Bayes class 5

Sean van der Merwe

# Multivariate simulation

In this class we are going to look at issues that come up when doing multivariate simulation.

At all times we must keep in mind the goal of stochastic simulation:

To draw a representative sample from the distribution of interest

Just like with real experimental samples, the more complex the distribution we are drawing from the more samples are required in order to represent the full picture of the object of interest.

# Exploring the mountain

Use the following code in RStudio to follow along more easily with the explanations to come:

plotly::plot\_ly(z=~volcano) |> plotly::add\_surface()

In Bayes, the most common target of simulation is the posterior distribution of the parameters of a model. Given a decent sample size, this distribution will resemble a mountain (formally it approaches a multivariate normal as $n\rightarrow \infty $). However, for a finite sample size, the mountain will only look smooth from a distance and have all sorts of interesting features up close.

The goal is to explore the mountain and understand all its quirks, including all the rubble at the bottom.

# Univariate simulation first

We have already discussed univariate simulation, here is a recap of some key points:

1. Computers simulate pseudo random uniform(0,1) numbers, as many as we need.
2. These numbers have the full appearance of independence – i.i.d.
3. Given uniform numbers we can get to any univariate distribution.
	1. For all distributions which have a closed form inverse CDF, we simply plug the uniform values into that inverse CDF.
	2. For other standard distributions there are tricks and approaches that produce neat and accurate simulations quickly, usually using simple transformations.
	3. For non-standard distributions there are various useful algorithms, such as sampling-importance resampling, acceptance-rejection, and discretisation.
		1. Discretisation is the simplest approach and always produces simulations quickly, but can be very inaccurate in some cases.

However, multivariate distributions don’t have an inverse CDF in the traditional sense, which means that we need entirely new approaches.

## Why do multivariate distributions not have inverse CDFs?

# Chains

With multivariate distributions in general there are no reliable approaches to produce i.i.d. samples, but this doesn’t matter. We don’t need i.i.d. samples, we need representative samples. A painter painting a mountain doesn’t generally use independent pencil dots, they use brush strokes, and yet they still capture the essence of the mountain in their painting. Thus, attention shifts to approaches of simulation that might not be independent but still able to correctly cover the target distribution.

Modern simulation approaches simulate chains with varying levels of autocorrelation, where each value simulated depends on 1 or more previous values, directly or indirectly.

Why does extreme autocorrelation pose a challenge when trying to draw a representative sample?

In extreme cases of autocorrelation $\left(ρ>0.99\right)$ we can discard every 2nd simulated vector and still retain 99% of the information. This process is called thinning and is most often required when simulating complex models via Gibbs sampling (as used by JAGS and BUGS). Stan does **not** require thinning.

## Burn in, warm up, and initial values

Simulating via chains requires that each chain be started at a valid point in the distribution. These points can sometimes be guessed by the software, but not always. Starting at a point too far away from the center of the target distribution will result in a complete failure to converge to the target distribution since you will be in a region where the distribution is almost flat. In short, you can’t explore a mountain if you start so far away that you can’t even see the mountain. Sometimes simpler parameter estimation approaches, like the method of moments or OLS regression, can help generate reasonable starting values.

Regardless, the simulation approach needs time to explore the distribution outline before simulating for real, and we don’t want the exploration to mess with the output, so we throw it away afterwards. When exploring a mountain, it is a good idea to first size it up and prepare properly, but the preparation time is not truly part of the journey.

## Convergence

Remember that in Bayes we are not interested in convergence in the mathematical sense (unless you just want the posterior mode). Instead, we are trying to get a picture of the whole distribution. Thus, we want to see our simulations converge in distribution.

We must check for two things: within-chain convergence and between-chain convergence.

Given a trace plot showing how the simulations move around, how would you assess within-chain convergence?

And what about between-chain convergence?

## Other checks

If you go a step further and simulate from the posterior predictive distribution then you can do many more checks, can you suggest any?

## Example

Ask the lecturer to show an example of a joint distribution, and how simulations approximate it.

Use the back of this page to make notes.