8.4 The Posterior via Markov Chain Monte Carlo

Zongyi Liu

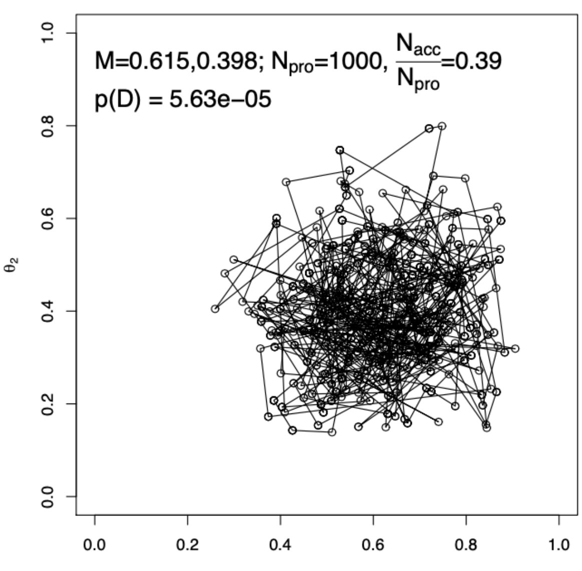
2023-06-17

8.4 The Posterior via Markov Chain Monte Carlo

Imagine that we have a likelihood function and a prior distribution function that cannot be handled adequately by formal analysis, so we cannot determine the posterior distribution via formal mathematics alone.

Imagine also that the parameter space is too big to be adequately spanned by a dense grid of points, so we cannot determine the posterior distribution via dense grid approximation.

An alternative approach to approximating the posterior distribution, as introduced in Chapter 7, is to generate a large number of representative values from the posterior distribution, and estimate the posterior from those representative values. Even though our present application, involving only two parameters on a limited domain, can be addressed using grid approximation, it is highly instructive to apply the Metropolis algorithm also. This will be our first application of the Metropolis algorithm to a two-dimensional parameter space.



8.4.1 Metropolis Algorithm

As introduced before, the random walk starts at some arbitrary point in the parameter space. We propose a jump to a new point in parameter space. The proposed jump is randomly generated from a **proposal distribution**, from which we assume it is easy to generate values.

The proposed jump is definitely accepted if the posterior is more dense at the proposed position than the current position, and the proposed jump is probabilistically accepted if the posterior is less dense at the proposed position than at the current position. As we defined before (7.1):

$$p_{\text{move}} = \min\left(\frac{P(\theta_{\text{proposed}})}{P(\theta_{\text{current}})}, 1\right)$$

The figure above shows the Metropolis algorithm applied to a case in which the prior dis- tribution is a product of beta distributions on each proportion.

The trajectory shown in the figure above excludes an initial "burn-in" period, which may have been unduly affected by the arbitrary starting position.

In this case, the proposal distribution was a symmetric bivariate normal with standard deviations of 0.2.

8.4.2 Gibbs Sampling

The Metropolis algorithm is very general and broadly applicable. One problem with it, however, is that the proposal distribution must be properly tuned to the posterior distribution if the algorithm is to work well. If the proposal distribution is too narrow or too broad, a large proportion of proposed jumps will be rejected and/or the trajectory will get bogged down in a localized region of the parameter space.

The Gibbs sampling could be helpful to solve such problem.

Gibbs sampling will allow us to generate a sample from the joint posterior, $p(\theta_1, \theta_2, \theta_3|D)$, if we are able to gener- ate samples from all of the conditional posterior distributions, $p(\theta_1|\theta_2, \theta_3, D)$, $p(\theta_2|\theta_1, \theta_3, D)$, and $p(\theta_3|\theta_1, \theta_3, D))$.

Here, doing formal analysis to determine the conditional posterior distributions can be difficult or impossible.

The procedure for Gibbs sampling is a type of random walk through parameter space, like the Metropolis algorithm. The walk starts at some arbitrary point, and at each point in the walk, the next step depends only on the current position, and on no previous positions.

The plot below can show the two steps of Gibbs sampling:

1. The top panel shows a random generation of a value for θ_1, conditional on a value for θ_2. The heavy lines show a slice through the posterior at the conditional value of θ_2, and the large 2 dot shows a random value of θ_1 sampled from the conditional density.
2. The bottom panel shows a random generation of a value for θ_2, conditional on the value for θ_1 de- termined by the previous step. The heavy lines show a slice through the posterior at the conditional value of θ_1, and the large dot shows a random value of θ_2 sampled from the conditional density.

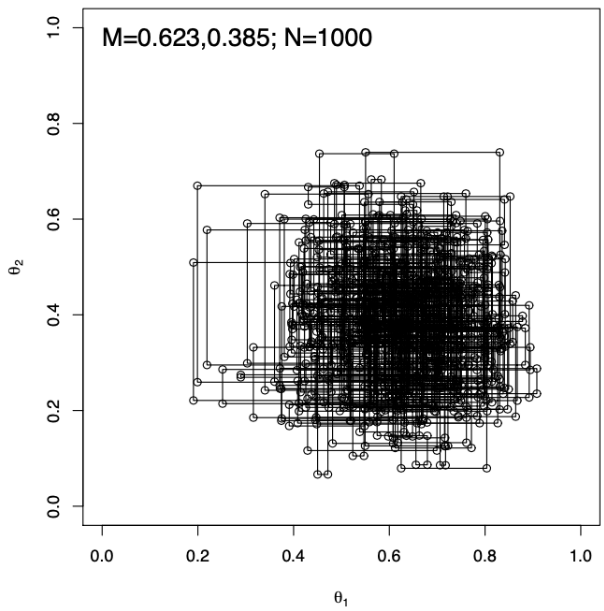
To accomplish Gibbs sampling, we must determine the conditional posterior distribution for each parameter, by definition of conditional probability:

$$\begin{aligned} p(\theta_1|\theta_2, D) &= p(\theta_1, \theta_2|D)/p(\theta_2|D) \\ &= p(\theta_1, \theta_2|D) \bigg/ \int d\theta_1\, p(\theta_1, \theta_2|D) \end{aligned}$$

In our current application, the joint posterior is a product of beta distributions, thus

$$
\begin{aligned}
p(\theta_1|\theta_2, D) &= p(\theta_1, \theta_2|D) \Big/ \int d\theta_1 \; p(\theta_1, \theta_2|D) \\
&= \frac{\text{beta}(\theta_1|z_1+a_1, N_1-z_1+b_1)\,\text{beta}(\theta_2|z_2+a_2, N_2-z_2+b_2)}{\int d\theta_1 \,\text{beta}(\theta_1|z_1+a_1, N_1-z_1+b_1)\,\text{beta}(\theta_2|z_2+a_2, N_2-z_2+b_2)} \\
&= \frac{\text{beta}(\theta_1|z_1+a_1, N_1-z_1+b_1)\,\text{beta}(\theta_2|z_2+a_2, N_2-z_2+b_2)}{\text{beta}(\theta_2|z_2+a_2, N_2-z_2+b_2)} \\
&= \text{beta}(\theta_1|z_1+a_1, N_1-z_1+b_1)
\end{aligned}
$$

We can plot it as follows:



8.4.2.1 Disadvantages of Gibbs Sampling

Here are some disadvantages of Gibbs Sampling:

1. We must be able to derive the conditional probabilities of each parameter on the other, and be able to generate samples from those conditional probabilities.
2. Because it only changes one parameter value at a time, its progress can be stalled by highly correlated parameters.