Adaptive Squeezed Rejection Sampling

Amy Chen

October 23, 2018

Overview

Adaptive squeezed rejection sampling is a method of drawing points from a target distribution, and goes a step further than rejection sampling by utilizing an automatic envelope generation strategy for squeezed rejection sampling.

Suppose we are interested in drawing points from $f(x)$, a concave function. Let g denote another density from which we know how to sample and for which we can easily calculate $g(x)$. Let e denote an envelope such that $e(x) = \frac{g(x)}{\alpha} \geq f(x) \forall x$ for which $f(x) > 0$ for a given constant $\alpha \leq 1$. It is simpler to generate this envelope function in the log space. Take n points on $log(f(x))$ and connect their tangent lines to determine $log(e(x))$. This ensures that when exponentiated, the envelope function encompasses $f(x)$.

We will also define a squeeze function $s(x)$ such that $s(x) \leq f(x) \forall x$ for which $f(x) > 0$. Using the selected points from generating the envelope function, connect the points to determine $log(s(x))$. This ensures that when exponentiated, the squeeze function is below $f(x)$.

Then adaptive rejection sampling can be completed in the following steps:

1. Sample $Y \sim g$
2. Sample $U \sim Unif(0, 1)$
3. If $U \leq \frac{s(Y)}{e(Y)}$, keep Y
4. If $U > \frac{s(Y)}{e(Y)}$ and $U \leq \frac{f(Y)}{e(Y)}$, keep Y
5. Otherwise, reject Y
6. Repeat for desired sample size

Demonstration

Suppose we would like to estimate $S = E[x^2]$ where X has density proportional to $q(x) = e^{\frac{-|x|^3}{3}}$

Target Function

Let the target function be $f(x) = e^{\frac{-|x|^3}{3}}$
Then $log(f(x)) = \frac{-|x|^3}{3}$

Envelope Function

Select points $(-1, -\frac{1}{3})$, $(0, 0)$, and $(1, -\frac{1}{3})$ from $log(f(x))$
By computing the tangent lines at each point, finding the points of intersection, and merging the functions, we get the log of the envelope function,

$$log(e(x)) = \begin{cases} 0 & \text{if } -\frac{2}{3} < x < \frac{2}{3} \\ \frac{2}{3} - |x| & otherwise \end{cases}$$

Exponentiate to get the envelope function,

$$e(x) = \begin{cases} 1 & \text{if } -\frac{2}{3} < x < \frac{2}{3} \\ e^{\frac{2}{3}} - |x| & otherwise \end{cases}$$

Squeezing Function

Let the log of the squeezing function be $log(s(x)) = -\frac{|x|}{3}$ if $-1 < x < 1$

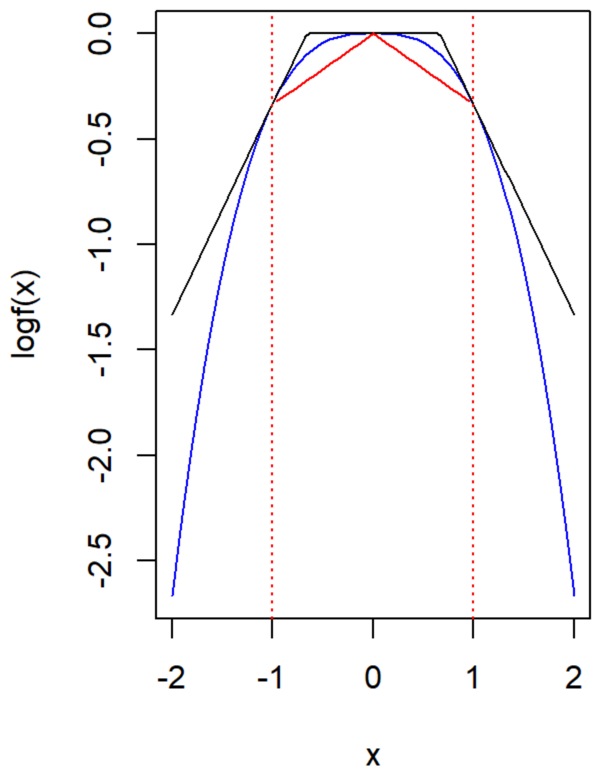
Exponentiate to get the squeezing function $s(x) = e^{-\frac{|x|}{3}}$ if $-1 < x < 1$

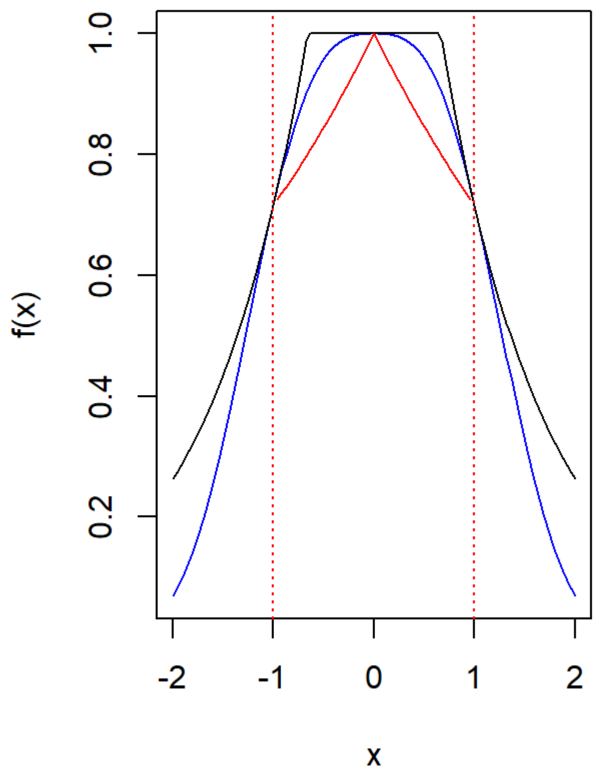
```
logf <- function(x) {
  -abs(x^3)/3
}
loge <- function(x) {
  ifelse( (x>-2/3)&(x<2/3), 0, 2/3-abs(x) );
}
logs <- function(x) {
  ifelse( (x>-1)&(x<1), -abs(x)/3, NA );
}
```

```
f <- function(x) {
  exp(logf(x))
}
e <- function(x) {
  exp(loge(x));
}
s <- function(x) {
  exp(logs(x));
}
```

```
par(mfrow=c(1,2))
curve(logf(x), from = -2, to = 2, col = "blue")
curve(loge(x), add = T)
curve(logs(x), add = T, col = "red")
abline(v = -1, lty = 3,col = "red")
abline(v = 1, lty = 3,col = "red")

curve(f(x), from = -2, to = 2, col = "blue")
curve(e(x), add = T)
curve(s(x), add = T, col = "red")
abline(v = -1, lty = 3,col = "red")
abline(v = 1, lty = 3,col = "red")
```





Finding Inverse CDF G^{-1}

$\int_{-\infty}^{\infty} e(x)\ dx = \int_{-\infty}^{-\frac{2}{3}} e^{\frac{2}{3}+x}\ dx + \int_{-\frac{2}{3}}^{\frac{2}{3}} e^{0}\ dx + \int_{\frac{2}{3}}^{\infty} e^{\frac{2}{3}-x}\ dx = \frac{10}{3}$

So the normalizing constant is $\frac{3}{10}$

Then the CDF of g is

$$G(x) = \begin{cases} \frac{3}{10}e^{\frac{2}{3}+x} & \text{if } x < -\frac{2}{3} \\ \frac{3}{10}x + \frac{1}{2} & \text{if } -\frac{2}{3} \le x \le \frac{2}{3} \\ 1 - \frac{3}{10}e^{\frac{2}{3}-x} & \text{if } x > \frac{2}{3} \end{cases}$$

Take the inverse of $G(x)$,

$$G^{-1}(u) = \begin{cases} log(\frac{10}{3}u) - \frac{2}{3} & \text{if } 0 < u < \frac{3}{10} \\ \frac{10}{3}(u - \frac{1}{2}) & -\frac{3}{10} \le u \le \frac{7}{10} \\ \frac{2}{3} - log(\frac{10}{3}(1-u)) & \frac{7}{10} < u < 1 \end{cases}$$

```
Ginv <- function(u) {
  ifelse(u<3/10, log(u*10/3)-2/3, ifelse(u>7/10, 2/3-log((1-u)*10/3),(u-1/2)*10/3));
}
```

Adaptive Rejection Sampling

Below is a function for performing rejection sampling with a sample size of n points.

```
# adaptive rejection sampling function
ars <- function(n) {
  x <- rep(NA, n);
  # number of points accepted
  ct <- 0;
  # number of points sampled
  total <- 0;
  # number of points caught by squeeze
  squeeze <- 0;

  while(ct < n) {
    y <- Ginv(runif(1));
    u <- runif(1);
    # check squeeze range
    if(y > -1 && y < 1) {
      # under squeeze
      if(u < s(y)/e(y)) {
          ct <- ct + 1;
          x[ct] <- y;
          squeeze <- squeeze + 1;
      }
      # above squeeze
      else {
        # under f
        if(u < f(y)/e(y)) {
          ct <- ct + 1;
          x[ct]<-y;
        }
      }
    }
    # outside squeeze but under f
    else if(u < f(y)/e(y)) {
      ct <- ct + 1;
      x[ct] <- y;
    }

    total <- total + 1;
  }

  list(x = x, acratio_sx = squeeze/total, acratio = ct/total);
}
```

Choose a sample size of 100,000. Below are a few points drawn using this method.

```
samp_size = 100000
set.seed(920)
ars_points <- ars(samp_size)
head(ars_points$x)
```

```
## [1]  0.2950836  1.3434853  1.1435056 -1.3004348 -0.3099414  0.1288534
```

The theoretical evelope ratio is $\frac{\int_{-\infty}^{\infty} f(x)\ dx}{\int_{-\infty}^{\infty} e(x)\ dx}$, the proportion of points in f that are in e.

```
integrate(f, lower = -Inf, upper = Inf)$value / integrate(e, lower = -Inf, upper = Inf)
$value
```

```
## [1] 0.7727395
```

For this simulation, the envelope ratio is

```
ars_points$acratio
```

```
## [1] 0.7754643
```

The theoretical squeeze ratio is $\frac{\int_{-1}^{1} s(x)\ dx}{\int_{-\infty}^{\infty} e(x)\ dx}$, the proportion of points in s that are in e.

```
integrate(s, lower = -1, upper = 1)$value / integrate(e, lower = -Inf, upper = Inf)$valu
e
```

```
## [1] 0.5102436
```

For this simulation, the squeeze ratio is

```
ars_points$acratio_sx
```

```
## [1] 0.512706
```

Calculate $E[x^2]$

Now that we have our points from the sample, square each accepted x, and take the mean to get $E[x^2]$.

```
mean(ars_points$x^2)
```

```
## [1] 0.7762001
```