به نام خدا

Performance Evaluation of Computer Systems

Prof. Ali Movaghar

Fall 2022

Performance Modeling and Design of Computer Systems

4- GENERATING RANDOM VARIABLES FOR SIMULATION

Inverse-Transform Method

- This method assumes that
  - We know the **c.d.f.** of the random variable X
  - This distribution is easily invertible

Inverse-Transform Method to generate r.v. X:

1. Generate $u \in U(0, 1)$.
2. Return $X = F_X^{-1}(u)$.

A Standard Uniform Random Variable Y

- Let us assume that Y is a random variable uniformly distributed between 0 and 1. That is,

$$F_X(x) = P[X \leq x] = \begin{cases} 0, & \text{if } x < 0, \\ x, & \text{if } 0 \leq x \leq 1, \\ 1, & \text{if } x > 1. \end{cases}$$

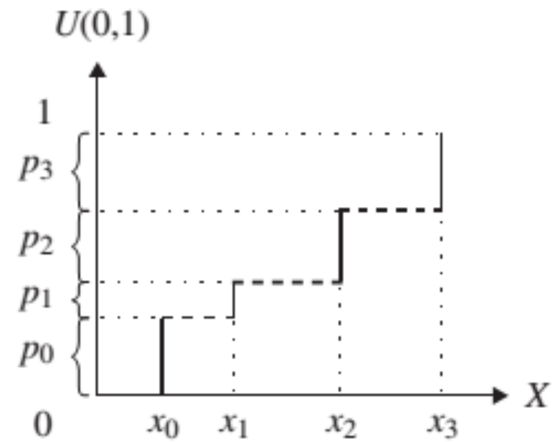
Generating a Random Variable X with Distribution $G(.)$

- Define $X = G^{-1}(Y)$
- Then,

$$\begin{aligned} F_X(x) &= P[X \leq x] \\ &= P[G^{-1}(Y) \leq x] \\ &= P[Y \leq G(x)] \\ &= G(x). \end{aligned}$$

Discrete Case

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



Solution:

1. Arrange $x_0, \ldots, x_k$ s.t. $x_0 < x_1 < \ldots < x_k$.
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 $0 \leq \ell \leq k$.

Discrete Case

- If X can take on many values,
  - We therefore need closed-form expressions for $\sum_{i=0}^{l} p_i$ for all l.
  - Equivalently, we need a closed form for $F_X(x) = P\{X \leq x\}$ for any x.
  - Then, we could do the same thing as in the continuous case.

Continuous Case

- Let $F(x) = 1\text{-}e^{-\lambda x}$ then we solve $y = 1\text{-}e^{-\lambda x}$
- then $1\text{-}y = e^{-\lambda x}$ or $x = -\, 1/\lambda \, Log(1\text{-}y)$

Accept-Reject Method

- Useful when we only know the **p.d.f.**, $f_X(\cdot)$.
- Throw away some of the instances until the desired **p.d.f.** is met
- **Example**
  – We want to generate P and we know how to generate Q

$$P = \begin{cases} 1 & \text{with prob } p_1 = 0.36 \\ 2 & \text{with prob } p_2 = 0.24 \\ 3 & \text{with prob } p_3 = 0.40 \end{cases}. \qquad Q = \begin{cases} 1 & \text{with prob } q_1 = 0.33 \\ 2 & \text{with prob } q_2 = 0.33 \\ 3 & \text{with prob } q_3 = 0.33 \end{cases}.$$

Discrete Case

- Let c be a constant such that $c > 1$ and

$$\frac{p_j}{q_j} \leq c,\ \forall j \text{ s.t. } p_j > 0.$$

<u>***Accept-Reject Algorithm to generate discrete r.v. P***</u>:

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.

Discrete Case

- **Proof**
- We want to prove that,
  - $\boldsymbol{P}$ {ends up being set to j (as opposed to some other value)} = p_j
  - P{P ends up being set to j} = $\frac{Fraction\ of\ time\ j\ is\ generated\ and\ accepted}{Fraction\ of\ time\ any\ value\ is\ accepted}$

- Fraction of time j is generated and accepted

  = **P** { j is generated } · **P** { j is accepted given j is generated}

  = $q_j . \frac{p_j}{cq_j} = \frac{p_j}{c}$

- Fraction of time any value is accepted

  = $\sum_j (Fraction\ of\ time\ j\ is\ generated\ and\ is\ accepted)$

  = $\sum_j \frac{p_j}{c} = \frac{1}{c}$
- So,

  P {P ends up being set to j} = $\frac{\frac{p_j}{c}}{\frac{1}{c}} = p_j$

Continuous Case

- We now use **.p.d.f.** rather than **.p.m.f.**

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,

$$f_Y(t) > 0 \Longleftrightarrow f_X(t) > 0.$$

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.

Generating Normal Random Variable

Goal: Generate $N \sim \text{Normal}(0, 1)$.

Idea: It will be enough to generate $X = |N|$ and then multiply N by -1 with probability 0.5.

- So,

$$f_X(t) = \frac{2}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}, \quad 0 < t < \infty$$

- Let $Y \sim Exp(1)$

$$f_Y(t) = e^{-t}, \quad 0 < t < \infty$$

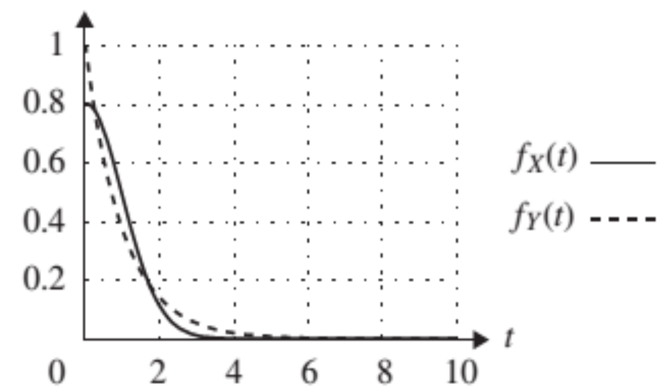


Figure 4.4. Solid line shows $f_X(t)$. Dashed line shows proposed $f_Y(t)$.

Determine c

$$\frac{f_X(t)}{f_Y(t)} = \frac{2}{\sqrt{2\pi}} e^{-\frac{t^2}{2}+t} = \sqrt{\frac{2}{\pi}} e^{t-\frac{t^2}{2}}$$

So, the maximum value occurs when $t - \frac{t^2}{2}$ is maximized.

$$0 = \frac{d}{dt}\left(t - \frac{t^2}{2}\right) = 1 - t \quad \Rightarrow \quad t = 1$$

So,

$$c = \frac{f_X(1)}{f_Y(1)} = \sqrt{\frac{2e}{\pi}} \approx 1.3.$$

Thus we only need 1.3 iterations on average!