Monte Carlo Methods

Louie Dinh

May 2, 2016

Inversion Sampling

If we want to generate samples from a distribution where we have the inverse CDF, F^{-1}, then we can do this easily. Key idea is that if we take U ~ unif(0,1) and generate $X = F^{-1}(U), then we will have samples from the desired distribution.

Proof:

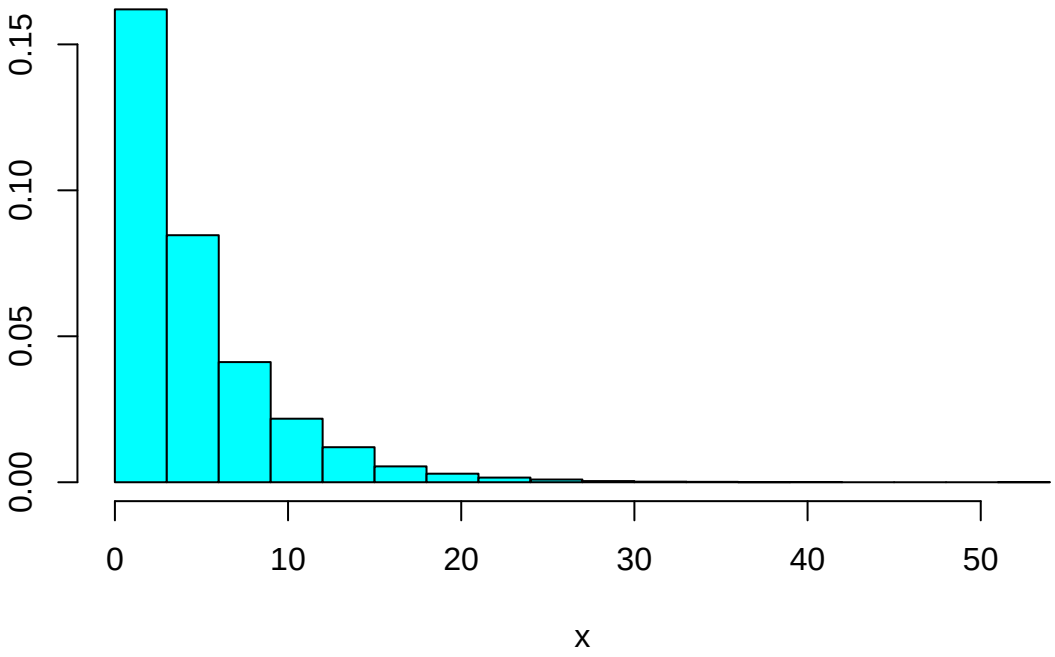
Let $X = F^{-1}(U)$.

$$F_X(x) = Pr(X <= x) = Pr(F^{-1}(U) <= x) = Pr(F(F^{-1}(U)) <= F(x)) = Pr(U <= F(x)) = F(x)$$

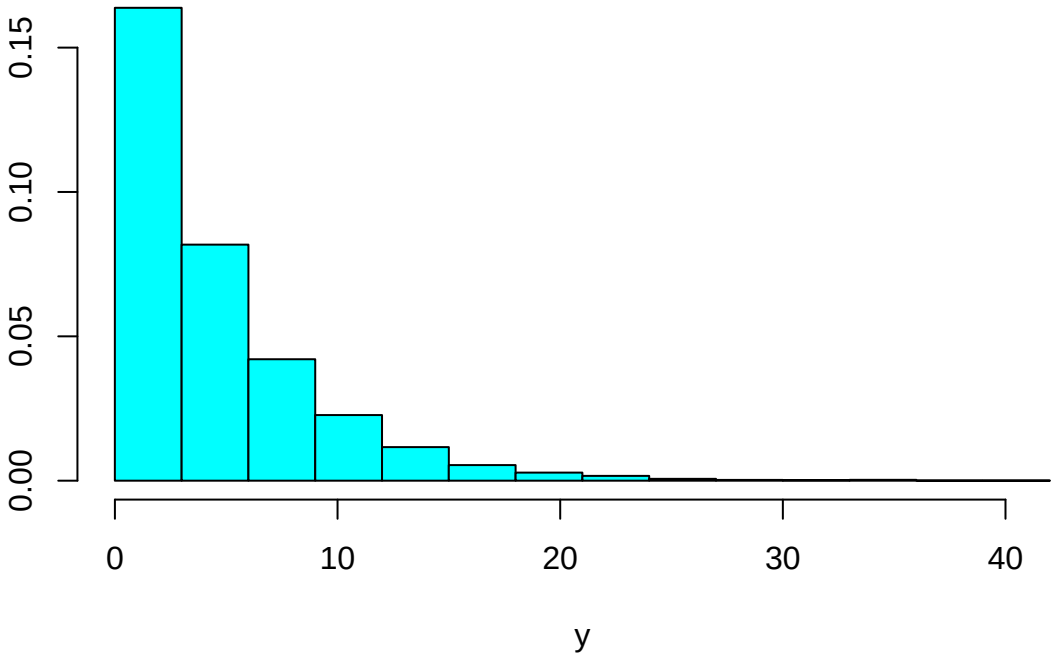
The first equality is by the definition of CDFs. The second is subtituting in X. The third is applying F to both sides. The fourth is simplification. The fifth is based on PDF of uniform.

Simulation Example:

```
# Sampled geom.
u <- runif(n=10000)
x <- qgeom(u, prob=.2)
# Reference geom.
y <- rgeom(n=10000, prob=.2)

library(MASS)
truehist(x, h=3)
```



```
truehist(y, h=3)
```



Rejection Sampling

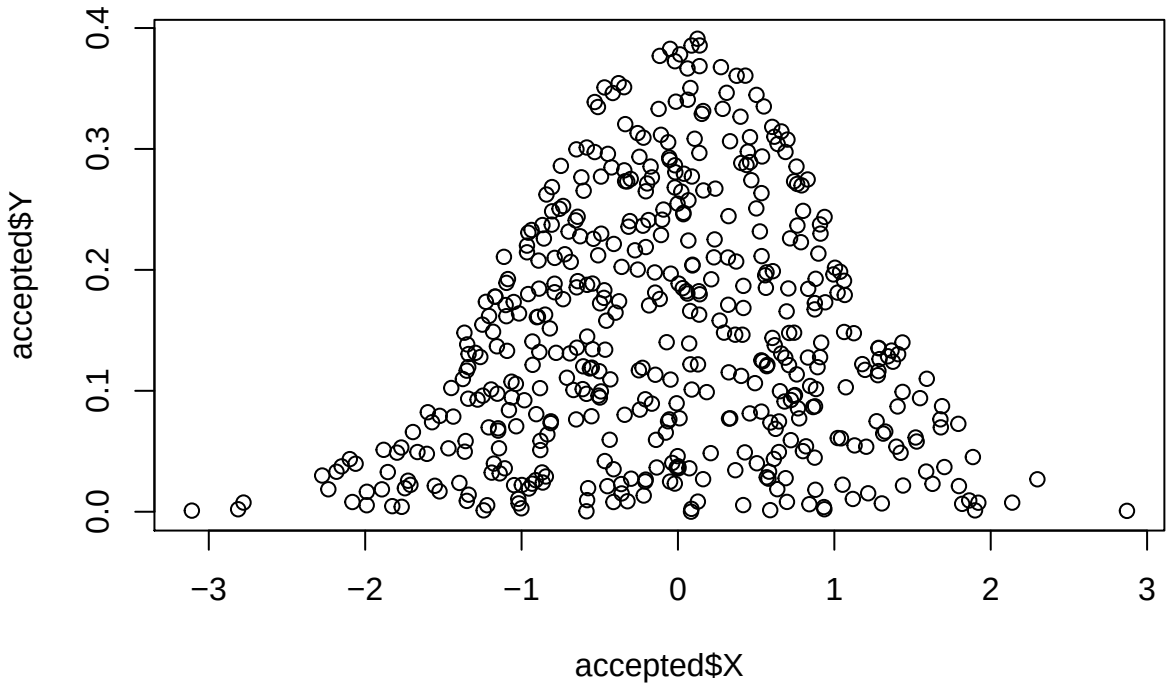
In rejection sampling, we have an un-normalized distribution function, p_hat(X) = 1/Z * p_hat(X), where Z is the appropriate normalizing constant to make the distribution sum to one.

We then need a proposed distribution q, and a scaler k such that kq is always greater than p_hat. Then we sample from q to get a number z_0. Then we sample uniformly from [0, kq(z_0)], to get a uniform distribution under the curve kq. We accept if kq(z_0) < p_hat(z_0), otherwise reject. The resulting samples will be samples from the normalized p.

Let's do something stupid. We will sample from normal using a scaled normal. We use the standard normal as the target distribution and 2 * the standard normal as our proposal distribution.

```
# Draw a sample from our proposed distribution, z.
# Draw a sample uniformly from 0 up to u = q(z).
# If u < p(z), then keep it! We have a uniform sample from under p at z.
# Otherwise reject

sample_proposed <- function() {
  rnorm(n=1)
}

proposed_density <- function(z) {
  2 * dnorm(x = z)
}

target_density <- function(z) {
  dnorm(x = z)
}
```

```
NSAMPLES = 1000
accepted <- data.frame()
rejected <- data.frame()
for(i in 1:NSAMPLES) {
  z = sample_proposed()
  u = runif(n=1, min=0, max=proposed_density(z))
  if(u <= target_density(z)){
    # Accept the sample
    accepted <- rbind(accepted, c(z, u))
  } else {
    rejected <- rbind(rejected, c(z, u))
  }
}
colnames(accepted) <- c("X", "Y")

plot(accepted$X, accepted$Y)
```




```
# Verify this is right by counting the % of samples between -2 and 2. Should be ~96%
pct_twosd <- sum(abs(accepted$X) <= 2) / nrow(accepted)
sprintf("Percentage of samples between two standard deviations of mean is %.2f", pct_twosd)
```

```
## [1] "Percentage of samples between two standard deviations of mean is 0.97"
```

Rejection sampling becomes intractable in higher dimensions corresponding to the difficulty of tightly approximating the targte distribution.

Importance Sampling

Allows one to approximate expectations of a function of z, where the probability distribution z cannot be evaluated exactly. Suppose we cannot sample easily from p(z) but we can evaluate p(z) for any z.

An easy way to approximate the expected value is to create a uniform grid, and then perform the summation

$$E[f] \ = \sum_{l=1}^{L} p(z^l) f(z^l)$$

This doesn't work well in higher dimensions because most probability distributions of interest have localized mass. The uniform grid strategy would be highly inefficient. Ideally we would want to sample from there p(z) is large, or ideally where f(z)p(z) is large.

Once again, we use proposal distribution q.

$$E[f] = \int f(z)p(z)dz = \int f(z)\frac{p(z)}{q(z)}q(z)dz = \int (f(z)\frac{p(z)}{q(z)})q(z)dz \approx \frac{1}{L}\sum_{l=1}^{L} f(z^l)\frac{p(z^l)}{q(z^l)}$$