathbf{E}'\mathbf{D}^{-1}\boldsymbol{\varepsilon}\right).$$

To sample the log-volatility $\mathbf{h}$, we first compute the "orthogonalized" innovations: $\widetilde{\boldsymbol{\varepsilon}}_t = \mathbf{L}(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta})$ for $t = 1, \ldots, T$. It is easy to check that $\mathbb{E}(\widetilde{\boldsymbol{\varepsilon}}_t \,|\, \mathbf{a}, \mathbf{h}, \boldsymbol{\beta}) = \mathbf{0}$ and

$$\text{Var}(\widetilde{\boldsymbol{\varepsilon}}_t \,|\, \mathbf{a}, \mathbf{h}, \boldsymbol{\beta}) = \mathbf{L}(\mathbf{L}\mathbf{D}_t^{-1}\mathbf{L})^{-1}\mathbf{L}' = \mathbf{D}_t.$$

Hence, $(\widetilde{\varepsilon}_{it} \,|\, \mathbf{a}, \mathbf{h}, \boldsymbol{\beta}) \sim \mathcal{N}(0, \mathrm{e}^{h_{it}})$. Therefore, we can apply the auxiliary mixture sampler in Section 7.1.1 to each of the series $\widetilde{\varepsilon}_{i1}, \ldots, \widetilde{\varepsilon}_{iT}$ for $i = 1, \ldots, n$.

The other steps are now standard. For example, to sample $\boldsymbol{\beta}$, we rewrite (8.7) as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \widetilde{\boldsymbol{\Sigma}})$ and $\widetilde{\boldsymbol{\Sigma}} = \text{diag}(\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_T)$ is a block-diagonal matrix. Together with the prior $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \mathbf{V}_{\boldsymbol{\beta}})$, we have

$$(\boldsymbol{\beta} \,|\, \mathbf{y}, \mathbf{a}, \mathbf{h}) \sim \mathcal{N}(\widehat{\boldsymbol{\beta}}, \mathbf{K}_{\boldsymbol{\beta}}^{-1}),$$

where

$$\mathbf{K}_{\boldsymbol{\beta}} = \mathbf{V}_{\boldsymbol{\beta}}^{-1} + \mathbf{X}'\widetilde{\boldsymbol{\Sigma}}^{-1}\mathbf{X}, \quad \widehat{\boldsymbol{\beta}} = \mathbf{K}_{\boldsymbol{\beta}}^{-1}\left(\mathbf{V}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}_0 + \mathbf{X}'\widetilde{\boldsymbol{\Sigma}}^{-1}\mathbf{y}\right).$$

Note that $\widetilde{\boldsymbol{\Sigma}}^{-1} = \text{diag}(\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_T^{-1})$ with $\boldsymbol{\Sigma}_t^{-1} = \mathbf{L}'\mathbf{D}_t^{-1}\mathbf{L}$.

8.3.2 Empirical Example

We illustrate the estimation of the VAR-SV model using a dataset of US quarterly observations on real GDP growth, CPI inflation and unemployment from 1959Q1 to 2013Q4. The following MATLAB script `VAR_SV.m` implements the Gibbs sampler. Note that the variable `L_id` is constructed to store the index of the lower trianglar elements of $\mathbf{L}$. At every iteration, the matrix $\mathbf{L}$ is updated using the latest draw of $\mathbf{a}$: `L(L_id) = a;`.

```
p = 2;          % if p > 4, need to change Y0 and Y below
nsim = 20000; burnin = 1000;
    % load data
load 'data_Q.csv'; % 1959Q1 to 2013Q4
    % GDP growth, CPI inflation and unemployment rate
data = data_Q(:,[1 2 10]);
Y0 = data(1:4,:);  % save the first 4 obs as the initial conditions
Y = data(5:end,:);
[T n] = size(Y);
y = reshape(Y',T*n,1);
k = n*p+1;         % # of coefficients in each equation
m = n*(n-1)/2;     % # of free elements in L
    % prior
a0 = zeros(m,1); iVa = eye(m);
beta0 = zeros(n*k,1);
    % precision for coefficients = 1; for intercepts = 1/10
tmp = ones(k*n,1);  tmp(1:p*n+1:k*n) = 1/10;
iVbeta = sparse(1:k*n,1:k*n,tmp);
nu_h = 3*ones(n,1); S_h = .1^2*ones(n,1);
b0 = ones(n,1); iB0 = eye(n)/10;
    % compute and define a few things
tmpY = [Y0(end-p+1:end,:); Y];
X_tilde = zeros(T,n*p);
for i=1:p
    X_tilde(:,(i-1)*n+1:i*n) = tmpY(p-i+1:end-i,:);
end
X_tilde = [ones(T,1) X_tilde];
X = SURform2(X_tilde,n);
L_id = nonzeros(tril(reshape(1:n^2,n,n),-1));
L = eye(n);
LiDL = zeros(T*n,n);
    % initialize for storage
store_h = zeros(nsim,T,n);
store_beta = zeros(nsim,k*n);
```

```
store_sigh2 = zeros(nsim,n);
    % initialize the Markov chain
beta = (X'*X)\(X'*y);
h0 = log(mean(reshape((y - X*beta).^2,n,T),2));
h = repmat(h0',T,1);
sigh2 = .1*ones(n,1);
a = zeros(m,1);
for isim = 1:nsim + burnin
        % sample beta
    L(L_id) = a;
    bigL = kron(speye(T),L);
    LiDL = bigL'*sparse(1:T*n,1:T*n,reshape(1./exp(h)',1,T*n))*bigL;
    XLiDL = X'*LiDL;
    Kbeta = iVbeta + XLiDL*X;
    beta_hat = Kbeta\(iVbeta*beta0 + XLiDL*y);
    beta =  beta_hat + chol(Kbeta,'lower')'\randn(n*k,1);

        % sample h
    U = reshape(y - X*beta,n,T)';
    s = (L*U')';
    ystar = log(s.^2 + .0001);
    for i=1:n
        h(:,i) = SVRW(ystar(:,i),h(:,i),h0(i),sigh2(i));
    end

        % sample a
    E = zeros(T*n,m);
    count_E = 0;
    for ii=1:n-1
        E(ii+1:n:end,count_E+1:count_E+ii) = -U(:,1:ii);
        count_E = count_E+ii;
    end
    iD = sparse(1:T*n,1:T*n,reshape(1./exp(h)',1,T*n));
    Ka = iVa + E'*iD*E;
    a_hat = Ka\(iVa * a0 + E'*iD*reshape(U',T*n,1));
    a = a_hat + chol(Ka,'lower')'\randn(m,1);

        % sample sigh2
    e2 = (h - [h0'; h(1:end-1,:)]).^2;
    sigh2 = 1./gamrnd(nu_h + T/2, 1./(S_h + sum(e2)'/2));

        % sample h0
    Kh0 = iB0 + sparse(1:n,1:n,1./sigh2);
```

```
    h0_hat = Kh0\(iB0*b0 + h(1,:)'./sigh2);
    h0 = h0_hat + chol(Kh0,'lower')'\randn(n,1);

        % store the parameters
    if isim > burnin
        isave = isim - burnin;
        store_beta(isave,:) = beta';
        store_h(isave,:,:) = h;
        store_sigh2(isave,:) = sigh2';
    end
end
```

We report the time-varying volatility of the three equations expressed as standard deviations in Figure 8.4. The volatility of the GDP equation decreases substantially in the early 1980s, the timing of which matches the onset of the Great Moderation. The volatility remains low until the Great Recession. The volatility of the unemployment equation follows a similar pattern. The volatility of the inflation equation is high in the 1970s and the early 1980s, and there is a spike in the middle of the Great Recession.

These results show that the error variances of all three equations have substantial time variation. Extending the standard VAR with constant variance to one with stochastic volatility is therefore empirically relevant.

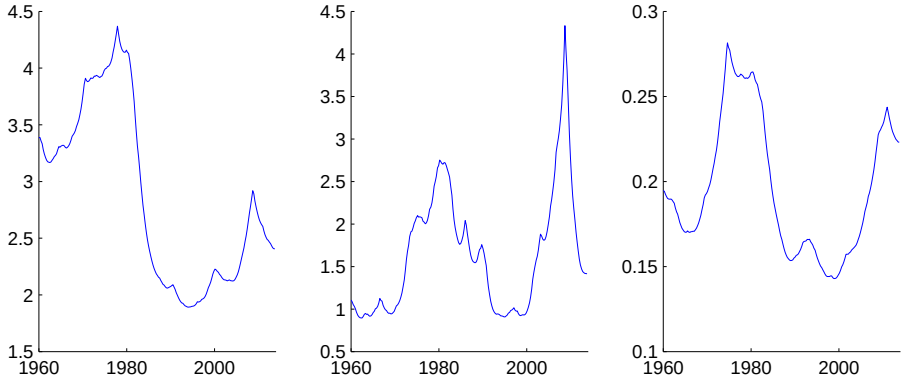

Figure 8.4: Time-varying volatility of the GDP growth equation (left panel), inflation equation (middle panel) and unemployment equation (right panel) expressed as standard deviations.

Bibliography

An, S. and Schorfheide, F. (2007), 'Bayesian analysis of DSGE models', *Econometric Reviews* **26**(2-4), 113–172.

Ardia, D., Basturk, N., Hoogerheide, L. and van Dijk, H. K. (2012), 'A comparative study of Monte Carlo methods for efficient evaluation of marginal likelihood', *Computational Statistics and Data Analysis* **56**(11), 3398–3414.

Banbura, M., Giannone, D. and Reichlin, L. (2010), 'Large Bayesian vector auto regressions', *Journal of Applied Econometrics* **25**(1), 71–92.

Berument, H., Yalcin, Y. and Yildirim, J. (2009), 'The effect of inflation uncertainty on inflation: Stochastic volatility in mean model within a dynamic framework', *Economic Modelling* **26**(6), 1201–1207.

Bianchi, F. (2013), 'Regime switches, agents' beliefs, and post-World War II U.S. macroeconomic dynamics', *The Review of Economic Studies* **80**(2), 463–490.

Canova, F. (1993), 'Modelling and forecasting exchange rates with a Bayesian time-varying coefficient model', *Journal of Economic Dynamics and Control* **17**, 233–261.

Carter, C. K. and Kohn, R. (1994), 'On Gibbs sampling for state space models', *Biometrika* **81**, 541–553.

Celeux, G., Forbes, F., Robert, C. P. and Titterington, D. M. (2006), 'Deviance information criteria for missing data models', *Bayesian Analysis* **1**(4), 651–674.

Chan, J. C. C. (2013), 'Moving average stochastic volatility models with application to inflation forecast', *Journal of Econometrics* **176**(2), 162–172.

Chan, J. C. C. (2016), 'Specification tests for time-varying parameter models with stochastic volatility', *Econometric Reviews* . Forthcoming.

Chan, J. C. C. (2017), 'The stochastic volatility in mean model with time-varying parameters: An application to inflation modeling', *Journal of Business and Economic Statistics* **35**(1), 17–28.

Chan, J. C. C. and Eisenstat, E. (2015), 'Marginal likelihood estimation with the Cross-Entropy method', *Econometric Reviews* **34**(3), 256–285.

Chan, J. C. C. and Grant, A. L. (2015), 'Pitfalls of estimating the marginal likelihood using the modified harmonic mean', *Economics Letters* **131**, 29–33.

Chan, J. C. C. and Grant, A. L. (2016), 'On the observed-data deviance information criterion for volatility modeling', *Journal of Financial Econometrics* **14**(4), 772–802.

Chan, J. C. C. and Jeliazkov, I. (2009), 'Efficient simulation and integrated likelihood estimation in state space models', *International Journal of Mathematical Modelling and Numerical Optimisation* **1**(1/2), 101–120.

Chan, J. C. C., Koop, G. and Potter, S. M. (2013), 'A new model of trend inflation', *Journal of Business and Economic Statistics* **31**(1), 94–106.

Chan, J. C. C., Koop, G. and Potter, S. M. (2016), 'A bounded model of time variation in trend inflation, NAIRU and the Phillips curve', *Journal of Applied Econometrics* **31**(3), 551–565.

Chan, J. C. C. and Kroese, D. P. (2012), 'Improved cross-entropy method for estimation', *Statistics and Computing* **22**(5), 1031–1040.

Chib, S. (1995), 'Marginal likelihood from the Gibbs output', *Journal of the American Statistical Association* **90**, 1313–1321.

Chib, S. and Greenberg, E. (1994), 'Bayes inference in regression models with ARMA(p, q) errors', *Journal of Econometrics* **64**(1–2), 183–206.

Chib, S. and Greenberg, E. (1995), 'Understanding the Metropolis-Hastings algorithm', *The American Statistician* **49**(4), 327–335.

Chib, S. and Jeliazkov, I. (2001), 'Marginal likelihood from the Metropolis-Hastings output', *Journal of the American Statistical Association* **96**, 270–281.

Clark, P. K. (1987), 'The cyclical component of US economic activity', *The Quarterly Journal of Economics* **102**(4), 797–814.

Clark, T. E. (2011), 'Real-time density forecasts from Bayesian vector autoregressions with stochastic volatility', *Journal of Business and Economic Statistics* **29**(3), 327–341.

Cogley, T. and Sargent, T. J. (2001), 'Evolving post-world war II US inflation dynamics', *NBER Macroeconomics Annual* **16**, 331–388.

Cogley, T. and Sargent, T. J. (2005), 'Drifts and volatilities: Monetary policies and outcomes in the post WWII US', *Review of Economic Dynamics* **8**(2), 262–302.

de Jong, P. and Shephard, N. (1995), 'The simulation smoother for time series models', *Biometrika* **82**, 339–350.

Deborah, G. and Strachan, R. W. (2009), 'Nonlinear impacts of international business cycles on the U.K.—a Bayesian smooth transition VAR approach', *Studies in Nonlinear Dynamics and Econometrics* **14**(1), 1–33.

Djegnéné, B. and McCausland, W. J. (2014), 'The HESSIAN method for models with leverage-like effects', *Journal of Financial Econometrics* . Forthcoming.

Durbin, J. and Koopman, S. J. (2002), 'A simple and efficient simulation smoother for state space time series analysis', *Biometrika* **89**, 603–615.

Engle, R. F., Lilien, D. M. and Robins, R. P. (1987), 'Estimating time varying risk premia in the term structure: the ARCH-M model', *Econometrica* **55**(2), 391–407.

Friel, N. and Pettitt, A. N. (2008), 'Marginal likelihood estimation via power posteriors', *Journal Royal Statistical Society Series B* **70**, 589–607.

Frühwirth-Schnatter, S. and Wagner, H. (2008), 'Marginal likelihoods for non-Gaussian models using auxiliary mixture sampling', *Computational Statistics and Data Analysis* **52**(10), 4608–4624.

Frühwirth-Schnatter, S. and Wagner, H. (2010), 'Stochastic model specification search for Gaussian and partial non-Gaussian state space models', *Journal of Econometrics* **154**, 85–100.

Früwirth-Schnatter, S. (1994), 'Data augmentation and dynamic linear models', *Journal of Time Series Analysis* **15**, 183–202.

Gelfand, A. E. and Dey, D. K. (1994), 'Bayesian model choice: Asymptotics and exact calculations', *Journal of the Royal Statistical Society Series B* **56**(3), 501–514.

Geweke, J. (1992), Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments, *in* 'Bayesian Statistics', Oxford University Press, pp. 169–193.

Geweke, J. (1993), 'Bayesian treatment of the independent Student-t linear model', *Journal of Applied Econometrics* **8**, S19–S40.

Geweke, J. (1999), 'Using simulation methods for Bayesian econometric models: inference, development, and communication', *Econometric Reviews* **18**(1), 1–73.

Grant, A. L. and Chan, J. C. C. (2017*a*), 'A Bayesian model comparison for trend-cycle decompositions of output', *Journal of Money, Credit and Banking* . Forthcoming.

Grant, A. L. and Chan, J. C. C. (2017*b*), 'Reconciling output gaps: Unobserved components model and Hodrick-Prescott filter', *Journal of Economic Dynamics and Control* **75**, 114–121.

Han, C. and Carlin, B. P. (2001), 'Markov chain Monte Carlo methods for computing Bayes factors: a comparative review', *Journal of the American Statistical Association* **96**, 1122–1132.

Harvey, A. C. (1985), 'Trends and cycles in macroeconomic time series', *Journal of Business and Economic Statistics* **3**(3), 216–227.

Hodrick, R. J. and Prescott, E. C. (1980), 'Postwar US business cycles: An empirical investigation', *Carnegie Mellon University discussion paper* **451**.

Hodrick, R. J. and Prescott, E. C. (1997), 'Postwar US business cycles: An empirical investigation', *Journal of Money, Credit, and Banking* **29**(1), 1–16.

Kass, R. E. and Raftery, A. E. (1995), 'Bayes factors', *Journal of the American Statistical Association* **90**(430), 773–795.

Kim, S., Shepherd, N. and Chib, S. (1998), 'Stochastic volatility: Likelihood inference and comparison with ARCH models', *Review of Economic Studies* **65**(3), 361–393.

Koop, G. (2013), 'Forecasting with medium and large Bayesian VARs', *Journal of Applied Econometrics* **28**(2), 177–203.

Koop, G. and Korobilis, D. (2010), 'Bayesian multivariate time series methods for empirical macroeconomics', *Foundations and Trends in Econometrics* **3**(4), 267–358.

Koop, G. and Korobilis, D. (2012), 'Forecasting inflation using dynamic model averaging', *International Economic Review* **53**(3), 867–886.

Koop, G., Leon-Gonzalez, R. and Strachan, R. W. (2010), 'Dynamic probabilities of restrictions in state space models: an application to the Phillips curve', *Journal of Business and Economic Statistics* **28**(3), 370–379.

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

Koop, G. and Potter, S. M. (1999), 'Bayes factors and nonlinearity: evidence from economic time series', *Journal of Econometrics* **88**(2), 251–281.

Koopman, S. J. and Hol Uspensky, E. (2002), 'The stochastic volatility in mean model: Empirical evidence from international stock markets', *Journal of Applied Econometrics* **17**(6), 667–689.

Kroese, D. P. and Chan, J. C. C. (2014), *Statistical Modeling and Computation*, Springer, New York.

Li, Y. and Yu, J. (2012), 'Bayesian hypothesis testing in latent variable models', *Journal of Econometrics* **166**(2), 237–246.

Li, Y., Zeng, T. and Yu, J. (2012), 'Robust deviance information criterion for latent variable models', *SMU Economics and Statistics Working Paper Series* .

Li, Y., Zeng, T. and Yu, J. (2014), 'A new approach to Bayesian hypothesis testing', *Journal of Econometrics* **178**, 602–612.

Liu, Z., Waggoner, D. F. and Zha, T. (2011), 'Sources of macroeconomic fluctuations: A regime-switching DSGE approach', *Quantitative Economics* **2**(2).

Luo, S. and Startz, R. (2014), 'Is it one break or ongoing permanent shocks that explains US real GDP?', *Journal of Monetary Economics* **66**, 155–163.

McCausland, W. J. (2012), 'The HESSIAN method: Highly efficient simulation smoothing, in a nutshell', *Journal of Econometrics* **168**(2), 189–206.

McCausland, W. J., Miller, S. and Pelletier, D. (2011), 'Simulation smoothing for state-space models: A computational efficiency analysis', *Computational Statistics and Data Analysis* **55**(1), 199–212.

Millar, R. B. (2009), 'Comparison of hierarchical Bayesian models for overdispersed count data using DIC and Bayes factors', *Biometrics* **65**(3), 962–969.

Morley, J. C., Nelson, C. R. and Zivot, E. (2003), 'Why are the Beveridge-Nelson and unobserved-components decompositions of GDP so different?', *Review of Economics and Statistics* **85**(2), 235–243.

Moura, G. V. and Turatti, D. E. (2014), 'Efficient estimation of conditionally linear and Gaussian state space models', *Economics Letters* **124**(3), 494–499.

Mumtaz, H. and Zanetti, F. (2013), 'The impact of the volatility of monetary policy shocks', *Journal of Money, Credit and Banking* **45**(4), 535–558.

Perron, P. and Wada, T. (2009), 'Let's take a break: Trends and cycles in US real GDP', *Journal of Monetary Economics* **56**(6), 749–765.

Primiceri, G. E. (2005), 'Time varying structural vector autoregressions and monetary policy', *Review of Economic Studies* **72**(3), 821–852.

Rubinstein, R. Y. (1997), 'Optimization of computer simulation models with rare events', *European Journal of Operational Research* **99**, 89–112.

Rubinstein, R. Y. (1999), 'The cross-entropy method for combinatorial and continuous optimization', *Methodology and Computing in Applied Probability* **2**, 127–190.

Rubinstein, R. Y. and Kroese, D. P. (2004), *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization Monte-Carlo Simulation, and Machine Learning*, Springer-Verlag, New York.

Rue, H. (2001), 'Fast sampling of Gaussian Markov random fields with applications', *Journal of the Royal Statistical Society Series B* **63**(2), 325–338.

Rue, H., Martino, S. and Chopin, N. (2009), 'Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace', *Journal of the Royal Statistical Society Series B* **71**(2), 319–392.

Schwarz, G. (1978), 'Estimating the dimension of a model', *The Annals of Statistics* **6**(2), 461–464.

Sims, C. A. (1980), 'Macroeconomics and reality', *Econometrica* **48**, 1–48.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002), 'Bayesian measures of model complexity and fit', *Journal of the Royal Statistical Society Series B* **64**(4), 583–639.

Stock, J. H. and Watson, M. W. (2007), 'Why has U.S. inflation become harder to forecast?', *Journal of Money Credit and Banking* **39**(s1), 3–33.

Verdinelli, I. and Wasserman, L. (1995), 'Computing Bayes factors using a generalization of the Savage-Dickey density ratio', *Journal of the American Statistical Association* **90**(430), 614–618.

Watson, M. W. (1986), 'Univariate detrending methods with stochastic trends', *Journal of Monetary Economics* **18**(1), 49–75.