Bayesian Inference: Metropolis-Hastings Sampling

Ilker Yildirim
Department of Brain and Cognitive Sciences
University of Rochester
Rochester, NY 14627

August 2012

References: Most of the material in this note was taken from: (1) Lynch, S. M. (2007). *Introduction to Applied Bayesian Statistics and Estimation for Social Scientists*. New York: Springer; and (2) Taylan Cemgil's lecture slides on Monte Carlo Methods (http://www.cmpe.boun.edu.tr/courses/cmpe58n/fall2009/)

1 Introduction

When performing Bayesian inference, we aim to compute and use the full posterior joint distribution over a set of random variables. Unfortunately, this often requires calculating intractable integrals. In such cases, we may give up on solving the analytical equations, and proceed with sampling techniques based upon Markov Chain Monte Carlo (MCMC) methods. When using MCMC methods, we estimate the posterior distribution and the intractable integrals using simulated samples from the posterior distribution.

In a separate Computational Cognition Cheat Sheet, we cover Gibbs sampling, another MCMC method. When using Gibbs sampling, the first step is to analytically derive the posterior conditionals for each of the random variables [e.g., $p(X_1|X_2, X_3)$, $p(X_2|X_1, X_3)$, and $p(X_3|X_1, X_2)$]. Then we simulate posterior samples from the target joint posterior by iteratively sampling a value for a random variable from its corresponding posterior conditional while all other variables are fixed to their current values. Gibbs sampling is perhaps the most frequently used MCMC technique.

However, there are several limitations to it. First, even if we have the full posterior joint density function, it may not be possible or practical to derive the conditional distributions for each of the random variables in the model. Second, even if we have the posterior conditionals for each variable, it might be that they are not of a known form, and therefore there is not a straightforward way to draw samples from them. Finally, there are cases in which Gibbs sampling will be very inefficient. That is, the "mixing" of the Gibbs sampling chain might be very slow, meaning that the algorithm may spend a long time exploring a local region with high density, and thus take very long to explore all regions with significant probability mass. For example, when the cross-correlation of the posterior conditional distributions between variables is high, successive samples become very highly correlated and sample values change very slowly from one iteration to the next, resulting in chains that basically do not mix.

The Metropolis-Hastings (MH) algorithm simulates samples from a probability distribution by making use of the full joint density function and (independent) proposal distributions

Algorithm 1 Metropolis-Hastings algorithm

Initialize $x^{(0)} \sim q(x)$
for iteration $i = 1, 2, \ldots$ **do**
  Propose: $x^{cand} \sim q(x^{(i)}|x^{(i-1)})$
  Acceptance Probability:
    $\alpha(x^{cand}|x^{(i-1)}) = \min\{1, \frac{q(x^{(i-1)}|x^{cand})\pi(x^{cand})}{q(x^{cand}|x^{(i-1)})\pi(x^{(i-1)})}\}$
  $u \sim$ Uniform $(u; 0, 1)$
  if $u < \alpha$ **then**
    Accept the proposal: $x^{(i)} \leftarrow x^{cand}$
  else
    Reject the proposal: $x^{(i)} \leftarrow x^{(i-1)}$
  end if
end for

for each of the variables of interest. Algorithm 1 provides the details of a generic MH algorithm.

The first step is to initialize the sample value for each random variable (this value is often sampled from the variable's prior distribution). The main loop of Algorithm 1 consists of three components: (1) Generate a proposal (or a candidate) sample x^{cand} from the proposal distribution $q(x^{(i)}|x^{(i-1)})$; (2) Compute the acceptance probability via the acceptance function $\alpha(x^{cand}|x^{(i-1)})$ based upon the proposal distribution and the full joint density $\pi(\cdot)$; (3) Accept the candidate sample with probability α, the acceptance probability, or reject it with probability $1 - \alpha$.

Proposal Distribution: The MH algorithm starts with simulating a "candidate" sample x^{cand} from the proposal distribution $q(\cdot)$. Note that samples from the proposal distribution are not accepted automatically as posterior samples. These candidate samples are accepted probabilistically based on the acceptance probability $\alpha(\cdot)$. There are mainly two kinds of proposal distributions, symmetric and asymmetric. A proposal distribution is a symmetric distribution if $q(x^{(i)}|x^{(i-1)}) = q(x^{(i-1)}|x^{(i)})$. Straightforward choices of symmetric proposals include Gaussian distributions or Uniform distributions centered at the current state of the chain. For example, if we have a Gaussian proposal, then we have $x^{cand} = x^{(i-1)} +$ Normal$(0, \sigma)$. Because the pdf for Normal$(x^{cand} - x^{(i-1)}; 0, \sigma) =$ Normal$(x^{(i-1)} - x^{cand}; 0, \sigma)$, this is a symmetric proposal. This proposal distribution randomly perturbs the current state of the chain, and then either accepts or rejects the pertubed value. Algorithms of this form are called "Random-walk Metropolis algorithm."

Random-walk MH algorithms are the most common MH algorithms. However, we may choose to (or need to) work with asymmetric proposal distributions in certain cases. For example, we may choose a proposal distribution that is inherently asymmetric, such as the log-normal density, which is skewed towards larger values. In other cases, we may need to work with asymmetric proposal distributions to accommodate for particular constraints in

our models. For example, if we wish to estimate the posterior distribution for a variance parameter, we require that our proposal does not generate values smaller than 0.

Acceptance function: Intuitively, the MH acceptance function is designed to strike a balance between the following two constraints: (1) The sampler should tend to visit higher probability areas under the full joint density (this constraint is given by the ratio $\frac{\pi(x^{cand})}{\pi(x^{(i-1)})}$); (2) The sampler should explore the space and avoid getting stuck at one site (e.g., the sampler can reverse its previous move in the space; this constraint is given by the ratio $\frac{q(x^{(i-1)}|x^{cand})}{q(x^{cand}|x^{(i-1)})}$). It is important that the MH acceptance function has this particular form because this form ensures that the MH algorithm satisfies the condition of *detailed balance*, which guarantees that the stationary distribution of the MH algorithm is in fact the target posterior that we are interested in (see Gilks et al., 1996, for more details).

Importantly, note that the acceptance function can be asymmetric (e.g., $\alpha(x^{(i)}|x^{(i-1)})) \neq \alpha(x^{(i-1)}|x^{(i)}))$ irrespective of the proposal distribution. Let's derive the acceptance function in the case of symmetric proposals:

$$\alpha(x^{(i)}|x^{(i-1)}) = \min\{1, \frac{q(x^{(i-1)}|x^{(i)})\pi(x^{(i)})}{q(x^{(i)}|x^{(i-1)})\pi(x^{(i-1)})}\} = \min\{1, \frac{\pi(x^{(i)})}{\pi(x(i-1))}\} \tag{1}$$

where $\pi(\cdot)$ is the full joint density. This result is intuitive. When the proposal distribution is symmetric — $q(x^{(i)}|x^{(i-1)}) = q(x^{(i-1)}|x^{(i)})$ — the acceptance probability becomes proportional to how likely each of the current state $x^{(i-1)}$ and the proposed state $x^{(i)}$ are under the full joint density.

Are Gibbs sampling and MH sampling related? Yes. In fact, Gibbs sampling is a special case of MH sampling where proposal distributions are the posterior conditionals. Recall that all proposals are accepted in Gibbs sampling, which implies that the acceptance probability is always 1. The algebra below shows that the acceptance function is equal to 1 for Gibbs sampling:

$$\begin{aligned}
&\alpha(x_n^{cand}, x_{-n}^{(i-1)}|x_n^{(i-1)}, x_{-n}^{(i-1)}) \\
&= \min\{1, \frac{q(x_n^{(i-1)}, x_{-n}^{(i-1)}|x_n^{cand}, x_{-n}^{(i-1)})p(x_n^{cand}, x_{-n}^{(i-1)})}{q(x_n^{cand}, x_{-n}^{(i-1)}|x_n^{(i-1)}, x_{-n}^{(i-1)})p(x_n^{(i-1)}, x_{-n}^{(i-1)})}\} \\
&= \min\{1, \frac{p(x_n^{(i-1)}|x_{-n}^{(i-1)})p(x_n^{cand}, x_{-n}^{(i-1)})}{p(x_n^{cand}|x_{-n}^{(i-1)})p(x_n^{(i-1)}, x_{-n}^{(i-1)})}\} \\
&= \min\{1, \frac{p(x_n^{(i-1)}|x_{-n}^{(i-1)})p(x_n^{cand}|x_{-n}^{(i-1)})p(x_{-n}^{(i-1)})}{p(x_n^{cand}|x_{-n}^{(i-1)})p(x_n^{(i-1)}|x_{-n}^{(i-1)})p(x_{-n}^{(i-1)})}\} \\
&= 1
\end{aligned} \tag{2}$$

Here, we used the fact that the proposal distributions for Gibbs sampling are the posterior conditionals [i.e., $q(x_n^{(i-1)}, x_{-n}^{(i-1)}|x_n^{cand}, x_{-n}^{(i-1)}) = p(x_n^{(i-1)}|x_{-n}^{(i-1)})$]. We also made use of the chain rule, where we wrote the full joint distribution as the product of two terms [e.g., $p(x_n^{cand}, x_{-n}^{(i-1)}) = p(x_n^{cand}|x_{-n}^{(i-1)})p(x_{-n}^{(i-1)})$].

Accept/Reject a proposal: Finally, we accept a given proposal with the acceptance probability α which is the outcome of the acceptance function described above. The `min` operator in the acceptance function makes sure that the acceptance probability α is never larger than 1. Operationally, we draw a random number uniformly between 0 and 1, and if this value is smaller than α, we accept the proposal; otherwise we reject it.

2 A simple model of covariation

We illustrate the MH algorithm on a very simple example. Consider two streams of observations $x_{1:N}$ and $y_{1:N}$. We are interested in modeling the correlation ρ between these two streams. We model the observations given the correlation as a bivariate Gaussian distribution:

$$x_i, y_i|\rho \sim \text{ Normal}(\mu, \Sigma) \tag{3}$$

where $\mu = [\mu_x \quad \mu_y]$ and $\Sigma = \begin{bmatrix} \sigma_{xx} & \rho \\ \rho & \sigma_{yy} \end{bmatrix}$. For simplicity, we assume that we know $\mu_x, \mu_y = 0$ and $\sigma_{xx}, \sigma_{yy} = 1$. In this case, the likelihood function takes the following form:

$$p(x_i, y_i|\rho) = \prod_{i=1}^{N} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\{-\frac{1}{2(1-\rho^2)}[x_i^2 - 2\rho x_i y_i + y_i^2]\} \tag{4}$$

For a fuller specification of the model, we need to specify a prior distribution over the correlation parameter ρ. A non-informative prior for covariance matrices is the "Jeffreys" prior (see Gelman et al., 1995), which is of the form $1/|\Sigma|^{3/2}$. In our case, the Jeffreys prior takes the following form:

$$p(\rho) = \frac{1}{|\Sigma|^{3/2}} = \frac{1}{\begin{vmatrix} 1 & \rho \\ \rho & 1 \end{vmatrix}^{3/2}} = 1/(1-\rho^2)^{3/2} \tag{5}$$

Using Bayes rule, we write the posterior distribution for the correlation parameter ρ in the following way:

$$p(\rho|x_{1:N}, y_{1:N}) \propto 1/(1-\rho^2)^{3/2} \prod_{i=1}^{N} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\{-\frac{1}{2(1-\rho^2)}[x_i^2 - 2\rho x_i y_i + y_i^2]\} \tag{6}$$

3 Inference with a MH sampler

The posterior in Equation 6 doesn't appear to be of any known form. Therefore, Gibbs sampling is not straightforward. Instead, we develop a random-walk MH algorithm to infer the posterior distribution $p(\rho|x_{1:N}, y_{1:N})$.

The sole latent variable in our model is ρ. There are two things we need to specify to fully develop an MH sampler, namely the proposal distribution and the acceptance function.

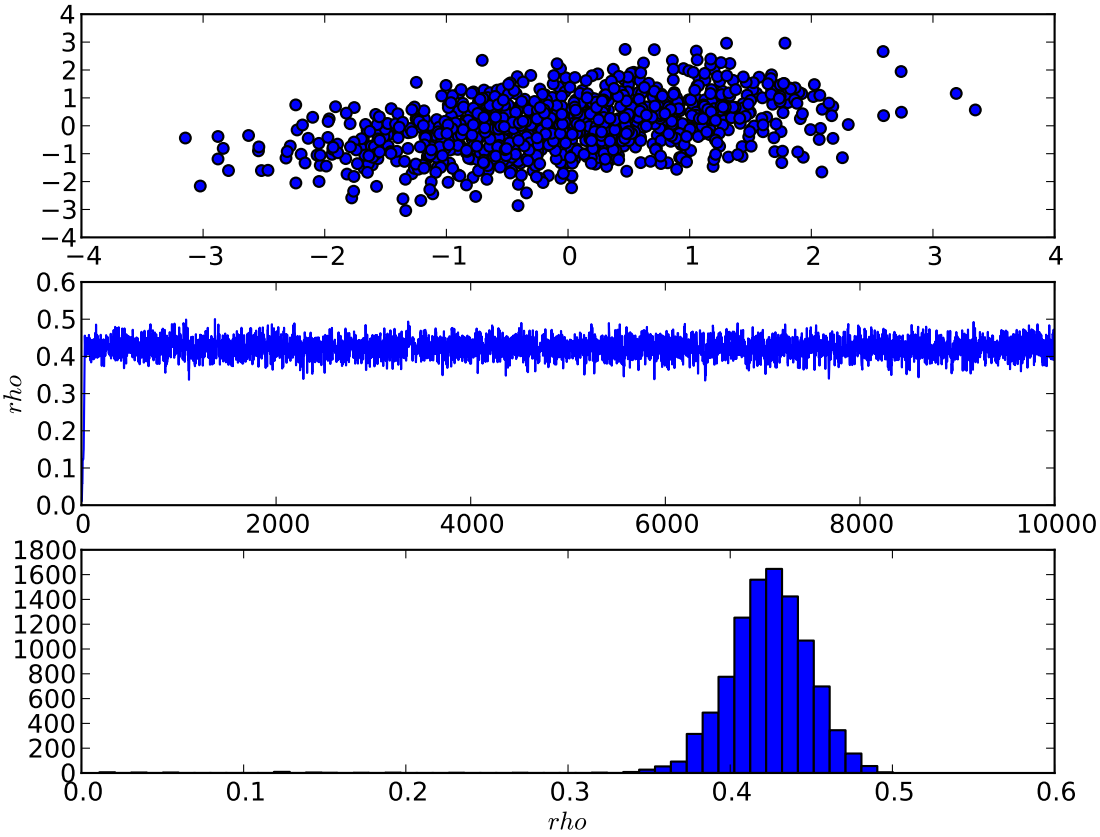

Figure 1: (Top row) Random data generated using the Python function numpy.multivariate_normal with $N = 1000$. (Middle row) A trace plot for ρ. (Bottom row) A histogram plot for the posterior distribution of ρ based upon the samples in the chain. The mean of this distribution is 0.42 and the standard deviation is 0.03.

Here we choose to work with a symmetric proposal distribution because it makes the algorithm more straightforward, both conceptually and computationally. In particular, we use a Uniform distribution centered at the current value of ρ with an overall width of 0.14.

$$\rho^{cand} \sim \text{ Uniform}(\rho^{(i-1)} - 0.07, \rho^{(i-1)} + 0.07) \tag{7}$$

Note that our choice of proposal distribution is not unique. We could have chosen a wider or a narrower Uniform distribution. Furthermore, we could have worked with a Gaussian distribution or perhaps an asymmetric proposal distribution. A good rule of thumb is that a good proposal distribution maintains the acceptance rate of proposals in a reasonable range. That is, if nearly all candidate samples are being rejected (low acceptance rate) or if nearly all candidates are being accepted (high acceptance rate), then it is likely that the proposal distribution is either too wide or too narrow, respectively.

Under this symmetric proposal distribution, the acceptance function can be obtained using Equation 1:

$$\alpha(\rho^{(i)}|\rho^{(i-1)}) = \min\{1, \frac{p(\rho^{(i)}|x_{1:N}, y_{1:N})}{p(\rho^{(i-1)}|x_{1:N}, y_{1:N})}\} \tag{8}$$

where $p(\rho|x_{1:N}, y_{1:N})$ is given in Equation 6.

We simulated a single chain for 10000 steps. Figure 1 illustrates the results. A trace plot for ρ is shown in the middle row. The thick pencil pattern (Gelman et al., 1995) indicates that the chain converged almost immediately. The bottom row shows the posterior distribution $p(\rho|x_{1:N}, y_{1:N})$ in a histogram plot based upon the posterior samples. An implementation of this algorithm written in the Python programming language can be found on the Computational Cognition Cheat Sheet website.

Gelman, A., Carlin, J. B., Stern. H. S., & Rubin, D. B. (1995). *Bayesian Data Analysis*. London: Chapman and Hall.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall.