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

THIS DOCUMENT IS NOT DONE

In the previous two "Why Do They Work?" write-ups, I showed why the Metropolis algorithm worked and why the Metropolis-Hastings algorithm works. Both proofs required the Principle of Detailed Balance.

Now we are going to see why the Gibbs Sampling (GS) algorithm works.

The Metropolis-Hastings and Gibbs Sampling Algorithms

The 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.

"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:

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

The clamped appropriate ratio is always 1! So you always move. So let us summarize the Gibbs sampling (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 ***from a probability distribution that is hard to sample.*** The Gibbs sampling algorithm says to ***make the proposed move using 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" 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 is 2^{50}. There is absolutely no way to exhaustively sample that space.

As a second example, I looked ahead to what will be our final example, which is 106 babies vaccinated for Hepatitis B. For each baby, there are multiple blood samples, and the hepatitis "titre" is measured in each blood sample, and there is going to be a slope and an an intercept for each baby. So that is 212 parameters. There is no way that that is of interest to report, so there are four more parameters that capture the distribution of those 212 parameters, so in total there are 216 parameters. You have a space with 216 parameters (instead of an electoral college with 50 states), and these are now continuous parameters, not yes or no outcomes.

The bottom line was that high-dimensional probability distributions are (a) easy to come up with interesting, real-world examples, and (b) impossible to exhaustively sample.

So we need some notation for a high dimensional probability distribution. How about this. 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:

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

It is for high-dimensional probability distributions that Gibbs sampling has one more twist, and why it is not just a circular solution with no benefit.

Gibbs Sampling of High-Dimensional Probability Distributions

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