Performance Modeling and Design of Computer Systems- Ch 4 Generating Random Variables for Simulation

Debobroto Das Robin

Kent State University

Spring 2020

drobin@kent.edu

Overview

1. Introduction
2. Inverse-Transform Method
   - Continuous Case
   - Discrete Case
3. Accept-Reject Method
   - Discrete Case
   - Continuous Case

Random Variable Generation

- **Problem** : Assume a system in which
  - **interarrival times of jobs** are well modeled by an **Exponential distribution**
  - **Job sizes** (service requirements) are well modeled by a **Normal distribution**.
  - We want to simulate the system
- We need to be able to generate instances of
  - **Exponential distribution** &
  - **Job sizes** (service requirements) are well modeled **Normal distribution**.
- **Solution** : 2 Basic maethods for generating random variables
  - Assuming we already have a generator of **Uniform(0,1)** $(u \epsilon U(0,1))$ random variables
- 2 methods are
  - Inverse-Transform Method
  - Accept-Reject Method

Performance Modeling and Design of Computer Systems- Ch 4 Generating Random Variables for Simulation

Debobroto Das Robin

Introduction

Inverse-Transform Method

Continuous Case

Discrete Case

Accept-Reject Method

Idea: Map each element u generated by uniform distribution to some x of desired distribution

- This method assumes that we know the
  - c.d.f. (cumulative distribution function), $F_X(x) = P(X \leq x)$, of the random variable X that we are trying to generate, and
  - that this distribution is easily invertible, namely that we can get x from $F_X(x)$
- 2 variations
  - Continuous
  - Discrete

Idea : map each $u \, \epsilon \, U(0,1)$ generated to some x, where c.d.f is of X is F_x.

- We already have uniform **random variable** generator system
- $u \, \epsilon \, U(0,1) \rightarrow x \, \epsilon \, X(0, \inf)$
- we need to find the inverse transform that maps u to x
- we want

$$u = P\{0 < U < u\} = P\{0 < X < x\} = F_x(x) \qquad (1)$$

$$u = F_x(x) \implies x = {F_x}^{-1}(u) \qquad (2)$$

- Algorithm **Inverse-Transform Method to generate r.v. X**
  - Generate $u \, \epsilon \, U(0,1)$
  - $X = {F_x}^{-1}(u)$

Inverse-Transform Method

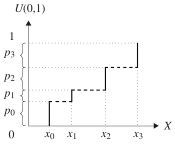
Continuous Case: Example exponential distribution

- Assume we want to get random variable x that follows exponential distribution
- $F(x) = u$
  $\Rightarrow 1 - e^{-\lambda x} = u$
  $\Rightarrow -\lambda x = ln(1-u)$
  $\Rightarrow x = -\dfrac{1}{\lambda} ln(1-u)$
- Given $u \,\epsilon\, U(0,1)$, get x from the formula
- x is an instance of $X \sim Exp(\lambda)$

Inverse-Transform Method
Discrete Case

- Same basic idea as continuous case
- We want X such that

$$X = \begin{cases} x_0 & \text{with prob } p_0 \\ x_1 & \text{with prob } p_1 \\ \dots & \\ x_k & \text{with prob } p_k \end{cases}$$



.2. Generating a discrete random variable with 4 values.

- Solution

1. Arrange $x_0, \dots, x_k$ s.t. $x_0 < x_1 < \dots < x_k$.
2. Generate $u \in U(0,1)$.
3. If $0 < u \leq p_0$, then output x_0.
   If $p_0 < u \leq p_0 + p_1$, then output x_1.
   If $p_0 + p_1 < u \leq p_0 + p_1 + p_2$, then output x_2.
   If $\sum_{i=0}^{\ell-1} p_i < u \leq \sum_{i=0}^{\ell} p_i$, then output x_ℓ, where $0 \leq \ell \leq k$.

Inverse-Transform Method

Practical Consideration for Solution

- Issues in the algorithm described in prev. page
  - the eqn. is not closed form
  - If X can take too many values $\rightarrow$ too many iterations
- Solution: Find an inverse transform like continuous case

Accept-Reject Method
Basics

- Applicable when,
  - we do not know the c.d.f., $F_X x$, but only know the p.d.f., $f_X()$
  - Ex: we want to generate a random variable from Normal distribution whose c.d.f. is unknown
- Idea: generate instances of the desired random variable, but throwing away (rejecting) some of the generated instances until the desired p.d.f (or p.m.f.) is met.
- Two cases
  - Discrete
  - Continuous

Accept-Reject Method

Discrete Case

- Given: Efficient method for generating random variable Q with probability mass function q_j, j :discrete, where $q_j = PQ = j$
- Output: Random variable P with probability mass function p_j, j :discrete, where $p_j = PP = j$
- Requirement: For all j , we must have $q_j > 0 \iff p_j > 0$. That is, P and Q take on the same set of values
- Example

$$Q = \begin{cases} 1 & \text{with prob } q_1 = 0.33 \\ 2 & \text{with prob } q_2 = 0.33 \\ 3 & \text{with prob } q_3 = 0.33 \end{cases}$$

$$P = \begin{cases} 1 & \text{with prob } p_1 = 0.36 \\ 2 & \text{with prob } p_2 = 0.24 \\ 3 & \text{with prob } p_3 = 0.40 \end{cases}.$$

Accept-Reject Method

Discrete Case: algorithm

- Algorithm

Accept-Reject Algorithm to generate discrete r.v. P:

1. Find r.v. Q s.t. $q_j > 0 \Leftrightarrow p_j > 0$.
2. Generate an instance of Q, and call it j.
3. Generate r.v. $U \in (0, 1)$.
4. If $U < \frac{p_j}{cq_j}$, return $P = j$ and stop; else return to step 2.

Here c $>$1

- Proof details in book page 74

Accept-Reject Method

Continuous Case

- Given: We know how to generate Y with probability density function $f_Y(t)$.
- Goal: To generate X with p.d.f. $f_X(t)$.
- Requirement: For all t, $f_Y(t) > 0 \iff f_X(t) > 0$
- Algorithm

Accept-Reject Algorithm to generate continuous r.v. X:

1. Find continuous r.v. Y s.t. $f_Y(t) > 0 \Leftrightarrow f_X(t) > 0$. Let c be a constant such that

$$\frac{f_X(t)}{f_Y(t)} \leq c, \ \forall t \text{ s.t. } f_X(t) > 0.$$

2. Generate an instance t of Y.
3. With probability $\frac{f_X(t)}{c \cdot f_Y(t)}$, return $X = t$ (i.e. "accept t" and stop). Else reject t and return to step 2.