2WB05 Simulation Lecture 8: Generating random variables

Marko Boon
http://www.win.tue.nl/courses/2WB05

Technische Universiteit **Eindhoven** University of Technology

January 7, 2013

Outline

1. How do we generate random variables?
2. Fitting distributions

Generating random variables

How do we generate random variables?

- Sampling from continuous distributions
- Sampling from discrete distributions

Continuous distributions

Inverse Transform Method

Let the random variable X have a continuous and increasing distribution function F. Denote the inverse of F by F^{-1}. Then X can be generated as follows:

- Generate U from $U(0, 1)$;
- Return $X = F^{-1}(U)$.

If F is not continuous or increasing, then we have to use the *generalized* inverse function

$$F^{-1}(u) = \min\{x : F(x) \geq u\}.$$

Continuous distributions

Examples

- $X = a + (b - a)U$ is uniform on (a, b);
- $X = -\ln(U)/\lambda$ is exponential with parameter λ;
- $X = (-\ln(U))^{1/a}/\lambda$ is Weibull, parameters a and λ.

Unfortunately, for many distribution functions we do not have an easy-to-use (closed-form) expression for the inverse of F.

Technische Universiteit Eindhoven University of Technology

Continuous distributions

Composition method

This method applies when the distribution function F can be expressed as a mixture of other distribution functions $F_1, F_2, \ldots$,

$$F(x) = \sum_{i=1}^{\infty} p_i F_i(x),$$

where

$$p_i \geq 0, \qquad \sum_{i=1}^{\infty} p_i = 1$$

The method is useful if it is easier to sample from the F_i's than from F.

- First generate an index I such that $P(I = i) = p_i, \qquad i = 1, 2, \ldots$
- Generate a random variable X with distribution function F_I.

Continuous distributions

Examples

- Hyper-exponential distribution:

$$F(x) = p_1 F_1(x) + p_2 F_2(x) + \cdots + p_k F_k(x), \qquad x \geq 0,$$

where $F_i(x)$ is the exponential distribution with parameter μ_i, $i = 1, \ldots, k$.

- Double-exponential (or Laplace) distribution:

$$f(x) = \begin{cases} \frac{1}{2}e^{x}, & x < 0; \\ \frac{1}{2}e^{-x}, & x \geq 0, \end{cases}$$

where f denotes the density of F.

Continuous distributions

Convolution method

In some case X can be expressed as a sum of independent random variables $Y_1, \ldots, Y_n$, so

$$X = Y_1 + Y_2 + \cdots + Y_n.$$

where the Y_i's can be generated more easily than X.

Algorithm:

- Generate independent $Y_1, \ldots, Y_n$, each with distribution function G;
- Return $X = Y_1 + \cdots + Y_n$.

Continuous distributions

Example

If X is Erlang distributed with parameters n and μ, then X can be expressed as a sum of n independent exponentials Y_i, each with mean $1/\mu$.

Algorithm:

- Generate n exponentials $Y_1, \dots, Y_n$, each with mean μ;
- Set $X = Y_1 + \cdots + Y_n$.

More efficient algorithm:

- Generate n uniform $(0, 1)$ random variables $U_1, \dots, U_n$;
- Set $X = -\ln(U_1 U_2 \cdots U_n)/\mu$.

Continuous distributions

Acceptance-Rejection method

Denote the density of X by f. This method requires a function g that *majorizes* f,

$$g(x) \geq f(x)$$

for all x. Now g will not be a density, since

$$c = \int_{-\infty}^{\infty} g(x)dx \geq 1.$$

Assume that $c < \infty$. Then $h(x) = g(x)/c$ *is* a density. Algorithm:

1. Generate Y having density h;
2. Generate U from $U(0, 1)$, independent of Y;
3. If $U \leq f(Y)/g(Y)$, then set $X = Y$; else go back to step 1.

The random variable X generated by this algorithm has density f.

Validity of the Acceptance-Rejection method

Note

$$P(X \leq x) = P(Y \leq x | Y \text{accepted}).$$

Now,

$$P(Y \leq x, Y \text{accepted}) = \int_{-\infty}^{x} \frac{f(y)}{g(y)} h(y) dy = \frac{1}{c} \int_{-\infty}^{x} f(y) dy,$$

and thus, letting $x \to \infty$ gives

$$P(Y \text{accepted}) = \frac{1}{c}.$$

Hence,

$$P(X \leq x) = \frac{P(Y \leq x, Y \text{accepted})}{P(Y \text{accepted})} = \int_{-\infty}^{x} f(y) dy.$$

Continuous distributions

Note that the number of iterations is geometrically distributed with mean c.

How to choose g?

- Try to choose g such that the random variable Y can be generated rapidly;
- The probability of rejection in step 3 should be small; so try to bring c close to 1, which mean that g should be close to f.

Continuous distributions

Example

The Beta(4,3) distribution has density

$$f(x) = 60x^3(1-x)^2, \qquad 0 \le x \le 1.$$

The maximal value of f occurs at $x = 0.6$, where $f(0.6) = 2.0736$. Thus, if we define

$$g(x) = 2.0736, \qquad 0 \le x \le 1,$$

then g majorizes f. Algorithm:

1. Generate Y and U from $U(0,1)$;
2. If $U \le \dfrac{60Y^3(1-Y)^2}{2.0736}$, then set $X = Y$; else reject Y and return to step 1.

Continuous distributions

Normal distribution

Methods:

- Acceptance-Rejection method
- Box-Muller method

Continuous distributions

Acceptance-Rejection method

If X is $N(0,1)$, then the density of $|X|$ is given by

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2}, \qquad x > 0.$$

Now the function

$$g(x) = \sqrt{2e/\pi}\, e^{-x}$$

majorizes f. This leads to the following algorithm:

1. Generate an exponential Y with mean 1;
2. Generate U from $U(0,1)$, independent of Y;
3. If $U \leq e^{-(Y-1)^2/2}$, then accept Y; else reject Y and return to step 1.
4. Return $X = Y$ or $X = -Y$, both with probability $1/2$.

Technische Universiteit Eindhoven University of Technology

Continuous distributions

Box-Muller method

If U_1 and U_2 are independent $U(0, 1)$ random variables, then

$$\begin{aligned} X_1 &= \sqrt{-2 \ln U_1} \cos(2\pi U_2) \\ X_2 &= \sqrt{-2 \ln U_1} \sin(2\pi U_2) \end{aligned}$$

are independent standard normal random variables.

This method is implemented in the function `nextGaussian()` in `java.util.Random`

Discrete distributions

Discrete version of Inverse Transform Method

Let X be a discrete random variable with probabilities

$$P(X = x_i) = p_i, \qquad i = 0, 1, \dots, \qquad \sum_{i=0}^{\infty} p_i = 1.$$

To generate a realization of X, we first generate U from $U(0, 1)$ and then set $X = x_i$ if

$$\sum_{j=0}^{i-1} p_j \leq U < \sum_{j=0}^{i} p_j.$$

Discrete distributions

So the algorithm is as follows:

- Generate U from $U(0, 1)$;
- Determine the index I such that

$$\sum_{j=0}^{I-1} p_j \leq U < \sum_{j=0}^{I} p_j$$

  and return $X = x_I$.

The second step requires a *search*; for example, starting with $I = 0$ we keep adding 1 to I until we have found the (smallest) I such that

$$U < \sum_{j=0}^{I} p_j$$

Note: The algorithm needs exactly one uniform random variable U to generate X; this is a nice feature if you use variance reduction techniques.

Discrete distributions

Array method: when X has a finite support

Suppose $p_i = k_i/100$, $i = 1, \ldots, m$,
where k_i's are integers with $0 \le k_i \le 100$

Construct array $A[i]$, $i = 1, \ldots, 100$ as follows:
set $A[i] = x_1$ for $i = 1, \ldots, k_1$
set $A[i] = x_2$ for $i = k_1 + 1, \ldots, k_1 + k_2$, etc.

Then, first sample a random index I from $1, \ldots, 100$:
$I = 1 + \lfloor 100U \rfloor$ and set $X = A[I]$

Technische Universiteit Eindhoven University of Technology

Bernoulli

Two possible outcomes of X (success or failure):

$$P(X = 1) = 1 - P(X = 0) = p.$$

Algorithm:

- Generate U from $U(0, 1)$;
- If $U \leq p$, then $X = 1$; else $X = 0$.

Discrete distributions

Discrete uniform

The possible outcomes of X are $m, m+1, \ldots, n$ and they are all equally likely, so

$$P(X = i) = \frac{1}{n-m+1}, \qquad i = m, m+1, \ldots, n.$$

Algorithm:

- Generate U from $U(0, 1)$;
- Set $X = m + \lfloor (n-m+1)U \rfloor$.

Note: No search is required, and compute $(n-m+1)$ ahead.

Geometric

A random variable X has a geometric distribution with parameter p if

$$P(X = i) = p(1-p)^i, \qquad i = 0, 1, 2, \ldots;$$

X is the number of failures till the first success in a sequence of Bernoulli trials with success probability p.

Algorithm:

- Generate independent Bernoulli(p) random variables $Y_1, Y_2, \ldots$; let I be the index of the first successful one, so $Y_I = 1$;
- Set $X = I - 1$.

Alternative algorithm:

- Generate U from $U(0, 1)$;
- Set $X = \lfloor \ln(U)/\ln(1-p) \rfloor$.

Technische Universiteit Eindhoven University of Technology

Discrete distributions

Binomial

A random variable X has a binomial distribution with parameters n and p if

$$P(X = i) = \binom{n}{i} p^i (1-p)^{n-i}, \qquad i = 0, 1, \ldots, n;$$

X is the number of successes in n independent Bernoulli trials, each with success probability p.

Algorithm:

- Generate n Bernoulli(p) random variables $Y_1, \ldots, Y_n$;
- Set $X = Y_1 + Y_2 + \cdots + Y_n$.

Alternative algorithms can be derived by using the following results.

Discrete distributions

Let $Y_1, Y_2, \ldots$ be independent geometric(p) random variables, and I the smallest index such that

$$\sum_{i=1}^{I+1} (Y_i + 1) > n.$$

Then the index I has a binomial distribution with parameters n and p.

Let $Y_1, Y_2, \ldots$ be independent exponential random variables with mean 1, and I the smallest index such that

$$\sum_{i=1}^{I+1} \frac{Y_i}{n - i + 1} > -\ln(1 - p).$$

Then the index I has a binomial distribution with parameters n and p.

Discrete distributions

Negative Binomial

A random variable X has a negative-binomial distribution with parameters n and p if

$$P(X = i) = \binom{n+i-1}{i} p^n (1-p)^i, \qquad i = 0, 1, 2, \ldots;$$

X is the number of failures before the n-th success in a sequence of independent Bernoulli trials with success probability p.

Algorithm:

- Generate n geometric(p) random variables $Y_1, \ldots, Y_n$;
- Set $X = Y_1 + Y_2 + \cdots + Y_n$.

Technische Universiteit Eindhoven University of Technology

Discrete distributions

Poisson

A random variable X has a Poisson distribution with parameter λ if

$$P(X = i) = \frac{\lambda^i}{i!}e^{-\lambda}, \qquad i = 0, 1, 2, \ldots;$$

X is the number of events in a time interval of length 1 if the inter-event times are independent and exponentially distributed with parameter λ.

Algorithm:

- Generate exponential inter-event times $Y_1, Y_2, \ldots$ with mean 1; let I be the smallest index such that

$$\sum_{i=1}^{I+1} Y_i > \lambda;$$

- Set $X = I$.

Discrete distributions

Poisson (alternative)

- Generate U(0,1) random variables $U_1, U_2, \ldots$;
  let I be the smallest index such that
  $$\prod_{i=1}^{I+1} U_i < e^{-\lambda};$$
- Set $X = I$.

Fitting distributions

Input of a simulation

Specifying distributions of random variables (e.g., interarrival times, processing times) and assigning parameter values can be based on:

- Historical numerical data
- Expert opinion

In practice, there is sometimes real data available, but often the only information of random variables that is available is their mean and standard deviation.

Fitting distributions

Empirical data can be used to:

- construct empirical distribution functions and generate samples from them during the simulation;
- fit theoretical distributions and then generate samples from the fitted distributions.

Fitting distributions

Moment-fitting

Obtain an approximating distribution by fitting a *phase-type distribution* on the mean, $E(X)$, and the coefficient of variation,

$$c_X = \frac{\sigma_X}{E(X)},$$

of a given positive random variable X, by using the following simple approach.

Technische Universiteit Eindhoven University of Technology

Moment-fitting

Coefficient of variation less than 1: Mixed Erlang

If $0 < c_X < 1$, then fit an $E_{k-1,k}$ distribution as follows. If

$$\frac{1}{k} \leq c_X^2 \leq \frac{1}{k-1},$$

for certain $k = 2, 3, \ldots$, then the approximating distribution is with probability p (resp. $1 - p$) the sum of $k - 1$ (resp. k) independent exponentials with common mean $1/\mu$. By choosing

$$p = \frac{1}{1 + c_X^2}\left(kc_X^2 - \sqrt{k(1 + c_X^2) - k^2 c_X^2}\right), \qquad \mu = \frac{k - p}{E(X)},$$

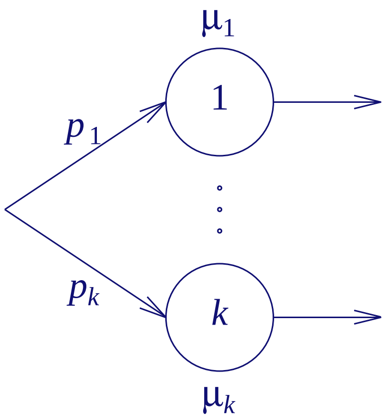
the $E_{k-1,k}$ distribution matches $E(X)$ and c_X.

Moment-fitting

Coefficient of variation greater than 1: Hyperexponential

In case $c_X \geq 1$, fit a $H_2(p_1, p_2; \mu_1, \mu_2)$ distribution.

Phase diagram for the $H_k(p_1, \ldots, p_k; \mu_1, \ldots, \mu_k)$ distribution:



Moment-fitting

But the H_2 distribution is not uniquely determined by its first two moments. In applications, the H_2 distribution with *balanced means* is often used. This means that the normalization

$$\frac{p_1}{\mu_1} = \frac{p_2}{\mu_2}$$

is used. The parameters of the H_2 distribution with balanced means and fitting $E(X)$ and c_X (≥ 1) are given by

$$p_1 = \frac{1}{2}\left(1 + \sqrt{\frac{c_X^2 - 1}{c_X^2 + 1}}\right), \qquad p_2 = 1 - p_1,$$

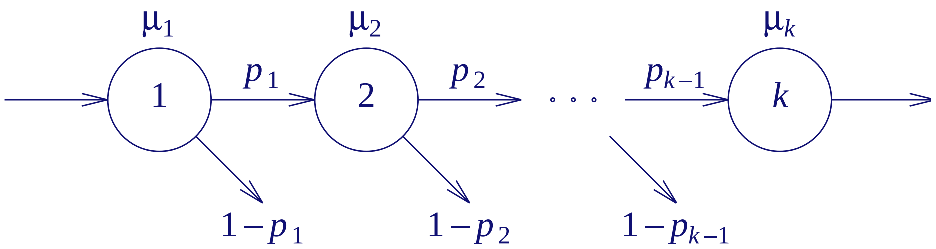
$$\mu_1 = \frac{2p_1}{E(X)}, \qquad \mu_2 = \frac{2p_2}{E(X)}.$$

Technische Universiteit
Eindhoven
University of Technology

Moment-fitting

In case $c_X^2 \geq 0.5$ one can also use a Coxian-2 distribution for a two-moment fit.

Phase diagram for the Coxian-k distribution:



The following parameter set for the Coxian-2 is suggested:

$$\mu_1 = 2/E(X), \quad p_1 = 0.5/c_X^2, \quad \mu_2 = \mu_1 p_1.$$

Moment-fitting

Fitting nonnegative discrete distributions

Let X be a random variable on the non-negative integers with mean $E(X)$ and coefficient of variation c_X. Then it is possible to fit a discrete distribution on $E(X)$ and c_X using the following families of distributions:

- Mixtures of Binomial distributions;
- Poisson distribution;
- Mixtures of Negative-Binomial distributions;
- Mixtures of geometric distributions.

This fitting procedure is described in Adan, van Eenige and Resing (see Probability in the Engineering and Informational Sciences, 9, 1995, pp 623-632).

Fitting distributions

Adequacy of fit

- Graphical comparison of fitted and empirical curves;
- Statistical tests (*goodness-of-fit tests*).

Technische Universiteit
Eindhoven
University of Technology