Generating Random Variables On Computers
EECS 501, J. Fessler

Most computers have random number subroutines that can generate random variables having a Uniform[0,1] distribution. These subroutines actually produce a cyclic sequence of numbers, so in fact they are not exactly independent, and are therefore often called "pseudo-random" numbers. For short sequences of numbers one can usually safely ignore the dependencies. For long sequences however, such as when quantifying the probability of bit errors of a communication system where one might have $p \approx 10^{-11}$ for example, one must be very careful to check the cycle length of the random number generator.

This note addresses the question: how can we convert uniform random variables into a random variables having a desired distribution function $F_X(x)$? We would like to find a transformation $X = g(U)$, where $U \sim$ Uniform[0, 1].

First consider the transformation $Y = g(X)$, where $g(x) = F_X(x)$. In other words, suppose we map X through its own distribution function. For simplicity, assume that X is a continuous random variable. Then $F_X(\cdot)$ will be monotone increasing (no flat segments). Since $0 \leq F_X(x) \leq 1$, the range of Y is clearly $[0, 1]$. Following the usual derivation, for $y \in [0, 1]$ we have

$$\begin{aligned}
F_Y(y) &= P[Y \leq y] && \text{definition of } F_Y \\
&= P[F_X(X) \leq y] && \text{since } Y = g(X) \text{ and } g(x) = F_X(x) \\
&= P[X \leq F_X^{-1}(y)] && \text{monotonicity of } F_X(\cdot) \\
&= F_X(F_X^{-1}(y)) && \text{definition of } F_X \\
&= y && \text{monotonicity of } F_X(\cdot).
\end{aligned}$$

Note that if $F_Y(y) = y$ for $y \in [0, 1]$, then Y has a Uniform[0,1] distribution. Thus mapping a random variable through its own distribution function $F_X(\cdot)$ yields a random variable with a uniform distribution on $[0, 1]$.

It turns out that the reverse is true as well: if we map a uniform random variable "backwards" through the inverse of a desired distribution function, i.e. $X = F_X^{-1}(U)$, then we generate a random variable having that desired distribution. Again, we assume that F_X is monotone increasing. Then

$$\begin{aligned}
F_X(x) &= P[X \leq x] && \text{definition of } F_X \\
&= P[F_X^{-1}(U) \leq x] && \text{since } X = g(U) \text{ and } g(u) = F_X^{-1}(u) \\
&= P[U \leq F_X(x)] && \text{monotonicity of } F_X(\cdot) \\
&= F_U(F_X(x)) && \text{definition of } F_U \\
&= F_X(x) && \text{since } F_U(u) = u \text{ for } u \in [0, 1] \text{ assuming } U \text{ is uniform R.V.}
\end{aligned}$$

Exercise: *modify the above derivation to accommodate arbitrary random variables, rather than just continuous random variables.*

But what happens if F_X or F_X^{-1} is a complicated function, or has no closed form expression (such as is the case for the Gaussian distribution)? One approach is the "rejection method," described in many texts, e.g. p. 158 of Leon-Garcia: "Probability and random processes for electrical engineering." Fortunately, for the Gaussian case there is another approach (p. 254 of Leon-Garcia):

- Generate U_1 and U_2, two independent random variables Uniform[0,1].
- Let $R = \sqrt{-2 \log U_1}$ and $\Theta = 2\pi U_2$.
- Let $X = R \cos \Theta$ and $Y = R \sin \Theta$.

Then using the Jacobian technique in Section 3.4 of Stark and Woods, one can show that X and Y are independent Gaussian random variables with mean 0 and variance 1. Thus one can generate Gaussian random variables from uniform random variables fairly efficiently.