Adaptive Rejection Sampling

Katherine Kempfert, Eric Chu, Jiahui Zhao

December 18, 2020

Introduction

Adaptive rejection sampling is a method for efficiently sampling from univariate probability density function which is log-concave. It is particularly useful when the distribution interested is computationally expensive. In this project, our group implemented the Adaptive Rejection Sampler (ars) based on algorithms discussed in the paper *Adaptive Rejection Sampling for Gibbs Sampling* by W. R. GILKS. More details about the algorithm is shown in Methods section.

Methods

To make sure tangent lines can be used as upper bounds, our function `ars` assumes the input density is log concave, that is, $h(x) = log(g(x))$ is a concave function. After initializing valid x abscissas, we calculate the intersections of tangent lines using

$$z_j = \frac{h(x_{j+1}) - h(x_j) - x_{j+1}h'(x_{j+1}) + x_j h'(x_j)}{h'(x_j) - h'(x_{j+1})}$$

For $z \in [z_{j-1}, z_j]$ and $j = 1, 2, ..., k$, the upper bound is defined and calculated as

$$u_k(x) = h(x_j) + (x - x_j)h'(x_j)$$

The sampling density $s_k(x)$, which we will use to draw samples from is

$$s_k(x) = \frac{\exp u_k(x)}{\int_D \exp u_k(x')dx'}$$

Observations will be sampled as follows. First, we find the interval to which x will be sampled from by selecting one of the piece of the piece-wise exponential density curves, which have been normalized using the denominator in above function $s_k(x)$. Then we randomly generate a value u_1 from $Uniform(0, 1)$ distribution, and find the largest interval index, i, such that the total integral value from the lowest interval of x to the upper bound of that interval is smaller than u_1.

We then use the Inverse CDF method to actually draw x^* within the $i_t h$ interval.

The CDF, given that x belongs to a particular interval, is computed as

$$S(x) = P(X \leq x | x \in [z_{j-1}, z_j]) = \frac{\int_{z_{j-1}}^{x} \exp u_k(x')dx'}{\int_{z_{j-1}}^{z_j} \exp u_k(x')dx'}$$

$S(x)$ is a value between 0 and 1. The denomination of $S(x)$ is a normalizing constant and can be denoted as C.

$$C = \int_{z_{j-1}}^{z_j} \exp u_k(x')dx'$$

The numerator can be integrated as follows,

$$\begin{aligned}
\int_{z_{j-1}}^{x} \exp u_k(z)dz &= \int_{z_{j-1}}^{x} \exp(h(x_j) + (z - x_j)h'(x_j))dz \\
&= \exp(h(x_j) - x_j h'(x_j)) \int_{z_{j-1}}^{x} \exp(zh'(x_j))dz \\
&= \exp(h(x_j) - x_j h'(x_j)) \cdot \frac{\exp(zh'(x_j))}{h'(x_j)}\Big|_{z_{j-1}}^{x} \\
&= \frac{\exp(h(x_j) - x_j h'(x_j))}{h'(x_j)} \left(\exp(xh'(x_j)) - \exp(z_{j-1}h'(x_j))\right).
\end{aligned}$$

Then, we randomly generate a value from $Uniform(0, 1)$ distribution, u_2.

$$\frac{\int_{z_{j-1}}^{x^*} \exp u_k(z)dz}{C} = u_2$$

and,

$$\int_{z_{j-1}}^{x^*} \exp u_k(z)dz = u_2 \times C$$

The inverse CDF can be computed as

$$\frac{e^{h(x_j) - x_j h'(x_j)}}{h'(x_j)}(e^{x^* h'(x_j)} - e^{z_{j-1}h'(x_j)}) = u_2 \times C$$

$$x^* = \frac{1}{h'(x_j)} log(\frac{u_2 \times C \times h'(x_j)}{e^{h(x_j) - x_j h'(x_j)}} + e^{z_{j-1}h'(x_j)})$$

After getting a new sample x^*, we perform the squeezing test and rejection test with a randomly generated uniform value w.

We accept x^* when either $w \leq e^{l_k(x^*) - u_k(x^*)}$ (squeezing test) or $w \leq e^{h(x^*) - u_k(x^*)}$ (rejection test) is met. However, we only include x^* in T_k to form T_{k+1} when the rejection test is evaluated.

We repeat the above algorithm until we get enough samples.

Implementation

Test

Conclusion