Sampling Methods

Patrick Lam

Outline

Inverse CDF Method

Rejection Sampling

Outline

Inverse CDF Method

Rejection Sampling

Inverse CDF Method

Suppose we want to sample from some **continuous** distribution $f(x)$, but we only know the CDF $F(x)$ and we are unable to take derivatives.

We can sample from $f(x)$ if we can sample from a Uniform(0,1) distribution and know the **inverse CDF** $F^{-1}(u)$, where

$$\begin{aligned} F(x) &= u \\ F^{-1}(u) &= x \end{aligned}$$

Repeat the following two steps m times:

1. Draw a random value from the Uniform(0,1) distribution. Call this value u.
2. Compute $F^{-1}(u)$ to get a value x. x is a draw from $f(x)$.

An Example

Suppose our target density (the one we want to sample from) is the triangle density:

$$f(x) = \begin{cases} 8x & \text{if } 0 \leq x < 0.25 \\ \frac{8}{3} - \frac{8}{3}x & \text{if } 0.25 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Now suppose we didn't know $f(x)$, but we did know the CDF F(x):

$$F(x) = \begin{cases} 0 & \text{if } x < 0 \\ 4x^2 & \text{if } 0 \leq x < 0.25 \\ \frac{8}{3}x - \frac{4}{3}x^2 - \frac{1}{3} & \text{if } 0.25 \leq x \leq 1 \\ 1 & \text{if } x > 1 \end{cases}$$

If we stick in a value of x into $F(x)$, we get some value u in the interval [0,1] (which corresponds to $P(X \leq x)$).

Now we need to find $F^{-1}(u)$ such that if we stick in a value of u, we get the corresponding x value.

To do so, we simply set $F(x) = u$ and solve for x.

$$F^{-1}(u) = \begin{cases} \frac{\sqrt{u}}{2} & \text{if } 0 \leq u < 0.25 \\ 1 - \frac{\sqrt{3(1-u)}}{2} & \text{if } 0.25 \leq u \leq 1 \end{cases}$$

For this problem, $F^{-1}(u)$ has a restricted domain of $[0, 1]$ because there are no solutions for $u \notin [0, 1]$. Since u is drawn from the Uniform(0,1) distribution, we do not have to worry about it.

Now we can sample using the inverse cdf method.

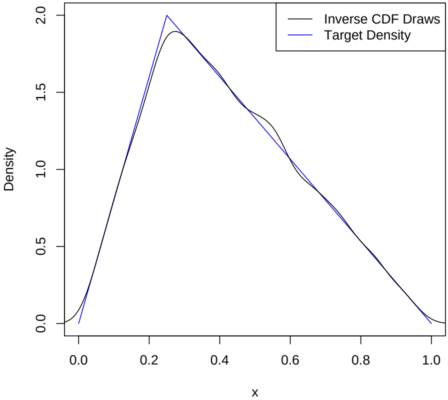
1. Draw m random values from the Uniform(0,1) distribution. Call these values $\mathbf{u}$.

```
> m <- 10000
> u <- runif(m, 0, 1)
```

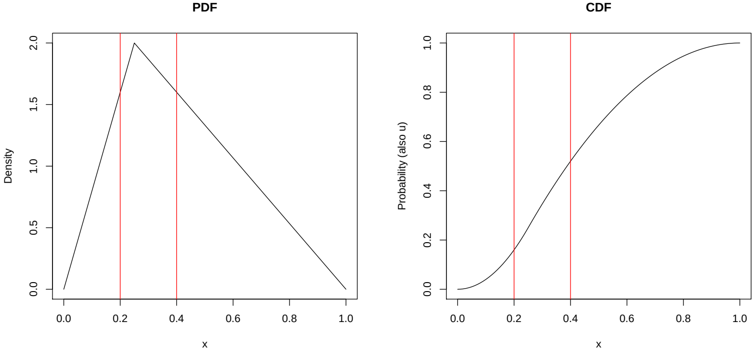
2. Compute $F^{-1}(\mathbf{u})$ to get values of $\mathbf{x}$. The values in $\mathbf{x}$ are draws from $f(x)$.

```
> invcdf.func <- function(u) {
+     if (u >= 0 && u < 0.25)
+         sqrt(u)/2
+     else if (u >= 0.25 && u <= 1)
+         1 - sqrt(3 * (1 - u))/2
+ }
> x <- unlist(lapply(u, invcdf.func))
```

We can compare the density of our draws to the target density $f(x)$.



Why Does It Work?



The areas with more density on the PDF (for example, the interval [0.2,0.4]) have a steeper "slope" on the CDF, so they cover more of the [0,1] space of u, and thus will be drawn more often.

Outline

Inverse CDF Method

Rejection Sampling

Rejection Sampling

Suppose again that we want to sample from our target density $f(x)$, which in our example is the triangle density.

For the rejection sampling, we need to pick a candidate density $g(x)$ such that $f(x) \leq Mg(x)$ for all x, where M is a constant.

Repeat the following steps until we get m accepted draws:

1. Draw a candidate x_c from $g(x)$.
2. Calculate an acceptance probability α for x_c.

$$\alpha = \frac{f(x_c)}{Mg(x_c)}$$

3. Draw a value u from the Uniform(0,1) distribution.
4. Accept x_c as a draw from $f(x)$ if $\alpha \geq u$. Otherwise, reject x_c and go back to step 1.

An Example

Target Density $f(x)$ (the triangle density):

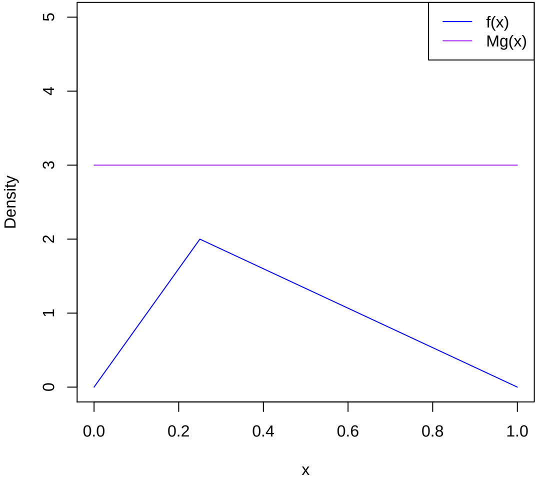
```
> f.x <- function(x) {
+     if (x >= 0 && x < 0.25)
+         8 * x
+     else if (x >= 0.25 && x <= 1)
+         8/3 - 8 * x/3
+     else 0
+ }
```

For our candidate density $g(x)$, let's use the Uniform(0,1) density:

```
> g.x <- function(x) {
+     if (x >= 0 && x <= 1)
+         1
+     else 0
+ }
```

Let's set $M = 3$ because I know from guess and check that $f(x)$ is never greater than $Mg(x)$, which is 3 for all $x \in [0, 1]$.

```
> M <- 3
```

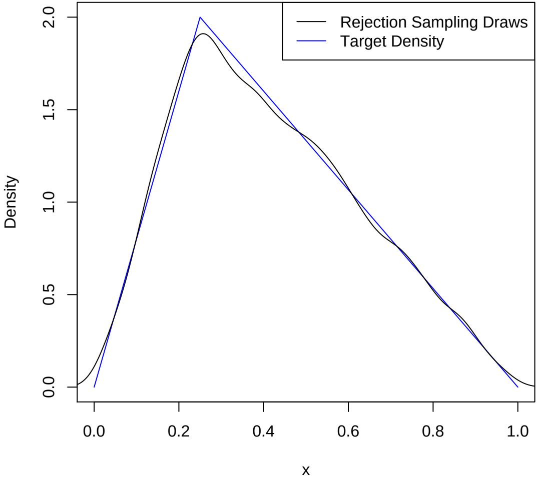


f(x)
Mg(x)
Density
0
1
2
3
4
5
0.0
0.2
0.4
0.6
0.8
1.0
x

Let's do rejection sampling!

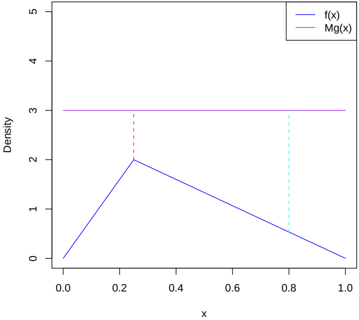
1. Draw a candidate x_c from $g(x)$.
2. Calculate an acceptance probability $\alpha = \frac{f(x_c)}{Mg(x_c)}$ for x_c.
3. Draw a value u from the Uniform(0,1) distribution.
4. Accept x_c as a draw from $f(x)$ if $\alpha \geq u$. Otherwise, reject x_c and go back to step 1.

```
> m <- 10000
> n.draws <- 0
> draws <- c()
> x.grid <- seq(0, 1, by = 0.01)
> while (n.draws < m) {
+     x.c <- runif(1, 0, 1)
+     accept.prob <- f.x(x.c)/(M * g.x(x.c))
+     u <- runif(1, 0, 1)
+     if (accept.prob >= u) {
+         draws <- c(draws, x.c)
+         n.draws <- n.draws + 1
+     }
+ }
```



Rejection Sampling Draws
Target Density
Density
x
0.0
0.5
1.0
1.5
2.0
0.0
0.2
0.4
0.6
0.8
1.0

Why Does It Work?



The difference between $f(x)$ and $Mg(x)$ at places with higher density (i.e. around $x = 0.25$) is smaller than at places with lower density (i.e. around $x = 0.8$), so the acceptance probability at $x = 0.25$ is higher and more draws of $x = 0.25$ are accepted.

There are an infinite number of candidate densities $g(x)$ and constants M that we can use.

The only difference between them is computation time.

If $g(x)$ is significantly different in shape than $f(x)$ or if $Mg(x)$ is significantly greater than $f(x)$, then more of our candidate draws will be rejected.

If $f(x) = Mg(x)$, then all our draws will be accepted.

A version of rejection sampling forms the basis for the Metropolis-Hastings algorithm that we will use later to sample from (possibly multivariate) posteriors without knowing the normalizing constant.