7.1 A Simple Case of the Metropolis Algorithm

Zongyi Liu

2023-06-15

There are cases that our prior beliefs about θ can not be adequately represented by a beta distribution, or by any function that yields an analytically solvable posterior function.

Moreover, the model typically has more than one parameters; for example, if we set up a grid on each parameter that has 1,000 values, then the six dimensional parameter space would have $1,000^6$ combinations of parameter values.

This chapter mainly deal with those problem.

The method described in this chapter assumes that the prior distribution is specified by a function that is easily evaluated. This simply means that if you specify a value for θ, then the value of $p(\theta)$ is easily determined

7.1 A Simple Case of the Metropolis Algorithm

Our goal in Bayesian inference is to get a good handle on the posterior distribution over the parameters. One way to do that is to sample a large number of representative points from the posterior, and then, from those points, compute descriptive statistics about the distribution.

But suppose we did not know the analytical formulas for the mean and standard deviation, and we did not have a direct way to calculate cumulative probabilities.

The question that we are going to solve is how can we sample a large number of representative values from a distribution.

7.1.1 A Politician Stumbles Upon the Metropolis Algorithm

This is a case where a politician lives on a chain of islands, and he is constantly traveling from island to island. He has three choices:

1. Stay on the current island
2. Move to the adjacent island to the west
3. Move to the adjacent island to the east

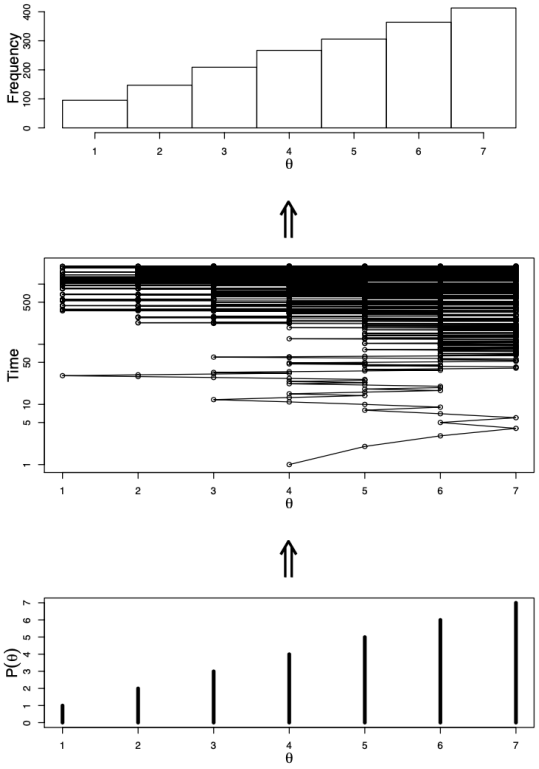
His goal is to visit all the islands proportionally to their relative population, and he has a fair coin to flip to help decide whether to propose the adjacent island to the east or the adjacent island to the west. If the proposed island has a larger population than the current island, then he definitely goes to the proposed island.

However, in the long run, the probability that the politician is on any one of the islands in the chain exactly matches the relative population of the island.

7.1.2 A Random Walk

Suppose there are seven islands with populations as plotted below. If the $\theta_{current} = 4$, then the coin would propose to move to right, $\theta_{proposed} = 5$. Because the relative population at the proposed position is greater than the relative population at the current position, the proposed move is accepted.

In the long run, the trajectory would be like the middle one in plots, and frequency would converge to the real population density in the upper part of the plots:



7.1.3 General Properties of a Random Walk

The graphs below show the probability that the moving politician is at each value of θ, at any given time step (t), the politician is at just one particular position. To approximate the target distribution, we let the politician meander around for many time steps while we keep track of where he has been.

After proposing to move, we then decide whether or not to accept it, we would like to move based on the probability $p_{move} = P(\theta_{proposed})/P(\theta_{current})$, where $P(\theta)$ is the value of the target distribution at θ, then we could get:

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

When the proposed probability is larger than the current probability, $p_{move} = 1$.

7.1.4 Why We Care

Here are points to notice when doing the random-walk process

1. We must be able to generate a random value from the proposal distribution (to create $\theta_{proposed}$).
2. We must be able to evaluate the target distribution at any proposed position (to compute $P(\theta_{proposed})/P(\theta_{current})$).
3. We must be able to generate a random value from a uniform distribution (to make a move according to p_{move}).

By doing those three things, we are able to do indirectly something we could not necessarily do directly: We can generate random samples from the target distribution.

By using MCMC techniques, we can do Bayesian inference in rich and complex models.

7.1.5 Why It Works

This section helps to explain why the algorithm works. We'll see that the relative transition probabilities, between adjacent positions, exactly match the relative values of the target distribution. Extrapolate that result across all the positions, and you can see that, in the long run, each position will be visited proportionally to its target value.

The ratio of transition probability is

$$\frac{p(\theta \rightarrow \theta+1)}{p(\theta+1 \rightarrow \theta)} = \frac{.5 \min\left(P(\theta+1)/P(\theta), 1\right)}{.5 \min\left(P(\theta)/P(\theta+1), 1\right)}$$

$$= \begin{cases} \frac{1}{P(\theta)/P(\theta+1)} & \text{if } P(\theta+1) > P(\theta) \\ \frac{P(\theta+1)/P(\theta)}{1} & \text{if } P(\theta+1) < P(\theta) \end{cases}$$

$$= \frac{P(\theta+1)}{P(\theta)}$$

This equation tells us that during transitions back and forth between adjacent positions, the relative probability of the transitions exactly matches the relative values of the target distribution.

In the long run, adjacent positions will be visited proportionally to their relative values in the target distribution.

Consider the probability of transitioning from position θ to some other position. The proposal distribution, in the present simple scenario, considers only positions $\theta+1$ and $\theta-1$. If the proposed position is not accepted, we stay at the current position θ. The probability of moving to position $\theta-1$ is the probability of proposing that position times the prob- ability of accepting the move if it is proposed: $.5min(P(\theta-1)/P(\theta), 1)$. The probability of moving to position $\theta+1$ is the probability of proposing that position times the probability of accepting the move if it is proposed: $.5min(P(\theta+1)/P(\theta), 1)$. The probability of staying at position θ is simply the complement of those two move-away probabilities: $.5[1 - min(P(\theta-1)/P(\theta), 1)] + .5[1 - min(P(\theta+1)/P(\theta), 1)]$.

We can put it into the matrix:

$$\begin{bmatrix} \ddots & p(\theta-2 \to \theta-1) & 0 & 0 & 0 \\ \ddots & p(\theta-1 \to \theta-1) & p(\theta-1 \to \theta) & 0 & 0 \\ 0 & p(\theta \to \theta-1) & p(\theta \to \theta) & p(\theta \to \theta+1) & 0 \\ 0 & 0 & p(\theta+1 \to \theta) & p(\theta+1 \to \theta+1) & \ddots \\ 0 & 0 & 0 & p(\theta+2 \to \theta+1) & \ddots \end{bmatrix}$$

which equals

$$\begin{bmatrix} \ddots & .5\min\left(\frac{P(\theta-1)}{P(\theta-2)}, 1\right) & 0 & 0 & 0 \\ \ddots & \begin{matrix}.5\left[1-\min\left(\frac{P(\theta-2)}{P(\theta-1)}, 1\right)\right] \\ +.5\left[1-\min\left(\frac{P(\theta)}{P(\theta-1)}, 1\right)\right]\end{matrix} & .5\min\left(\frac{P(\theta)}{P(\theta-1)}, 1\right) & 0 & 0 \\ 0 & .5\min\left(\frac{P(\theta-1)}{P(\theta)}, 1\right) & \begin{matrix}.5\left[1-\min\left(\frac{P(\theta-1)}{P(\theta)}, 1\right)\right] \\ +.5\left[1-\min\left(\frac{P(\theta+1)}{P(\theta)}, 1\right)\right]\end{matrix} & .5\min\left(\frac{P(\theta+1)}{P(\theta)}, 1\right) & 0 \\ 0 & 0 & .5\min\left(\frac{P(\theta)}{P(\theta+1)}, 1\right) & \begin{matrix}.5\left[1-\min\left(\frac{P(\theta)}{P(\theta+1)}, 1\right)\right] \\ +.5\left[1-\min\left(\frac{P(\theta+2)}{P(\theta+1)}, 1\right)\right]\end{matrix} & \ddots \\ 0 & 0 & 0 & .5\min\left(\frac{P(\theta+1)}{P(\theta+2)}, 1\right) & \ddots \end{bmatrix}$$

Consider a matrix T, the value in its r^{th} row and c^{th} column is denoted T_{rc}, we can then multiply the matrix on its left side by a row vector w, which yields another row vector. The c^{th} component of the product wT is $\sum_r w_r T_{rc}$.

Matrix multiplication is a very useful procedure for keeping track of position probabilities. We can see that the θ component of wT is

$$\begin{aligned} \sum_r w_r T_{r\theta} = {} & P(\theta-1)/Z \times .5\min\left(\frac{P(\theta)}{P(\theta-1)}, 1\right) \\ & + P(\theta)/Z \times \left(.5\left[1-\min\left(\frac{P(\theta-1)}{P(\theta)}, 1\right)\right] + .5\left[1-\min\left(\frac{P(\theta+1)}{P(\theta)}, 1\right)\right]\right) \\ & + P(\theta+1)/Z \times .5\min\left(\frac{P(\theta)}{P(\theta+1)}, 1\right) \end{aligned}$$

To simplify that equation, we can consider four cases,

1. $P(\theta) > P(\theta-1)$ and $P(\theta) > P(\theta-1)$
2. $P(\theta) > P(\theta-1)$ and $P(\theta) < P(\theta-1)$
3. $P(\theta) < P(\theta-1)$ and $P(\theta) > P(\theta-1)$
4. $P(\theta) < P(\theta-1)$ and $P(\theta) < P(\theta-1)$

Then the equation above would become

$$
\begin{aligned}
\sum_r w_r T_{r\theta} &= P(\theta-1)/Z \times .5 \\
&\quad + P(\theta)/Z \times \left(.5\left[1-\left(\frac{P(\theta-1)}{P(\theta)}\right)\right] + .5\left[1-\left(\frac{P(\theta+1)}{P(\theta)}\right)\right]\right) \\
&\quad + P(\theta+1)/Z \times .5 \\
&= .5\,P(\theta-1)/Z \\
&\quad + .5P(\theta)/Z - .5P(\theta)/Z\frac{P(\theta-1)}{P(\theta)} + .5P(\theta)/Z - .5P(\theta)/Z\frac{P(\theta+1)}{P(\theta)} \\
&\quad + .5\,P(\theta+1)/Z \\
&= P(\theta)/Z
\end{aligned}
$$

We have shown that the target distribution is stable under the Metropolis algorithm, for our special case of island hopping. To prove that the Metropolis algorithm realizes the target distribution, we would need to show that the process actually gets us to the target distribution regardless of where we start.