Chapter 4

Non-conjugate multi-parameter problems

In this chapter we will study some multi-parameter problems in which the prior distribution does not have to be conjugate. Inferences are made by using techniques which simulate realisations from the posterior distribution. These methods are generally referred to as *Markov Chain Monte Carlo* techniques, and often abbreviated to MCMC. There are many different MCMC techniques, but we only have time to look briefly at two of the most fundamental. The first is the *Gibbs sampler*, which was at the forefront of the recent MCMC revolution, and the second is generally known as *Metropolis-Hastings* sampling. In fact, MCMC schemes based on the combination of these two fundamental techniques are still at the forefront of MCMC research.

4.1 Why is inference not straightforward in non-conjugate problems?

Example 4.1

Consider again the problem in section 2.2 in which we have a random sample from a normal distribution where both the mean μ and the precision τ are unknown, that is, $X_i|\mu,\tau \sim N(\mu, 1/\tau)$, $i = 1, 2, \ldots, n$ (independent). In this section, we showed that a NGa prior for $(\mu,\tau)^T$ was conjugate, that is, if we used a $NGa(b, c, g, h)$ prior distribution for $(\mu,\tau)^T$ then the posterior was a $NGa(B, C, G, H)$ distribution. But what if a $NGa(b, c, g, h)$ prior distribution does not adequately represent our prior beliefs? Suppose instead that our prior beliefs are represented by independent priors for the parameters, with

$$\mu \sim N\left(b, \frac{1}{c}\right) \qquad \text{and} \qquad \tau \sim Ga(g, h)$$

for known values b, c, g and h. What is the posterior distribution for $(\mu,\tau)^T$?

Solution

Previously we have seen that the likelihood function is

$$f(\boldsymbol{x}|\mu,\tau) = \left(\frac{\tau}{2\pi}\right)^{n/2} \exp\left[-\frac{n\tau}{2}\left\{s^2 + (\bar{x}-\mu)^2\right\}\right].$$

Using Bayes Theorem, the posterior density is

$$\begin{aligned}\pi(\mu,\tau|\boldsymbol{x}) &\propto \pi(\mu,\tau)\, f(\boldsymbol{x}|\mu,\tau)\\ &\propto \pi(\mu)\,\pi(\tau)\, f(\boldsymbol{x}|\mu,\tau)\end{aligned}$$

and so, for $\mu \in \mathbb{R}$, $\tau > 0$

$$\begin{aligned}\pi(\mu,\tau|\boldsymbol{x}) &\propto \left(\frac{c}{2\pi}\right)^{1/2} \exp\left\{-\frac{c}{2}(\mu-b)^2\right\} \times \frac{h^g \tau^{g-1} e^{-h\tau}}{\Gamma(g)}\\ &\qquad \times \tau^{\frac{n}{2}} \exp\left[-\frac{n\tau}{2}\left\{s^2 + (\bar{x}-\mu)^2\right\}\right]\\ &\propto \tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x}-\mu)^2\right]\right\}.\end{aligned}$$

This is not a $NGa(B,C,G,H)$ density as it cannot be written in the form

$$k\,\tau^{G-\frac{1}{2}} \exp\left\{-\frac{\tau}{2}\left[C(\mu-B)^2 + 2H\right]\right\}$$

for any choice of B, C, G and H. In fact $\pi(\mu,\tau|\boldsymbol{x})$ is not the density of any standard distribution.

Can we plot this density? Before we do, we need to know the constant of proportionality, k. As the posterior density must integrate to one

$$\int_{-\infty}^{\infty}\int_{0}^{\infty} k\tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x}-\mu)^2\right]\right\} d\tau\, d\mu = 1$$

$$k^{-1} = \int_{-\infty}^{\infty}\int_{0}^{\infty} \tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x}-\mu)^2\right]\right\} d\tau\, d\mu$$

Therefore the posterior density is, for $\mu \in \mathbb{R}$, $\tau > 0$

$$\pi(\mu,\tau|\boldsymbol{x}) = \frac{\tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x}-\mu)^2\right]\right\}}{\displaystyle\int_{-\infty}^{\infty}\int_{0}^{\infty} \tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x}-\mu)^2\right]\right\} d\tau\, d\mu}.$$

What is the posterior mean of μ and of τ? What are their marginal distributions? How

can we calculate the moments $E(\mu^{m_1}\tau^{m_2}|\boldsymbol{x})$ of this posterior distribution? Now

$$\begin{aligned}\pi(\mu|\boldsymbol{x}) &= \int_0^\infty \pi(\mu,\tau|\boldsymbol{x})\,d\tau \\ &= \frac{\displaystyle\int_0^\infty \tau^{g+\frac{n}{2}-1}\exp\left\{-\frac{c}{2}(\mu-b)^2-h\tau-\frac{n\tau}{2}\left[s^2+(\bar{x}-\mu)^2\right]\right\}d\tau}{\displaystyle\int_{-\infty}^\infty\int_0^\infty \tau^{g+\frac{n}{2}-1}\exp\left\{-\frac{c}{2}(\mu-b)^2-h\tau-\frac{n\tau}{2}\left[s^2+(\bar{x}-\mu)^2\right]\right\}d\tau\,d\mu}\end{aligned}$$

and

$$\begin{aligned}\pi(\tau|\boldsymbol{x}) &= \int_{-\infty}^\infty \pi(\mu,\tau|\boldsymbol{x})\,d\mu \\ &= \frac{\displaystyle\int_{-\infty}^\infty \tau^{g+\frac{n}{2}-1}\exp\left\{-\frac{c}{2}(\mu-b)^2-h\tau-\frac{n\tau}{2}\left[s^2+(\bar{x}-\mu)^2\right]\right\}d\mu}{\displaystyle\int_{-\infty}^\infty\int_0^\infty \tau^{g+\frac{n}{2}-1}\exp\left\{-\frac{c}{2}(\mu-b)^2-h\tau-\frac{n\tau}{2}\left[s^2+(\bar{x}-\mu)^2\right]\right\}d\tau\,d\mu}.\end{aligned}$$

In general, the moments are

$$\begin{aligned}E(\mu^{m_1}\tau^{m_2}|\boldsymbol{x}) &= \int_{-\infty}^\infty\int_0^\infty \mu^{m_1}\tau^{m_2}\pi(\mu,\tau|\boldsymbol{x})\,d\tau\,d\mu \\ &= \frac{\displaystyle\int_{-\infty}^\infty\int_0^\infty \mu^{m_1}\tau^{m_2}\times\tau^{g+\frac{n}{2}-1}\exp\left\{-\frac{c}{2}(\mu-b)^2-h\tau-\frac{n\tau}{2}\left[s^2+(\bar{x}-\mu)^2\right]\right\}d\tau\,d\mu}{\displaystyle\int_{-\infty}^\infty\int_0^\infty \tau^{g+\frac{n}{2}-1}\exp\left\{-\frac{c}{2}(\mu-b)^2-h\tau-\frac{n\tau}{2}\left[s^2+(\bar{x}-\mu)^2\right]\right\}d\tau\,d\mu}.\end{aligned}$$

These integrals cannot be determined analytically, though it is possible to use numerical integration methods or approximations (for large n). However, in general, the accuracy of the numerical approximation to the integral deteriorates as the dimension of the integral increases.

Comment

The above shows how not using a conjugate prior distribution can cause many basic problems such as plotting the posterior density or determining posterior moments. But having to use conjugate priors is far too restrictive for many real data analyses: (i) our prior beliefs may not be captured using a conjugate prior; (ii) most models for complex data do not have conjugate priors. It was for these reasons that until relatively recently (say mid-1990s), practical Bayesian inference for real complex problems was either not feasible or only undertaken by the dedicated few prepared to develop bespoke computer code to numerically evaluate all the integrals etc.

4.2 Simulation-based inference

One way to get around the problem of having to work out integrals (like those in the previous section) is to base inferences on simulated realisations from the posterior distribution. This is the fundamental idea behind MCMC methods. If we could simulate from

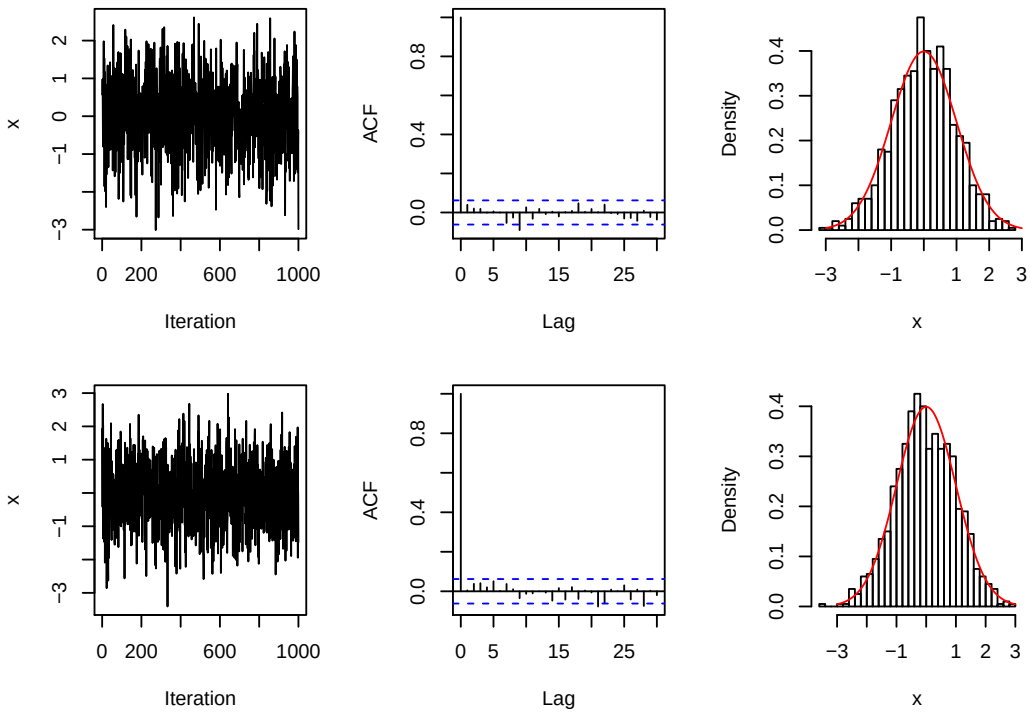

Figure 4.1: Summaries of the 1K realisations from the black box simulator

the posterior distribution then we could use a very large sample of realisations to determine posterior means, standard deviations, correlations, joint densities, marginal densities etc.

As an example, imagine you wanted to know about the standard normal distribution – its shape, its mean, its standard deviation – but didn't know any mathematics so that you couldn't derive say the distribution's zero mean and unit variance. However you've been given a "black box" which can simulate realisations from this distribution. Here we'll use the `R` function `rnorm()` as the black box simulator. If you decide to generate 1K realisations the output might look something like the top row of Figure 4.1. The top left plot shows the trace plot of the output, that is, the realisations from the black box sampler in the order they are produced. The next plot along the top row shows the autocorrelation (ACF) plot. This shows how correlated the realisations are at different lags. We know that the simulator `rnorm()` produces independent realisations and so the (sample) correlation between say consecutive values $corr(x_i, x_{i+1})$ will be almost zero. This is also the case for correlations at all positive lags. Finally the lag 0 autocorrelation $corr(x_i, x_i)$ must be one (by definition). The sample ACF plot is consistent with all of these "claims". Finally the top right plot is a density histogram of the realisations. This too is consistent with the standard normal density (which is also shown). We can also estimate various quantities of the standard normal distribution; for example:

```
 1st Qu.    Median      Mean  3rd Qu.  St.Dev.
-0.65240 -0.00130 -0.00192  0.64810  0.96049
```

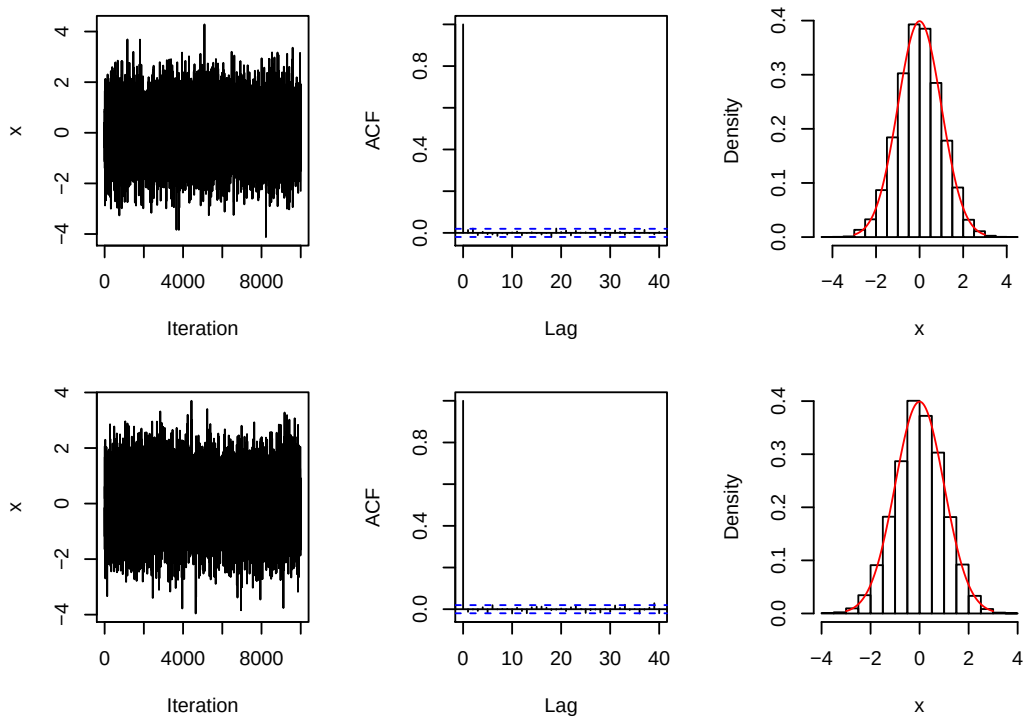

Figure 4.2: Summaries of the 10K realisations from the black box simulator

Here we see that the mean and median are around zero and the standard deviation is around one.

The second row of plots in the figure is another collection of 1K realisations from the black box simulator. These look very similar to those on the top row but are slightly different due to the stochasticity (random nature) of the simulator. This output has the following numerical summaries:

```
 1st Qu.    Median      Mean  3rd Qu.   St.Dev.
-0.69880 -0.09637 -0.03274  0.67330  0.99599
```

Again these numerical summaries are slightly different but essentially tell the same story. In fact we know from previous modules that there is sample variability in estimates of means from random samples. So if we use the simulator again (twice) to obtain 10K realisations, we will get even more similar looking output; see Figure 4.2. The numerical summaries from these outputs are

```
  1st Qu.   Median      Mean  3rd Qu.   St.Dev.
-0.66820 -0.00048  0.00370  0.68070  0.99593
-0.67130  0.01354  0.01008  0.67920  1.00691
```

Here we have much less sampling variability in our estimates due to the larger sample size. In fact we can estimate any "population" quantity to any required accuracy simply by simulating a large enough collection of realisations.

These analyses show how we can make inferences, calculate means, variances, densities etc by using realisations from a distribution. In the rest of this chapter, we will look into how we can construct algorithms for simulating from (complex) posterior distributions, from which we can then make inferences.

4.3 Motivation for MCMC methods

The example in section 4.1 showed that using non-conjugate priors can be problematic. MCMC methods address this problem by providing an algorithm which simulates realisations from the posterior distribution.

We consider a generic case where we want to simulate realisations of two random variables X and Y with joint density $f(x, y)$. This joint density can be factorised as

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

and so we can simulate from $f(x, y)$ by first simulating $X = x$ from $f(x)$, and then simulating $Y = y$ from $f(y|x)$. On the other hand, we can write

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

and so simulate $Y = y$ from $f(y)$ and then $X = x$ from $f(x|y)$.

We have already seen that dealing with conditional posterior distributions is straightforward when the prior is semi-conjugate, so let's assume that simulating from $f(y|x)$ and $f(x|y)$ is straightforward. The key problem with using either of the above methods is that, in general, we can't simulate from the marginal distribution, $f(x)$ and $f(y)$.

For the moment, suppose we can simulate from the marginal distribution for X, that is, we have an $X = x$ from $f(x)$. We can now simulate a $Y = y$ from $f(y|x)$ to give a pair (x, y) from the bivariate density. Given that this pair is from the bivariate density, the y value must be from the marginal $f(y)$, and so we can simulate an $X = x'$ from $f(x|y)$ to give a new pair (x', y) also from the joint density. But now x' is from the marginal $f(x)$, and so we can simulate a $Y = y'$ from $f(y|X = x')$ to give a new pair (x', y') also from the joint density. And we can keep going.

This alternate sampling from conditional distributions defines a bivariate Markov chain, and the above is an intuitive explanation for why $f(x, y)$ is its stationary distribution. Thus being able to simulate easily from conditional distributions is key to this methodology.

4.4 The Gibbs sampler

Suppose we want to generate realisations from the posterior density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_p)^T$, and that we can simulate from the full conditional distributions (FCDs)

$$\pi(\theta_i|\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_p, \boldsymbol{x}) = \pi(\theta_i|\cdot), \qquad i = 1, 2, \ldots, p.$$

The Gibbs sampler follows the following algorithm:

1. Initialise the iteration counter to $j = 1$.
   Initialise the state of the chain to $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})^T$.
2. Obtain a new value $\boldsymbol{\theta}^{(j)}$ from $\boldsymbol{\theta}^{(j-1)}$ by successive generation of values
$$\begin{aligned}
\theta_1^{(j)} &\sim \pi(\theta_1|\theta_2^{(j-1)}, \theta_3^{(j-1)}, \ldots, \theta_p^{(j-1)}, \boldsymbol{x}) \\
\theta_2^{(j)} &\sim \pi(\theta_2|\theta_1^{(j)}, \theta_3^{(j-1)}, \ldots, \theta_p^{(j-1)}, \boldsymbol{x}) \\
&\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
\theta_p^{(j)} &\sim \pi(\theta_p|\theta_1^{(j)}, \theta_2^{(j)}, \ldots, \theta_{p-1}^{(j)}, \boldsymbol{x})
\end{aligned}$$
3. Change counter j to $j + 1$, and return to step 2.

This algorithm defines a homogeneous Markov chain as each simulated value depends only on the previous simulated value and not on any other previous values or the iteration counter j. It can be shown that $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ is the stationary distribution of this chain and so if we simulate realisations by using a Gibbs sampler, eventually the the Markov chain will converge to the required posterior distribution.

4.4.1 Processing output from a Gibbs sampler

Burn-in period

First we have to determine how many iterations are needed before the Gibbs sampler has reached its stationary distribution. This is known as the *burn-in* period. There are many diagnostic tests available to help determine how long this is but, in general, the most effective method is simply to look at a trace plot of the posterior sample and detect the point after which the realisations look to be from the same distribution. Figure 4.3 illustrates typical output from a Gibbs sampler. Here we see the output when using three different starting points. Initially there is a transient stage and then the distribution of the output becomes the same for each chain (perhaps after iteration 500).

Once the Gibbs sampler has reached its stationary distribution, all subsequent iterates are realisations from the posterior distribution. Suppose that $\boldsymbol{\theta} = (\mu, \tau)^T$ and we have run the Gibbs sampler for N iterations after convergence giving a posterior sample
$$\{(\mu^{(1)}, \tau^{(1)}), (\mu^{(2)}, \tau^{(2)}), \ldots, (\mu^{(N)}, \tau^{(N)})\}.$$
We can use this sample to calculate any features of the posterior distribution. For example, we can estimate the marginal posterior densities $\pi(\mu|\boldsymbol{x})$ and $\pi(\tau|\boldsymbol{x})$ by using histograms of the $\mu^{(j)}$ and of the $\tau^{(j)}$ respectively. Of course, as N is finite we cannot determine these densities exactly. We can also estimate other features of the posterior distribution such as the posterior means, variances and correlation by using their equivalents in the posterior sample: $\bar{\mu}$, $\bar{\tau}$, s_μ^2, s_τ^2 and $r_{\mu\tau}$. Again these estimates will not be exact as N is finite. However, we can make them as accurate as we want by taking a sufficiently large posterior sample, that is, by taking N large enough.

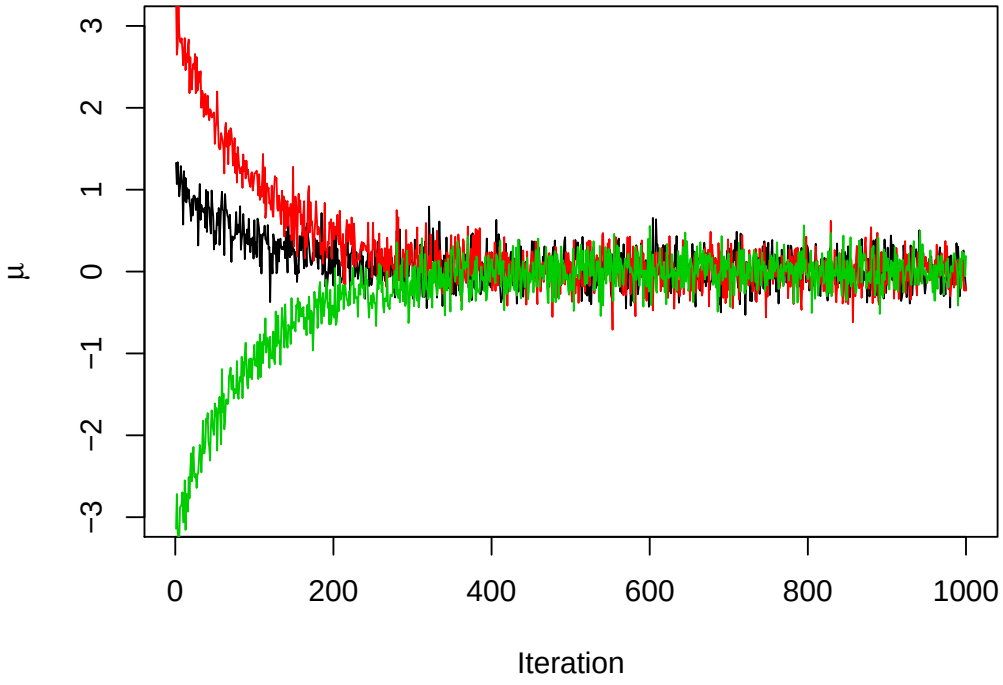

Figure 4.3: Demonstration of burn-in using a Gibbs sampler and three initial values

Dealing with autocorrelation

Another problem with the output from a Gibbs sampler (after convergence) is that it is not a random sample. It should not be surprising that successive realisations from the sampler are autocorrelated, after all the output is a realisation from a Markov chain! To understand the dependence structure, we look at the sample autocorrelation function for each variable. For example, the sample autocorrelation function (ACF) for μ at lag k is

$$r(k) = Corr(\mu^{(j)}, \mu^{(j+k)}).$$

Looking at a plot of this ACF can give an idea as to how much to *thin* the output before it becomes un-autocorrelated (the sample autocorrelations at lags 1,2,3,... are small). Thinning here means not taking every realisation by say taking say every mth realisation. In general, an appropriate level of thinning is determined by the largest lag m at which any of the variables have a non-negligible autocorrelation. If doing this leaves a (thinned) posterior sample which is too small then the original Gibbs sampler should be re-run (after convergence) for a sufficiently large number of iterations until the thinned sample is of the required size.

To get a clearer idea of how thinning works, we now look at some output from a moving average $MA(1)$ process. Figure 4.4 shows this output together with its sample autocorrelation function. Theoretically output from this process should have zero autocorrelations at lags greater than one, and this is what we see (with sample noise) in the figure. If we now thin the output by taking every other value then it's clear that the autocorrelations at non-zero lags should be zero. The lower two plots show the thinned output and its

sample autocorrelation function. This ACF plot suggests that the thinned output is not autocorrelated.

We now look at some output from an autoregressive $AR(1)$ process. Figure 4.5 shows this output together with its sample autocorrelation function. As mentioned previously, theoretically output from this process has autocorrelations which decrease geometrically, and this is what we see (with sample noise) in the figure. The smallest lag after which the autocorrelations are negligible is around 7–10. If we now thin the output by taking every 10th value then we should get output which is almost un-autocorrelated. The lower two plots show the thinned output and its sample autocorrelation function and the ACF plot is consistent with the thinned output being un-autocorrelated.

If the MCMC output is un-autocorrelated then the accuracy of $\bar{\mu}$ is roughly $\pm 2s_\mu/\sqrt{N}$. However if the Markov chain followed an autoregressive $AR(1)$ process, its autocorrelations would decrease geometrically, with $r(k) \simeq r(1)^k$, $k \geq 2$. In this case it can be shown that the accuracy of $\bar{\mu}$ is roughly $\pm 2s_\mu/\sqrt{N\{1-r(1)\}^2}$, that is, because the process has autocorrelation, the amount of information in the data is equivalent to a random sample with size $N_{eff} = N\{1-r(1)\}^2$. This effective random sample size calculation gets more complicated for processes with non-zero higher order autocorrelations and this is why we usually adopt the simplistic method of thinning. It's worth noting that, in general, MCMC output with positive autocorrelations has $N_{eff} < N$. Also sometimes MCMC output with some negative autocorrelations can have $N_{eff} > N$.

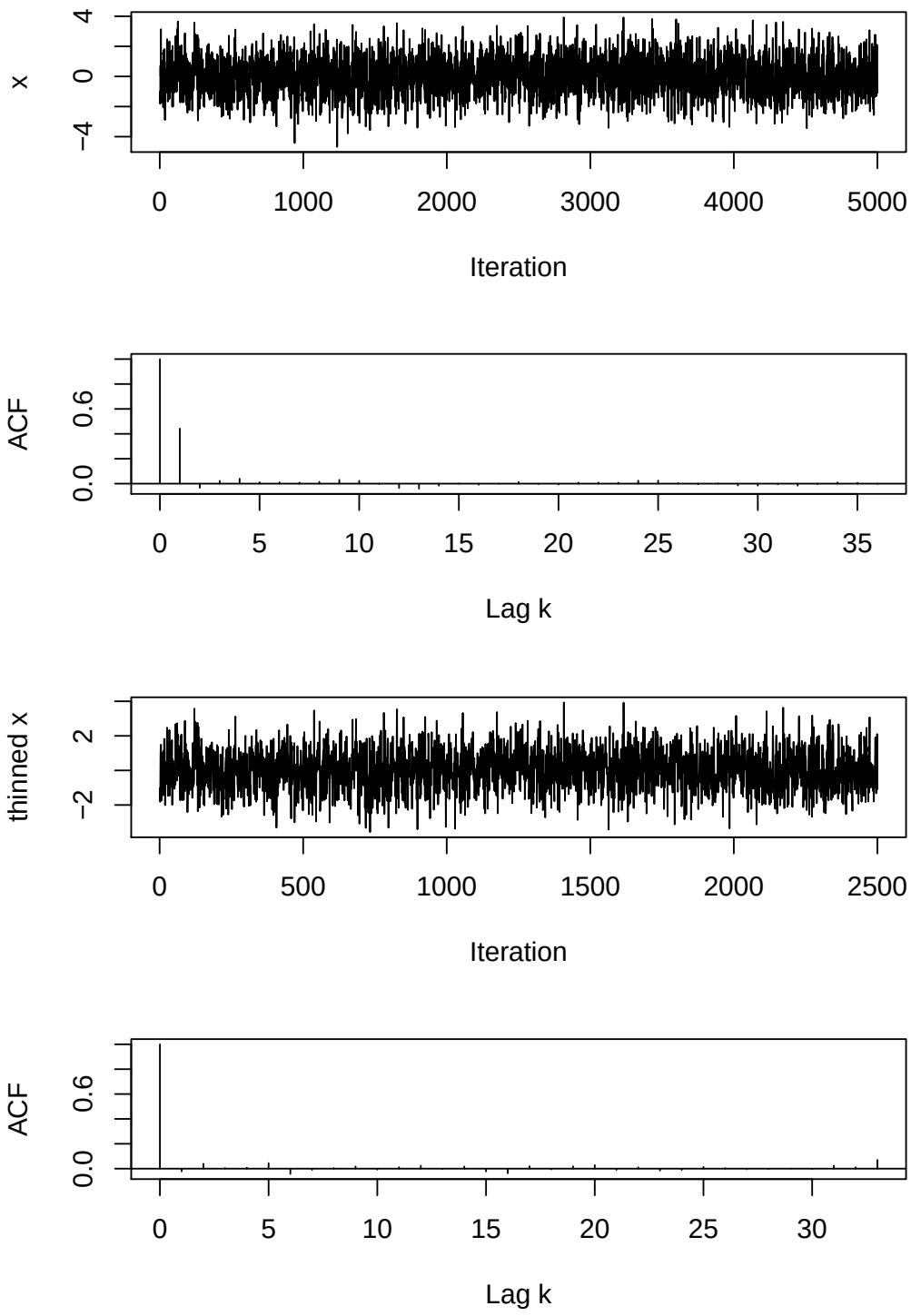

Figure 4.4: Effect of thinning output from a $MA(1)$ process

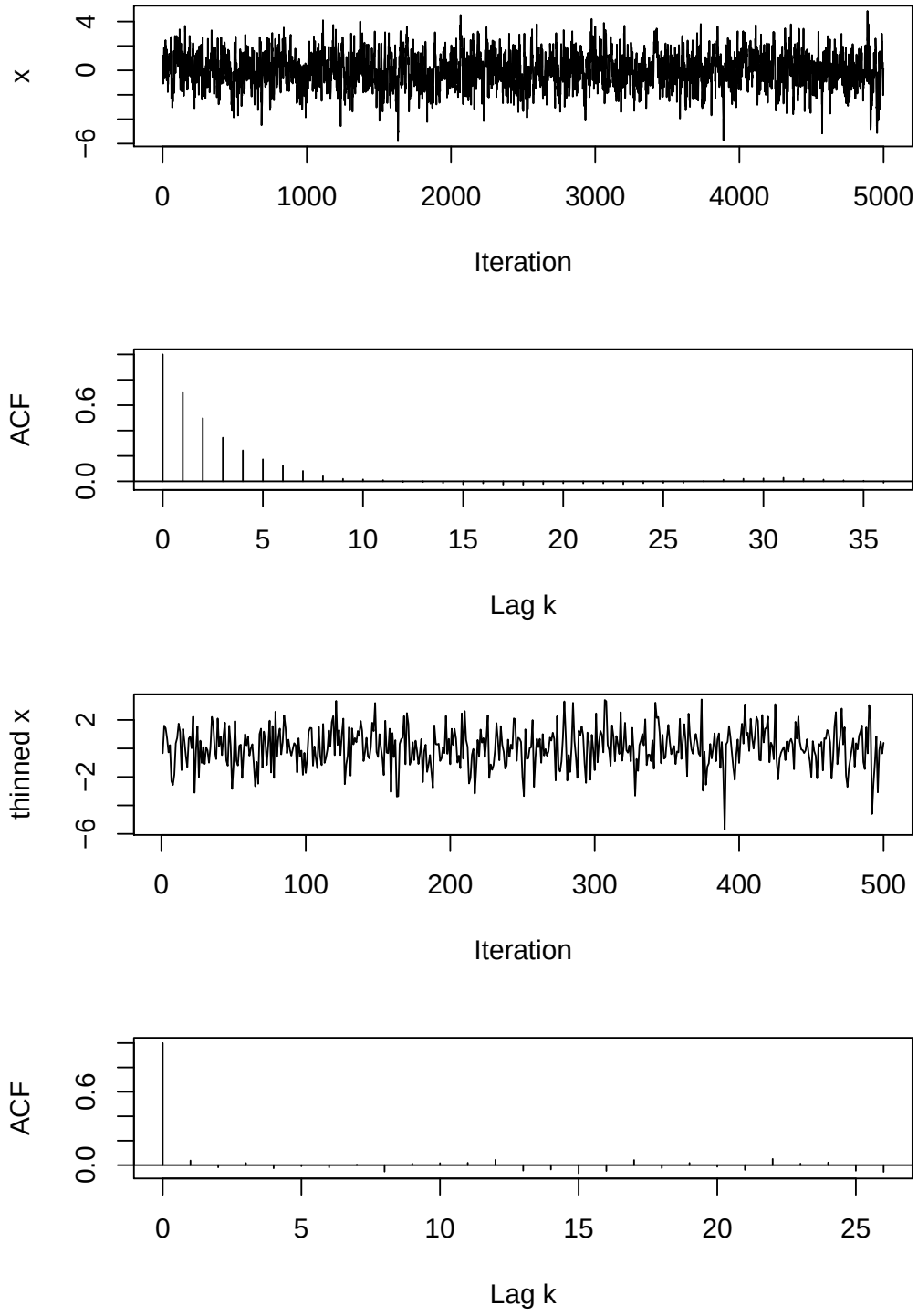

Figure 4.5: Effect of thinning output from a $AR(1)$ process

Strategy

1. Determine the *burn-in* period, after which the Gibbs sampler has reached its stationary distribution. This may involve thinning the posterior sample as slowly snaking trace plots may be due to high autocorrelations rather than a lack of convergence.
2. After this, determine the level of thinning needed to obtain a posterior sample whose autocorrelations are roughly zero.
3. Repeat steps 1 and 2 several times using different initial values to make sure that the sample really is from the stationary distribution of the chain, that is, from the posterior distribution.

Accuracy of posterior summaries

Each time we run an MCMC scheme, we obtain a different sample from the posterior distribution. Suppose that after burn-in and thinning, we have a large sample with N unautocorrelated values, say $\mu_1, \ldots, \mu_N$. In order to determine the accuracy of the sample mean and standard deviation estimates of the posterior mean and standard deviation we need to make some assumption about the posterior distribution. If the data sample size n is large then the posterior distribution will be approximately normal. So we will think of our MCMC output as being a random sample from the posterior distribution.

Suppose the posterior output has sample mean $\bar{\mu}$ and standard deviation s_μ. We need to know the accuracy of these estimates of $M = E(\mu|\boldsymbol{x})$ and $\Sigma = SD(\mu|\boldsymbol{x})$. We saw in Example 3.3 that the asymptotic posterior distribution about the mean and precision $(\mu, \tau)^T$ using a random sample from a normal $N(\mu, 1/\tau)$ distribution was

$$\mu|\boldsymbol{x} \sim N(\bar{x}, s^2/n), \qquad \tau|\boldsymbol{x} \sim N\{1/s^2, 2/(ns^4)\}, \qquad \text{independently.}$$

Rewriting this result in terms of the MCMC sample mean $\bar{\mu}$, standard deviation s_μ and the parameters they estimate gives posterior distributions

$$M \sim N(\bar{\mu}, s_\mu^2/N), \qquad \Sigma^{-2} \sim N\{1/s_\mu^2, 2/(Ns_\mu^4)\}, \qquad \text{independently.}$$

Therefore an approximate 95% HDI for M is

$$\bar{\mu} \pm z_{0.025}\frac{s_\mu}{\sqrt{N}} \simeq \bar{\mu} \pm \frac{2s_\mu}{\sqrt{N}}$$

since $z_{0.025} \simeq 2$.

Also, from the posterior distribution for Σ^{-2}, we have

$$\begin{aligned} & P\left(\frac{1}{s_\mu^2} - 2\sqrt{\frac{2}{Ns_\mu^4}} < \Sigma^{-2} < \frac{1}{s_\mu^2} + 2\sqrt{\frac{2}{Ns_\mu^4}}\right) \simeq 0.95 \\ \Longrightarrow \quad & P\left(\frac{1 - 2\sqrt{2/N}}{s_\mu^2} < \Sigma^{-2} < \frac{1 + 2\sqrt{2/N}}{s_\mu^2}\right) \simeq 0.95 \\ \Longrightarrow \quad & P\left(\frac{s_\mu}{\sqrt{1 + 2\sqrt{2/N}}} < \Sigma < \frac{s_\mu}{\sqrt{1 - 2\sqrt{2/N}}}\right) \simeq 0.95 \end{aligned}$$

Therefore a 95% confidence interval for Σ is

$$s_\mu \left(1 \pm 2\sqrt{2/N}\right)^{-1/2} \simeq s_\mu \left(1 \pm \frac{1}{2} \times 2\sqrt{2/N}\right) = s_\mu \pm s_\mu \sqrt{\frac{2}{N}}.$$

It can be shown that these accuracy calculations are fairly accurate even when the posterior distribution (from which we have the MCMC sample) is not particularly normal.

4.4.2 Bayesian inference using a Gibbs sampler

Example 4.2

Construct a Gibbs sampler for the posterior distribution in Example 4.1.

Solution

We first need to determine the conditional posterior density for μ and for τ.

The conditional posterior density for μ is, for $\mu \in \mathbb{R}$

$$\begin{aligned}
\pi(\mu|\tau, \boldsymbol{x}) &= \frac{\pi(\mu, \tau|\boldsymbol{x})}{\pi(\tau|\boldsymbol{x})} \\
&\propto \pi(\mu, \tau|\boldsymbol{x}) \\
&\propto \tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu - b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x} - \mu)^2\right]\right\} \\
&\propto \exp\left\{-\frac{c}{2}(\mu - b)^2 - \frac{n\tau}{2}(\bar{x} - \mu)^2\right\}
\end{aligned}$$

after moving all multiplicative constants not involving μ into the proportionality sign, and so after completing the square in the exponent

$$\pi(\mu|\tau, \boldsymbol{x}) \propto \exp\left\{-\frac{c + n\tau}{2}\left(\mu - \frac{bc + n\tau\bar{x}}{c + n\tau}\right)^2\right\}, \mu \in \mathbb{R}$$

that is

$$\mu|\tau, \boldsymbol{x} \sim N\left(\frac{bc + n\tau\bar{x}}{c + n\tau}, \frac{1}{c + n\tau}\right).$$

Note that we could have obtained this result by using Example 1.3 with $d = c$.

The conditional posterior density for τ is, for $\tau > 0$

$$\begin{aligned}
\pi(\tau|\mu, \boldsymbol{x}) &= \frac{\pi(\mu, \tau|\boldsymbol{x})}{\pi(\mu|\boldsymbol{x})} \\
&\propto \pi(\mu, \tau|\boldsymbol{x}) \\
&\propto \tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu - b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x} - \mu)^2\right]\right\} \\
&\propto \tau^{g+\frac{n}{2}-1} \exp\left\{-\left[h + \frac{n}{2}\left\{s^2 + (\bar{x} - \mu)^2\right\}\right]\tau\right\}
\end{aligned}$$

after moving all multiplicative constants not involving τ into the proportionality sign, and so

$$\tau|\mu, \boldsymbol{x} \sim Ga\left(g + \frac{n}{2}, h + \frac{n}{2}\left\{s^2 + (\bar{x} - \mu)^2\right\}\right).$$

We also need to initialise the algorithm. We might use the prior mean ($\mu^{(0)} = b$, $\tau^{(0)} = g/h$) or the mle ($\mu^{(0)} = \bar{x}$, $\tau^{(0)} = 1/s^2$). Alternatively, if we wanted to compare different runs of the Gibbs sampler, we might simulate $(\mu^{(0)}, \tau^{(0)})$ from the prior distribution.

Therefore the Gibbs sampler is

1. Initialise the iteration counter to $j = 1$.
   Initialise the state of the chain by taking $(\mu^{(0)}, \tau^{(0)})$ as $(b, g/h)$ or $(\bar{x}, 1/s^2)$ or as a random draw from the prior distribution.
2. Obtain new values $\mu^{(j)}$ and $\tau^{(j)}$ from $\mu^{(j-1)}$ and $\tau^{(j-1)}$ via
$$\mu^{(j)} \sim N\left(\frac{bc + n\tau^{(j-1)}\bar{x}}{c + n\tau^{(j-1)}}, \frac{1}{c + n\tau^{(j-1)}}\right)$$
$$\tau^{(j)} \sim Ga\left(g + \frac{n}{2}, h + \frac{n}{2}\left\{s^2 + \left(\bar{x} - \mu^{(j)}\right)^2\right\}\right)$$
3. Change counter j to $j + 1$, and return to step 2.

Comments

Notice that, since μ and τ independent *a priori*, $\mu|\tau \sim N(b, 1/c)$. Therefore, given τ, the normal prior for μ is conjugate. Similarly, $\tau|\mu \sim Ga(g, h)$ and so, given μ, the gamma prior for τ is conjugate. Therefore, both conditional priors (for $\mu|\tau$ and $\tau|\mu$) are conjugate. Such priors are called *semi-conjugate*.

Producing and analysing output from this Gibbs sampler

The `R` function `gibbsNormal` in the library `nclbayes` implements this Gibbs sampling algorithm. The library also contains the functions `mcmcProcess` which can be used to remove the burn–in and thin the output, and `mcmcAnalysis` which analyses the MCMC output. Let us consider the case in which the data have size $n = 100$, mean $\bar{x} = 15$ and standard deviation $s = 4.5$ and the prior distribution has $\mu \sim N(10, 1/100)$ and $\tau \sim Ga(3, 12)$, independently. The following code produces output from this Gibbs sampler, initialising at $(10, 0.25)$ and then analyses the output.

```
library(nclbayes)
posterior=gibbsNormal(N=1000,initial=c(10,0.25),
    priorparam=c(10,1/100,3,12),n=100,xbar=15,s=4.5)
posterior2=mcmcProcess(input=posterior,burnin=10,thin=1)
```

```
op=par(mfrow=c(2,2))
plot(posterior,col=c(1:length(posterior)),main="All realisations")
plot(posterior,type="l",main="All realisations")
plot(posterior2,col=c(1:length(posterior2)),main="After deleting first 10")
plot(posterior2,type="l",main="After deleting first 10")
par(op)

mcmcAnalysis(posterior,rows=2,show=F)
mcmcAnalysis(posterior2,rows=2,show=F)
```

The first block of code runs the function `gibbsNormal`, with initial values `initial` taken as the prior means, to obtain the output. The next block then uses the function `mcmcProcess` to post-process the Gibbs output by deleting an initial `burnin` $= 10$ values and then not thinning by taking `thin` $= 1$. The next block of code produces the plots in Figure 4.6. After this the code analyses the Gibbs sampler output and produces the plots in Figure 4.7.

In Figure 4.6, the top left plot shows the values produced by the Gibbs sampler: notice the initial value $(\mu^{(0)}, \tau^{(0)}) = (b = 10, g/h = 0.25)$ appears at the top left part of the plot and the other values towards the bottom right part (in different colours). The top right plot is another representation of this output but here each consecutive pair $(\mu^{(j)}, \tau^{(j)})$ are joined by a line. This clearly shows that all pairs after the first one remain in the same vicinity (the bottom right part). The lower plots are the equivalent ones to the upper plots but only use the Gibbs sampler output after deleting the first 10 pairs.

In Figure 4.7, the top two rows of plots summarise the Gibbs sampler output using all realisations and the bottom two rows of plots are the equivalent plots but only use the Gibbs sampler output after deleting the first 10 pairs. The first column of plots shows the trace plot of the output, that is, the values for each variable (μ and τ) as the sampler iterates from its first value to its final value. The top two first column plots clearly show the initial value $(\mu^{(0)}, \tau^{(0)}) = (b = 10, g/h = 0.25)$, after which the subsequent values all look to be from the same distribution. In particular, there looks to be no change in the range of values or in the mean value. The bottom two first column plots emphasise these points; here the first 10 values have been deleted as burn-in, though probably we needed only to delete the first 2 or 3 values. Note that the benefit of using `R` code that runs quickly is that adopting a conservative strategy which deletes too many values as burn-in, does not have significant time implications (though this is not a sensible strategy if the code is very slow and takes months to run!).

The second column shows the autocorrelation function for each variable. Note that the spike at lag 0 is due to $r(0) = Corr(\mu^{(j)}, \mu^{(j)}) = 1$. The plots show that the autocorrelations at all other lags are negligible, and so no thinning is needed. The final column shows histograms of the Gibbs sampler output. If using a burn-in of 10 iterations is okay (and it is here!) then the subsequent output can be taken as our posterior sample and therefore the lower two histograms will be good estimates of the marginal densities: good because the output is (almost) uncorrelated and the sample size is quite large.

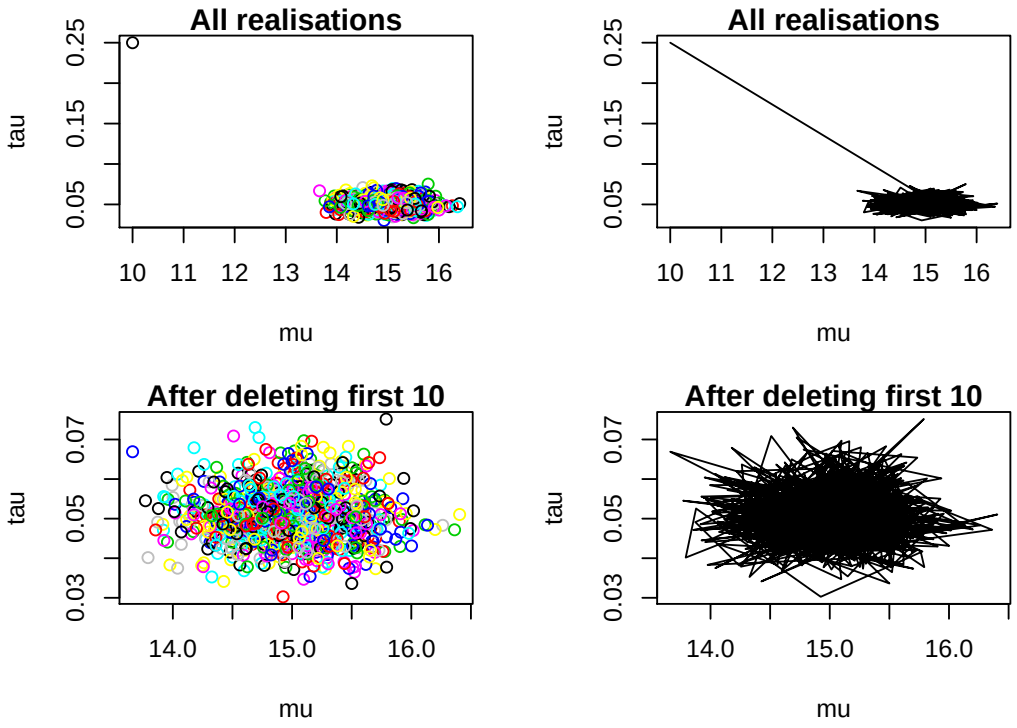

Figure 4.6: Progress of the MCMC scheme

If we use the command

```
mcmcAnalysis(posterior2,rows=2)
```

that is, don't use the `show=F` option, then the function will produce the plots and also various useful numerical summaries. In this run of the Gibbs sampler it gave

```
N = 990 iterations
       mu              tau
 Min.   :13.66   Min.   :0.03025
 1st Qu.:14.68   1st Qu.:0.04663
 Median :15.01   Median :0.05068
 Mean   :14.99   Mean   :0.05110
 3rd Qu.:15.29   3rd Qu.:0.05557
 Max.   :16.41   Max.   :0.07515
Standard deviations:
         mu          tau
0.448562708 0.006743413
```

We can also calculate other features of the joint posterior distribution such as its correlation

$$Corr(\mu, \tau|\boldsymbol{x}) = -0.002706$$

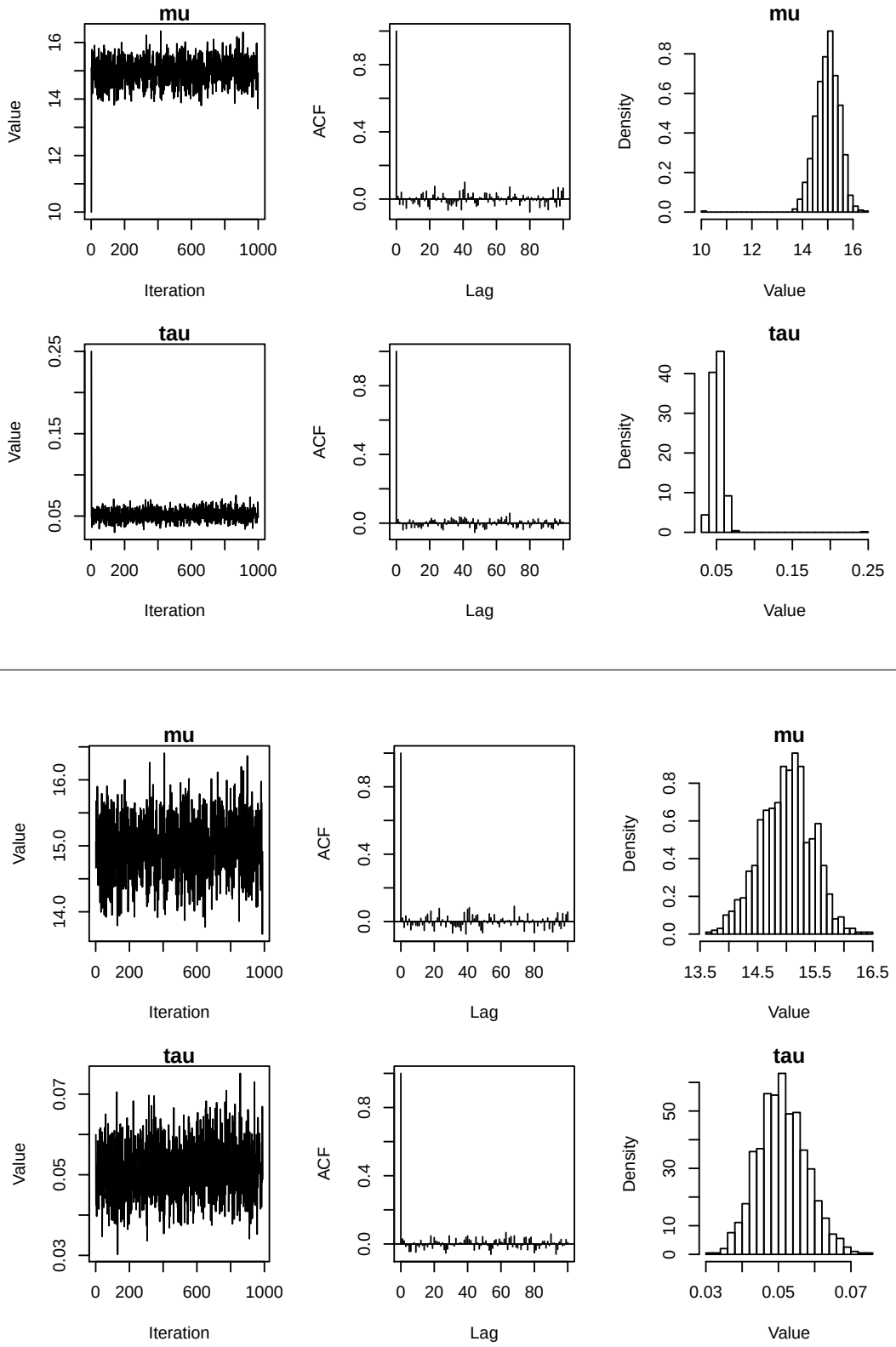

Figure 4.7: Trace plots, autocorrelation plots and histograms of the Gibbs sampler output. Upper plots: all realisations. Lower plots: after deleting the first 10 iterations

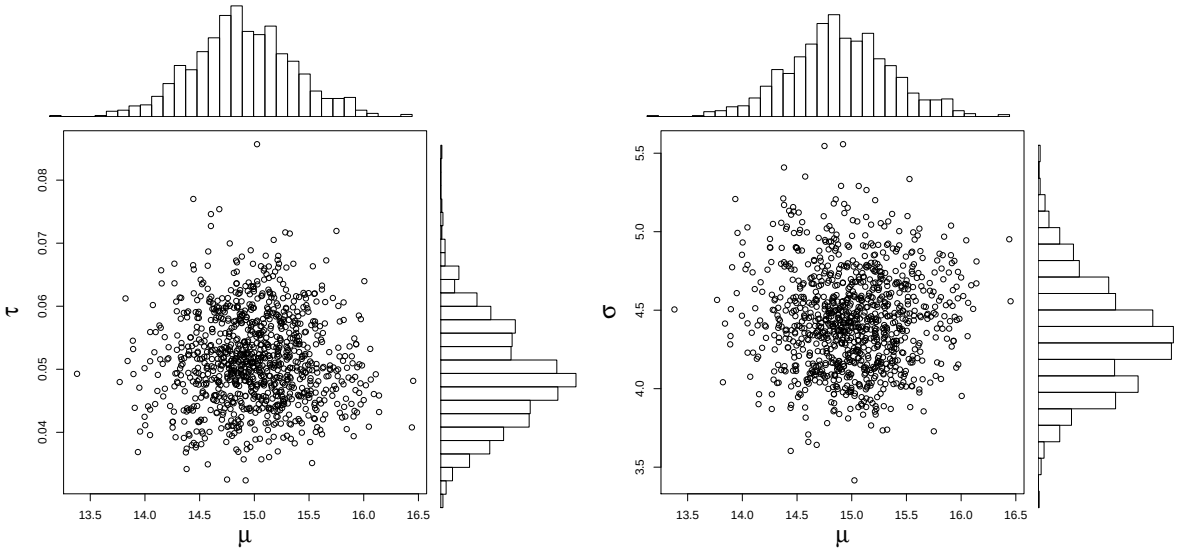

Figure 4.8: Plot of the bivariate posterior sample and their marginal distributions. Left plot: $(\mu, \tau)^T$; right plot: $(\mu, \sigma)^T$.

using the command `cor(posterior2)`, and summarise the posterior sample with a plot of its values and its marginal distributions; see Figure 4.8. We can also determine $100(1-\alpha)\%$ equi-tailed confidence intervals as follows. Suppose we have N realisations from our Gibbs sampler. If we sort the values into increasing order then the confidence interval will have end points which are $N\alpha/2$th and $N(1-\alpha/2)$th values. The `nclbayes` package has a function `mcmcCi` to do this. In this case we would use `mcmcCi(posterior2,level=0.95)` and, for this output, obtain the 95% confidence intervals as

$$\begin{aligned} \mu: &\quad (14.071,\ 15.803) \\ \tau: &\quad (0.03818,\ 0.06499). \end{aligned}$$

Now we have a sample from the posterior distribution, we can determine the posterior distribution for any function of the parameters. For example, if we want the posterior distribution for $\sigma = 1/\sqrt{\tau}$ then we can easily obtain realisations of σ as $\sigma^{(j)} = 1/\sqrt{\tau^{(j)}}$, from which we can produce a plot of its values and its marginal distributions (see Figure 4.8) and also obtain its numerical summaries

```
> sigma=1/sqrt(posterior2[,2])
> summary(sigma)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.648   4.242   4.442   4.453   4.631   5.750
> sd(sigma)
[1] 0.2977992
> quantile(sigma,probs=c(0.025,0.975))
    2.5%    97.5%
3.922480 5.109632
```

Summary

We can use the (converged and thinned) MCMC output to do the following.

- Obtain the posterior distribution for any (joint) functions of the parameters, such as $\sigma = 1/\sqrt{\tau}$ or $(\theta_1 = \mu - \tau, \theta_2 = e^{\mu+\tau/2})^T$
- Look at bivariate posterior distributions via scatter plots
- Look at univariate marginal posterior distributions via histograms or boxplots
- Obtain numerical summaries such as the mean, standard deviation and confidence intervals for single variables and correlations between variables.

Example 4.3

Gibbs sampling can also be used when using a conjugate prior. Construct a Gibbs sampler for the problem in Example 2.2 analysing Cavendish's data on the earth's density. Recall this assumed the data were a random sample from a normal distribution with unknown mean μ and precision τ, that is, $X_i|\mu,\tau \sim N(\mu, 1/\tau)$, $i = 1, 2, \ldots, n$ (independent), and took a conjugate NGa prior distribution for $(\mu, \tau)^T$.

Solution

The joint posterior density is, for $\mu \in \mathbb{R}$, $\tau > 0$

$$\pi(\mu, \tau|\boldsymbol{x}) \propto \tau^{G-\frac{1}{2}} \exp\left\{-\frac{\tau}{2}\left[C(\mu - B)^2 + 2H\right]\right\}$$

where

$$B = \frac{bc + n\bar{x}}{c+n}, \qquad C = c + n,$$
$$G = g + \frac{n}{2}, \qquad H = h + \frac{cn(\bar{x} - b)^2}{2(c+n)} + \frac{ns^2}{2}.$$

Therefore the conditional posterior density for μ is, for $\mu \in \mathbb{R}$

$$\begin{aligned}
\pi(\mu|\tau, \boldsymbol{x}) &= \frac{\pi(\mu, \tau|\boldsymbol{x})}{\pi(\tau|\boldsymbol{x})} \\
&\propto \pi(\mu, \tau|\boldsymbol{x}) \\
&\propto \tau^{G-\frac{1}{2}} \exp\left\{-\frac{\tau}{2}\left[C(\mu - B)^2 + 2H\right]\right\} \\
&\propto \exp\left\{-\frac{C\tau}{2}(\mu - B)^2\right\}
\end{aligned}$$

after moving all multiplicative constants not involving μ into the proportionality sign, and so

$$\mu|\tau, \boldsymbol{x} \sim N\left(B, \frac{1}{C\tau}\right).$$

(Actually we already knew this from the definition of the NGa distribution.)

The conditional posterior density for τ is, for $\tau > 0$

$$\begin{aligned}\pi(\tau|\mu, \boldsymbol{x}) &= \frac{\pi(\mu, \tau|\boldsymbol{x})}{\pi(\mu|\boldsymbol{x})} \\ &\propto \pi(\mu, \tau|\boldsymbol{x}) \\ &\propto \tau^{G-\frac{1}{2}} \exp\left\{-\frac{\tau}{2}\left[C(\mu - B)^2 + 2H\right]\right\}\end{aligned}$$

and so

$$\tau|\mu, \boldsymbol{x} \sim Ga\left(G + \frac{1}{2},\ H + \frac{C}{2}(\mu - B)^2\right).$$

We will initialise the algorithm using the prior means: $\mu^{(0)} = b$ and $\tau^{(0)} = g/h$. Therefore the Gibbs sampler is

1. Initialise the iteration counter to $j = 1$.
   Initialise the state of the chain to $\mu^{(0)} = b$ and $\tau^{(0)} = g/h$.
2. Obtain new values $\mu^{(j)}$ and $\tau^{(j)}$ from $\mu^{(j-1)}$ and $\tau^{(j-1)}$ via

$$\begin{aligned}\mu^{(j)} &\sim N\left(B, \frac{1}{C\tau^{(j-1)}}\right) \\ \tau^{(j)} &\sim Ga\left(G + \frac{1}{2},\ H + \frac{C}{2}\left(\mu^{(j)} - B\right)^2\right)\end{aligned}$$

3. Change counter j to $j + 1$, and return to step 2.

The R function gibbsNormal2 in the library nclbayes implements this Gibbs sampling algorithm. Consider again the analysis of Cavendish's measurements on the earth's density in Example 2.2. These data gave $n = 23$, $\bar{x} = 5.4848$, $s = 0.1882$ and this information was combined with a $NGa(b = 5.41, c = 0.25, g = 2.5, h = 0.1)$ prior distribution to give a $NGa(B = 5.4840, C = 23.25, G = 14, H = 0.5080)$ posterior distribution. Here we analyse the data using a Gibbs sampler and verify that it gives the same results. For example, we know that the marginal posterior distributions are

$$\mu|\boldsymbol{x} \sim t_{2G=28}(B = 5.4840, H/(GC) = 0.001561)$$

and

$$\tau|\boldsymbol{x} \sim Ga(G = 14, H = 0.5080),$$

and so we can compare the Gibbs output with these distributions. The following code runs this Gibbs sampler for this problem.

```
library(nclbayes)
posterior=gibbsNormal2(N=1010,initial=c(5.41,25),
    priorparam=c(5.41,0.25,2.5,0.1),n=23,xbar=5.4848,s=0.1882)
posterior2=mcmcProcess(input=posterior,burnin=10,thin=1)

mcmcAnalysis(posterior,rows=2,show=F)
mcmcAnalysis(posterior2,rows=2,show=F)
```

Figure 4.9 shows the summary of the Gibbs sampler output after deleting the first 1000 iterations as burn-in. The traceplots look like the sampler has converged: they indicate a well mixing chain with similar means and variances in different sections of the chain. Also the autocorrelation plots show that no thinning is needed.

These realisations from the posterior distribution can be summarised using R function mcmcAnalysis as

```
N = 1000 iterations
       mu              tau
 Min.   :5.342   Min.   :10.44
 1st Qu.:5.460   1st Qu.:22.81
 Median :5.484   Median :27.70
 Mean   :5.485   Mean   :28.09
 3rd Qu.:5.510   3rd Qu.:32.80
 Max.   :5.619   Max.   :53.66
Standard deviations:
        mu        tau
0.03936649 7.39551702
```

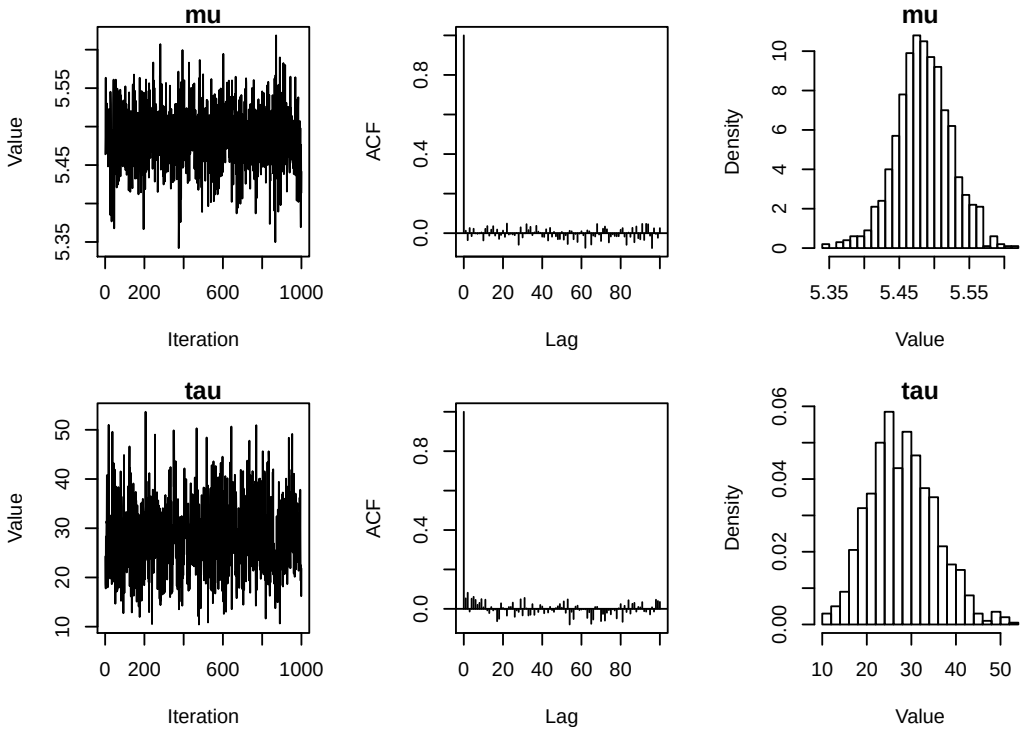

Figure 4.9: Trace plots, autocorrelation plots and histograms of the Gibbs sampler output

These posterior summaries are pretty accurate as we know the correct summaries are

$$E(\mu|\boldsymbol{x}) = B = 5.4840, \qquad SD(\mu|\boldsymbol{x}) = \sqrt{\frac{H}{(G-1)C}} = 0.04100$$

$$E(\tau|\boldsymbol{x}) = \frac{G}{H} = 27.559, \qquad SD(\tau|\boldsymbol{x}) = \frac{\sqrt{G}}{H} = 7.3655.$$

We could obtain even more accurate estimates for these posterior summaries by running the sampler for more iterations. Figure 4.10 shows that the histograms of the Gibbs sampler output are also very close to the (known) marginal posterior densities. These results confirm that our Gibbs sampler is working correctly and does indeed produce realisations from the correct posterior distribution.

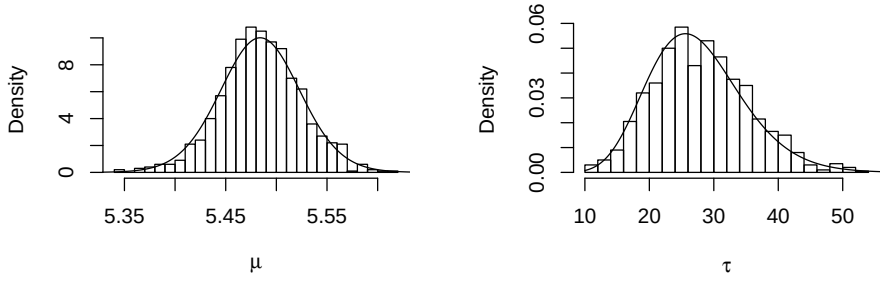

Figure 4.10: Histograms of the Gibbs sampler output and the (known) marginal densities

Example 4.4

Suppose we have a random sample of size n from a gamma $Ga(\alpha, \lambda)$ distribution in which both the index $\alpha > 0$ and scale parameter $\lambda > 0$ are unknown, that is, $X_i|\alpha, \lambda \sim Ga(\alpha, \lambda)$, $i = 1, 2, \ldots, n$ (independent). We shall assume independent prior distributions for these parameters, with $\alpha \sim Ga(a, b)$ and $\lambda \sim Ga(c, d)$ for known values a, b, c and d. Determine the posterior density for $(\alpha, \lambda)^T$ and hence the posterior conditional densities for $\alpha|\lambda$ and $\lambda|\alpha$.

Solution

The likelihood function is

$$\begin{aligned} f(\boldsymbol{x}|\alpha,\lambda) &= \prod_{i=1}^{n} \frac{\lambda^{\alpha} x_i^{\alpha-1} e^{-\lambda x_i}}{\Gamma(\alpha)} \\ &= \frac{\lambda^{n\alpha} \left(\prod x_i\right)^{\alpha-1} e^{-n\bar{x}\lambda}}{\Gamma(\alpha)^n} \\ &= \frac{\lambda^{n\alpha} \bar{x}_g^{n(\alpha-1)} e^{-n\bar{x}\lambda}}{\Gamma(\alpha)^n} \end{aligned}$$

where $\bar{x}_g = \sqrt[n]{\prod x_i}$ is the geometric mean of the data. Using Bayes Theorem, the posterior density is

$$\pi(\alpha,\lambda|\boldsymbol{x}) \propto \pi(\alpha,\lambda)\, f(\boldsymbol{x}|\alpha,\lambda)$$

and so, for $\alpha > 0$, $\lambda > 0$

$$\begin{aligned} \pi(\alpha,\lambda|\boldsymbol{x}) &\propto \alpha^{a-1} e^{-b\alpha} \times \lambda^{c-1} e^{-d\lambda} \times \frac{\lambda^{n\alpha} \bar{x}_g^{n(\alpha-1)} e^{-n\bar{x}\lambda}}{\Gamma(\alpha)^n} \\ &\propto \frac{\alpha^{a-1} \bar{x}_g^{n\alpha} \lambda^{c+n\alpha-1}}{\Gamma(\alpha)^n} \exp\left\{-b\alpha - (d+n\bar{x})\lambda\right\}. \end{aligned}$$

This is not the density of a standard distribution. The conditional posterior density for α is, for $\alpha > 0$

$$\begin{aligned} \pi(\alpha|\lambda,\boldsymbol{x}) &= \frac{\pi(\alpha,\lambda|\boldsymbol{x})}{\pi(\lambda|\boldsymbol{x})} \\ &\propto \pi(\alpha,\lambda|\boldsymbol{x}) \\ &\propto \frac{\alpha^{a-1} \bar{x}_g^{n\alpha} \lambda^{c+n\alpha-1}}{\Gamma(\alpha)^n} \exp\left\{-b\alpha - (d+n\bar{x})\lambda\right\} \\ &\propto \frac{\alpha^{a-1} \bar{x}_g^{n\alpha} \lambda^{n\alpha} e^{-b\alpha}}{\Gamma(\alpha)^n} \\ &\propto \frac{\alpha^{a-1} e^{-(b-n\log\bar{x}_g - n\log\lambda)\alpha}}{\Gamma(\alpha)^n}. \end{aligned}$$

This looks like it might be a gamma density but (i) we would require $b - n\log\bar{x}_g - n\log\lambda > 0$ which can't be guaranteed (ii) the divisor term is wrong: it would have to be $\Gamma(a)$ for this to be a gamma density.

The conditional posterior density for λ is, for $\lambda > 0$

$$\begin{aligned} \pi(\lambda|\alpha,\boldsymbol{x}) &= \frac{\pi(\alpha,\lambda|\boldsymbol{x})}{\pi(\alpha|\boldsymbol{x})} \\ &\propto \pi(\alpha,\lambda|\boldsymbol{x}) \\ &\propto \frac{\alpha^{a-1} \bar{x}_g^{n\alpha} \lambda^{c+n\alpha-1}}{\Gamma(\alpha)^n} \exp\left\{-b\alpha - (d+n\bar{x})\lambda\right\} \\ &\propto \lambda^{c+n\alpha-1} e^{-(d+n\bar{x})\lambda}, \end{aligned}$$

that is, $\lambda|\alpha,\boldsymbol{x} \sim Ga(c+n\alpha,\, d+n\bar{x})$.

The full conditional distribution (FCD) for λ is a standard distribution and so it is straightforward to simulate from this distribution. However, the FCD for α is not a standard distribution and not easy to simulate from. Therefore we cannot use a Gibbs sampler to simulate from the posterior $\pi(\alpha,\lambda|\boldsymbol{x})$. Fortunately, there are other methods available to help us in these situations.

4.5 Metropolis-Hastings sampling

The Gibbs sampler is a very powerful tool but is only useful if the full conditional distributions (FCDs) are standard distributions (which are easy to simulate from). Fortunately there is a class of methods which can be used when the FCDs are non-standard. These methods are known as Metropolis-Hastings schemes.

Suppose we want to simulate realisations from the posterior density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ and all of the FCDs are non-standard. Suppose further that we have a *proposal distribution* with density $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$, which is easy to simulate from. This distribution gives us a way of proposing new values $\boldsymbol{\theta}^*$ from the current value $\boldsymbol{\theta}$.

Consider the following algorithm:

1. Initialise the iteration counter to $j = 1$, and initialise the chain to $\boldsymbol{\theta}^{(0)}$.
2. Generate a *proposed* value $\boldsymbol{\theta}^*$ using the proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(j-1)})$.
3. Evaluate the *acceptance probability* $\alpha(\boldsymbol{\theta}^{(j-1)}, \boldsymbol{\theta}^*)$ of the proposed move, where
$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{x})\, q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\pi(\boldsymbol{\theta}|\boldsymbol{x})\, q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}\right\}.$$
4. Set $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^*$ with probability $\alpha(\boldsymbol{\theta}^{(j-1)}, \boldsymbol{\theta}^*)$, and set $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^{(j-1)}$ otherwise.
5. Change the counter from j to $j + 1$ and return to step 2.

In other words, at each stage, a new value is generated from the proposal distribution. This is either accepted, in which case the chain moves, or rejected, in which case the chain stays where it is. Whether or not the move is accepted or rejected depends on an acceptance probability which itself depends on the relationship between the density of interest and the proposal distribution. Note that the posterior density $\pi(\cdot|\boldsymbol{x})$ only enters into the acceptance probability as a ratio, and so the method can be used when it is known up to a scaling constant, that is,

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*)\, f(\boldsymbol{x}|\boldsymbol{\theta}^*)\, q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\pi(\boldsymbol{\theta})\, f(\boldsymbol{x}|\boldsymbol{\theta})\, q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}\right\},$$

since

$$\pi(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{\pi(\boldsymbol{\theta})\, f(\boldsymbol{x}|\boldsymbol{\theta})}{f(\boldsymbol{x})}.$$

It can be shown that the above algorithm defines a Markov chain with $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ as its stationary distribution.

Notice that the above description holds for all possible proposal distributions (subject to them generating realisations from the full parameter space). But are some choices better than others? We now discuss some commonly used proposal distributions.

4.5.1 Symmetric chains (Metropolis method)

The simplest case is the Metropolis sampler and uses a symmetric proposal distribution, that is, one with $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$, $\forall\ \boldsymbol{\theta}, \boldsymbol{\theta}^*$. In this case the acceptance probability simplifies to

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{x})}{\pi(\boldsymbol{\theta}|\boldsymbol{x})}\right\},$$

and hence does not involve the proposal density at all. Consequently proposed moves which will take the chain to a region of higher posterior density are always accepted, while moves which take the chain to a region of lower posterior density are accepted with probability proportional to the ratio of the two densities — moves which will take the chain to a region of very low density will be accepted with very low probability. Note that any proposal of the form $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = f(|\boldsymbol{\theta}^* - \boldsymbol{\theta}|)$ is symmetric, where $f(\cdot)$ is some zero mean density function, as $|\boldsymbol{\theta}^* - \boldsymbol{\theta}| = |\boldsymbol{\theta} - \boldsymbol{\theta}^*|$. In this case, the proposal value is a symmetric displacement from the current value. This motivates the following.

Random walk proposals

Consider the random walk proposal in which the proposed value $\boldsymbol{\theta}^*$ depends on the current value $\boldsymbol{\theta}$ via

$$\boldsymbol{\theta}^* = \boldsymbol{\theta} + \boldsymbol{w},$$

where $\boldsymbol{w}$ is a random $p \times 1$ vector from the zero mean density $f(\cdot)$ which is *symmetric* about its mean, and is independent of the state of the chain. We can generate our *proposal* value by first simulating an *innovation* $\boldsymbol{w}$, and then set the *proposal* value to $\boldsymbol{\theta}^* = \boldsymbol{\theta} + \boldsymbol{w}$. Clearly $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = f(\boldsymbol{\theta}^* - \boldsymbol{\theta}) = f(\boldsymbol{w})$. Also $\boldsymbol{\theta} = \boldsymbol{\theta}^* - \boldsymbol{w}$ and so $q(\boldsymbol{\theta}|\boldsymbol{\theta}^*) = f(\boldsymbol{\theta} - \boldsymbol{\theta}^*) = f(-\boldsymbol{w})$. However, as $f(\cdot)$ is a zero mean symmetric density, we have that $f(\boldsymbol{w}) = f(-\boldsymbol{w})$ and so $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$.

But, what distribution should we use for $f(\cdot)$? A distribution which is simple and easy to simulate from would be good, with obvious choices of the uniform or normal distributions, though the normal distribution is generally better, but is a bit more expensive to simulate. What variance should we use for the distribution we choose? This choice will affect the acceptance probability, and hence the overall proportion of accepted moves. If the variance of the innovation is too low, then most proposed values will be accepted, but the chain will move very slowly around the space — the chain is said to be too "cold". On the other hand, if the variance of the innovation is too large, very few proposed values will be accepted, but when they are, they will often correspond to quite large moves — the chain is said to be too "hot". Theoretically it has been shown that the optimal acceptance rate is around 0.234 — this is an asymptotic result (for large samples of data) — but experience suggests that an acceptance rate of around 20–30% is okay. Thus, the variance of the innovation distribution should be "tuned" to get an acceptance rate of around this level.

Normal random walk proposals

A symmetric normal random walk proposal takes the form $\theta^*|\theta \sim N(\theta, k^2)$ for some innovation size $k > 0$. This is a symmetric proposal because $\theta^* = \theta + w$, where $w \sim N(0, k^2)$ has a density which is symmetric about zero. Also the proposal ratio is

$$\frac{q(\theta|\theta^*)}{q(\theta^*|\theta)} = \frac{\frac{1}{\sqrt{2\pi k^2}} \exp\left\{-\frac{(\theta-\theta^*)^2}{2k^2}\right\}}{\frac{1}{\sqrt{2\pi k^2}} \exp\left\{-\frac{(\theta^*-\theta)^2}{2k^2}\right\}} = 1.$$

Uniform random walk proposals

A symmetric uniform random walk proposal takes the form $\theta^*|\theta \sim U(\theta - a, \theta + a)$ for some innovation size $a > 0$. This is a symmetric proposal because $\theta^* = \theta + w$, where $w \sim U(-a, a)$ has a density which is symmetric about zero. Also

$$\frac{q(\theta|\theta^*)}{q(\theta^*|\theta)} = \frac{1/(2a)}{1/(2a)} = 1.$$

Example 4.5

Suppose the posterior distribution is a standard normal distribution, with density $\phi(\cdot)$. Construct a Metropolis–Hastings algorithm which samples this posterior distribution by using a uniform random walk proposal. Examine how the acceptance rate for this algorithm depends on the width of the uniform distribution.

Solution

We have $\pi(\theta|\boldsymbol{x}) = \phi(\theta)$ and will use a symmetric uniform random walk proposal $\theta^*|\theta \sim U(\theta - a, \theta + a)$ for some choice of $a > 0$. The acceptance probability is

$$\begin{aligned}\alpha(\theta, \theta^*) &= \min\left\{1, \frac{\pi(\theta^*|\boldsymbol{x})}{\pi(\theta|\boldsymbol{x})}\right\} = \min\left\{1, \frac{\phi(\theta^*)}{\phi(\theta)}\right\} \\ &= \min\left\{1, e^{(\theta^2-\theta^{*2})/2}\right\}.\end{aligned}$$

Therefore the algorithm is

1. Initialise the iteration counter to $j = 1$, and initialise the chain to $\theta^{(0)} = 0$ say.
2. Generate a proposed value $\theta^* \sim U(\theta^{(j-1)} - a, \theta^{(j-1)} + a)$.

3. Evaluate the acceptance probability $\alpha(\theta^{(j-1)}, \theta^*) = \min\left\{1, e^{(\theta^{(j-1)^2} - \theta^{*2})/2}\right\}$
4. Set $\theta^{(j)} = \theta^*$ with probability $\alpha(\theta^{(j-1)}, \theta^*)$, and set $\theta^{(j)} = \theta^{(j-1)}$ otherwise.
5. Change the counter from j to $j+1$ and return to step 2.

Of course, in practice we would never simulate from a standard normal distribution using this M–H algorithm as there are much more efficient methods (like the one used in `rnorm`). The purpose here was to illustrate the general method using a very simple choice of posterior distribution.

The R function `metropolis` in the library `nclbayes` implements this Metropolis algorithm. The following code runs this algorithm, taking the population mean as its initial value and taking $a = 6$:

```
posterior=metropolis(N=10000,initial=0,a=6)
mcmcAnalysis(posterior,rows=1,show=F)
```

Figure 4.11 shows the output from runs of the algorithm for 10k iterations using different values of a. The top row uses $a = 0.6$ and this chain is too "cold": the innovations are too small and are generally accepted. The acceptance rate for this chain was 0.881. Notice that the autocorrelations are too high and this chain would have to be thinned. Increasing the size of the innovations to $a = 6$ gives the output on the middle row. The autocorrelations are much lower and the acceptance rate was 0.260 (nearer the asymptotic 0.234 M–H acceptance rate). Increasing the size still further to $a = 60$ gives the output on the bottom row. This chain is too "hot" with few proposed values being accepted (acceptance rate 0.027), but when they are, it results in a fairly large move to the chain. This gives fairly high autocorrelations and this chain would have to be thinned.

Normal random walk proposals

Suppose we decide to use a normal random walk with $f(\cdot) = N_p(0, \Sigma)$ and so the proposal distribution is

$$\boldsymbol{\theta}^* | \boldsymbol{\theta} \sim N_p(\boldsymbol{\theta}, \Sigma).$$

Tuning this random walk requires us to choose a value for the covariance matrix Σ. If the posterior distribution is approximately normally distributed (as it is with large data samples) then researchers have shown that the optimal choice is

$$\Sigma = \frac{2.38^2}{p} Var(\boldsymbol{\theta}|\boldsymbol{x}).$$

In practice, of course, we don't know the posterior variance $Var(\boldsymbol{\theta}|\boldsymbol{x})$. However, we could first run the MCMC algorithm substituting in the (generally much larger) prior variance $Var(\boldsymbol{\theta})$. If this chain doesn't converge quickly then we can use its output to get a better idea of $Var(\boldsymbol{\theta}|\boldsymbol{x})$ and run the MCMC code again – this will have more appropriate values for the parameter variances and correlations.

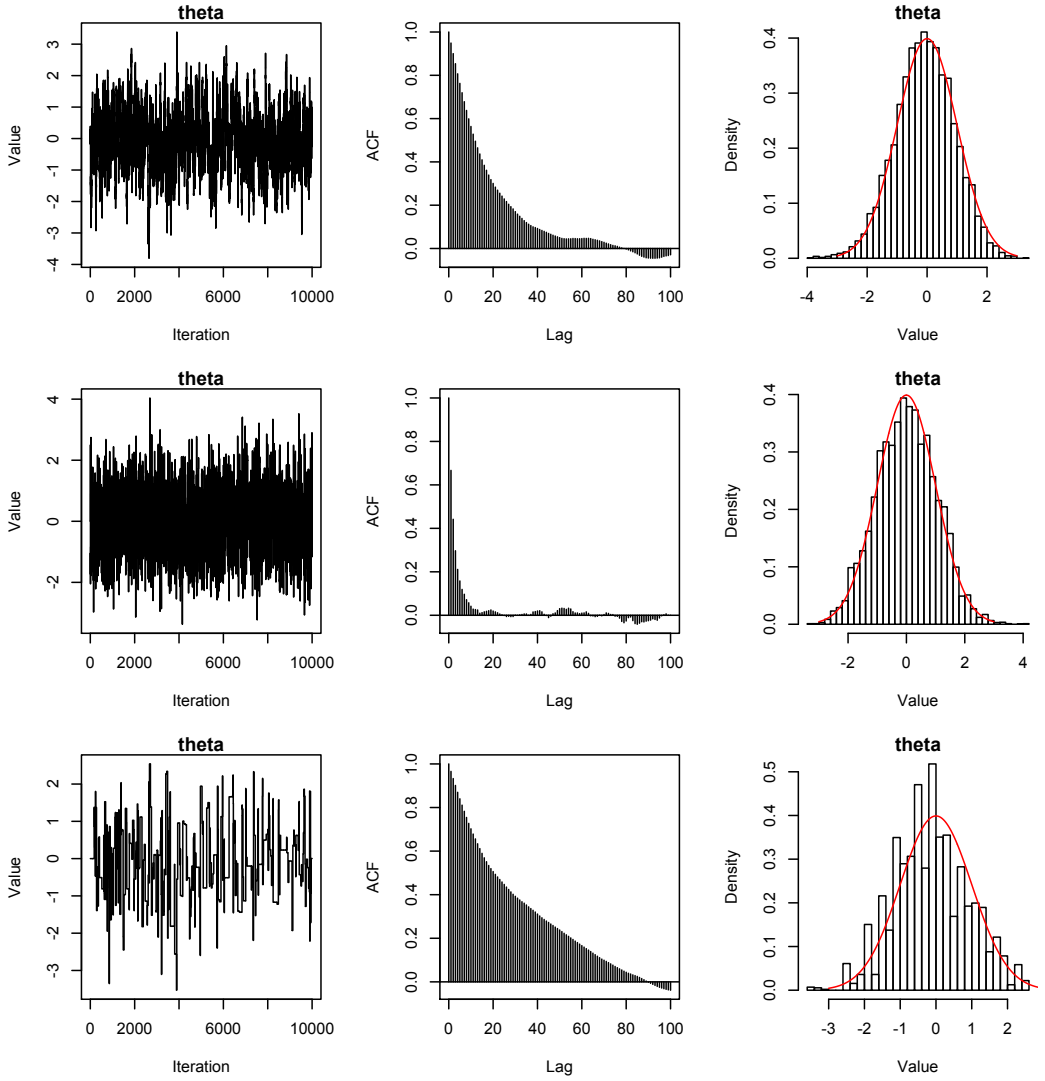

Figure 4.11: Trace plots, autocorrelation plots and histograms of the output from a Metropolis–Hastings sampler using a $U(-a, a)$ random walk proposal. Top row: $a = 0.6$; middle row: $a = 6$; bottom row: $a = 60$

It has been shown from experience that it is not vital to get an extremely accurate value for Σ. Often just getting the correct order of magnitude for its elements will be sufficient, that is, using say 0.1 rather than 0.01 or 1.

4.5.2 Independence chains

In this case, the proposal is formed independently of the position of the chain, and so $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = f(\boldsymbol{\theta}^*)$ for some density $f(\cdot)$. Here the acceptance probability is

$$\begin{aligned}\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) &= \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{x})\, f(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}|\boldsymbol{x})\, f(\boldsymbol{\theta}^*)}\right\} \\ &= \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{x})}{f(\boldsymbol{\theta}^*)} \Big/ \frac{\pi(\boldsymbol{\theta}|\boldsymbol{x})}{f(\boldsymbol{\theta})}\right\},\end{aligned}$$

and we see that the acceptance probability can be increased by making $f(\cdot)$ as similar to $\pi(\cdot|\boldsymbol{x})$ as possible. In this case, the higher the acceptance probability, the better.

Bayes Theorem via independence chains

One possible choice for the proposal density is the prior density. The acceptance probability is then

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{f(\boldsymbol{x}|\boldsymbol{\theta}^*)}{f(\boldsymbol{x}|\boldsymbol{\theta})}\right\},$$

and hence depends only on the likelihood ratio of the proposal and the current value.

4.6 Hybrid methods

We have now seen how we can use the Gibbs sampler to sample from multivariate distributions provided that we can simulate from the full conditional distributions. We have also seen how we can use Metropolis-Hastings methods to sample from awkward FCDs. If we wish, we can combine these in order to form hybrid Markov chains whose stationary distribution is a distribution of interest.

4.6.1 Componentwise transitions

Given a posterior distribution with full conditional distributions that are awkward to sample from directly, we can define a Metropolis-Hastings scheme for each full conditional distribution, and apply them to each component in turn for each iteration. This is like the Gibbs sampler, but each component update is a Metropolis-Hastings update, rather than a direct simulation from the full conditional distribution. Each of these steps will require its own proposal distribution. The algorithm is as follows:

1. Initialise the iteration counter to $j = 1$.
   Initialise the state of the chain to $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})^T$.
2. Let $\boldsymbol{\theta}_{-i}^{(j)} = \left(\theta_1^{(j)}, \ldots, \theta_{i-1}^{(j)}, \theta_{i+1}^{(j-1)}, \ldots, \theta_p^{(j-1)}\right)^T$, $i = 1, 2, \ldots, p$.
   Obtain a new value $\boldsymbol{\theta}^{(j)}$ from $\boldsymbol{\theta}^{(j-1)}$ by successive generation of values

- $\theta_1^{(j)} \sim \pi(\theta_1|\boldsymbol{\theta}_{-1}^{(j)}, \boldsymbol{x})$ using a Metropolis–Hastings step with proposal distribution $q_1(\theta_1|\theta_1^{(j-1)}, \boldsymbol{\theta}_{-1}^{(j)})$
- $\theta_2^{(j)} \sim \pi(\theta_2|\boldsymbol{\theta}_{-2}^{(j)}, \boldsymbol{x})$ using a Metropolis–Hastings step with proposal distribution $q_2(\theta_2|\theta_2^{(j-1)}, \boldsymbol{\theta}_{-2}^{(j)})$

  $\vdots$
- $\theta_p^{(j)} \sim \pi(\theta_p|\boldsymbol{\theta}_{-p}^{(j)}, \boldsymbol{x})$ using a Metropolis–Hastings step with proposal distribution $q_p(\theta_p|\theta_p^{(j-1)}, \boldsymbol{\theta}_{-p}^{(j)})$

3. Change counter j to $j+1$, and return to step 2.

This is in fact the original form of the Metropolis algorithm. Note that the distributions $\pi(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})$ are just the FCDs.

Suppose we decide to use normal random walks for these M–H steps, that is, take $q_i(\theta_i^*|\theta_i, \boldsymbol{\theta}_{-i}^{(j)})$ is a $N(\theta_i, \Sigma_{ij})$ density. What is the appropriate value for Σ_{ij}? As the proposal in step j is targeting the conditional posterior density $\pi(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})$, the optimal choice of Σ_{ij} is

$$\Sigma_{ij} = \frac{2.38^2}{1} Var(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x}) = 2.38^2 Var(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x}).$$

As these (conditional) posterior variances are not known before running the MCMC code, a sensible strategy might be to replace it with the (probably much larger) prior conditional variance $Var(\theta_i|\boldsymbol{\theta}_{-i}^{(j)})$ or even the prior marginal variance $Var(\theta_i)$. Again, recall that these values are to be used as a guide, generally to get the order of magnitude for the innovation variance.

4.6.2 Metropolis within Gibbs

Given a posterior distribution with full conditional distributions, some of which may be simulated from directly, and others which have Metropolis-Hastings updating schemes, the Metropolis within Gibbs algorithm goes through each in turn, and simulates directly from the full conditional, or carries out a Metropolis-Hastings update as necessary. This algorithm is, in fact, just the above algorithm but uses the full conditional distributions as the proposal distributions when they are easy to simulate from. To see this, suppose that we can simulate from the FCD $\pi(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})$ and use this as the proposal distribution, that is, take $\theta_i^* \sim \pi(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})$. Then the acceptance probability for this step is

$$\begin{aligned}
\alpha(\theta_i, \theta_i^*) &= \min\left\{1, \frac{\pi(\theta_i^*|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})}{\pi(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})} \frac{q(\theta_i|\theta_i^*, \boldsymbol{\theta}_{-i}^{(j)})}{q(\theta_i^*|\theta_i, \boldsymbol{\theta}_{-i}^{(j)})}\right\} \\
&= \min\left\{1, \frac{\pi(\theta_i^*|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})}{\pi(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})} \frac{\pi(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})}{\pi(\theta_i^*|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})}\right\} \\
&= \min(1, 1) \\
&= 1,
\end{aligned}$$

that is, we always accept the proposal from the FCD.

Example 4.6

Construct an MCMC scheme for the problem in Example 4.4 where we had a random sample of size n from a gamma $Ga(\alpha, \lambda)$ distribution and independent gamma $Ga(a, b)$ and $Ga(c, d)$ prior distributions for α and λ respectively. Recall that the FCDs were

$$\pi(\alpha|\lambda, \boldsymbol{x}) \propto \frac{\alpha^{a-1} e^{(-b+n\log\bar{x}_g+n\log\lambda)\alpha}}{\Gamma(\alpha)^n}, \quad \alpha > 0$$

and

$$\pi(\lambda|\alpha, \boldsymbol{x}) \propto \lambda^{c+n\alpha-1}\, e^{-(d+n\bar{x})\lambda}, \quad \lambda > 0.$$

Solution

We have that $\lambda|\alpha, \boldsymbol{x} \sim Ga(c + n\alpha,\ d + n\bar{x})$ but the distribution of $\alpha|\lambda, \boldsymbol{x}$ is non-standard. Therefore we will use a Metropolis within Gibbs algorithm which uses a Gibbs step for λ and a M–H step for α. In the M–H step, we will use a normal random walk proposal distribution, with $\alpha^*|\alpha \sim N(\alpha, \Sigma_\alpha)$, in which the proposal α^* has acceptance probability $\min(1, A)$, where

$$\begin{aligned} A &= \frac{\pi(\alpha^*|\lambda, \boldsymbol{x})}{\pi(\alpha|\lambda, \boldsymbol{x})} \\ &= \frac{\alpha^{*a-1} e^{(-b+n\log\bar{x}_g+n\log\lambda)\alpha^*}}{\Gamma(\alpha^*)^n} \times \frac{\Gamma(\alpha)^n}{\alpha^{a-1} e^{(-b+n\log\bar{x}_g+n\log\lambda)\alpha}}, \quad \alpha^* > 0 \\ &= \left(\frac{\alpha^*}{\alpha}\right)^{a-1} \left\{\frac{\Gamma(\alpha)}{\Gamma(\alpha^*)}\right\}^n e^{(-b+n\log\bar{x}_g+n\log\lambda)(\alpha^*-\alpha)}, \quad \alpha^* > 0, \end{aligned}$$

and zero otherwise.

We need to find a value for α or λ to initialise the MCMC algorithm. We could use a value simulated from the prior or use the prior mean. Alternatively we could use the mle but this is rather complicated to determine. However, the moment estimates are rather more straightforward, and equating first and second population and sample moments gives

$$\text{mean}: \quad \frac{\tilde{\alpha}}{\tilde{\lambda}} = \bar{x} \qquad\qquad \text{variance}: \quad \frac{\tilde{\alpha}}{\tilde{\lambda}^2} = s^2$$

and so

$$\tilde{\lambda} = \frac{\bar{x}}{s^2} \qquad\qquad \tilde{\alpha} = \frac{\bar{x}^2}{s^2}.$$

Therefore the MCMC algorithm is

1. Initialise the iteration counter to $j = 1$, and initialise the chain to $\alpha^{(0)} = (\bar{x}/s)^2$.
2. Obtain a new value
$$\lambda^{(j)} \sim Ga(c + n\alpha^{(j-1)},\ d + n\bar{x})$$
3. Generate a proposed value $\alpha^* \sim N(\alpha^{(j-1)}, \Sigma_\alpha)$
4. Evaluate the acceptance probability $\min(1, A)$ at $\alpha^* = \alpha^*$, $\alpha = \alpha^{(j-1)}$ and $\lambda = \lambda^{(j)}$
5. Set $\alpha^{(j)} = \alpha^*$ with probability $\min(1, A)$, and set $\alpha^{(j)} = \alpha^{(j-1)}$ otherwise.
6. Change the counter from j to $j + 1$ and return to step 2.

The R function `mwgGamma` in the library `nclbayes` implements this Metropolis within Gibbs algorithm. The following code produces posterior output from an analysis of a dataset with $n = 50$, $\bar{x} = 0.62$, $\bar{x}_g = 0.46$ and $s = 0.4$, with prior beliefs represented by $a = 2$, $b = 1$, $c = 3$ and $d = 1$, and uses a normal random walk proposal with variance $\Sigma_\alpha = 0.9^2$ as this gives a reasonable acceptance probability of 0.237. The initial value is taken as the moment estimate $\tilde{\alpha} = (\bar{x}/s)^2$.

```
posterior=mwgGamma(N=20000,initial=(0.46/0.4)^2,innov=0.9,
    priorparam=c(2,1,3,1),n=50,xbar=0.62,xgbar=0.46,show=TRUE)
mcmcAnalysis(posterior,rows=2,show=F)
posterior2=mcmcProcess(input=posterior,burnin=10,thin=20)
mcmcAnalysis(posterior2,rows=2)
```

The upper plots in Figure 4.12 show all the output of this MCMC scheme and the lower plots show the output after deleting the first 10 iterations as burn–in and then thinning by only taking every 20th iterate to reduce the autocorrelations.

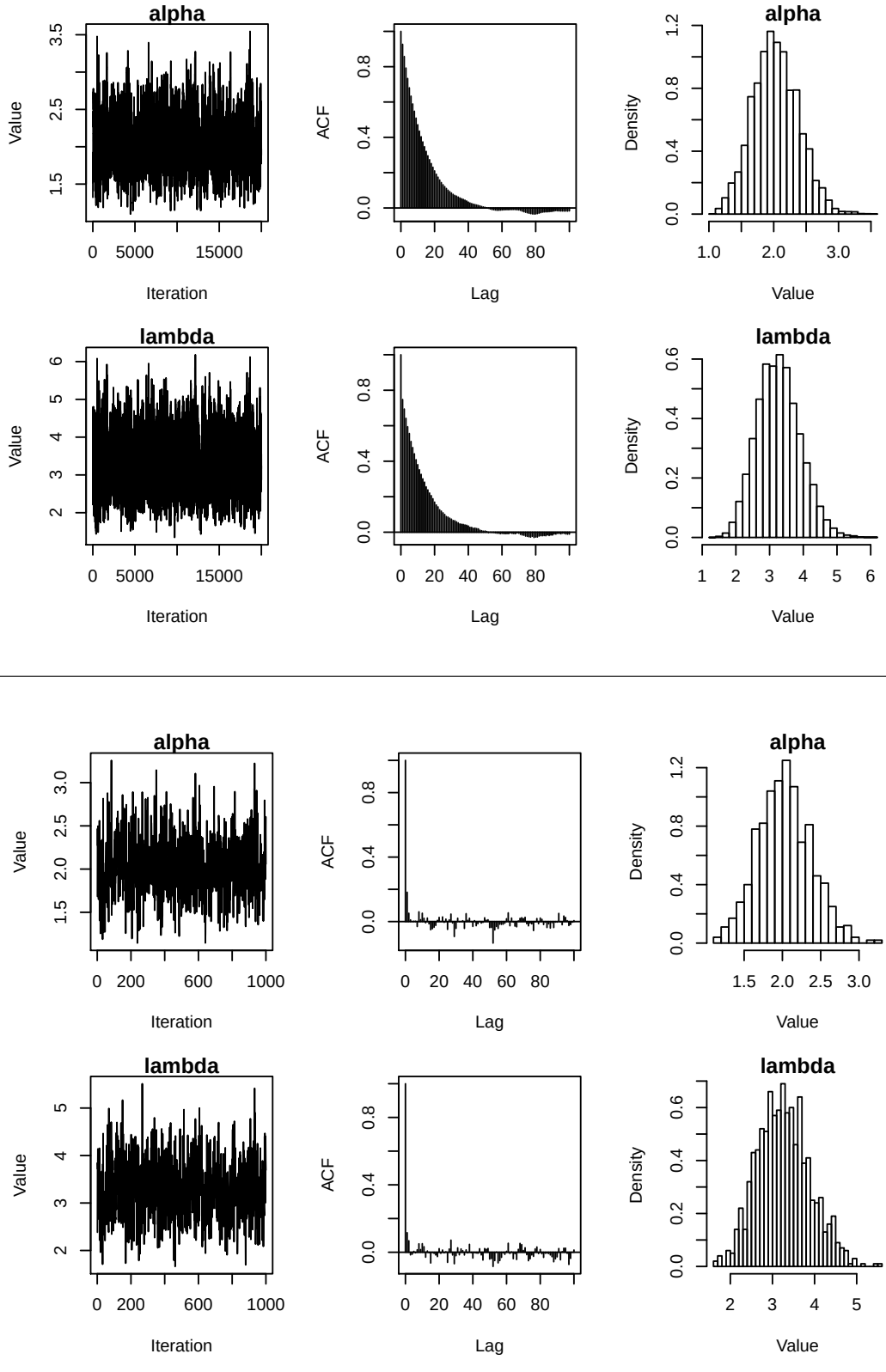

Figure 4.12: Trace plots, autocorrelation plots and histograms of the Metropolis with Gibbs output. Upper plots: all realisations. Lower plots: with burn-in = 10, thin = 20.

Comments

1. If you're unsure whether the proposal distribution is symmetric then it's quite straightforward to examine the proposal ratio. In this last example, we have proposal $\alpha^*|\alpha \sim N(\alpha, \Sigma_\alpha)$ and so

$$\frac{q(\alpha|\alpha^*)}{q(\alpha^*|\alpha)} = \frac{\frac{1}{\sqrt{2\pi\Sigma_\alpha}} \exp\left\{-\frac{(\alpha-\alpha^*)^2}{2\Sigma_\alpha}\right\}}{\frac{1}{\sqrt{2\pi\Sigma_\alpha}} \exp\left\{-\frac{(\alpha^*-\alpha)^2}{2\Sigma_\alpha}\right\}} = 1.$$

2. A normal random walk proposal α^* is not accepted if it is negative as, in this case, $A = 0$. This can be wasteful. An alternative is to use a proposal distribution which only generates positive proposal values, such as $\alpha^*|\alpha \sim LN(\log\alpha, \Sigma_\alpha)$. Using this skewed proposal distribution, we have

$$\frac{q(\alpha|\alpha^*)}{q(\alpha^*|\alpha)} = \frac{\frac{1}{\alpha\sqrt{2\pi\Sigma_\alpha}} \exp\left\{-\frac{(\log\alpha-\log\alpha^*)^2}{2\Sigma_\alpha}\right\}}{\frac{1}{\alpha^*\sqrt{2\pi\Sigma_\alpha}} \exp\left\{-\frac{(\log\alpha^*-\log\alpha)^2}{2\Sigma_\alpha}\right\}} = \frac{\alpha^*}{\alpha}.$$

   Therefore the acceptance probability for a proposed value α^* is $\min(1, B)$ where

$$\begin{aligned} B &= \frac{\pi(\alpha^*|\lambda, \boldsymbol{x})}{\pi(\alpha|\lambda, \boldsymbol{x})} \times \frac{q(\alpha|\alpha^*)}{q(\alpha^*|\alpha)} \\ &= \frac{\alpha^*\,\pi(\alpha^*|\lambda, \boldsymbol{x})}{\alpha\,\pi(\alpha|\lambda, \boldsymbol{x})}. \end{aligned}$$

   The acceptance probability is still quite straightforward to calculate, and with this proposal distribution we never reject proposal values that are inconsistent with the parameter space (here $\alpha > 0$). Incidentally, $\log X \sim N(\mu, \sigma^2)$ if $X \sim LN(\mu, \sigma^2)$ and so using a log-normal proposal is the same as using a normal random walk on the log scale, that is, a normal random walk for $\log\alpha$. Also log-normal proposals are easy to simulate because if $Y \sim N(\mu, \sigma^2)$ then $e^Y \sim LN(\mu, \sigma^2)$.

3. Dealing with a constraint such as $\alpha > 0$ in optimisation methods or here in MCMC methods, can be solved by re-parameterising the model. Here, for example, we could work with $A = \log\alpha$ and obtain realisations from the posterior distribution for A. Once we have these realisations we can easily obtain realisations from the posterior distribution for $\alpha = e^A$. Working in A rather than α means we have to simulate realisations from the conditional posterior

$$\pi_A(A|\lambda, \boldsymbol{x}) = \pi_\alpha(e^A|\lambda, \boldsymbol{x}) \times \left|\frac{d}{da}\, e^A\right| = e^A\,\pi_\alpha(e^A|\lambda, \boldsymbol{x})$$

   using (2.1). If we also use a normal random walk for proposing new values for A (as

it's unconstrained) then a proposal A^* is accepted with probability $\min(1, C)$ where

$$\begin{aligned} C &= \frac{\pi(A^*|\lambda, \boldsymbol{x})}{\pi(A|\lambda, \boldsymbol{x})} \times \frac{q(A|A^*)}{q(A^*|A)} \\ &= \frac{\pi(A^*|\lambda, \boldsymbol{x})}{\pi(A|\lambda, \boldsymbol{x})} \qquad \text{since the proposal distribution is symmetric about zero} \\ &= \frac{e^{A^*}\,\pi_\alpha(e^{A^*}|\lambda, \boldsymbol{x})}{e^{A}\,\pi_\alpha(e^{A}|\lambda, \boldsymbol{x})} \\ &= \frac{\alpha^*\,\pi_\alpha(\alpha^*|\lambda, \boldsymbol{x})}{\alpha\,\pi_\alpha(\alpha|\lambda, \boldsymbol{x})}. \end{aligned}$$

Notice that the acceptance probabilities B and C are the same, that is, there is no (algorithmic) difference between using a log-normal random walk for a positive parameter or working on the log-scale and using a symmetric normal random walk.

4.7 Summary

(i) Bayesian inference can be complicated when not using a conjugate prior distribution.

(ii) One solution is to use Markov chain Monte Carlo (MCMC) methods.

(iii) These work by producing realisations from the posterior distribution by constructing a Markov chain which has the posterior distribution as its stationary distribution.

(iv) The MCMC methods we have studied are the Gibbs sampler, Metropolis within Gibbs algorithm and the Metropolis–Hastings algorithm.

(v) When obtaining output from these algorithms, we need to assess whether there needs to be a burn-in and whether the output needs to be thinned (by looking at traceplots and autocorrelation plots) using `mcmcAnalysis` and `mcmcProcess`.

(vi) The (converged and thinned) MCMC output are realisations from the posterior distribution. It can be used to

- obtain the posterior distribution for any (joint) functions of the parameters (such as $\sigma = 1/\sqrt{\tau}$ or $(\theta_1 = \mu - \tau, \theta_2 = e^{\mu+\tau/2})^T$);
- look at bivariate posterior distributions via scatter plots;
- look at univariate marginal posterior distributions via histograms or boxplots;
- obtain numerical summaries such as the mean, standard deviation and confidence intervals for single variables and correlations between variables.

(vii) Equi-tailed posterior confidence intervals can be determined from the MCMC output using `mcmcCi`.

4.8 Learning objectives

By the end of this chapter, you should be able to:

- explain why not using a conjugate prior generally causes problems in determining the posterior distribution
- describe the Gibbs sampler, explain why it is a Markov chain and give an outline as to why its stationary distribution is the posterior distribution
- describe the issues of processing MCMC output (burn-in, autocorrelation, thinning etc.) and interpret numerical/graphical output
- derive the full conditional densities for any posterior distribution and name these distributions if they are "standard" distributions given in the notes or on the exam paper
- describe a Metropolis-Hastings algorithm in general terms and when using either symmetric or non-symmetric random walk proposals or independence proposals
- describe the hybrid methods *componentwise transitions* and *Metropolis within Gibbs*
- provide a detailed description of **any** of the MCMC algorithms as they apply to generating realisations from **any** posterior distribution