Chapter 3

Bayesian Computation

Notation and Setup:

We use the generic notation $\mathbf{x}$ to denote the parameters. We will use the notation $\pi(\mathbf{x})$ to denote the posterior distribution. We are suppressing the data part. This $\pi(\mathbf{x})$ is sometimes called the **target** distribution.

- $\mathbf{x}$: Parameters, earlier notation was $\boldsymbol{\theta}$.
- $f(\mathbf{x})$: Non-normalized posterior density. $f(\mathbf{x})$ = Likelihood $\times$ Prior.
- $\pi(\mathbf{x}) = \frac{f(\mathbf{x})}{\int f(\mathbf{x})d\mathbf{x}}$: Normalized posterior density.
- $b(\mathbf{x})$: Any generic function of interest.

Problem: Evaluate features of $b(\mathbf{x})$, e.g.,

$$E_\pi(b) = \int b(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} = \frac{\int b(\mathbf{x})f(\mathbf{x})d\mathbf{x}}{\int f(\mathbf{x})d\mathbf{x}}. \tag{3.1}$$

Evaluating the above under a multidimensional non-normalized situation is quite difficult. That is why we want to develop machinery. Let us start with a very simple example.

♡ **Example** 3.11. Suppose that $X_1, \ldots, X_n \sim N(\theta, 1)$ and the prior is $\pi(\theta) = \frac{1}{\pi}\frac{1}{1+\theta^2}$. Here the posterior is

$$\begin{aligned} \pi(\theta|\mathbf{x}) &\propto \tfrac{1}{\pi}\tfrac{1}{1+\theta^2}\textstyle\prod_{i=1}^n \tfrac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(x_i-\theta)^2} \\ &\propto \tfrac{1}{1+\theta^2}e^{-\frac{1}{2}n(\theta-\bar{x})^2} \end{aligned}$$

Suppose that $a = \bar{x}$ then

$$\pi(\theta|\mathbf{x}) \propto \frac{1}{1+\theta^2} e^{-\frac{1}{2}n(\theta-a)^2} \tag{3.2}$$

Now instead of writing the posterior density with θ in the argument, we use the more convenient notation

$$\pi(x) \propto \frac{1}{1+x^2} e^{-\frac{1}{2}n(x-a)^2} \tag{3.3}$$

Now we recognise the following:

- $f(x) = \frac{1}{1+x^2} e^{-\frac{1}{2}n(x-a)^2}$.
- $\pi(x) = f(x) / \int_{-\infty}^{\infty} f(x)dx$.

□

3.1 Numerical Integration

Suppose the problem is to evaluate

$$I = \int_a^b h(x)dx.$$

(In our context $h(x) = b(x)\pi(x)$.) Then the *trapezoidal rule* is the following. Let $x_1 = a$ and $x_n = b$. We take a grid $a = x_1 < x_2 < \cdots < x_n = b$. Then use

$$\hat{I} = \frac{1}{2} \sum_{i=1}^{n-1} (x_{i+1} - x_i)\{h(x_i) + h(x_{i+1})\},$$

to approximate I. There are other variations using what are called Simpson's rule, Newton-Coates formulae etc. However, as our models become more complex it becomes increasingly difficult to perform the required integrations by this method. If the dimension of the posterior distribution is quite low e.g., 1, 2 or 3 this method may work well.

3.2 Laplace Approximation

Here the posterior is first approximated by a normal distribution. Then the integral (3.1) can be evaluated sometimes analytically by replacing $\pi(\mathbf{x})$ by its normal approximation. Let $L_n(\mathbf{x}) =$

$\log \pi(\mathbf{x})$. Let $\mathbf{x}_0$ denote the mode of $\pi(x)$, i.e.

$$L_n'(\mathbf{x}_0) = \left.\frac{\partial L_n'(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0} = 0.$$

Assume the existence and positive-definiteness of

$$\Sigma = \left(-L_n''(\mathbf{x}_0)\right)^{-1}$$

where $L_n''(\mathbf{x}_0)$ is the Hessian matrix with elements

$$\left[L_n''(\mathbf{x}_0)\right]_{ij} = \left.\frac{\partial^2 L_n(\mathbf{x})}{\partial x_i \partial x_j}\right|_{\mathbf{x}=\mathbf{x}_0}.$$

Once we are able to find $\mathbf{x}_0$ and Σ we can replace $\pi(\mathbf{x})$ by the normal distribution with mean $\mathbf{x}_0$ and variance-covariance matrix Σ.

Under suitable conditions the normal approximation will be good and features of $\pi(\mathbf{x})$ (e.g. the mean and variance etc.) are approximated using the normal approximation.

3.3 Monte Carlo Integration

Suppose that we can draw samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}$ from $\pi(\mathbf{x})$. Then we can estimate

$$E_\pi[b(\mathbf{x})] \approx \bar{b}_N = \frac{1}{N}\sum_{i=1}^{N} b\left(\mathbf{x}^{(i)}\right). \tag{3.4}$$

We can use the laws of large numbers to show that $\bar{b}_N$ approaches $E_\pi[b(\mathbf{x})]$. This is *Monte Carlo integration.* This is the basic idea in statistics:

features of an unknown distribution can be discovered once a large enough random sample from that distribution has been obtained.

♡ **Example** 3.12. Suppose that $\mathbf{x}$ is at least two dimensional and we are interested in the ratio: $b(\mathbf{x}) = \frac{x_1}{x_2}$, then

$$E_\pi\left[\frac{x_1}{x_2}\right] \approx \frac{1}{N}\sum_{i=1}^{N}\frac{x_1^{(i)}}{x_2^{(i)}}$$

□

♡ **Example** 3.13. Kernel Density Estimates: Select a 'kernel' $k(\cdot)$ such that $\int_{-\infty}^{\infty} k(z)dz = 1$, e.g., the standard normal density. Given the samples $x^{(1)}, x^{(2)}, \ldots, x^{(N)}$, form

$$\hat{f}_N(z) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{h} k\left(\frac{z - x^{(i)}}{h}\right).$$

Evaluate this at a number of grid points (z) on the support and plot it. Thats how kernel density estimates are formed. h is called the window width or the smoothing parameter. There are many different optimal choices for h. Recommended one is $h = 1.06sN^{-1/5}$ where s is the sample standard deviation. A software package like S-Plus can do this with a command or two. □

3.4 Non-iterative Monte Carlo

The methods based on importance sampling, rejection method and weighted bootstrap method fall in this category.

Importance Sampling: Suppose that $g(\mathbf{x})$ is an importance sampling density (ISD) for $\pi(\mathbf{x})$. The density g should have two properties.

1. g resembles π, i.e., g looks like π
2. g is easy to sample from.

Implicitly we also assume that g and π have the same support. See Figure 3.1. Then we can evaluate $E_\pi(b)$ as follows.

$$E_\pi(b) = \frac{\int b(\mathbf{x})f(\mathbf{x})d\mathbf{x}}{\int f(\mathbf{x})d\mathbf{x}} = \frac{\int b(\mathbf{x})\frac{f(\mathbf{x})}{g(\mathbf{x})}g(\mathbf{x})d\mathbf{x}}{\int \frac{f(\mathbf{x})}{g(\mathbf{x})}g(\mathbf{x})d\mathbf{x}} = \frac{\int b(\mathbf{x})w(\mathbf{x})g(\mathbf{x})d\mathbf{x}}{\int w(\mathbf{x})g(\mathbf{x})d\mathbf{x}}.$$

Suppose that we have N samples, $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}$ from the distribution $g(\mathbf{x})$. Clearly,

$$\hat{b} \stackrel{def}{=} \frac{\sum b(\mathbf{x}^{(i)})w_i}{\sum w_i}, \text{ where } w_i = \frac{f(\mathbf{x}^{(i)})}{g(\mathbf{x}^{(i)})},$$

provide a Monte Carlo integration for $E_\pi(b)$. Notice that in effect we have done two Monte Carlo integrations: one for the numerator and the other for the denominator. For $\hat{b}$ to be a good estimate of $E_\pi(b)$ we need w's to be well behaved, i.e., should be roughly equal. This is very hard to do when dimensionality is high. We can refine the estimator of b in two ways.

Rejection Sampling: It is a method for generating i.i.d. samples from $\pi(\mathbf{x})$ once we can generate i.i.d. samples from an importance sampling density $g(\mathbf{x})$.

Suppose that,

$$M = \sup_{\mathbf{x}} \frac{f(\mathbf{x})}{g(\mathbf{x})}$$

is available. For practical purposes the supremum in the above definition can be replaced by maximum.

The **rejection** method has the following steps.

Draw $\mathbf{x} \sim g$ and $u \sim U(0,1)$ independently. Retain $\mathbf{x}$ if $u \leq \frac{f(\mathbf{x})}{M\, g(\mathbf{x})}$ otherwise generate another sample from g and repeat.

The quantity:

$$\alpha(x) = \frac{f(\mathbf{x})}{M\, g(\mathbf{x})}$$

is called the acceptance probability of a candidate x. Note also that in order to implement the method we do *not* need the normalising constant in $\pi(x)$.

Now we show that a sample drawn this way has the exact distribution π.

Proof First see:

$$P\left(\mathbf{x} \text{ retained } | \mathbf{x} \sim g\right) = \frac{f(\mathbf{x})}{Mg(\mathbf{x})}.$$

Hence

$$P\left(\mathbf{x} \text{ retained }\right) = \int \frac{f(\mathbf{x})}{Mg(\mathbf{x})} g(\mathbf{x}) d\mathbf{x} = \frac{1}{M} \int f(\mathbf{x}) d\mathbf{x}.$$

Our aim is to prove that $\mathbf{x}$ retained through the rejection method has the distribution π.

$$\begin{aligned} P\left(\mathbf{x} < c | \mathbf{x} \text{ retained }\right) &= \frac{P(\mathbf{x}<c \text{ and } \mathbf{x} \text{ retained })}{P(\mathbf{x} \text{ retained })} \\ &= \frac{\int_{\mathbf{x}<c} \int_0^{\frac{f(\mathbf{x})}{Mg(\mathbf{x})}} g(\mathbf{x}) du d\mathbf{x}}{\frac{1}{M} \int f(\mathbf{x}) d\mathbf{x}} \\ &= \frac{\int_{\mathbf{x}<c} f(\mathbf{x}) d\mathbf{x}}{\int f(\mathbf{x}) d\mathbf{x}} \\ &= \int_{\mathbf{x}<c} \pi(\mathbf{x})\, d\mathbf{x}, \end{aligned}$$

which shows that the retained $\mathbf{x}$ has the exact distribution π as was claimed. □

Draw N samples and form the estimator $\bar{b}_N$ in (3.4).

♡ **Example** 3.14. Suppose that we are working with proper priors. Let g be the prior distribution. Then

$$M = \sup_{\mathbf{x}} \frac{f(\mathbf{x})}{g(\mathbf{x})} = \sup \frac{\text{Likelihood } \times \text{ Prior}}{\text{Prior}} = \sup \text{ Likelihood }.$$

Hence M will be attained at the maximum likelihood estimate (mle). However, it is not a good idea in high dimension. For simple problems this is ideal. □

♡ **Example** 3.15. Return to the normal-Cauchy example. Take $g(\mathbf{x}) = \frac{1}{\pi}\frac{1}{1+x^2}$, the prior distribution. Find out the M and the acceptance probability. Write computer codes for generating from $\pi(x)$ as given in (3.3) using the rejection method. □

Weighted Bootstrap: Draw m samples, (m is usually huge) $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}$ from the distribution $g(\mathbf{x})$. Calculate the weights w_i and then calculate the normalized weights $q_i = \frac{w_i}{\sum w_i}$. Note that $\mathbf{x}^{(i)}, q_i$ define a discrete probability distribution. A random sample $\mathbf{x}$ drawn from this discrete distribution is an approximate sample from π. Draw N random samples this way and form $\bar{b}_N$ given by (3.4).

♡ **Example** 3.16. Can experiment with the normal distribution in one dimension. Take π to be the standard normal distribution and g to be a normal distribution with zero mean and variance σ^2. We need to take $\sigma^2 > 1$ for good results. Why? □

How do we get a g? By using the Laplace approximation shown earlier.

Remark: It is important to understand the points about all the sampling ideas described above. These together are called Sampling-Importance-Re-sampling *SIR* methodology.

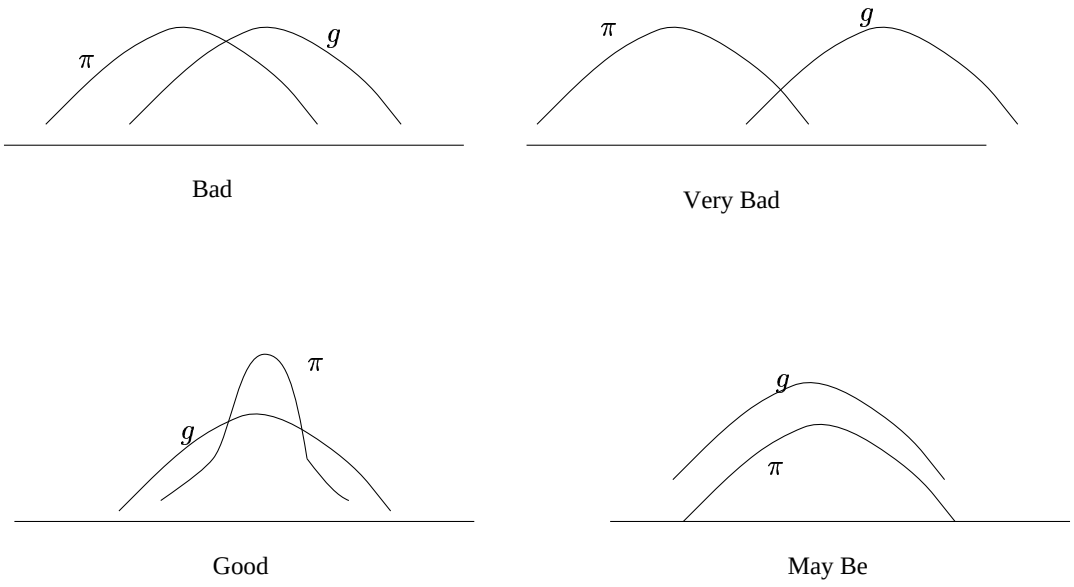

Figure 3.1: How different choices of g influences the computation.

3.5 Markov Chain Monte Carlo

As models become more complex in high dimension the posterior distributions become analytically intractable. All of the previous methods fail. The solution is to consider iterative Monte Carlo or Markov chain Monte Carlo. Simulate a Markov chain with stationary distribution given by the posterior distribution, $\pi(\mathbf{x})$. Features of π are discovered (accurately) by forming *ergodic averages* as in (3.4). It turns out that $\bar{b}_N$ still accurately estimates $E_\pi(b)$ if we generate samples using a Markov chain. Theorems like the CLT and laws of large numbers can be proven.

3.5.1 Notions of Markov Chains

A Markov chain is generated by sampling

$$x^{(t+1)} \sim p(x|x^{(t)}).$$

This p is called the *transition kernel* of the Markov chain. So $x^{(t+1)}$ depends only on $x^{(t)}$, not on $x^{(0)}, x^{(1)}, \ldots, x^{(t-1)}$.

♡ **Example** 3.17.

$$x^{(t+1)} \sim N(x^{(t)}, 1)$$

□

Stationarity As $t \to \infty$, the Markov chain converges in distribution to its *stationary* distribution. This is also called its invariant distribution.

Irreducibility *Irreducible* means any set of states can be reached from any other state in a finite number of moves (or transitions).

Ergodicity Suppose that we have an irreducible Markov chain with stationary distribution $\pi(x)$. Then we have an *ergodic* theorem:

$$\begin{aligned} \bar{b}_N &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N} b\left(\mathbf{x}^{(i)}\right) \\ &\to E_\pi[b(\mathbf{x})] \text{ as } N \to \infty. \end{aligned}$$

Can we estimate the standard error of $\bar{b}_N$? Yes! the numerical standard error is given by:

$$\text{nse}\left(\bar{b}_N\right) = \sqrt{\frac{1}{N}\text{var}_\pi(b)\left\{1 + 2\sum_{t=1}^{N-1}\rho_t(b)\right\}}$$

where $\rho_t(b)$ is the lag$-t$ correlation in $\{b(x^{(i)})\}$. Hence we can make nse as small as we like by increasing N. For independent sampling, the correlation term disappears. The correlation term can decrease the nse but usually increases it. The nse may not be finite for general Markov chains, but is finite when the Markov chain is 'good': geometrically ergodic.

3.5.2 Metropolis-Hastings

The most general algorithm known as the Metropolis-Hastings algorithm is given as follows.

1. Start anywhere and say we are at $\mathbf{x}^{(t)} = \mathbf{x}$.
2. Generate $\mathbf{y}$ from $q(\mathbf{y}|\mathbf{x})$. $\mathbf{y}$ is called a *candidate point* and q is called a *proposal distribution.*
3. Calculate $\alpha(\mathbf{x}, \mathbf{y}) = \min\left\{1, \frac{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})}{\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})}\right\}$
4. Accept $\mathbf{x}^{(t+1)} = \mathbf{y}$ with probability $\alpha(\mathbf{x}, \mathbf{y})$.
5. Else set $\mathbf{x}^{(t+1)} = \mathbf{x}$.

Note the involvement of π : the target density. It only enters through the ratio $\frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}$. Hence we do not need to know the normalizing constant to implement the algorithm.

Special cases:

1. $q(\mathbf{y}|\mathbf{x}) = q(\mathbf{x}|\mathbf{y})$: Metropolis Algorithm.
2. $q(\mathbf{y}|\mathbf{x}) = g(\mathbf{y})$: Independence Sampler.
3. $q(\mathbf{y}|\mathbf{x}) = \prod \pi(y_i|\mathbf{y}_{<i}, \mathbf{x}_{>i}) \Rightarrow \alpha(\mathbf{x}, \mathbf{y}) = 1$: Gibbs Sampler.

Generally it is easier to analyse the first two versions theoretically. The last version, the Gibbs sampler however, is much easier to implement. We will try to understand each of the above.

Metropolis Algorithm: Here the proposal distribution is symmetric. i.e.,

$$q(\mathbf{y}|\mathbf{x}) = q(\mathbf{x}|\mathbf{y}).$$

A special case of this is

$$q(\mathbf{y}|\mathbf{x}) = q(|y - x|).$$

Note that for this choice we have

$$\alpha(\mathbf{x}, \mathbf{y}) = \min\left\{1, \frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}\right\}$$

This is called the *Random-Walk* Metropolis algorithm. Proposals depend on the current value.

♡ **Example** 3.18. Return to the normal-Cauchy example. Take $q(\mathbf{y}|\mathbf{x})$ to be the normal with mean x and variance σ^2. Write computer codes for generating from $\pi(x)$ as given in (3.3). □

♡ **Example** 3.19. Suppose that the target distribution is $\pi(x)$ is the standard normal distribution. Take $q(\mathbf{y}|\mathbf{x})$ to be the normal with mean x and variance σ^2. Experiment different values for σ^2. Calculate the percentage of acceptance. Form the estimator for the mean of the distribution. □

♡ **Example** 3.20. Modify the above program to generate from $\Gamma(\alpha, \beta)$ distribution for given α and β using the same proposal distribution. □

♡ **Example** 3.21. Write programs to generate from $\pi(\mathbf{x}) = N_p(\mathbf{0}, I)$, the p dimensional normal distribution. □

Independence Sampler Here we choose

$$q(\mathbf{y}|\mathbf{x}) = g(\mathbf{y}).$$

Proposals are drawn from a fixed density g, just as we had done in the importance sampling case. Independence samplers are either very good or very bad. We can get independent samples from here by choosing $g = \pi$. Then $\alpha(\mathbf{x}, \mathbf{y}) = 1$ and the chain will never reject a candidate. However, the above is perhaps a dream and hard to achieve in practice. Tails of $g(\cdot)$ must be heavier than tails of $\pi(\cdot)$ for geometric convergence.

♡ **Example** 3.22. $\pi(x) = N(0, 1)$ is the standard normal distribution. $q(y|x) = g(\mathbf{y}) = N(0, \sigma^2)$. □

♡ **Example** 3.23. Return to the normal-Cauchy example. Take $q(\mathbf{y}|\mathbf{x}) = \frac{1}{\pi}\frac{1}{1+y^2}$, the standard Cauchy distribution. Note that it does not depend on $\mathbf{x}$. Write computer codes for generating from $\pi(x)$ as given in (3.3). □

3.5.3 Gibbs Sampler

Let $\mathbf{x} = (x_1, \ldots, x_k)^T$. Suppose that we start the algorithm at $\mathbf{x}^{(0)}$. This could be any point in the support of π. Given $\mathbf{x}^{(t)}$ we iterate one step as follows. To sample from the k dimensional

distribution $\pi(\mathbf{x})$, the Gibbs sampler makes a Markov transition from $\mathbf{x}^{(t)}$ to $\mathbf{x}^{(t+1)}$ as follows.

$$\begin{aligned} x_1^{(t+1)} &\sim \pi(x_1|x_2^{(t)}, x_3^{(t)}, \cdots, x_k^{(t)}) \\ x_2^{(t+1)} &\sim \pi(x_2|x_1^{(t+1)}, x_3^{(t)}, \cdots, x_k^{(t)}) \\ \vdots \quad &\vdots \quad \vdots \\ x_k^{(t+1)} &\sim \pi(x_k|x_1^{(t+1)}, x_2^{(t+1)}, \cdots, x_{k-1}^{(t+1)}). \end{aligned}$$

The densities on the right hand sides of the above are called *complete conditional distributions* or the *full conditional distributions*. Note that always the most up-to-date version $\mathbf{x}$ is used.

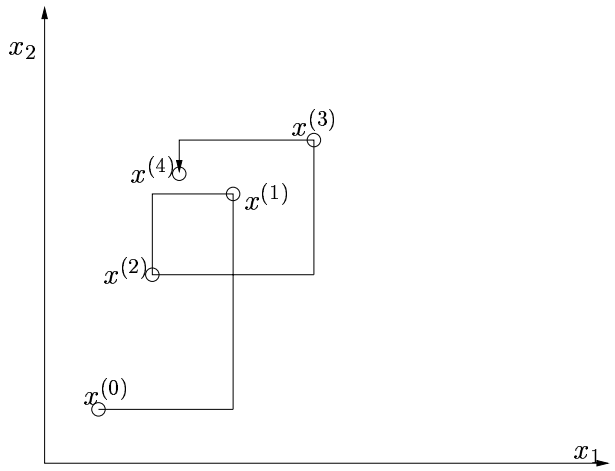

Figure 3.2: Diagram showing how the Gibbs sampler works.

♡ **Example** 3.24. Suppose $y_i \overset{iid}{\sim} N(\mu, \sigma^2), i = 1, 2, \ldots, n$. Assume flat prior for μ and σ^2 as $\pi(\mu, \sigma^2) = \sigma^{-2} I(\sigma^2 > 0)$. The posterior distribution of μ and σ^2 is given by

$$\pi(\mu, \sigma^2|\mathbf{y}) \propto \frac{1}{(\sigma^2)^{n/2+1}} \exp\left\{-\frac{1}{2}\sum\left(\frac{y_i - \mu}{\sigma^2}\right)^2\right\}.$$

The Gibbs sampler for this problem samples μ and σ^2 alternatively. It samples μ from the distribution, $N(\bar{y}, \sigma^2/n)$ and $\frac{1}{\sigma^2}$ from the distribution $\Gamma(\frac{n}{2}, \frac{1}{2}\sum(y_i - \mu)^2)$. Write a computer program to implement this Gibbs sampler. Check your answers from the exact results. □

♡ **Example** 3.25. Try writing the complete conditional distributions for the pump failure data. Then write a Gibbs sampler program for sampling from the posterior distribution. □

In the above examples the complete conditional distributions are easy to sample from because they are standard. Alternative strategies are used when some of the complete conditionals are non-

standard. When they are *log-concave*, i.e., second derivative of the log density is strictly decreasing, the *adaptive rejection sampling* can be used. C programs are available to do this. Otherwise a Metropolis step can be used to update any complete conditional which is non-standard.

3.5.4 Implementation Issues

Theory suggests that if we run the chain long enough, it will converge to the stationary distribution. However, sometimes there can be slow convergence. Hence, some efforts are made to 'diagnose' convergence. Note that even if your Markov chain gives good convergence diagnostics you can not make the claim that you have *proved* that it has converged. Rather these diagnostics are treated as a 'feel-good-factor'.

Diagnostics 1: Visual plots Do a time series plot of the different components of the Markov chain. Calculate autocorrelations and see how quickly they die down as you increase the lag. For a fast mixing Markov chain they should die quite rapidly. Rerun the Markov chain with different starting points and overlay the time series of the same component. The replicates should criss-cross many times. If they do not, you should investigate whether there is multi-modality or something.

How many Chains? Advice in the literature is conflicting.

- One very long run. Geyer (1992), Raftery and Lewis (1992). Reaches part other schemes can not. If you are a proabilist, you would prefer this.
- Several long runs. Gelman and Rubin (1992). Gives indication of convergence. Argue that a single chain can be misleading.

Single Chain: Raftery and Lewis (1992) consider the problem of calculating the number of iterations necessary to estimate a posterior quantile from a single run of a Markov chain. They propose a 2-state Markov chain model fitting procedure based upon pilot analysis of output from the original chain.

Suppose that we wish to estimate a particular posterior quantile for some function b of a parameter (or set of parameters) $\mathbf{X}$, i.e., we wish to estimate u such that

$$P(b(\mathbf{X}) \leq u) = q$$

for some pre-specified q and so that, given our estimate $\hat{u}$, $P(b(\mathbf{X}) \leq \hat{u})$ lies within $\pm r$ of the true value, say, with probability p.

Raftery and Lewis propose a method to calculate n_0, the initial number of iterations to discard (which we call the 'burn-in'), k, the thinning size, i.e., every k^{th} iterate of the Markov chain and n the number of further iterations required to estimate the above probability to within the required accuracy.

The details of the methodology can be found on the cited reference. Basically the methodology construct a discrete two state Markov chain $Z^t = I_{b^t=b(\mathbf{X}^t)\leq u}$ from $\mathbf{X}^t$. Then find the eigenvalues of this Markov chain to find the answers.

They also suggest looking at a quantity called the dependence factor:

$$I = \frac{n_0 + n}{n_{min}}$$

where n_{min} is the number of initial iterations performed to produce the estimates. Ideally, this factor should be close to 1.

Multi Chain: Gelman and Rubin (1992) propose a method which assesses convergence by monitoring the ratio of two variance estimates. In particular their method uses multiple replications and is based upon a comparison, for scalar functions of $\mathbf{X}$, of the within sample variance for each of m parallel chains, and the between sample variance of different chains. This is essentially a classical analysis of variance.

The method consists of analysing the independent sequences to form a distributional estimate for what is known about the target random variable, given the observations simulated so far. This distributional estimate, based upon the *Student's t* distribution, is somewhere between the starting and target distributions, and provides a basis for an estimate of how close the process is to convergence and, in particular, how much we might expect the estimate to improve with more simulations. The method proceeds as follows. We begin by independently simulating $m \geq 2$ sequences of length $2n$, each beginning at different starting points which are over dispersed with respect to the stationary distribution. We discard the first n iterations and retain only the last n. Then for each scalar functional of interest, b, we calculate B/n, the variance between the m sequence means, which we denote by $\bar{b}_{i.}$. Thus we define

$$\frac{B}{n} = \frac{1}{m-1}\sum_{i=1}^{m}(\bar{b}_{i.} - \bar{b}_{..})^2,$$

where

$$\bar{b}_{i.} = \frac{1}{n}\sum_{t=n+1}^{2n} b_i^t, \quad \bar{b}_{..} = \frac{1}{m}\sum_{i=1}^{m} b_{i.}^t$$

and $b_i^t = b(\mathbf{x}_i^t)$ is the t^{th} observation of b from chain i. We then calculate W, the mean of the m within-sequence variances, s_i^2. Thus, W is given by

$$W = \frac{1}{m}\sum_{i=1}^{m} s_i^2,$$

where

$$s_i^2 = \frac{1}{n-1}\sum_{t=n+1}^{2n} (\bar{b}_{i.} - \bar{b}_{..})^2.$$

We can then estimate the target variance by a weighted average of W and B, given by

$$\hat{\sigma}^2 = \frac{n-1}{n}W + \frac{1}{n}B,$$

which overestimates the true value. We can improve upon the $N(\hat{\mu}, \hat{\sigma}^2)$ estimate of the stationary distribution by allowing for the variability of both $\hat{\mu}$ and $\hat{\sigma}^2$ and thus we obtain a t-distribution with with mean $\hat{\mu}$ and variance

$$\hat{V} = \hat{\sigma}^2 + \frac{B}{mn}$$

with a certain df to be estimated. Gelman and Rubin diagnostic is then calculates as

$$R = \frac{\hat{V}}{\hat{\sigma}^2}.$$

As $n \to \infty$ this should shrink to 1. For more details see the cited references.