Generating discrete Random variable

Arabin Kumar Dey

Assistant Professor

Department of Mathematics
Indian Institute of Technology Guwahati

Talk at Indian Institute of Technology Guwahati

6-th january, 2012

The Inverse transform method

- Consider a r.v.X with pmf
  $P(X = x_j) = p_j \ , \ \ j = 0, 1, \cdots , \sum_j p_j = 1$

- To generate a value from this distribution, first we generate a random number U and set

$$X = \begin{cases} x_0, & \text{if } \ U < p_0 \\ x_1, & \text{if } \ p_0 \leq U < p_0 + p_1 \\ \vdots & \\ x_j & \textit{if } \ \sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^{j} p_i \\ \vdots & \end{cases}$$

- This is called the inverse transform method.
- Proof to be outlined in class.

$$
\begin{aligned}
P(X = x_j) &= P(\sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^{j} p_i) \\
&= P(U < \sum_{i=0}^{j} p_i) - P(U < \sum_{i=0}^{j-1} p_i) \\
&= \sum_{i=0}^{j} p_i - \sum_{i=0}^{j-1} p_i \\
&= p_j
\end{aligned}
$$

Some remarks

- The method can be written algorithmically as generate a random number U
  if $U < p_0$, set $X = x_0$ and STOP
  if $U < p_0 + p_1$, set $X = x_1$ and STOP
  if $U < p_0 + p_1 + p_2$, set $X = x_2$ and STOP
  $\vdots$
- If the $X_i's$ are ordered like $X_0 < X_1 < X_2 < \cdots$ so that the cdf $F(X_k) = \sum_{i=0}^{k} p_i$ and that
  X equals to X_j if $F(X_{j-1}) \leq U < F(X_j)$.
  Therefore, after generating U, we determine the value of X by looking for the interval.$[F(x_{j-1}), F(X_j)]$ in which it lies (or equivalently finding the inverse of U).

Illustrative example

Suppose we want to simulate from the discrete distribution with $P(X = 1) = 0.20$, $P(X = 2) = 0.25$, $P(X = 3) = 0.40$, $P(X = 4) = 0.15$

We do the following:

generate a random number U

if $U < 0.20$, set $X = 1$ and STOP

if $U < 0.45$, set $X = 2$ and STOP

if $U < 0.85$, set $X = 3$ and STOP

otherwise set X = 4.

It is suggested the following could be more efficient:

generate a random number U

if $U < 0.40$, set $X = 3$ and STOP

if $U < 0.65$, set $X = 2$ and STOP

if $U < 0.85$, set $X = 1$ and STOP

otherwise set X = 4.

Generating a discrete uniform random variable

- In the discrete uniform distribution, we have equal probabilities: $P(X = j) = \frac{1}{n}$, for $j = 1, 2, \ldots, n$.
- To simulate from this distribution, we generate a random number U and then set
  $X = j \quad \text{if } \frac{j-1}{n} \leq U < \frac{j}{n}$.
- This condition is equivalent to if $j - 1 \leq nU < j$, that is
  X=Int(nU)+1 ,
  where Int(X) is the greatest integer part of X.

Generating a geometric random variable

- In a geometric distribution with parameter p, we have $P(X = i) = pq^{i-1}$,for $i \geq 1$.
- Note that the cumulative probability $P(X \leq j-1) = \sum_{i=1}^{j-1} P(X = i) = 1 - q^{j-1}$.
- It can be shown that with a random number U, then X=Int$(\frac{log(U)}{log(q)}) + 1$.
  is indeed geometric with parameter p.

Generate U and set X=j
if $\quad 1-q^{j-1} \leq U < 1-q^{j}$
$U \sim$ Uniform(0,1)$\Longrightarrow 1-U \sim$ Uniform(0,1)
$q^{j} < 1-U \leq q^{j-1}$

$$\begin{aligned}
X &= \min\{j| \quad q^{j} < 1-U\} \\
&= \min\{j| \quad j * logq < log(1-U)\} \\
&= \min\{j| \quad j > \frac{log(1-U)}{logq}\} \\
&= Int(\frac{log(1-U)}{logq}) + 1 \\
&= Int(\frac{log(U)}{logq}) + 1
\end{aligned}$$

Generating a Poisson random variable

- For the case of the Poisson, we exploit the recursion property $p_{i+1} = \frac{\lambda}{i+1} p_i \quad$ for $i \geq 0$.
- The following steps can then be followed to generate from a Poisson with parameter λ:

  step 1 : generate a random number U.
  step 2 : set $i = 0$, $p = e^{-\lambda}$ and $F = p$.
  step 3 : if $U < F$, set $X = i$ and STOP.
  step 4 : set $p = \frac{\lambda p}{i+1}$, $F = F + p$, and $i = i + 1$.
  step 5 : return to step 3.
- Note that F is indeed the cdf $F(i) = P(X \leq i)$.
- It can be shown that the average number of searches grows with the square root of λ.(proof to be discussed!)

Generating a Binomial random variables

- Just as in the Poisson case, we exploit the recursion property for the Binomial distribution: $P(X = i+1) = \frac{n-i}{i+1}\frac{p}{1-p}P(X = i)$.
- The following steps can then be followed to generate a Binomial random variable with parameters n and probability of success p:
  step 1: generate a random number U.
  step 2: set $c = \frac{p}{1-p}$, $i = 0$, $pr = (1-p)^n$, and F=pr.
  step 3: if $U < F$, set $X = i$ and STOP.
  step 4: reset $pr = [\frac{c(n-i)}{i+1}]pr$, $F = F + pr$, and $i = i + 1$.
  step 5: return to step 3.
- As an exercise, try to write an R routine for generating a Binomial random variable following the above steps.
- Another approach to simulate from a Binomial(n, p) is to use the interpretation that it is equal to the number of success in n independent Bernoulli trials.

$$Bernoulli(p) = \begin{cases} X = 1 & with \quad probability \quad p \\ X = 0 & with \quad probability \quad 1-p \end{cases}$$

Let $X_1, X_2, X_3, \cdots\cdots, X_n$ are all Bernoulli trials,
Consider $X_1 + X_2 + X_3 + \cdots\cdots + X_n = X$,
So $X \sim$ Binomial (n, p).

The composition approach

- Consider now simulating from a distribution with mass function $P(X = j) = \alpha p_j^{(1)} + (1-\alpha)p_j^{(2)}, j \geq 0, 0 < \alpha < 1.$
- If X_1 and X_2 are the random variables with respective mass functions $p_j^{(1)}$ and $p_j^{(2)}$, then

$$X = \begin{cases} X_1 & with \quad probability \quad \alpha \\ X_2 & with \quad probability \quad 1-\alpha \end{cases}$$

- One approach then to generate from this mixture distribution is :

  step 1 : genrate a random number U_1

  step 2 : generate from X_1 and X_2 distributions.

  step 3 : if $U < \alpha$, set $X = X_1$.

  step 4 : else if $U > \alpha$, set $X = X_2$.

Example 4g

- Consider the example of generating X from a distribution with mass function

$$p_j = P(X = j) = \begin{cases} 0.05, & for \quad j = 1,2,3,4,5 \\ 0.15, & for \quad j = 6,7,8,9,10 \end{cases}$$

- Note that this is equivalent to
$P(X = j) = 0.5p_j^{(1)} + 0.5p_j^{(2)}$,
where $p_j^{(1)} = 0.10$, for $j = 1, 2, \cdots\cdots, 10$ and

$$p_j^{(2)} = \begin{cases} 0, & for \quad j = 1,2,3,4,5 \\ 0.2, & for \quad j = 6,7,8,9,10 \end{cases}$$

- Thus, first generate a random number U, and then generate from the discrete uniform over $1, 2, \cdots 10$ if $U < 0.5$ and from the discrete uniform over 6,7,8,9,10 otherwise.

Example 4g

- Consider the example of generating X from a distribution with mass function

$$p_j = P(X = j) = \begin{cases} 0.05, & for \quad j = 1, 2, 3, 4, 5 \\ 0.15, & for \quad j = 6, 7, 8, 9, 10 \end{cases}$$

- Note that this is equivalent to
$P(X = j) = 0.5p_j^{(1)} + 0.5p_j^{(2)}$,
where $p_j^{(1)} = 0.10$, for $j = 1, 2, \cdots\cdots, 10$ and

$$p_j^{(2)} = \begin{cases} 0, & for \quad j = 1, 2, 3, 4, 5 \\ 0.2, & for \quad j = 6, 7, 8, 9, 10 \end{cases}$$

- Thus, first generate a random number U, and then generate from the discrete uniform over $1, 2, \cdots 10$ if $U < 0.5$ and from the discrete uniform over 6,7,8,9,10 otherwise.

Mixture distributions

- In the case where the distribution function of X is given by $F(x) = \sum_{i=1}^{n} \alpha_i F_i(x)$,
  where F_i , $i = 1, \cdots, n$ are distribution functions, we have what we call a mixture distribution.
- To simulate from such a mixture distribution,
  step 1 : simulate a random variable I, equal to i with probability α_i, for $i = 1, 2, \cdots, 10$.
  step 2 : simulate from the distribution F_i.
- This is also called the composition method. Generate U
  $U < \alpha_1 \longrightarrow X = X_1$
  $U < \alpha_1 + \alpha_2 \longrightarrow X = X_2$
  $\vdots$
  $U < \sum_{i=1}^{j} \alpha_i \longrightarrow X = X_j$
  $\vdots$

R codes for simulating from known discrete distributions

- In *R*, there are many functions that generate discrete random variables. Most of them start with *r*.
- Here are a few of them :
  rbinom - binomial
  rnbinom - negative binomial
  rpois - Poisson
  rgeom - geometric
  rhyper - hypergeometric