MCMC algorithms for fitting Bayesian models

We have seen that direct simulation techniques are often suitable for fitting Bayesian models with some particular non-informative priors (recall your Linear Model notes). However, the real advantage of Bayesian statistics is the flexibility it offers to fit hierarchical models with more general priors. Unfortunately, direct simulation methods often turn out to be inadequate in such settings. More complex algorithms, known as Markov Chain Monte Carlo algorithms need to be designed; these are precisely what are used by WinBUGS for model fitting. These algorithms can also fit the simplest Bayesian models (for example, the cases you have already handled with non-informative priors). We describe below the Gibbs sampler and the Metropolis Hastings algorithms – two of the most popular MCMC algorithms.

The Gibbs sampler We will lay down the general principles and illustrate with the Normal linear model. First, we collect the data into a vector $Y$ and all our model parameters in a vector $\theta$. Then, note that the posterior distribution of $\theta$ will be given by,

$$ f(\theta|Y) \propto f(\theta) \times f(Y|\theta), $$

where $f(\theta)$ is the prior distribution and $f(Y|\theta)$ is the likelihood. Let us get more explicit letting $\theta = (\theta_1, \ldots, \theta_p)$ be the parameters in our model. The Gibbs sampler will start a Markov Chain with a set of initial values $\theta_0 = (\theta_{01}, \ldots, \theta_{0p})$ and then perform the $i^{th}$ iteration, say for $i = 1, \ldots, M$, by updating successively from the full conditional distributions:

$$ \theta_{i1} \sim f(\theta_1|\theta_{i-1,2}, \ldots, \theta_{i-1,p}, Y) $$

$$ \theta_{i2} \sim f(\theta_2|\theta_{i1,1}, \theta_{i-1,3}, \ldots, \theta_{i-1,p}, Y) $$

$$ \ldots $$

(the generic $k^{th}$ element) $\theta_{ik} \sim f(\theta_k|\theta_{i,1,\ldots,k-1}, \theta_{i-1,k+1}, \ldots, \theta_{i-1,p}, Y)$

$$ \ldots $$

$$ \theta_{ip} \sim f(\theta_p|\theta_{i,1,\ldots,i-1,p-1}, Y) $$

The completion of the above loop results in a single iterate of the Gibbs sampler with an update of $\theta_1 = (\theta_{11}, \ldots, \theta_{1p})$. This is repeated $M$ times to obtain a Gibbs sample of vectors $\theta_1, \ldots, \theta_M$. 
From the theory of Markov chains it can be shown that such a chain will eventually converge to a stationary or equilibrium distribution which is precisely the posterior distribution \( f(\theta | Y) \). What this means from a practical standpoint is that if we sample long enough, in the above scheme, we will eventually be sampling from the posterior distribution itself. So, after we discard an initial set of samples (called burn-in) we retain the remaining samples as our posterior sample and carry out all inference on them.

Next let us consider the linear model example. We have the equation:

\[
Y = X\beta + \epsilon
\]

where \( Y \) is a \( n \times 1 \) vector of responses, \( X \) is a \( n \times p \) vector of covariates, \( \beta \) is the \( p \times 1 \) vector of regression parameters and \( \epsilon \) is a vector of i.i.d. errors, distributed as \( N(0, \sigma^2) \). All our inference will be implicitly conditioned on \( X \). Consider again a flat prior on \( \beta \), but an Inverted-Gamma prior, \( IG(a, b) \) (so \( 1/\sigma^2 \) has a Gamma distribution with mean \( = a/b \), variance \( = a/b^2 \)) on \( \sigma^2 \).

Therefore, our \( \theta \) is \((\beta, \sigma^2)\) and we need to update these. It can be easily computed that the full conditional distributions needed are given by:

\[
\begin{align*}
  f(\beta | Y, \sigma^2) &= N((X^T X)^{-1} X^T Y, \sigma^2 (X^T X)^{-1}) \\
  f(\sigma^2 | Y, \beta) &= IG(a + n/2, b + \frac{1}{2} (Y - X\beta)^T (Y - X\beta))
\end{align*}
\]

The latter is much easier to identify than the marginal distribution of \( \sigma^2 \). This is the primary benefit of the Gibbs sampler – it helps avoid computing unfriendly marginal distributions.

**Homework** Write your own \( \text{R} \) code to fit a linear model from the regression dataset you have been using from www.biostat.umn.edu/~sudiptob/pubh7440/LinearModelExample.txt, but now designing a MH sampler. First fit a classical linear model (OLS) using any standard statistical software. Next adopt a Bayesian approach using an Inverse Gamma \( IG(2, b) \) prior for \( \sigma^2 \) with a value of \( b \) that gives a reasonable prior estimate for \( \sigma^2 \) (from your OLS) and a flat prior for \( \beta \). Run 2 chains for convergence analysis and compare with your WinBUGS and Gibbs sampler results.

**Metropolis-Hastings** Note that in the above setting we have closed-form full conditional distributions for both the parameter components, \( \beta \) and \( \sigma^2 \). Sometimes, in more complex settings, some full conditional distributions are not available in closed form solution. In such cases, a Metropolis or a Metropolis-Hastings step is often needed. Although the M-H is a more versatile algorithm (in fact, the Gibbs sampler can be looked upon as a special case of the M-H), it is widely used as a step within the Gibbs sampler when a full
conditional sampling is not easy. We will not get into the details of the Metropolis algorithm, but provide a brief description of its use within the Gibbs sampler. Again, this is how WinBUGS performs its updates.

Suppose you want to draw from a full conditional distribution (univariate or multivariate) of the $k^{th}$ component, say $f(\theta_k | \theta_{-k}, Y)$, which is not easy to draw from. Then, a Metropolis step may be used to carry out the update, using the following steps:

Select a candidate distribution, say $g(\cdot, \nu)$, where $\nu$ may be its parameters that are fixed by the user.

Theoretically it can be anything, but in practice you choose either a Normal distribution if your parameter can be any real number, or a log-normal if it has positive support.

Draw $U \sim g(\cdot, \nu)$ and compute

$$r = \frac{f(U | \theta_{-k}, Y) g(\theta_k, \nu)}{f(\theta_k | \theta_{-k}, Y) g(U, \nu)}$$

Set the new value of $\theta_k$ as $U$ with probability minimum $(r, 1)$, otherwise retain the current value of $\theta_k$. 