Random Number Generation

Inverse CDF

Let $X$ have a continuous, strictly increasing cdf $F(x)$ for $a < x < b$, where we allow $a = -\infty$ and/or $b = +\infty$. If $U \sim \text{uniform}(0,1)$, then the cdf of $Y := F^{-1}(U)$ is $F(x)$. To see this, write

$$F_Y(y) = P(Y \leq y) = P(F^{-1}(U) \leq y) = P(U \leq F(y)) = \int_0^{F(y)} 1 \, du = F(y).$$

**Example** Let $X \sim \exp(\lambda = 1)$ so that $F(x) = 1 - e^{-x}$, $x \geq 0$. To find $F'(u)$, solve $u = F(x)$ for $x$, i.e., $u = 1 - e^{-x}$ or $e^x = 1 - u$ or $-x = \ln(1 - u)$ or $x = -\ln(1 - u)$. So $Y = -\ln(1 - U) \sim \exp(\lambda = 1)$. In addition, since $Y = 1 - U$ is also uniform$(0,1)$, it suffices to generate $Y \sim \text{uniform}(0,1)$ and use $Y = -\ln V$ instead. This is how the MATLAB function `expndinv` works.

**Example.** Let $X \sim \text{Cauchy}(\lambda = 1)$ with cdf $F(x) = \frac{1}{\pi} \tan^{-1} x + \frac{1}{2}$. From $u = \frac{1}{\pi} \tan^{-1} x + \frac{1}{2}$, $x = \tan[\pi(u - \frac{1}{2})]$. 

**Example.** Let $X \sim \text{N}(0,1)$ with cdf $F(x) = \int_{-\infty}^{x} e^{-t^2/2} \, dt / \sqrt{2\pi}$. We cannot solve $u = F(x)$ in closed form, although we could do it numerically, which would be slow. Note, however, that although MATLAB provides $F^{-1}$ (see `norminv`), MATLAB does not use this method.
**Mixtures**

Suppose $X$ has a density $f$ of the form

$$f(x) = \sum_{k=1}^{n} \phi_k f_k(x),$$

where the $\phi_k$ are positive and $\sum_{k=1}^{n} \phi_k = 1$, and $f_k$ is the pdf of a RV $Y_k$ that can be simulated easily. Generate a discrete RV $N$ with $P(N=k) = \phi_k$.

If $N=k$, generate $Y_k$ with density $f_k$ and put $X = Y_N = Y_k$. Then

$$P(X \leq x) = P(Y_N \leq x) = \sum_{k=1}^{n} P(Y_N \leq x | N=k) \phi_k$$

$$= \sum_{k=1}^{n} P(Y_k \leq x | N=k) \phi_k = \sum_{k=1}^{n} \left( \int_{-\infty}^{x} f_k(y) \, dy \right) \phi_k$$

and so

$$\frac{d}{dx} P(X \leq x) = \sum_{k=1}^{n} f_k(x) \phi_k = f(x) \text{ as required.}$$

**Discrete RVs**

To generate the discrete RV $N$ above, let $\text{UNI}n(i, a, b)$ and put $N := \min\{1 \leq k \leq n : \sum_{l=1}^{k} \phi_l \geq U \}$. Then since

$$N = l \iff \sum_{k=1}^{l} \phi_k \geq U > \sum_{k=1}^{l-1} \phi_k$$

and so $P(N=k) = \phi_k$ as required.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>$\phi_1$</td>
<td>$\phi_1 + \phi_2$</td>
<td>$\phi_1 + \phi_2 + \phi_3$</td>
<td>$\phi_1 + \phi_2 + \phi_3 + \phi_4$</td>
</tr>
</tbody>
</table>

In MATLAB, if $P = [\phi_1, \ldots, \phi_n]$, and $S = \text{cumsun}(P)$, and $U = \text{rand}(1)$, $i = \text{find}(U \leq S)$, then $N = i$ has $P(N=k) = \phi_k$. 

$RN-2$
Special Tricks

1) If \( U \) and \( V \) are independent uniform(0,1) and
\[
X := \sqrt{-2 \ln U} \cos(2\pi V), \quad Y := \sqrt{-2 \ln U} \sin(2\pi V),
\]
then it can be shown that \( X \) and \( Y \) are independent \( N(0,1) \) RVs.

2) If \( X_1, X_2, \ldots, X_n \) are iid \( \exp(\lambda) \), then \( Y := X_1 + \cdots + X_n \)
is Erlang \((n, \lambda)\).

3) If \( X_1, X_2, \ldots, X_n \) are iid \( N(0,1) \), then \( Y = X_1^2 + \cdots + X_n^2 \)
is chi-squared with \( n \) degrees of freedom.

4) If \( U_1, U_2, \ldots, U_n \) are iid \( \text{uniform}(0,1) \), then
by the central limit theorem \( Y := U_1 + \cdots + U_n \)
is approximately \( N(n/2, n/12) \). Hence,
\[
X := \frac{(Y - n/2)}{\sqrt{n/12}} \text{ is approximately } N(0,1).
\]

Rejection (or Acceptance/Rejection)

Let \( X \) and \( Y \) be RV's with respective pdfs \( p \) and \( q \).
Let \( f = \alpha p \) and \( g = \beta q \), where \( \alpha \) and \( \beta \) are arbitrary positive constants. Put
\[
K := \sup_x \frac{f(x)}{g(x)} < \infty
\]
so that \( f(x) \leq K g(x) \). Thus, \( f(x) > 0 \Rightarrow g(x) > 0 \).

Fix any \( K' \geq K \). For \( n = 1, 2, \ldots \), let \( U_n \) be iid \( \text{uniform}(0,1) \) and \( Y_n \) be iid with density \( g \).
Let \( N = \min \{ n \geq 1 : U_n \leq \frac{f(Y_n)}{K' g(Y_n)} \} \). Put \( X := Y_N \).
If we put $Z_n := I \{ U_n > \frac{f(y_n)}{K'g(y_n)} \}$, then

$$E[N] = \frac{1}{1-\Theta} = \frac{1}{\Theta(U_n \leq \frac{f(y_n)}{K'g(y_n)})}$$

Now, the acceptance probability can be found using the law of total probability:

$$\Theta(U_n \leq \frac{f(y_n)}{K'g(y_n)}) = \int \Theta(U_n \leq \frac{f(y_n)}{K'g(y_n)} | Y_n = y) q(y) dy$$

$$= \int \Theta(U_n \leq \frac{f(y)}{K'g(y)}) q(y) dy$$

$$= \int \frac{f(y)}{K'g(y)} q(y) dy$$

$$= \frac{1}{K'\beta} \int f(y) dy = \frac{\alpha}{K'\beta}$$

To minimize $E[N]$, maximize $\alpha/(K'\beta)$. Next,

$$\Theta(X \leq x) = \Theta(Y_N \leq x)$$

$$= \sum_{n=1}^{\infty} \Theta(Y_n \leq x | N=n) \Theta(N=n)$$

Now,

$$\Theta(Y_n \leq x | N=n) = \Theta(Y_n \leq x | N=n)$$
\[
= \mathbb{P}(Y_n \leq x \mid z_1 = 1, \ldots, z_{n-1} = 1, z_n = 0) \\
= \mathbb{P}(Y_n \leq x \mid z_n = 0).
\]

We then write

\[
\mathbb{P}(Y_n \leq x, z_n = 0) = \mathbb{P}(Y_n \leq x, U_n \leq \frac{f(Y_n)}{K'g(Y_n)}) \\
= \int \mathbb{P}(Y_n \leq u, U_n \leq \frac{f(Y_n)}{K'g(Y_n)} \mid Y_n = y) q(y) \, dy \\
= \int \mathbb{P}(Y_n \leq x, U_n \leq \frac{f(Y_n)}{K'g(Y_n)}) q(y) \, dy \\
= \int_{-\infty}^{\infty} \mathbb{P}(U_n \leq \frac{f(y)}{K'g(y)}) q(y) \, dy \\
= \int_{-\infty}^{\infty} \frac{f(y)}{K'g(y)} q(y) \, dy \\
= \int_{-\infty}^{\infty} \phi(\beta) d\gamma - \frac{\alpha}{K' \beta} \\
= \mathbb{P}(U_n \leq \frac{f(Y_n)}{K'g(Y_n)}) = \mathbb{P}(Z_n = 0)
\]

\[
= \mathbb{P}(Y_n \leq x \mid z_n = 0) = F_X(x) \\
\mathbb{P}(Y_n \leq x \mid N = n) = F_X(x) \\
= \mathbb{P}(Y_n \leq x) = \sum_{n=1}^{\infty} F_X(x) \mathbb{P}(N = n) = F_X(x)
\]

\textbf{Example}

\[
\text{Then } U_n \sim \text{unif}(0,1), Y_n \sim \text{uniform}[a, b], \quad \bar{f} := \max f(x) \quad a \leq x \leq b \\
g(x) = \bar{f} \quad \text{for } a \leq x \leq b \\
\text{Take } K' = K = \sup \frac{f(x)}{g(x)} = 1, \quad a \leq x \leq b
\]
The acceptance condition is

\[ U_n = \frac{f(Y_n)}{1 + f} \quad \text{or} \quad \frac{f}{1 + f} U_n \leq f(Y_n) \]

This is equivalent to having \( V_n = \frac{f}{1 + f} U_n \sim \text{unif}(0, f) \) independent of \( Y_n \sim \text{unif}[a, b] \). Thus, \((Y_n, V_n)\) is \( \text{unif}[a, b] \times [0, f] \) and the acceptance condition is that \((Y_n, V_n)\) lie on or below the curve \( f(x) \).

**Example**

\[
\begin{array}{c}
\text{In this case } q(y) = \frac{f_1}{f_1 + f_2} f_1(y) + \frac{f_2}{f_1 + f_2} f_2(y), \\
\text{where } f_1 \sim \text{unif}[a, c] \text{ and } f_2 \sim \text{unif}[c, b].
\end{array}
\]

**Prop.** In general, \((Y_n, K'g(Y_n) U_n)\) is uniformly distributed on \( A := \{(y, z) : 0 \leq z \leq K'g(y)\} \).

**Proof.** Let \( B \) be a bounded subset of \( A \). We must show that \( P((Y_n, K'g(Y_n) U_n) \in B) = \text{area}(B) / \text{area}(A) \), where \( \text{area}(A) = \int K'g(y) \, dy = K' \int q(y) \, dy = K' \beta \).

Now

\[
P((Y_n, K'g(Y_n) U_n) \in B) = \int P((y, K'g(y) U_n) \in B) q(y) \, dy
\]

\[
= \int \mathbb{1}_B(K'g(y) U_n \leq y) q(y) \, dy, \quad B_y := \{z : (y, z) \in B\}
\]

Note that \( z \in B_y \iff (y, z) \in B \cap A \Rightarrow 0 \leq z \leq K'g(y) \). Also,
\[ K'y(y) \cup \cup_{[0,K'y(y) \cup k]} \]. Thus,

\[ P((y, K'y(y) \cup k) \in B) = \int \left( \int I_{B_y(z)} \frac{1}{K'y(y)} \, dz \right) \eta(y) \, dy, \]

\[ = \int \left( \int I_{B_y(z)} \, dz \right) \frac{dy}{K'y} \quad \text{since} \quad B_y \subset [0, K'y(y)] \]

\[ = \frac{\text{area}(B)}{K'y}. \]
Simulating Random Vectors

Suppose \( f_{xyz}(x,y,z) \) is given. Write it as

\[
f_{xyz}(x,y,z) = f_{z|yx}(y|x)f_{y|x}(y|x)f_{x}(x)
\]

Then simulate \( X \sim f_{x}(x) \). If \( X=x \), simulate \( Y \) according to the one-dimensional pdf \( f_{y|x}(y|x) \) to get \( Y=y \). Then simulate \( Z \) according to the one-dimensional density \( f_{z|yx}(y|x) \).

Of course, this idea generalizes to simulate \( X_1, \ldots, X_n \). Note, however, that to generate \( X_n \), we must store the previously generated values of \( X_1=x_1, \ldots, X_{n-1}=x_{n-1} \). For large \( n \), the storage requirements become prohibitive.

Fortunately, in the Markov case, since

\[
f_{x_k|x_{k-1}, \ldots, x_1}(x_k|x_{k-1}, \ldots, x_1) = f_{x_k}(x_k|x_{k-1}),
\]

we only need to store the most recently generated value.

The Gaussian Case

To simulate \( X \sim N(0, C) \), let \( Y_1, \ldots, Y_n \) be iid \( N(0, 1) \) and put \( Y := [Y_1, \ldots, Y_n]' \). Then \( C^{1/2}Y \sim N(0, C) \) since \( E[(C^{1/2}Y)(C^{1/2}Y)'] = C^{1/2}C^{1/2} = I \). If \( \Lambda = P'C P \), \( C = P\Lambda^{1/2} P' \), or \( C = \Lambda^{1/2} P'P \Lambda^{1/2} \Rightarrow C^{1/2} = P\Lambda^{1/2} P' \) or \( C^{1/2} = \Lambda^{1/2} P'P \), where \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n) \) and \( P'P = PP' = I \).