Chapter 1 - Random Number Generation

Transformation of Random Variables. Box Muller Transformation.

Prof. Alex Alvarez, Ali Raisolsadat

School of Mathematical and Computational Sciences
University of Prince Edward Island

Transformation of Random Variables

The inverse transform method is an example of how to transform a uniform random variable into a random variable with a different distribution.

This is just one example of a possible transformation of random variables.

There are more complicated transformations of random variables that could be useful in some cases.

Transformation of Random Vectors

In some cases it might be convenient to transform a whole random vector from $\mathbb{R}^d$ into a new random vector from $\mathbb{R}^d$ in order to get random numbers/vectors with a desired distribution.

Perhaps the most noticeable example of this is the **Box-Muller transform** for the generation of normal random variables.

This comes as an application of a very general result from the textbook (**Theorem 1.34**).

Box-Muller Transform Algorithm

Algorithm

1. Generate $\Theta \sim U[0, 2\pi]$ and $U \sim U[0, 1]$ independently
2. Compute $R = \sqrt{-2\ln(U)}$
3. Compute $(X, Y) = (R\cos\Theta, R\sin\Theta)$
4. Return (X, Y)

The random variables X, Y are independent, standard normal random variables.

Remarks:

- Unfortunately, as we have discussed earlier in the course (see slides corresponding to the Inverse transform method) **we cannot transform a single uniformly distributed random variable into a single normal random variable easily**
- However, with the Box-Muller algorithm **we can transform two independent, uniformly distributed random variables into two independent standard normal random variables.**
- This type of transformations (from $\mathbb{R}^d$ to $\mathbb{R}^d$) are not very common and the Box-Muller algorithm is a very notable exception.
- Some textbooks also refer to this method as the **polar method**.

Box-Muller Transform Example

R Code

```r
n <- 1000
m <- n/2
X <- vector()
R <- vector()
U <- runif(m, min=0, max =1)
Theta <-
  runif(m, min=0, max = 2*pi)

for (i in c(1:m)) {
  R[i] <- sqrt(-2*log(U[i]))
  X[2*i-1]<- R[i]*cos(Theta[i])
  X[2*i]<- R[i]*sin(Theta[i])
}

hist(X)
```

Python Code

```python
import numpy as np
import matplotlib.pyplot as plt

n = 1000
m = n // 2
X = np.zeros(n)
R = np.zeros(m)
U = np.random.uniform(0, 1, m)
Theta = np.random.uniform(0, 2*np.pi, m)

for i in range(m):
    R[i] = np.sqrt(-2 * np.log(U[i]))
    X[2*i] = R[i] * np.cos(Theta[i])
    X[2*i+1] = R[i] * np.sin(Theta[i])

plt.hist(X, bins=30)
plt.show()
```