STATS 406F15 Lab 05

1 Review of simulating Discrete random variables via inversion

Discrete random variables can be generated by slicing up the interval (0, 1) into subintervals which define a partition of (0, 1):

$$(0, F(x_1)), (F(x_1), F(x_2)), (F(x_2), F(x_3)), ..., (F(x_{k-1}), 1)$$

and generating U = Uniform(0, 1) random variables, and seeing which subinterval U falls into.

Write a function to generate n random numbers from binomial distribution with m trials and p using inversion sampling. The mass function of binomial distribution is

$$P(X = k) = \binom{m}{k} p^k (1-p)^{m-k}$$

where $k = 0, 1, \ldots, n$. We set $n = 1000$, $p = 0.2$. Plot the histograms with $m = 5$ and $m = 50$, respectively.

Solution:

```
## Function to implement the inversion sampling
## for binomial distribution
binomial <- function(n, num_trials, prop)
{
    ## Generate n uninform numbers
    z <- runif(n)
    ## Obtain P(X=0), P(X=1), ..., P(X=num_trails)
    p <- choose(num_trials, seq(0, num_trials))
    * prop^seq(0, num_trials) * (1-prop)^(num_trials-seq(0, num_trials))
    ## Intialize the binomial numbers
    x <- rep(0, n)
    for (i in seq(1, n))
    {
        s <- 0  ## Initialize the sum
```

```
        k <- -1  ## Initialize
        ## While loop
        ## Finding the smallest k such that p[1]+...p[k] > z[i]
        while (s < z[i])
        {
            k <- k + 1
            s <- s + p[k+1]
        }
        x[i] <- k
    }

    return(x)
}

## Call binomial generator
x <- binomial(n=1000, num_trials=50, prop=0.2)
hist(x)
```

2 Basic Monte Carlo Integration

2.1 Moments of functions of random variables

Example: Let U and V be independent Uniform (-1,1). Compute $P(|U+V| \leq 1)$ using Monte Carlo method. Report the monte carlo error.

Solution: $P(|U+V| \leq 1) = E\left(1(|U+V| \leq 1)\right) = \int_{-1}^{1}\int_{-1}^{1} 1(|u+v| \leq 1)dudv$. Estimate this by generating $U_1, U_2, ..., U_N$ and $V_1, V_2, ..., V_N$ from the uniform distribution on (-1,1) and calculating

$$\frac{1}{N}\sum_{i=1}^{N} 1(|U_i + V_i| \leq 1).$$

```
rm(list=ls(all=TRUE)) ## clear out old variables from the memory

## The input parameter is #samples and level of significance
integral <- function(n, alpha)
{
    u1 <- runif(n,min=-1, max =1)
    u2 <- runif(n,min=-1, max =1)
    h<- (abs(u1 + u2) <= 1)
    integral <- mean(h)
    mc_error <- sqrt(var(h) / n)
z_alphaby2 <- qnorm(1- alpha/2)
```

```
ci <- c( integral - z_alphaby2*mc_error, integral + z_alphaby2*mc_error)
    output <- list(integral=integral, mc_error=mc_error, ci = ci )
    return(output)
}

print(integral(10000,.05))
```

2.2 Calculating arbitrary integrals

The integral of an arbitrary function h over the interval (a, b) is :

$$\int_a^b h(x)dx = (b-a)\int_a^b h(x)\frac{1}{b-a}dx = (b-a)E(h(V)),$$

where V follows Uniform(a,b).

Example: Compute $\int_0^{2\pi} \sin(x\cos(x))dx$.

Solution: This is equal to $2\pi E(\sin(V\cos(V)))$ where $V \sim U(0, 2\pi)$

```
rm(list=ls(all=TRUE))

## The input parameter is #samples
integral1 <- function(n)
{
    x <- runif(n=n, min=0, max=2*pi)
    integral <- mean(2*pi*sin(x*cos(x)))
    mc_error <- sqrt(var(2*pi*sin(x*cos(x))) / n)

    output <- list(integral=integral, mc_error=mc_error)
    return(output)
}
```

Example: Consider again $h(x) = sin(xcos(x))$. Compute the integral

$$\int_{-\infty}^{\infty} h(x)dx.$$

Solution:

```
#### Computation of the second integral
## The input parameter is #samples
integral2 <- function(n)
{
    x <- rnorm(n)
```

```
    integral <- mean(sin(x*cos(x)) / dnorm(x))
    mc_error <- sqrt(var(sin(x*cos(x)) / dnorm(x)) / n)

    output <- list(integral=integral, mc_error=mc_error)
    return(output)
}

print(integral2(10000))
```

3 Rejection Sampling

Write a function to generate n random numbers from standard normal distribution using rejection sampling. The density function of standard normal is:

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$$

Using Cauchy distribution as the trial (reference) distribution:

$$g(x) = \frac{1}{\pi(1 + x^2)}$$

Solution:

Consider the ratio:

$$h(x) = \frac{f(x)}{g(x)} = \sqrt{\pi/2}(1 + x^2) \exp(-\frac{x^2}{2})$$

Setting $h'(x) = 0$, we can obtain $x = \{-1, 0, +1\}$, where $x = -1, +1$ correspond to the two peaks. Then we set $M = h(1) = \sqrt{2\pi} \exp(-1/2)$.

```
## Density function of standard normal distribution
f <- function(x)
{
    y <- 1/sqrt(2*pi) * exp(-x^2/2)
    return(y)
}

## Density function of Cauchy distribution
g <- function(x)
{
    y <- 1 / (pi * (1 + x^2))
    return(y)
}
```

```r
## Rejection sampling for standard normal distribution
rejection_sampling_standard_normal <- function(n)
{
    ## Boundary M
    M <- f(1)/g(1)

    ## Initialize the output
    x <- rep(0, n)

    k <- 0 ## Number of accepted samples
    ## While loop until there are n accepted samples
    while (k <= n)
    {
        y <- rcauchy(1)
        u <- runif(1)
        if (u*M*g(y) <= f(y))
        {
            k <- k + 1
            x[k] <- y
        }
    }

    return(x)
}

## Call function
x <- rejection_sampling_standard_normal(1000)
hist(x, breaks=20)
```