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

Introduction

Inverse-Transform Method

Accept-Reject Method

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

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

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

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

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

Inverse-Transform Method

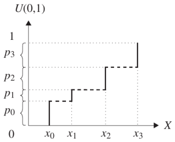
Continuous Case: Example exponential distribution

- 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 $0 < u \le p_0$, then output x_0.
   If $p_0 < u \le p_0 + p_1$, then output x_1.
   If $p_0 + p_1 < u \le p_0 + p_1 + p_2$, then output x_2.
   If $\sum_{i=0}^{\ell-1} p_i < u \le \sum_{i=0}^{\ell} p_i$, then output x_ℓ, where $0 \le \ell \le k$.

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