7 MCMC Methods for Statistical Inference

While recovering from an illness in 1946, Stan Ulam was playing solitaire. It, then, occurred to him to try to compute the chances that a particular solitaire laid out with 52 cards would come out successfully (Eckhard, 1987). After attempting exhaustive combinatorial calculations, he decided to go for the more practical approach of laying out several solitaires at random and then observing and counting the number of successful plays. This idea of selecting a statistical sample to approximate a hard combinatorial problem by a much simpler problem is at the heart of modern Monte Carlo simulation.

Stan Ulam soon realized that computers could be used in this fashion to answer questions of neutron diffusion and mathematical physics. He contacted John Von Neumann, who understood the great potential of this idea. Over the next few years, Ulam and VonNeumann developed many Monte Carlo algorithms, including importance sampling and rejection sampling. Enrico Fermi in the 1930's also used Monte Carlo in the calculation of neutron diffusion, and later designed the FERMIAC, a Monte Carlo mechanical device that performed calculations (Anderson, 1986). In the 1940's Nick Metropolis, a young physicist, designed new controls for the state-of-the-art computer (ENIAC) with Klari Von Neumann, John's wife. He was fascinated with Monte Carlo methods and this new computing device. Soon he designed an improved computer, which he named the MANIAC in the hope that computer scientists would stop using acronyms. During the time he spent working on the computing machines, many mathematicians and physicists (Fermi, Von Neumann, Ulam,Teller, Richtmyer, Bethe, Feynman, & Gamow) would go to him with their work problems. Eventually in 1949, he published the first public document on Monte Carlo simulation with Stan Ulam (Metropolis & Ulam, 1949). This paper introduces, among other ideas, Monte Carlo particle methods, which form the basis of modern sequential Monte Carlo methods such as bootstrap filters, condensation, and survival of the fittest algorithms (Doucet, deFreitas, & Gordon, 2001). Soon after, he proposed the Metropolis algorithm with the Tellers and the Rosenbluths (Metropolis et al., 1953).

Many papers on Monte Carlo simulation appeared in the physics literature after 1953. From an inference perspective, the most significant contribution was the generalization of the Metropolis algorithm by Hastings in 1970. Hastings and his student Peskun showed that Metropolis and the more general Metropolis-Hastings algorithms are particular instances of a large family of algorithms, which also includes the Boltzmann algorithm.

In the 1980's, two important MCMC papers appeared in the fields of computer vision and artificial intelligence (Geman & Geman, 1984; Pearl, 1987). Despite the existence of a few MCMC publications in the statistics literature at this time, it is generally accepted that it was only in 1990 that MCMC made the first significant impact in statistics (Gelfand & Smith, 1990). In the neural networks literature, the publication of Neal (1996) was particularly influential

This section provides an introduction to Markov chain Monte Carlo simulations and Bayesian inference, and where these two meet.

7.1 Markov Processes

Let x_t denote the value of some random variable at time or iteration t that takes values in the set $S = \{s_1, s_2, ...\}$. S is the state set. A Markov chain generates a series of samples

$[x_0, x_1, x_2, \ldots, x_T]$ by starting at some point x_0, and then following a series of stochastic steps. Markov chains satisfy the Markov property. The Markov property for a Markov chain of order 1 is written as

$$p(x_t|x_{t-1}, \ldots, x_2, x_1, x_0) = p(x_t|x_{t-1})$$

which means that the state of the chain at time/iteration t only depends on the state of the chain at time/iteration t-1. The history (how we get from x_0 to x_{t-1}) is not important.

The probability of moving from state i at time t-1 to state j at time t is called the transition probability: $p_{ij}(t) = \Pr(x_t = s_j|x_{t-1} = s_i)$. The ith row of the transition matrix is the conditional density of the outcomes (states) for the next iteration/time in the chain given the current state i. Transition probabilities only depends on the current state, not past states.

An invariant density (or stationary density), $\pi = [\pi_1....\pi_k]$, for discrete states with k states and transition matrix P, satisfies $[\pi_1....\pi_k]' = [\pi_1....\pi_k]P$. π_j can be interpreted as the probability that the chain is in state j at any time/iteration, independent of the previous state of the chain. Note that not all Markov chains have an invariant density.

For example if the transition matrix is given by

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} \tag{7.1}$$

the invariant density (or stationary density) is equal to $[\pi_1 \;\; \pi_2] = [0.75 \;\; 0.25]$.

For better clarification, let us go through the steps with some financial context. Let us assume the P matrix defined above contains the transition probability of moving from one market state to another. In this scenario, we can consider two states, $S = \{Bull, Bear\}$. The first row of P implies that the probability of moving from a bull market to a bear market is $P_{1,2} = 0.2$, while the probability of remaining in a bull market is $P_{1,1} = 0.8$. Similarly, the second row indicates that the probability of a bear market transitioning to a bull market is $P_{2,1} = 0.6$, while the probability of remaining in the bear market is $P_{2,2} = 0.4$. We repeat the notation in mathematical form as follows:

$$P_{ij}(t) = \Pr(x_t = S_j|x_{t-1} = S_i)$$

where x_t is the market state at time t. Solving for the invariant density, $[\pi_1 \;\; \pi_2]$, would amount to solving for:

$$\begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} = [\pi_1 \;\; \pi_2] \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix}$$

We will obtain the following equations:

$$0.8\pi_1 + 0.6\pi_2 = \pi_1 \Rightarrow \pi_2 = \frac{1}{3}\pi_1$$

$$0.2\pi_1 + 0.4\pi_2 = \pi_2$$

Note that the invariant density, π, contains the unconditional probabilities that the Markov chain, $\{x_t\}$, is at a given state at any time t. Since no two states can occur at the same time, we can include an added equation i.e:

$$\pi_1 + \pi_2 = 1$$

Combining the above relations, we can have:

$$\pi_1 + \frac{1}{3}\pi_1 = 1 \Rightarrow \frac{4}{3}\pi_1 = 1 \Rightarrow \pi_1 = \frac{3}{4}$$

$$\Rightarrow \pi = [\frac{3}{4}, \frac{1}{4}]$$

i.e the probability that the market would be in bull market at any given time is 0.75, and 0.25 for a bear market. We can verify this result with the second equation from the matrix (We only used one of the equations obtained from expanding the matrix form, along with the added equation):

$$0.2\pi_1 + 0.4\pi_2 = 0.2(0.75) + 0.4(0.25) = 0.15 + 0.1 = 0.25 = \pi_2$$

Thus we have outlined how the invariant density, π, functions with respect to the corresponding transition probability matrix P. Remember that π and P work as a pair; i.e π may be an invariant density for a matrix P_1 but not necessarily for a different matrix P_2. This makes sense in the real world context since, if the conditional probabilities are different, the unconditional ones (i.e the invariant density) may change as well.

If the states are continuous taking values in $\mathbb{R}$, then the *transition density* can be defined as $p(x, y) = p(y|x)$; the probability of moving from x at time t-1 to y at time t. Transition density $p(x, y)$ is sometimes called *transition kernel*. An invariant density (or stationary density), $\pi(y)$, for the transition kernel $p(x, y)$ is a density that satisfies

$$\pi(y) = \int_R \pi(x)p(x, y)dx$$

The basis of Markov Chain Monte Carlo (MCMC) algorithms is the construction of a transition kernel, $p(x, y)$, that has an invariant density equal to the target density (target density: an unknown density from which we would like to take draws). Given such a kernel, the process can be started at x_0 (arbitrary) and yield a draw x_1 from $p(x_0, x_1)$, x_2 from $p(x_1, x_2)$, ..., and x_G from $p(x_{G-1}, x_G)$, where G is the desired number of simulations. After a transient period, the distribution of the generated chain, $\{x^{(g)}\}_{g=1}^G$, is approximately equal to the target distribution. Surveys of MCMC methods include Chib (2001), Geweke (1997) and Robert and Casella (1999).

To approximate the distribution of an unknown parameter, $p(\theta)$ where $\theta \in \Theta$, using Markov chain simulation, we simulate a Markov process on Θ whose invariant density is $p(\theta)$. The steps are:

1. Create a Markov process (transition kernel) whose invariant distribution is $p(\theta)$ (the target density that you are looking for).

2. Start from a starting point, $\theta^{(0)}$, and run the simulation sufficiently long so the distribution of the simulated values, $\{\theta^{(g)}\}_{g=1}^G$, is close to the invariant distribution.

3. Use the generated chain, $\{\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(G)}\}$, as a sample from the target density, $p(\theta)$, from which we can make inference about θ.

The question is how to find a kernel that has the target density, $p(\theta)$, as its invariant distribution. It is remarkable that the Metropolis–Hasting (MH) algorithm gives us a general principle to find such kernels. We will start with a special case of MH algorithm, called Gibbs sampler.

Note that G is the number of simulated values, and N or T denote the number of observations. We can increase the precision of our MCMC technique by increasing G, even if the number of observations is fixed.

7.1.1 Gibbs Sampling

The Gibbs algorithm is a special case of the MH algorithm (discussed later) and can be useful when the target density is the joint density of the random variables. Suppose we are interested in finding the joint density of two random variables, (θ_1, θ_2). Gibbs sampler can be used when it is possible to sample from each conditional distribution. For example, suppose we wish to sample from a non-standard joint distribution, $p(\theta_1, \theta_2)$. Further suppose that the two conditional distributions, $p(\theta_1|\theta_2)$ and $p(\theta_2|\theta_1)$ are distributions for which simulation algorithms are known. The Gibbs algorithm can then be written as

1. Choose a starting value, $\theta_2^{(0)}$.
2. At the gth iteration, $g = 1, \ldots, G$, draw

$$\theta_1^{(g)} \text{ from } p(\theta_1|\theta_2^{(g-1)}),$$

$$\theta_2^{(g)} \text{ from } p(\theta_2|\theta_1^{(g)}),$$

until the desired number of iterations is obtained. Step 2 produces one draw from the Markov chain, $(\theta_1^{(g)}, \theta_2^{(g)})$. G is the number of draws to be used for inference.

It can be proved that the invariant distribution of the Gibbs kernel is the target distribution, $p(\theta_1, \theta_2)$. (Not proved here.)

Iterating on step 2, we collect the draws $\{(\theta_1^{(g)}, \theta_2^{(g)})\}_{g=1}^{G}$ that can be used to make inference about (θ_1, θ_2). For example, a point estimate of $E[\theta_1\theta_2]$ will be approximated by

$$\hat{E}[\theta_1\theta_2] \approx \frac{1}{G}\sum_{g=1}^{G} \theta_1^{(g)}\theta_2^{(g)}.$$

Remark. To calculate the exact value of $E[\theta_1\theta_2] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \theta_1\theta_2 f(\theta_1, \theta_2)d\theta_1 d\theta_2$, we need to know the joint density of (θ_1, θ_2) which is not of any known form.

7.1.2 Metropolis–Hasting Sampling

The MH algorithm is more general than the Gibbs sampler because it does not require that the full set of conditional distributions be available. For situations in which there is only one random variable or when both the joint density and the conditional densities are of unknown forms, the MH algorithm provides a generic method for sampling. We only need to be able to evaluate the target pdf at a point θ, but we do not need to know much else about the target distribution. The MH approach also includes Gibbs sampling as a special case.

To generate a sample from $p(\theta)$ using the MH algorithm, where θ may be a scalar or a vector random variable, we first need to find a kernel, $q(\theta, \eta)$, that has $p(.)$ as its invariant distribution. The idea here is using a *reversible* kernel $q(.,.)$ for which

$$p(\theta)q(\theta, \eta) = p(\eta)q(\eta, \theta) \tag{7.2}$$

This means the probability of going from state θ to η is the same as the probability of going from state η to θ. If a chain is reversible then we can identify the invariant distribution as $p(\theta)$. In fact, we can show that $p(\theta)$ is the invariant distribution of $q(.,.)$ if Equation 7.2 holds. See Karlsson (2004) for some more discussion.

If the kernel $q(\theta, \eta)$ is not reversible, we can convert it to a reversible kernel $p(\theta, \eta)$ by

$$p(\theta, \eta) = \alpha(\theta, \eta) \times q(\theta, \eta)$$

using

$$\alpha(\theta, \eta) = \min\{\frac{p(\eta)q(\eta, \theta)}{p(\theta)q(\theta, \eta)}, 1\}$$

This can be interpreted as: If the present state of the chain is θ, generate a value η from kernel $q(\theta, \eta)$ and make the move to η with probability $\alpha(\theta, \eta)$. If the move to η is rejected, the chain remains at θ. $\alpha(\theta, \eta)$ is the probability of accepting the new draw.

The steps of the MH algorithm can be written as

1. Choose a starting value, $\theta^{(0)}$.
2. At the gth iteration, $g = 1, \ldots, G$, generate η from $q(\theta^{(g-1)}, \eta)$
3. Draw u from the uniform distribution $U(0, 1)$. If

$$u \leq \alpha(\theta^{(g-1)}, \eta) = \min\{\frac{p(\eta)q(\eta, \theta^{(g-1)})}{p(\theta^{(g-1)})q(\theta^{(g-1)}, \eta)}, 1\}, \tag{7.3}$$

set $\theta^{(g)} = \eta$, otherwise, $\theta^{(g)} = \theta^{(g-1)}$ and return to step 2.

As with the Gibbs sampler, we iterate on this to obtain a set of draws $\{\theta^{(g)}\}_{g=1}^{G}$. Now we have a sample from the target density from which we can estimate various moments or other features of $p(\theta)$ and make inference about θ. Since any normalizing constant cancels out in these calculations (Equation 7.3), we only need to specify the kernel of the target density, $p(\theta)$.

Note: $q(x, y)$ is also called the proposal density and is set by the econometricians. Note that the proposal density can condition on the data as well as the last iteration of the parameter vector, $\theta^{(g-1)}$. A common proposal is the random walk proposal:

$$q(x, y) = N(y|x, \tau),$$

where τ is called the tunning parameter. In practice, we set τ such that the acceptance rate is between 20% and 60%.

Note: If the proposal is a symmetric function (e.g., the random walk proposal), then Equation (7.3) in step 3 reduces to

$$u \leq \min\{\frac{p(\eta)}{p(\theta^{(g-1)})}, 1\} \tag{7.4}$$

Note: Because the initial value, $\theta^{(0)}$, is arbitrary and is not drawn from the invariant distribution, some portion of the initial sample must be discarded; this portion is called *transient* or *burn-in* sample. See Figure 57.

Note: In both Gibbs sampling and the MH algorithm, it is important to assess convergence of the chain (examples in Figure 58). A good sampler will efficiently move through the parameter space and display low serial correlation between draws. A poor sampler will display very high autocorrelation over the draws and suggests the target density is not being explored efficiently. Some suggestions for detecting convergence include:

1. Rerun the chain for different startup values of $\theta^{(0)}$. You should obtain very similar results over different starting values.

2. Plot the autocorrelation function (ACF) of the random variable θ, $\{\theta^{(g)}\}_{g=1}^{G}$. If the chain is mixing well, the ACF will die out quickly.

3. Look at the time series plot (trace plot) of $\{\theta^{(g)}\}_{g=1}^{G}$. It should look random. If it looks more like a random walk then the chain is not mixing well and likely has not converged.

4. Perform statistical tests on the output. See the separated partial means test in Geweke (2003), Theorem 3.7.3.

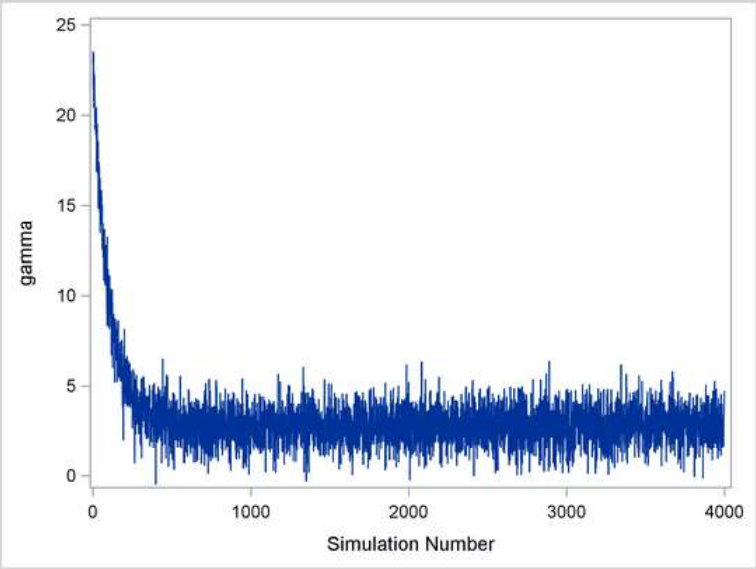

Figure 57: Effects of the initial point on the generated chain and the need for discarding the first few iterations as the burn-in sample.

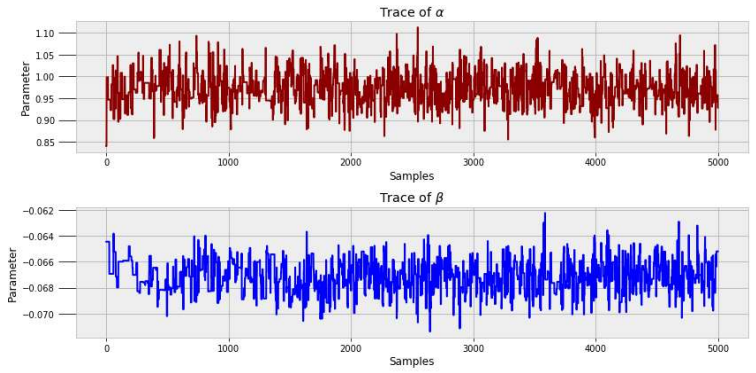

Figure 58: The chains are mixing well. The top panel looks better.

An Example of the MH algorithm:

X_1 and X_2 are multivariate standardized normal random variables with a correlation of 0.5:

$$(x_1, x_2) \sim N_2((\mu_1, \mu_2), \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}) \equiv N_2((0,0), \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}) \tag{7.5}$$

We are interested in draws from the conditional distribution of X_2 given $X_1 = -2$, $f(x_2|x_1 = -2)$. This is used heavily in risk management and known as stress testing.

We know the exact theoretical conditional distribution (a univariate normal density: $N(\mu_2 + \frac{\sigma_2}{\sigma_1}(x_1 - \mu_1), \sigma_2^2(1-\rho^2)) \equiv N(-1, 0.86^2)$). However, here we assume that this conditional density is not of any standard form and try to take draws from this target density using the MH algorithm. The only thing that we know is that the conditional density has the functional form:

$$p(x_2) = f(x_2|x_1 = -2) = \frac{N_2(-2, x_2)}{f(x_1 = -2)}$$

where

1. $N_2(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{(1-\rho^2)}} exp\{\frac{-1}{2(1-\rho^2)}[(\frac{x_1-\mu_1}{\sigma_1})^2 + (\frac{x_2-\mu_2}{\sigma_2})^2 - 2\rho\frac{(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_2}]\}$ is the pdf of the bivariate normal density at (x_1, x_2)

2. $f(x_1) = \frac{1}{\sqrt{2\pi}\sigma_1} exp\{\frac{-(x_1-\mu_1)^2}{2\sigma_1^2}\}$ is the pdf of the normal density at x_1.

To take draws from our target density using the MH algorithm, we need

- A starting point: arbitrary (e.g., $x_2^{(0)} = -2$)
- A proposal: A random walk, where the mean of proposed next step is the current location (e.g., $N(\eta|x_2^{(g-1)}, \tau)$). Note that the random walk normal proposal is symmetric, and this simplifies our calculations.
- The number of simulations as well as the size of the burn-in sample (e.g., $G = 1,000,000$ and burn-in$=5,000$).

The steps of the MH Algorithm:

1. Start from $x_2^{(0)} = -2$.
2. At the gth iteration, $g = 1, \ldots, G$, generate η from $N(x_2^{(g-1)}, \tau)$.
3. Draw u from the uniform distribution $U(0,1)$. If

$$u \leq \min\{\frac{N_2(-2, \eta)/f(X_1 = -2)}{N_2(-2, x_2^{(g-1)})/f(X_1 = -2)}, 1\} = \min\{\frac{p(\eta)}{p(x_2^{(g-1)})}, 1\}, \tag{7.6}$$

   set $x_2^{(g)} = \eta$, otherwise, $x_2^{(g)} = x_2^{(g-1)}$ and return to step 2.
4. Use the generated chain, $\{x_2^{(1)}, x_2^{(2)}, \ldots, x_2^{(G)}\}$, as a sample from the target density, the conditional density of X_2.

```r
X1 <- -2
cor <- 0.5

proposalfunction <- function(param){
  return(rnorm(1,mean = param, sd= 1))
}

joint_dist <- function(x) dmvnorm(x,c(0,0),matrix(c(1,cor,cor,1),2,2))

run_MCMC <- function(startvalue, iterations){
  chain = array(dim = c(iterations+1,2))
  chain[1,] = startvalue
  for (i in 1:iterations){
    proposal_x2 = proposalfunction(chain[i,2])
    full_proposal <- c(X1,proposal_x2)
    probab = joint_dist(full_proposal)/ joint_dist(chain[i,])

    if (runif(1) < probab){
      chain[i+1,] = full_proposal
    }else{
      chain[i+1,] = chain[i,]
    }
    print(i)
  }
  return(chain)
}

startvalue = c(X1,X1)
iterations <- 1000000
chain = run_MCMC(startvalue, iterations)

burnIn = 5000
result <- chain[burnIn:iterations+1,]
```

Figure 59: R commands for the MH algorithm.

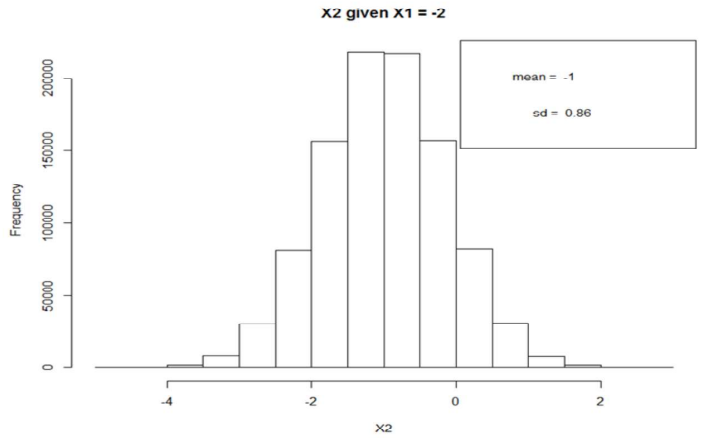

Figure 60: The empirical histogram of the draws from the conditional density $f(X_2|X_1 = -2)$ using the MH algorithm. Since the joint density of (X_1, X_2) is multivariate normal, the true conditional density can be written as $N(\mu_2 + \frac{\sigma_2}{\sigma_1}(x_1 - \mu_1), \sigma_2^2(1-\rho^2)) \equiv N(-1, 0.86^2)$. As we see the estimated density represents the true density very well.

7.2 Bayesian Inference

Consider a simple model

$$r_t = \mu + a_t, \quad a_t \overset{\text{i.i.d}}{\sim} N(0, \sigma_a^2), \quad t = 1, \ldots, T \tag{7.7}$$

From the classical point of view (frequentist school), the set of parameters, $\theta = (\mu, \sigma_a)$, are assumed to be unknown values, and we try to estimate them. Frequentist school does not consider parameter uncertainty. This is what we have done so far in this course. For example, in the moving average model above we use the Maximum Likelihood technique in order to estimate $\hat{\mu}$ and $\hat{\sigma_a}$.

From the subjective point of view (Bayesian school), the unknown parameters are regarded as random variables and assigned probability distributions to capture the parameter uncertainty.

Both Bayesian and classical approaches view the unknown model parameters fixed, but the Bayesians express their uncertainty about these parameters using probability. Before seeing the observations, $\{r_1, \ldots, r_T\}$, we assign prior distributions, $p(\theta)$, to the unknown parameters. A prior is used to summarize the uncertainty regarding θ before we observe the data and learn from it. One way of selecting priors is to plot the pdf of the observed data and conjecture the values of the parameter that are more or less likely. There are other techniques to determine the prior distributions. For our example above, one can define the following independent priors

$$p(\mu, \sigma_a^2) = p(\mu) \times p(\sigma_a^2) \equiv N(\mu|\mu_0, \sigma_0^2) \times IG(\sigma_a^2|\nu_0, s_0)$$

where $IG(.|.)$ represents the inverse gamma distribution. μ_0, σ_0, ν_0 and s_0 are known quantities, called the hyper-parameters.

Bayesian inference centers on the posterior distribution of the random variable θ conditional on having observed the data. Bayesian analysis is simply applying Bayes Theorem and updating a prior distribution to a posterior distribution. Bayes rule shows us how our prior beliefs are changed once we observed an event. The result is the posterior probability.

Using Bayes Theorem, we can obtain the posterior distribution of a set of parameters, θ, after observing the data from time 1 to T, $p(\theta|r_{1:T})$:

$$p(\theta|r_{1:T}) = \frac{p(\theta)p(r_{1:T}|\theta)}{\int p(\theta)p(r_{1:T}|\theta)d\theta} \propto p(\theta)p(r_{1:T}|\theta) \tag{7.8}$$

where $r_{1:T} = \{r_1, \ldots, r_T\}$, $p(\theta)$ denotes the prior density, and $p(r_{1:T}|\theta)$ is often called the likelihood. The denominator is the integrating constant. In most cases, we do not need to know this normalizing constant (e.g., the MH algorithm). Therefore, we can write the posterior proportional to the likelihood times the prior. That is, we update the prior beliefs to posterior beliefs after observing the data. All inference about θ is conducted through the posterior and conditions on the data, $r_{1:T}$.

For our example mentioned above, the posterior distribution of the unknown parameters can be written as

$$p(\mu, \sigma_a^2|r_{1:T}) \propto p(\mu) \times p(\sigma_a^2) \times \prod_{t=1}^{T} N(r_t|\mu, \sigma_a^2).$$

The posterior combines in one expression all the information that we have about θ; information about θ before the current data through the prior distribution, and the information contained in the current data through the likelihood function. It is updating the information we have about θ. Bayesian inference is based on what did occur as opposed to classical inference that is concerned about what might have happened. Bayesian approach is ex post and Bayesian inference is conditional on the observed data. See Figure 61.

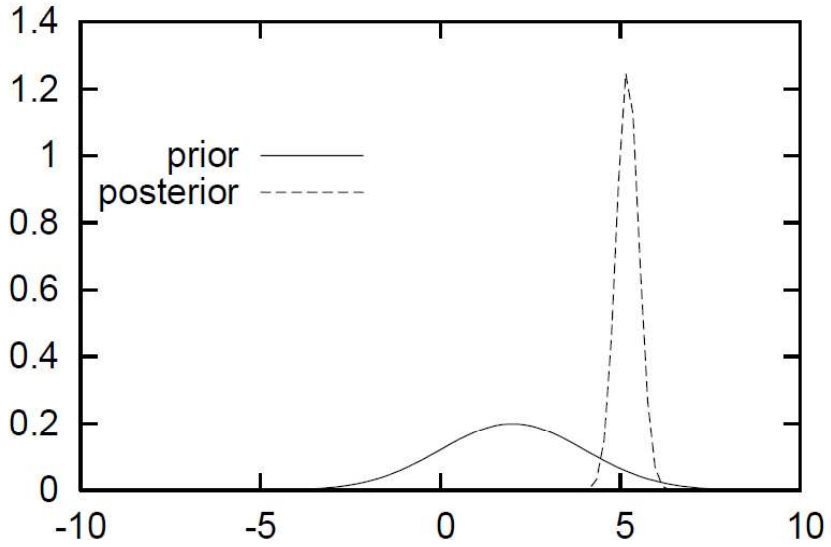

Figure 61: An example of the prior and posterior distributions.

Once we have the posterior density of the parameters, $p(\theta|r_{1:T})$, we can calculate the posterior expectation of any function of the parameters, $g(\theta)$, as

$$E(g(\theta)|r_{1:T}) = \int g(\theta)p(\theta|r_{1:T})d\theta \tag{7.9}$$

For example, the posterior mean of the parameters can be calculated as

$$E(\theta|r_{1:T}) = \int \theta p(\theta|r_{1:T})d\theta \tag{7.10}$$

As an example of deriving the posterior distribution, let us consider the model in Equation 7.7 when σ_a^2 is **known** and μ is unknown with a Gaussian prior. The data generating process in this case is

$$\begin{aligned} r_t &= \mu + a_t, \qquad t = 1, \ldots, T \\ a_t &\overset{\text{i.i.d}}{\sim} N(0, \sigma_a^2), \quad \sigma_a^2 \text{ is known.} \\ \mu &\sim N(\mu_0, \sigma_0^2) \qquad \text{prior for } \mu \end{aligned} \tag{7.11}$$

The posterior distribution of the unknown parameter μ can be derived as follows:

$$
\begin{aligned}
p(\mu|r_1,\ldots,r_T) &\propto p(\mu)\times\prod_{t=1}^{T} f(r_t|\mu)\\
&= N(\mu|\mu_0,\sigma_0^2)\times\prod_{t=1}^{T} N(r_t|\mu,\sigma_a^2)\\
&= \frac{1}{\sqrt{2\pi}\sigma_0}e^{-\frac{1}{2}(\frac{\mu-\mu_0}{\sigma_0})^2}\prod_{t=1}^{T}(\frac{1}{\sqrt{2\pi}\sigma_a}e^{-\frac{1}{2}(\frac{r_t-\mu}{\sigma_a})^2})\\
&= ...\\
&= \frac{1}{\sqrt{2\pi}\sigma_*}e^{-\frac{1}{2}(\frac{\mu-\mu_*}{\sigma_*})^2}\\
&\equiv N(\mu|\mu_*,\sigma_*^2)
\end{aligned}
$$

where

$$\mu_* = \sigma_*^2(\frac{T\bar{r}}{\sigma_a^2}+\frac{\mu_0}{\sigma_0^2}),\quad \sigma_*^2 = (\frac{T}{\sigma_a^2}+\frac{1}{\sigma_0^2})^{-1}$$

We can rewrite the posterior using the precision parameter $h_* = \frac{1}{\sigma_*^2}$ as

$$p(\mu|r_1,\ldots,r_T,\sigma_a) \equiv N(\mu|\mu_*,\frac{1}{h_*})$$

where the posterior mean is the weighted average of the sample mean ($\bar{r}$) and the prior mean (μ_0):

$$\mu_* = \frac{\frac{T}{\sigma_a^2}}{\frac{T}{\sigma_a^2}+\frac{1}{\sigma_0^2}}\bar{r}+\frac{\frac{1}{\sigma_0^2}}{\frac{T}{\sigma_a^2}+\frac{1}{\sigma_0^2}}\mu_0,$$

and the posterior precision is the weighted average of the sample precision ($\frac{1}{\frac{\sigma_a^2}{T}}$) and the prior precision ($\frac{1}{\sigma_0^2}$):

$$h_* = \frac{1}{\sigma_*^2} = \frac{T}{\sigma_a^2}+\frac{1}{\sigma_0^2}$$

In both equations, it is clear how increasing the number of observations, T, makes the prior impacts disappear.

In this example, because σ_a^2 was assumed to be known, we were able to simplify the posterior and derive a standard form for the posterior distribution of μ, by which we can easily make inference about μ. In general, it is not simple to derive a known standard form for the posterior distributions. In general cases that the posterior distribution is not of any standard form, we resort to MCMC techniques to approximate the posterior. For example, consider the following linear regression model with k regressors where the vector of the regression coefficients, β, and the variance of the shocks, σ^2, are both unknown:

$$y_t = x_t'\beta + a_t,\quad a_t \overset{\text{i.i.d}}{\sim} N(0,\sigma^2),\quad t=1,\ldots,T$$

where T is the number of observations, y_t is the observation at time t, $x_t = (x_{t,1},\ldots,x_{t,k})'$ is the vector of regressors, and $\beta = (\beta_1,\ldots,\beta_k)'$ is the vector of the regression coefficients. Here, $\theta = (\beta,\sigma^2)$.

In a Bayesian framework, we first need to define the priors. Here, we work with independent prior distributions: Multivariate normal distribution for β and inverse gamma distribution for σ^2.

$$p(\beta) \equiv N_k(\beta_0, B_0), \qquad p(\sigma^2) \equiv IG(\nu_0/2, s_0/2)$$

β_0, B_0, s_0 and ν_0 are hyper-parameters and are known quantities. The posterior distribution of the parameters then can be written as

$$\begin{aligned} p(\beta, \sigma^2|y_{1:T}, x_{1:T}) &\propto p(\beta, \sigma^2)p(y_{1:T}|(\beta, \sigma^2)) \\ &= p(\beta) \times p(\sigma^2) \times \prod_{t=1}^{T} N(y_t|x_t'\beta, \sigma^2) \\ &= \frac{1}{\sqrt{(2\pi)^k}|B_0|} e^{-\frac{1}{2}(\beta-\beta_0)'B_0^{-1}(\beta-\beta_0)} \times (\sigma^2)^{\frac{\nu_0}{2}-1} e^{-\frac{s_0}{2\sigma^2}} \times \prod_{t=1}^{T} (\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{y_t - x_t'\beta}{\sigma})^2}) \end{aligned}$$

which is not of any standard form! In the next section, we see how we can benefit from the MCMC techniques to make inference about the parameters while we don't have a closed-form posterior distribution.

7.3 Applications of MCMC Techniques

Consider the linear regression model discussed above. We illustrated that the joint posterior distributions of the parameters, $\theta = (\beta, \sigma^2)$, is not of any standard form. However, we can derive the conditional distribution of β and σ^2.

- The posterior distribution of β conditional on knowing σ^2 can be written as

$$\beta|y_{1:T}, x_{1:T}, \sigma^2 \sim N_k(\beta_*, B_*)$$

where

$$\beta_* = B_*^{-1}(\frac{\sum_{t=1}^{T} x_t y_t}{\sigma^2} + B_0^{-1}\beta_0)$$

$$B_* = (\frac{\sum_{t=1}^{T} x_t x_t'}{\sigma^2} + B_0^{-1})^{-1}$$

- The posterior distribution of σ^2 conditional on knowing β can be written as

$$\sigma^2|y_{1:T}, x_{1:T}, \beta \sim IG(\frac{T+\nu_0}{2}, \frac{s_0 + \sum_{t=1}^{T}(y_t - x_t'\beta)^2}{2})$$

Therefore, we can employ the Gibbs sampler in order to get the target density; the joint posterior distribution of (β, σ^2). The steps are as follows:

1. Choose a starting value, $\sigma^{2(0)}$.
2. At the gth iteration, $g = 1, \ldots, G$, draw

$$\beta^{(g)} \sim N_k(\beta_*, B_*),$$

where

$$\beta_* = B_*^{-1}(\frac{\sum_{t=1}^T x_t y_t}{\sigma^{2(g-1)}} + B_0^{-1}\beta_0) \text{ and } B_* = (\frac{\sum_{t=1}^T x_t x_t'}{\sigma^{2(g-1)}} + B_0^{-1})^{-1}$$

Then draw

$$\sigma^{2^{(g)}} \sim IG(\frac{T+\nu_0}{2}, \frac{s_0 + \sum_{t=1}^T (y_t - x_t'\beta^{(g)})^2}{2}),$$

We iterate on this to obtain a set of draws $\{\beta^{(g)}, \sigma^{2(g)}\}_{g=1}^G$, after dropping an initial burn-in. If the chain has converged then we have a sample from the joint posterior distribution from which we can estimate various moments or other features of $p(\beta, \sigma^2|y_{1:T})$. Some examples are:

- $\mathrm{E}[g(\theta)|y_{1:T}] \approx \frac{1}{G}\sum_{g=1}^G g(\theta^{(g)})$. $g(\theta) = \theta$ gives the posterior mean while the posterior variance can be found from $\mathrm{E}[\theta^2|y_{1:T}] - (\mathrm{E}[\theta|y_{1:T}])^2$.
- Quantiles $p(\theta < q_\alpha) = \alpha$, $\alpha \in (0,1)$. First sort the parameter draws from smallest to largest. The quantile q_α is estimated as the order statistics $[\alpha \times G]$ of the sorted draws where $[\ ]$ denotes the integer function. We can also do interpolation as we did in calculating the empirical quantiles in VaR Section.
- Density or histogram plots of the marginals can be constructed directly from the draws.

In R with package *MCMCpack*, use *MCMCregress()* to do the Gibbs sampling for the linear regression model. As an example, look at the regression equation

$$\log rv_t = \beta_0 + \beta_1 \log rv_{t-1} + \beta_2 r_t + a_t, \quad a_t \overset{\text{i.i.d}}{\sim} N(0, \sigma^2), \quad t = 1, \ldots, T$$

Here, $\beta = (\beta_0, \beta_1, \beta_2)'$, and $\theta = (\beta, \sigma)$. The priors are

$$p(\beta) \equiv N_k(\beta_0, B_0), \qquad p(\sigma^2) \equiv IG(\nu_0/2, s_0/2)$$

where $\beta_0 = (0,0,0)'$, $B_0 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$, $\nu_0 = 16$, and $s_0 = 4$.

We run the Gibbs sampler for $G = 1000$ iterations after dropping the first 100 iterations. Figure 62 illustrates the trace plots of the MCMC draws and the empirical densities of the draws of the four unknown parameters.

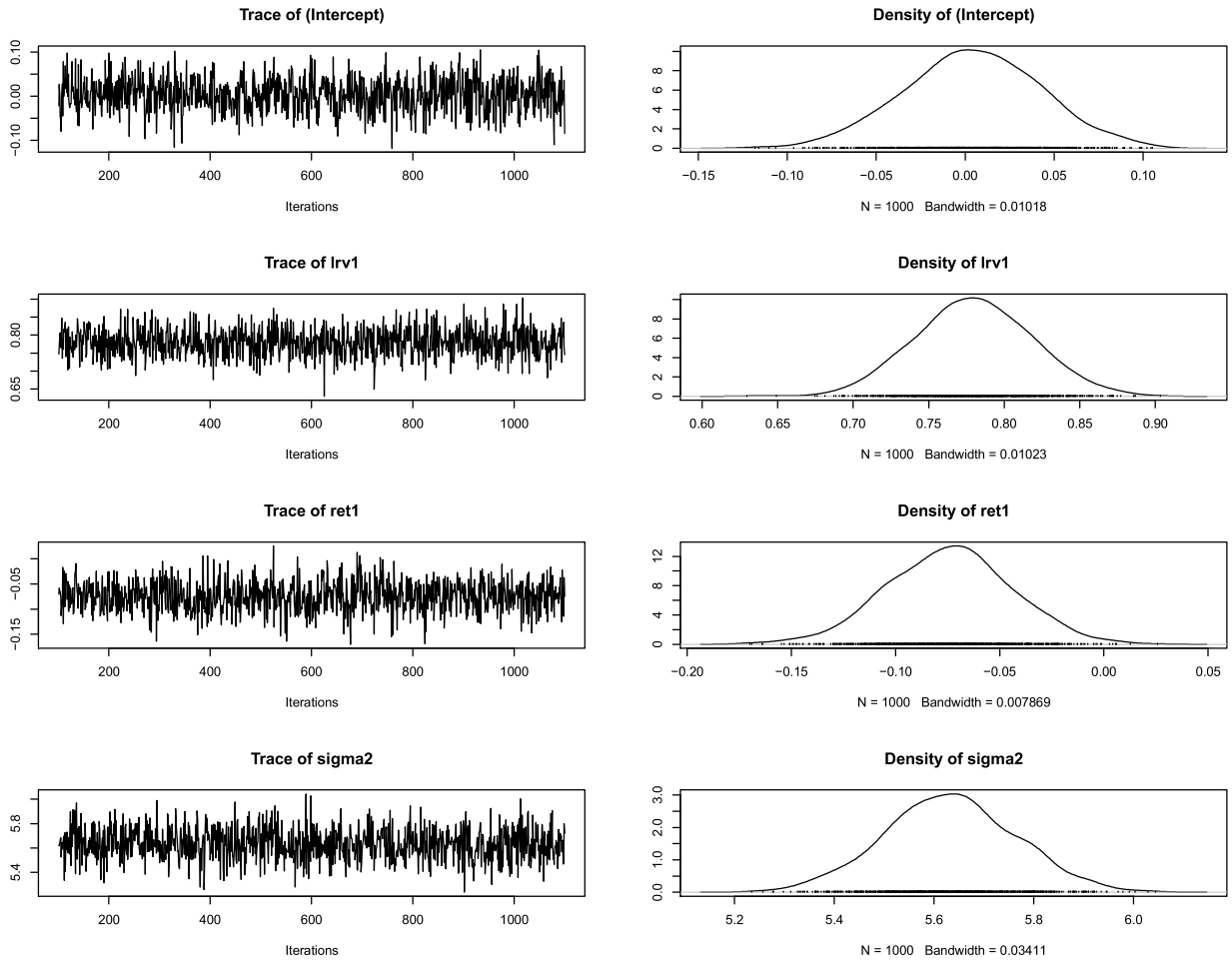

Figure 62: The trace plots for the parameters of the linear regression model using the Gibbs sampler. The dependent variable is the log-realized volatility. The regressors are the lag-1 log-realized volatility and the contemporaneous return.

R Demonstration: Realized Volatility Data

```
library("MCMCpack")

> setwd("C:/Users/ashamsi/Dropbox/LapTop-DeskTop/MQIM6602/Lectures/lec12-18") # set my working directory.

> d <- read.table("rtrv.dat", quote="")
> colnames(d) <- c("date", "r5m", "rd", "rv", "rv1", "rv2", "rv3")
> head(d)

# Use rv to create log rv
d$r5m = 100.*d$r5m
d$logrv = log(d$rv)
head(d)
summary(d)

plot(d$logrv,type="l")
plot(d$r5m,type="l")
plot(d$r5m,d$logrv)

 #create a matrix of lags (2 lags)
lags_logrv = embed( d$logrv,3)

head(lags_logrv)
dim(lags_logrv)
dim(d)

## bind all the dataset back together
m <- cbind(lags_logrv[,1],lags_logrv[,2],d[2:3653,2])
## check a few columns to make sure computations correct
print(m[1:10,])

## put matrix into a dataframe
data <- data.frame(m)
## add names
colnames(data) <- c("lrv","lrv1","ret1")

post_1 <- MCMCregress(lrv ~ lrv1+ret1,b0=0,B0=0.1,sigma.mu=100.0,sigma.var=100.0,burnin=100,mcmc=1000,
  +            data=data,marginal.likelihood="Chib95")

summary(post_1)

varnames(post_1)
dim(post_1)
print(post_1[,1])

## posterior plots
plot(post_1)
$
```

In more general models, we can use the results above after some manipulation. For example, consider a regression model with time series errors discussed before in this course:

$$y_t = x_t'\beta + e_t, \quad e_t = \phi_1 e_{t-1} + a_t, \quad a_t \overset{\text{i.i.d}}{\sim} N(0, \sigma^2), \quad t = 1, \ldots, T$$

We can convert this model to the following model:

$$y_{0,t} = x_{0,t}'\beta + a_t, \quad a_t \overset{\text{i.i.d}}{\sim} N(0, \sigma^2), \quad t = 1, \ldots, T$$

where $y_{0,t} = y_t - \phi_1 y_{t-1}$ and $x_{0,t} = x_t - \phi_1 x_{t-1}$. Conditional on knowing ϕ_1, we can simply get the posterior distribution of β and σ^2 following the steps above and using Gibbs sampler. Then conditional on knowing σ^2, we can get the closed-form posterior distribution of ϕ_1 using the following model (Similar to the model in Equation 7.11)

$$e_t = \phi_1 e_{t-1} + a_t, \quad a_t \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$$