Monte Carlo Methods — Why Do They Work? — Part III

In the previous two "Why Do They Work?" write-ups, I proved that the Metropolis algorithm worked and that the Metropolis-Hastings algorithm worked:

* https://brianhill.github.io/bayesian-statistics/resources/MonteCarloMethodsWhyDoTheyWork-II.nb.pdf
* https://brianhill.github.io/bayesian-statistics/resources/MonteCarloMethodsWhyDoTheyWork-III.nb.pdf

Both proofs required the Principle of Detailed Balance. Now we are going to prove that the Gibbs Sampling (GS) algorithm works, but all the hard work was actually in the second write-up, because Gibbs sampling is a special case of Metropolis-Hastings (and Metropolis is a special case of Metropolis-Hastings too).

To get going, and to leverage our previous write-up, we need to understand the relationship between Metropolis-Hastings and Gibbs sampling.

The Metropolis-Hastings and Gibbs Sampling Algorithms

The core of the Metropolis-Hastings algorithm (repeated *ad nauseam)* was:

Step 1: You are in bin i. You propose a random move from bin i to some other bin j. The probability of proposing the random move is denoted $g(j \mid i)$, which is read "g of j given i." (Note: If you want to recover Metropolis as a special case of Metropolis-Hastings, you put in that $g(i + 1 \mid i) = 0.5$ and $g(i - 1 \mid i) = 0.5$. In other words, you have 50% chance of moving to either of the nearest neighbors, and in Metropolis there is no chance of moving to anything but a nearest neighbor.)

Step 2: Compute the ***clamped*** appropriate ratio. The **clamped** appropriate ratio for going from $i \to j$ is $\min\left(\frac{p_j}{p_i}\ \frac{g(i \mid j)}{g(j \mid i)}, 1\right)$.

Step 3: Generate a random number between 0 and 1. If the number is less than clamped appropriate ratio, move to the proposed bin, and make a tally there. Otherwise stay in the current bin and make another tally in the current bin. Whatever bin you made the tally in is now the current bin, and you go back to step 1.

"Hold on to your papers, fellow scholars [apologies to Dr. Karoly Zsolnai-Feher]," here is what Gibbs sampling is:

You just choose $g(i \mid j) = p_i$.

Really, you have got to be kidding me. That is laughable and ridiculous. I'll get to why if you don't already see why. Anyway, let's see what happens to the clamped appropriate ratio:

We put $g(i \mid j) = p_i$ and $g(j \mid i) = p_j$ into the clamped appropriate ratio and you get:

$$\min\left(\frac{p_j}{p_i}\,\frac{g(i\,|\,j)}{g(j\,|\,i)},\,1\right) = \min\left(\frac{p_j}{p_i}\,\frac{p_i}{p_j},\,1\right) = \min(1,\,1) = 1$$

The clamped appropriate ratio is always 1! So you always move. So let us summarize the Gibbs sampling algorithm (to be repeated *ad nauseam):*

Step 1: The current bin is bin *i*. You propose a random move from bin *i* to some other bin, bin *j*. The probability of the proposed random move to bin *j* is just p_j, where p_j is the probability distribution you are trying to sample.

Step 2: You accept the proposed move and make a tally in the proposed bin. The proposed bin is now the current bin, and you go back to step 1.

Now I'll say why it laughable and ridiculous:

The whole point of any Monte Carlo method is to generate a representative set of samples ***for a probability distribution that is hard to sample.*** The Gibbs sampling algorithm says to ***make the proposed move by drawing from that very same probability distribution*** that you already had resigned yourself as being hard to sample. So it has "solved" the problem by requiring you to know the solution to the problem. ***This appears to be a circular solution with no benefit.***

The actual genius of the Gibbs sampling algorithm comes next.

High-Dimensional Probability Distributions

Back on Nov. 22, I launched the entire section on Monte Carlo Methods with my "Monte Carlo Methods Introduction" write-up:

* https://brianhill.github.io/bayesian-statistics/resources/MonteCarloMethodsIntroduction.nb.pdf

My example was a probability distribution that has 50 dimensions! It was the electoral college outcome. Now each state can only go one of two ways, so each axis was as simple as an axis can get. But still the total number of outcomes was 2^{50}. We did the math and saw that there is absolutely no way to exhaustively sample that space with any reasonable amount of computer time.

As a second example, I looked ahead to what will be our final example, which was a study of 106 babies vaccinated for Hepatitis B. For each baby, there are multiple blood samples, and the hepatitis "titre" is measured in each blood sample. The blood sample data is going to be fitted by a slope and

an intercept for each baby (in the study, they actually take the logarithm of the titre before doing the linear fit). So that is 212 parameters. There is no way you conclude an epidemiological paper by reporting 212 different numbers and error bars on each of them, so the authors introduced four more parameters that capture the distribution of those 212 parameters. So in total there are 216 parameters. The bottom line is that in this second example, you have a space with 216 parameters (instead of an electoral college with 50 states), and these are now continuous parameters (not just binary wins and losses). You can see that this is going to be a far worse probability distribution to sample than the electoral college problem, and that one was already practically impossible to exhaustively simple.

To summarize, we are motivated to study high-dimensional probability distributions with Monte Carlo methods because (a) they show up in interesting, real-world examples, and (b) they are impossible to exhaustively sample.

To study high-dimensional probability distributions, we need some new notation. Instead of bins labeled by an index i, the bins will now be labeled by d indices, i_1, i_2, i_3, ..., i_d. So the probability distribution we are trying to sample is now denoted:

$$p_{i_1\, i_2\, i_3 \ldots\, i_d}$$

The simplest case that has more than one-dimension would have $d = 2$, and the p's would look like:

$$p_{i_1\, i_2}$$

To make that even more concrete, we could have a really simple example in mind. How about the first axis represents whether a nodule that comes out of the potato harvester and is destined for the kitchen is a rock or a potato, and the second axis represents whether it is small, medium, or large. Then the six p's would be:

$p_{\text{rock small}}$
$p_{\text{rock medium}}$
$p_{\text{rock large}}$
$p_{\text{potato small}}$
$p_{\text{potato medium}}$
$p_{\text{potato large}}$

Of course, the total of all 6 p's has to add up to 1 (assuming there aren't other things in the sack that goes to the kitchen, like rotten turnips).

Anyway, examples aside, it is for multi-dimensional probability distributions that Gibbs sampling has one more twist, why it is not just a circular solution with no benefit, and why it is now in widespread use for sampling high-dimensional probability distributions.

Gibbs Sampling of High-Dimensional Probability Distributions

I HAVE LOTS MORE TO WRITE, BUT YOU CAN ALREADY SEE WHERE THIS IS GOING:

We are going to (a) move along only one axis at a time in the Gibbs sampling, and (b) assume that a substantial burden of calculating $p_{i_1 i_2 i_3 \ldots i_d}$ is simplified by the fact that along at least some of the axes, the p's factor.