Chapter 1 - Random Number Generation

The Inverse Transform Method

Prof. Alex Alvarez, Ali Raisolsadat

School of Mathematical and Computational Sciences
University of Prince Edward Island

Inverse Transform Method

General Quantile Function: Let F be a cumulative distribution function(c.d.f.). Then the inverse of F is defined as

$$F^{-1}(u) = \inf \{x \in \mathbb{R} | F(x) \geq u\}$$

Inverse Transform Method

General Quantile Function: Let F be a cumulative distribution function(c.d.f.). Then the inverse of F is defined as

$$F^{-1}(u) = \inf \{x \in \mathbb{R} | F(x) \geq u\}$$

Theorem: Let $F : \mathbb{R} \to [0, 1]$ be a c.d.f. and F^{-1} its inverse. If $U \sim \text{uniform}[0, 1]$ and we define $X = F^{-1}(U)$ then X has c.d.f. F.

Inverse Transform Method

General Quantile Function: Let F be a cumulative distribution function(c.d.f.). Then the inverse of F is defined as

$$F^{-1}(u) = \inf\{x \in \mathbb{R} | F(x) \geq u\}$$

Theorem: Let $F : \mathbb{R} \to [0, 1]$ be a c.d.f. and F^{-1} its inverse. If $U \sim \text{uniform}[0, 1]$ and we define $X = F^{-1}(U)$ then X has c.d.f. F.

Proof (Short Version):

$$P(X \leq a) = P(F^{-1}(U) \leq a) = P(\inf\{x \in \mathbb{R} | F(x) \geq U\} \leq a)$$

Since $\inf\{x \in \mathbb{R} | F(x) \geq U\} \leq a$ holds if and only if $F(a) \geq U$ then

$P(X \leq a) = P(F(a) \geq U) = F(a)$ therefore X has c.d.f. F.

This result is very general (applicable to both continuous and discrete distributions).

Proof (Long Version):

Let $U \sim \text{Uniform}(0,1)$ and define the generalized inverse of F as

$$F^{-1}(u) = \inf\{x \in \mathbb{R} \mid F(x) \geq u\}$$

Let

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

Then, for any $x \in \mathbb{R}$:

$$P(X \leq x) = P(F^{-1}(U) \leq x)$$

By the definition of the generalized inverse:

$$F^{-1}(U) \leq x \iff U \leq F(x)$$

Hence,

$$P(X \leq x) = P(U \leq F(x)) = F(x)$$

This shows that X has CDF F, regardless of whether F is continuous or discrete.

Algorithm (Inverse Transform Method)

Algorithm

1. Generate $U \sim U[0, 1]$
2. Define $X = F^{-1}(U)$
3. Return X

Algorithm (Inverse Transform Method)

Algorithm

1. Generate $U \sim U[0,1]$
2. Define $X = F^{-1}(U)$
3. Return X

Remarks:

- This method is more suited to continuous distributions, but it can also be applied to discrete distributions (See example 1.18 from the textbook).
- An important limitation of the method is that can only be applied to one-dimensional probability distributions.

Inverse Transform Example

Example: Generate a sample of 5 random numbers from a continuous random variable with probability density function

$$f(x) = \begin{cases} x^3/4 & \text{if } x \in [0,2] \\ 0 & \text{otherwise} \end{cases}$$

Inverse Transform Example

Example: Generate a sample of 5 random numbers from a continuous random variable with probability density function

$$f(x) = \begin{cases} x^3/4 & \text{if } x \in [0,2] \\ 0 & \text{otherwise} \end{cases}$$

Solution: First we find the cumulative distribution function F.

$$F(x) = \int_{-\infty}^{x} f(s)ds = \begin{cases} 0 & \text{if } x < 0 \\ x^4/16 & \text{if } 0 \leq x < 2 \\ 1 & \text{if } x \geq 2 \end{cases}$$

From here we get $F^{-1}(x) = \sqrt[4]{16x}$

Code

Code in R:

```r
n <- 5
U <- runif(n, min=0, max=1)
X <- (16*U)^(1/4)
print(X)
```

Code in Python:

```python
import numpy as np

n = 5
U = np.random.uniform(0, 1, n)
X = (16*U)**(1/4)
print(X)
```

Example (Exponential Distribution): Generate a sample of $n = 100$ random numbers from the exponential distribution with parameter $\lambda = 2$ using the inverse transform method.

Example (Exponential Distribution): Generate a sample of $n = 100$ random numbers from the exponential distribution with parameter $\lambda = 2$ using the inverse transform method.

In general the density for the exponential distribution with parameter λ is given by

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Then the cumulative distribution function is $F(x) = 1 - e^{-\lambda x}$ for $x \geq 0$.

The inverse function is $F^{-1}(x) = -\dfrac{\ln(1-x)}{\lambda}$

Code

Code in R:

```r
n <- 100
lambda <- 2
U <- runif(n, min=0, max=1)
X <- -log(1-U)/lambda
print(X)
hist(X)
```

Code in Python:

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

n = 100
lambda_ = 2
U = np.random.uniform(0, 1, n)
X = -np.log(1-U)/lambda_
print(X)
plt.hist(X, bins=10)
plt.show()
```

Inverse Transform Method: Exponential Distribution

Code

Code in R:

```r
n <- 100
lambda <- 2
U <- runif(n, min=0, max=1)
X <- -log(1-U)/lambda
print(X)
hist(X)
```

Code in Python:

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

n = 100
lambda_ = 2
U = np.random.uniform(0, 1, n)
X = -np.log(1-U)/lambda_
print(X)
plt.hist(X, bins=10)
plt.show()
```

Remark: Of course, without the constraint of using the inverse transform method, we could use R's or Python's built-in exponential generator.

Summary

- The inverse transform method is straightforward, efficient and very general.
- For the generation of continuous random variables, the inverse transform method is the method of choice in most cases, as long as F^{-1} can be found explicitly.
- Unfortunately, some important distributions do not admit a closed-form solution for F^{-1} (the normal distribution is an obvious example of this) therefore other methods would have to be applied in those cases.

Homework

1. Write down a computer program to generate a sample of 1000 random numbers from the probability distribution with density function

$$f(x) = \frac{3}{2}x^{-5/2}, \quad x \in [1, \infty), \quad 0 \text{ otherwise.}$$

2. Find the explicit inverse of the CDF for the **Weibull distribution**

$$F(x) = 1 - e^{-(x/\lambda)^k}, \; x \geq 0, \quad x \leq 0$$

3. Find the explicit inverse of the CDF for the **Cauchy distribution**

$$F(x) = \frac{1}{\pi} \arctan\left(\frac{x - x_0}{\gamma}\right) + \frac{1}{2}, \quad \gamma > 0$$

4. Find the explicit inverse of the CDF for the **Logistic distribution**

$$F(x) = \frac{1}{1 + e^{-(x-\mu)/s}}, \quad s > 0$$

and plot both the PDF and CDF. What is the difference between this distribution and the **Normal distribution**?