11.6 Sampling Method I: Random Number Generation

Note that a lot of distribution functions are introduced in Kotz, Balakrishman and Johnson (2000a, 2000b, 2000c, 2000d, 2000e).

The random draws discussed in this section are based on uniform random draws between zero and one.

11.6.1 Uniform Distribution: $U(0, 1)$

Properties of Uniform Distribution: The most heuristic and simplest distribution is uniform.

The **uniform distribution** between zero and one is given by:

$$f(x) = \begin{cases} 1, & \text{for } 0 < x < 1, \\ 0, & \text{otherwise.} \end{cases}$$

Mean, variance and the moment-generating function are given by:

$$\mathrm{E}(X) = \frac{1}{2}, \qquad \mathrm{V}(X) = \frac{1}{12}, \qquad \phi(\theta) = \frac{e^\theta - 1}{\theta}.$$

Use L'Hospital's theorem to derive $\mathrm{E}(X)$ and $\mathrm{V}(X)$ using $\phi(\theta)$.

In the next section, we introduce an idea of generating uniform random draws, which in turn yield the other random draws by the transformation of variables, the inverse transform algorithm and so on.

Uniform Random Number Generators: It is no exaggeration to say that all the random draws are based on a uniform random number.

Once uniform random draws are generated, the various random draws such as exponential, normal, logistic, Bernoulli and other distributions are obtained by transforming the uniform random draws.

Thus, it is important to consider how to generate a uniform random number.

However, generally there is no way to generate exact uniform random draws.

As shown in Ripley (1987) and Ross (1997), a deterministic sequence that appears at random is taken as a sequence of random numbers.

First, consider the following relation:

$$m = k - [k/n]n,$$

where k, m and n are integers.

$[k/n]$ denotes the largest integer less than or equal to the argument.

In Fortran 77, it is written as `m=k-int(k/n)*n`, where $0 \le m < n$.

m indicates the **remainder (余り)** when k is divided by n.

n is called the **modulus (商)**.

We define the right hand side in the equation above as:

$$k - [k/n]n \equiv k \bmod n.$$

Then, using the modular arithmetic we can rewrite the above equation as follows:

$$m = k \bmod n,$$

which is represented by: `m=mod(k,n)` in Fortran 77 and `m=k%n` in C language.

A basic idea of the uniform random draw is as follows.

Given x_{i-1}, x_i is generated by:

$$x_i = (ax_{i-1} + c) \bmod n,$$

where $0 \leq x_i < n$.

a and c are positive integers, called the **multiplier** and the **increment**, respectively.

The generator above have to be started by an initial value, which is called the **seed**.

$u_i = x_i/n$ is regarded as a uniform random number between zero and one.

This generator is called the **linear congruential generator (線形合同法)**.

Especially, when $c = 0$, the generator is called the **multiplicative linear congruential generator**.

This method was proposed by Lehmer in 1948 (see Lehmer, 1951).

If n, a and c are properly chosen, the period of the generator is n.

However, when they are not chosen very carefully, there may be a lot of serial correlation among the generated values.

Therefore, the performance of the congruential generators depend heavily on the choice of (a, c).

There is a great amount of literature on uniform random number generation.

See, for example, Fishman (1996), Gentle (1998), Kennedy and Gentle (1980), Law and Kelton (2000), Niederreiter (1992), Ripley (1987), Robert and Casella (1999), Rubinstein and Melamed (1998), Thompson (2000) and so on for the other congruential generators.

However, we introduce only two uniform random number generators.

Wichmann and Hill (1982 and corrigendum, 1984) describe a combination of three congruential generators for 16-bit computers.

The generator is given by:

$$x_i = 171 x_{i-1} \bmod 30269,$$

$$y_i = 172 y_{i-1} \bmod 30307,$$

$$z_i = 170 z_{i-1} \bmod 30323,$$

and

$$u_i = \left(\frac{x_i}{30269} + \frac{y_i}{30307} + \frac{z_i}{30323} \right) \bmod 1.$$

We need to set three seeds, i.e., x_0, y_0 and z_0, for this random number generator.
u_i is regarded as a uniform random draw within the interval between zero and one.
The period is of the order of 10^{12} (more precisely the period is 6.95×10^{12}).

For 32-bit computers, L'Ecuyer (1988) proposed a combination of k congruential generators that have prime moduli n_j, such that all values of $(n_j - 1)/2$ are relatively prime, and with multipliers that yield full periods.
Let the sequence from jth generator be $x_{j,1}, x_{j,2}, x_{j,3}, \cdots$.
Consider the case where each individual generator j is a maximum-period multiplicative linear congruential generator with modulus n_j and multiplier a_j, i.e.,

$$x_{j,i} \equiv a_j x_{j,i-1} \bmod n_j.$$

Assuming that the first generator is a relatively good one and that n_1 is fairly large, we form the ith integer in the sequence as:

$$x_i = \sum_{j=1}^{k} (-1)^{j-1} x_{j,i} \bmod (n_1 - 1),$$

where the other moduli $n_j,\ j = 2, 3, \cdots, k$, do not need to be large.

The normalization takes care of the possibility of zero occurring in this sequence:

$$u_i = \begin{cases} \dfrac{x_i}{n_1}, & \text{if } x_i > 0, \\ \dfrac{n_1 - 1}{n_1}, & \text{if } x_i = 0. \end{cases}$$

As for each individual generator j, note as follows.

Define $q = [n/a]$ and $r \equiv n \bmod a$, i.e., n is decomposed as $n = aq + r$, where $r < a$.

Therefore, for $0 < x < n$, we have:

$$\begin{aligned} ax \bmod n &= (ax - [x/q]n) \bmod n \\ &= \Big(ax - [x/q](aq + r)\Big) \bmod n \\ &= \Big(a(x - [x/q]q) - [x/q]r\Big) \bmod n \\ &= \Big(a(x \bmod q) - [x/q]r\Big) \bmod n. \end{aligned}$$

Practically, L'Ecuyer (1988) suggested combining two multiplicative congruential generators, where $k = 2$, $(a_1, n_1, q_1, r_1) = (40014, 2147483563, 53668, 12211)$ and $(a_2, n_2, q_2, r_2) = (40692, 2147483399, 52774, 3791)$ are chosen.

Two seeds are required to implement the generator.

The period of the generator proposed by L'Ecuyer (1988) is of the order of 10^{18} (more precisely 2.31×10^{18}), which is quite long and practically long enough.

L'Ecuyer (1988) presents the results of both theoretical and empirical tests, where the above generator performs well.

Furthermore, L'Ecuyer (1988) gives an additional portable generator for 16-bit computers.

Also, see L'Ecuyer(1990, 1998).

To improve the length of period, the above generator proposed by L'Ecuyer (1988) is combined with the shuffling method suggested by Bays and Durham (1976), and it is introduced as `ran2` in Press, Teukolsky, Vetterling and Flannery (1992a, 1992b).

11.6.2 Transforming $U(0,1)$: Continuous Type

In this section, we focus on a continuous type of distributions, in which density functions are derived from the uniform distribution $U(0,1)$ by transformation of variables.

Normal Distribution: $N(0,1)$: The normal distribution with mean zero and variance one, i.e, the standard normal distribution, is represented by:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2},$$

for $-\infty < x < \infty$.

Mean, variance and the moment-generating function are given by:

$$\mathrm{E}(X) = 0, \qquad \mathrm{V}(X) = 1, \qquad \phi(\theta) = \exp\Big(\frac{1}{2}\theta^2\Big).$$

The normal random variable is constructed using two independent uniform random variables.

This transformation is well known as the Box-Muller (1958) transformation and is shown as follows.

Let U_1 and U_2 be uniform random variables between zero and one.

Suppose that U_1 is independent of U_2.

Consider the following transformation:

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

where we have $-\infty < X_1 < \infty$ and $-\infty < X_2 < \infty$ when $0 < U_1 < 1$ and $0 < U_2 < 1$.

Then, the inverse transformation is given by:

$$u_1 = \exp\left(-\frac{x_1^2 + x_2^2}{2}\right), \qquad u_2 = \frac{1}{2\pi}\arctan\frac{x_2}{x_1}.$$

We perform transformation of variables in multivariate cases.

From this transformation, the Jacobian is obtained as:

$$\begin{aligned} J = \begin{vmatrix} \dfrac{\partial u_1}{\partial x_1} & \dfrac{\partial u_1}{\partial x_2} \\ \dfrac{\partial u_2}{\partial x_1} & \dfrac{\partial u_2}{\partial x_2} \end{vmatrix} &= \begin{vmatrix} -x_1\exp\Bigl(-\dfrac{1}{2}(x_1^2+x_2^2)\Bigr) & -x_2\exp\Bigl(-\dfrac{1}{2}(x_1^2+x_2^2)\Bigr) \\ \dfrac{1}{2\pi}\dfrac{-x_2}{x_1^2+x_2^2} & \dfrac{1}{2\pi}\dfrac{x_1}{x_1^2+x_2^2} \end{vmatrix} \\ &= -\frac{1}{2\pi}\exp\Bigl(-\frac{1}{2}(x_1^2+x_2^2)\Bigr). \end{aligned}$$

Let $f_x(x_1, x_2)$ be the joint density of X_1 and X_2 and $f_u(u_1, u_2)$ be the joint density of U_1 and U_2.

Since U_1 and U_2 are assumed to be independent, we have the following:

$$f_u(u_1, u_2) = f_1(u_1) f_2(u_2) = 1,$$

where $f_1(u_1)$ and $f_2(u_2)$ are the density functions of U_1 and U_2, respectively.

Note that $f_1(u_1) = f_2(u_2) = 1$ because U_1 and U_2 are uniform random variables between zero and one.

Accordingly, the joint density of X_1 and X_2 is:

$$\begin{aligned} f_x(x_1, x_2) &= |J| f_u\Big(\exp(-\frac{x_1^2 + x_2^2}{2}), \frac{1}{2\pi} \arctan \frac{x_2}{x_1}\Big) \\ &= \frac{1}{2\pi} \exp\Big(-\frac{1}{2}(x_1^2 + x_2^2)\Big) \\ &= \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2}x_1^2\Big) \times \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2}x_2^2\Big), \end{aligned}$$

which is a product of two standard normal distributions.

Thus, X_1 and X_2 are mutually independently distributed as normal random variables with mean zero and variance one.

See Hogg and Craig (1995, pp.177 – 178).

Therefore, to avoid computation of the sine, various algorithms have been invented (Ahrens and Dieter (1988), Fishman (1996), Gentle (1998), Marsaglia, MacLaren and Bray (1964) and so on).

Exponential Distribution: The exponential distribution with parameter β is written as:

$$f(x) = \begin{cases} \dfrac{1}{\beta} e^{-\frac{x}{\beta}}, & \text{for } 0 < x < \infty, \\ \\ 0, & \text{otherwise,} \end{cases}$$

for $\beta > 0$.

β indicates a scale parameter.

Mean, variance and the moment-generating function are obtained as follows:

$$\mathrm{E}(X) = \beta, \qquad \mathrm{V}(X) = \beta^2, \qquad \phi(\theta) = \frac{1}{1 - \beta\theta}.$$

The relation between the exponential random variable the uniform random variable is shown as follows:

When $U \sim U(0, 1)$, consider the following transformation:

$$X = -\beta \log(U).$$

Then, X is an exponential distribution with parameter β.

Because the transformation is given by $u = \exp(-x/\beta)$, the Jacobian is:

$$J = \frac{\mathrm{d}u}{\mathrm{d}x} = -\frac{1}{\beta} \exp\Bigl(-\frac{1}{\beta} x\Bigr).$$

By transforming the variables, the density function of X is represented as:

$$f(x) = |J| f_u\Big(\exp(-\frac{1}{\beta}x)\Big) = \frac{1}{\beta}\exp\Big(-\frac{1}{\beta}x\Big),$$

where $f(\cdot)$ and $f_u(\cdot)$ denote the probability density functions of X and U, respectively.

Note that $0 < x < \infty$ because of $x = -\beta\log(u)$ and $0 < u < 1$.

Thus, the exponential distribution with parameter β is obtained from the uniform random draw between zero and one.

When $\beta = 2$, the exponential distribution reduces to the chi-square distribution with 2 degrees of freedom.

Gamma Distribution: $G(\alpha,\beta)$: The gamma distribution with parameters α and β, denoted by $G(\alpha,\beta)$, is represented as follows:

$$f(x) = \begin{cases} \dfrac{1}{\beta^\alpha\Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}}, & \text{for } 0 < x < \infty, \\ 0, & \text{otherwise,} \end{cases}$$

for $\alpha > 0$ and $\beta > 0$, where α is called a **shape parameter** and β denotes a scale parameter.

$\Gamma(\cdot)$ is called the **gamma function**, which is the following function of α:

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} \, \mathrm{d}x.$$

The gamma function has the following features:

$$\Gamma(\alpha+1) = \alpha\Gamma(\alpha), \qquad \Gamma(1) = 1, \qquad \Gamma\Big(\frac{1}{2}\Big) = 2\Gamma\Big(\frac{3}{2}\Big) = \sqrt{\pi}.$$

Mean, variance and the moment-generating function are given by:

$$\mathrm{E}(X) = \alpha\beta, \qquad \mathrm{V}(X) = \alpha\beta^2, \qquad \phi(\theta) = \frac{1}{(1-\beta\theta)^\alpha}.$$

The gamma distribution with $\alpha = 1$ is equivalent to the exponential distribution shown in Section 11.6.2.

This fact is easily checked by comparing both moment-generating functions.

Now, utilizing the uniform random variable, the gamma distribution with parameters α and β are derived as follows.

The derivation shown in this section deals with the case where α is a positive integer, i.e., $\alpha = 1, 2, 3, \cdots$.

The random variables Z_1, Z_2, $\cdots$, Z_α are assumed to be mutually independently distributed as exponential random variables with parameter β, which are shown in Section 11.6.2.

Define $X = \sum_{i=1}^{\alpha} Z_i$.

Then, X has distributed as a gamma distribution with parameters α and β, where α should be an integer, which is proved as follows:

$$\begin{aligned}\phi_x(\theta) &= \mathrm{E}(e^{\theta X}) = \mathrm{E}(e^{\theta \sum_{i=1}^{\alpha} Z_i}) = \prod_{i=1}^{\alpha} \mathrm{E}(e^{\theta Z_i}) = \prod_{i=1}^{\alpha} \phi_i(\theta) = \prod_{i=1}^{\alpha} \frac{1}{1-\beta\theta} \\ &= \frac{1}{(1-\beta\theta)^{\alpha}},\end{aligned}$$

where $\phi_x(\theta)$ and $\phi_i(\theta)$ represent the moment-generating functions of X and Z_i, respectively.

Thus, sum of the α exponential random variables yields the gamma random variable with parameters α and β.

When α is large, we have serious problems computationally in the above algorithm, because α exponential random draws have to be generated to obtain one gamma random draw with parameters α and β.

When $\alpha = k/2$ and $\beta = 2$, the gamma distribution reduces to the chi-square distribution with k degrees of freedom.

Chi-Square Distribution: $\chi^2(k)$: The chi-square distribution with k degrees of freedom, denoted by $\chi^2(k)$, is written as follows:

$$f(x) = \begin{cases} \dfrac{1}{2^{k/2}\Gamma\left(\frac{k}{2}\right)} x^{\frac{k}{2}-1} e^{-\frac{1}{2}x}, & \text{for } 0 < x < \infty, \\ 0, & \text{otherwise,} \end{cases}$$

where k is a positive integer.

The chi-square distribution is equivalent to the gamma distribution with $\beta = 2$ and $\alpha = k/2$.

The chi-square distribution with $k = 2$ reduces to the exponential distribution with $\beta = 2$, shown in Section 11.6.2.

Mean, variance and the moment-generating function are given by:

$$\mathrm{E}(X) = k, \qquad \mathrm{V}(X) = 2k, \qquad \phi(\theta) = \frac{1}{(1-2\theta)^{k/2}}.$$

F Distribution: $F(m,n)$: The F distribution with m and n degrees of freedom, denoted by $F(m,n)$, is represented as:

$$f(x) = \begin{cases} \dfrac{\Gamma\left(\frac{m+n}{2}\right)}{\Gamma\left(\frac{m}{2}\right)\Gamma\left(\frac{n}{2}\right)}\left(\dfrac{m}{n}\right)^{\frac{m}{2}} x^{\frac{m}{2}-1}\left(1+\dfrac{m}{n}x\right)^{-\frac{m+n}{2}}, & \text{for } 0 < x < \infty, \\ 0, & \text{otherwise,} \end{cases}$$

where m and n are positive integers.

Mean and variance are given by:

$$\mathrm{E}(X) = \frac{n}{n-2}, \qquad \text{for } n > 2,$$
$$\mathrm{V}(X) = \frac{2n^2(m+n-2)}{m(n-2)^2(n-4)}, \qquad \text{for } n > 4.$$

The moment-generating function of F distribution does not exist.

One F random variable is derived from two chi-square random variables.

Suppose that U and V are independently distributed as chi-square random variables, i.e., $U \sim \chi^2(m)$ and $V \sim \chi^2(n)$.

Then, it is shown that $X = \dfrac{U/m}{V/n}$ has a F distribution with (m, n) degrees of freedom.

***t* Distribution: $t(k)$:** The t distribution (or Student's t distribution) with k degrees of freedom, denoted by $t(k)$, is given by:

$$f(x) = \frac{\Gamma\left(\frac{k+1}{2}\right)}{\Gamma\left(\frac{k}{2}\right)} \frac{1}{\sqrt{k\pi}} \left(1 + \frac{x^2}{k}\right)^{-\frac{k+1}{2}},$$

for $-\infty < x < \infty$, where k does not have to be an integer but conventionally it is a positive integer.

When k is small, the t distribution has fat tails.

The t distribution with $k = 1$ is equivalent to the Cauchy distribution.

As k goes to infinity, the t distribution approaches the standard normal distribution, i.e., $t(\infty) = N(0,1)$, which is easily shown by using the definition of e, i.e.,

$$\left(1 + \frac{x^2}{k}\right)^{-\frac{k+1}{2}} = \left(1 + \frac{1}{h}\right)^{-\frac{hx^2+1}{2}} = \left(\left(1 + \frac{1}{h}\right)^{h}\right)^{-\frac{1}{2}x^2} \left(1 + \frac{1}{h}\right)^{-\frac{1}{2}} \longrightarrow e^{-\frac{1}{2}x^2},$$

where $h = k/x^2$ is set and h goes to infinity (equivalently, k goes to infinity).

Thus, a kernel of the t distribution is equivalent to that of the standard normal distribution.

Therefore, it is shown that as k is large the t distribution approaches the standard normal distribution.

Mean and variance of the t distribution with k degrees of freedom are obtained as:

$$\mathrm{E}(X) = 0, \qquad \text{for } k > 1,$$

$$\mathrm{V}(X) = \frac{k}{k-2}, \qquad \text{for } k > 2.$$

In the case of the t distribution, the moment-generating function does not exist, because all the moments do not necessarily exist.

For the t random variable X, we have the fact that $\mathrm{E}(X^p)$ exists when p is less than k.

Therefore, all the moments exist only when k is infinity.

One t random variable is obtained from chi-square and standard normal random variables.

Suppose that $Z \sim N(0, 1)$ is independent of $U \sim \chi^2(k)$.

Then, $X = Z/\sqrt{U/k}$ has a t distribution with k degrees of freedom.

Marsaglia (1984) gives a very fast algorithm for generating t random draws, which is based on a transformed acceptance/rejection method, which will be discussed later.

Gentle, J.E., 1998, *Random Number Generation and Monte Carlo Methods*, Springer-Verlag.

Kennedy, Jr. W.J. and Gentle, J.E., 1980, *Statistical Computing* (Statistics: Textbooks and Monographs, Vol.33), Marcel Dekker.

Ripley, B.D., 1987, *Stochastic Simulation*, John Wiley & Sons.

11.7 Sampling Method II: Random Number Generation

11.7.1 Rejection Sampling (棄却法)

We want to generate random draws from $f(x)$, called the **target density (目的密度)**, but we consider the case where it is hard to sample from $f(x)$.

Now, suppose that it is easy to generate a random draw from another density $f_*(x)$, called the **sampling density (サンプリング密度)** or **proposal density (提案密度)**.

In this case, random draws of X from $f(x)$ are generated by utilizing the random draws sampled from $f_*(x)$.

Let x be the the random draw of X generated from $f(x)$.

Suppose that $q(x)$ is equal to the ratio of the target density and the sampling density, i.e.,

$$q(x) = \frac{f(x)}{f_*(x)}. \tag{1}$$

Then, the target density is rewritten as:

$$f(x) = q(x) f_*(x).$$

Based on $q(x)$, the acceptance probability is obtained.

Depending on the structure of the acceptance probability, we have three kinds of sampling techniques, i.e.,

rejection sampling (棄却法) in this section, **importance resampling (重点的リサンプリング法)** in Section 11.7.2 and the **Metropolis-Hastings algorithm (メトロポリス－ハスティング・アルゴリズム)** in Section **??**.

See Liu (1996) for a comparison of the three sampling methods.

Thus, to generate random draws of x from $f(x)$, the functional form of $q(x)$ should be known and random draws have to be easily generated from $f_*(x)$.

In order for rejection sampling to work well, the following condition has to be satisfied:

$$q(x) = \frac{f(x)}{f_*(x)} < c,$$

where c is a fixed value.

That is, $q(x)$ has an upper limit.

As discussed below, $1/c$ is equivalent to the acceptance probability.

If the acceptance probability is large, rejection sampling computationally takes a lot of time.

Under the condition $q(x) < c$ for all x, we may minimize c.

That is, since we have $q(x) < \sup_x q(x) \leq c$, we may take the supremum of $q(x)$ for c.

Thus, in order for rejection sampling to work efficiently, c should be the supremum of $q(x)$ with respect to x, i.e., $c = \sup_x q(x)$.

Let x^* be the random draw generated from $f_*(x)$, which is a candidate of the random draw generated from $f(x)$.

Define $\omega(x)$ as:

$$\omega(x) = \frac{q(x)}{\sup_z q(z)} = \frac{q(x)}{c},$$

which is called the **acceptance probability (採択確率)**.

Note that we have $0 \leq \omega(x) \leq 1$ when $\sup_z q(z) = c < \infty$.

The supremum $\sup_z q(z) = c$ has to be finite.

This condition is sometimes too restrictive, which is a crucial problem in rejection sampling.

A random draw of X is generated from $f(x)$ in the following way:

(i) Generate x^* from $f_*(x)$ and compute $\omega(x^*)$.

(ii) Set $x = x^*$ with probability $\omega(x^*)$ and go back to (i) otherwise.

In other words, generating u from a uniform distribution between zero and one, take $x = x^*$ if $u \leq \omega(x^*)$ and go back to (i) otherwise.

The above random number generation procedure can be justified as follows.

Let U be the uniform random variable between zero and one, X be the random variable generated from the target density $f(x)$,

X^* be the random variable generated from the sampling density $f_*(x)$, and x^* be the realization (i.e., the random draw) generated from the sampling density $f_*(x)$.

Consider the probability $P\big(X \leq x | U \leq \omega(x^*)\big)$, which should be the cumulative distribution of X, $F(x)$, from Step (ii).

The probability $P\big(X \leq x | U \leq \omega(x^*)\big)$ is rewritten as follows:

$$P\big(X \leq x | U \leq \omega(x^*)\big) = \frac{P\big(X \leq x, U \leq \omega(x^*)\big)}{P\big(U \leq \omega(x^*)\big)},$$

where the numerator is represented as:

$$\begin{aligned} P\big(X \leq x, U \leq \omega(x^*)\big) &= \int_{-\infty}^{x}\int_{0}^{\omega(t)} f_{u,*}(u,t)\,\mathrm{d}u\,\mathrm{d}t = \int_{-\infty}^{x}\int_{0}^{\omega(t)} f_u(u) f_*(t)\,\mathrm{d}u\,\mathrm{d}t \\ &= \int_{-\infty}^{x}\Big(\int_{0}^{\omega(t)} f_u(u)\,\mathrm{d}u\Big) f_*(t)\,\mathrm{d}t = \int_{-\infty}^{x}\Big(\int_{0}^{\omega(t)} \mathrm{d}u\Big) f_*(t)\,\mathrm{d}t \\ &= \int_{-\infty}^{x}\Big[u\Big]_{0}^{\omega(t)} f_*(t)\,\mathrm{d}t = \int_{-\infty}^{x} \omega(t) f_*(t)\,\mathrm{d}t = \int_{-\infty}^{x} \frac{q(t)}{c} f_*(t)\,\mathrm{d}t = \frac{F(x)}{c}, \end{aligned}$$

and the denominator is given by:

$$P\Big(U \le \omega(x^*)\Big) = P\Big(X \le \infty, U \le \omega(x^*)\Big) = \frac{F(\infty)}{c} = \frac{1}{c}.$$

In the numerator, $f_{u,*}(u, x)$ denotes the joint density of random variables U and X^*.

Because the random draws of U and X^* are independently generated in Steps (i) and (ii) we have $f_{u,*}(u, x) = f_u(u)f_*(x)$, where $f_u(u)$ and $f_*(x)$ denote the marginal density of U and that of X^*.

The density function of U is given by $f_u(u) = 1$, because the distribution of U is assumed to be uniform between zero and one.

Thus, the first four equalities are derived.

Furthermore, in the seventh equality of the numerator, since we have:

$$\omega(x) = \frac{q(x)}{c} = \frac{f(x)}{cf_*(x)},$$

$\omega(x)f_*(x) = f(x)/c$ is obtained.

Finally, substituting the numerator and denominator shown above, we have the following equality:

$$P\Big(X \le x|U \le \omega(x^*)\Big) = F(x).$$

Thus, the rejection sampling method given by Steps (i) and (ii) is justified.

The rejection sampling method is the most efficient sampling method in the sense of precision of the random draws, because using rejection sampling we can generate mutually independently distributed random draws.

However, for rejection sampling we need to obtain the c which is greater than or equal to the supremum of $q(x)$.

If the supremum is infinite, i.e., if c is infinite, $\omega(x)$ is zero and accordingly the candidate x^* is never accepted in Steps (i) and (ii).

See, for example, Boswell, Gore, Patil and Taillie (1993), O'Hagan (1994) and Geweke (1996) for rejection sampling.

11.7.2 Importance Resampling (重点的リサンプリング)

The **importance resampling** method also utilizes the sampling density $f_*(x)$, where we should choose the sampling density from which it is easy to generate random draws.

Let x_i^* be the ith random draw of x generated from $f_*(x)$.

The acceptance probability is defined as:

$$\omega(x_i^*) = \frac{q(x_i^*)}{\sum_{j=1}^{n} q(x_j^*)},$$

where $q(\cdot)$ is represented as equation (1).

To obtain a random draws from $f(x)$, we perform the following procedure:

(i) Generate x_j^* from the sampling density $f_*(x)$ for $j = 1, 2, \cdots, n$.

(ii) Compute $\omega(x_j^*)$ for all $j = 1, 2, \cdots, n$.

(iii) Generate a uniform random draw u between zero and one and take $x = x_j^*$ when $\Omega_{j-1} \le u < \Omega_j$, where $\Omega_j = \sum_{i=1}^{j} \omega(x_i^*)$ and $\Omega_0 \equiv 0$.

The x obtained in Step (iii) represents a random draw from the target density $f(x)$.

In Step (ii), all the probability weights $\omega(x_j^*)$, $j = 1, 2, \cdots, n$, have to be computed for importance resampling.

Thus, we need to generate n random draws from the sampling density $f_*(x)$ in advance.

When we want to generate more random draws (say, N random draws), we may repeat Step (iii) N times.

In the importance resampling method, there are n realizations, i.e., x_1^*, x_2^*, $\cdots$, x_n^*, which are mutually independently generated from the sampling density $f_*(x)$.

The cumulative distribution of $f(x)$ is approximated by the following empirical distribution:

$$P(X \leq x) = \int_{-\infty}^{x} f(t)\,\mathrm{d}t = \int_{-\infty}^{x} \frac{f(t)}{f_*(t)} f_*(t)\,\mathrm{d}t = \frac{\int_{-\infty}^{x} q(t) f_*(t)\,\mathrm{d}t}{\int_{-\infty}^{\infty} q(t) f_*(t)\,\mathrm{d}t}$$

$$\approx \frac{(1/n)\sum_{i=1}^{n} q(x_i^*) I(x, x_i^*)}{(1/n)\sum_{j=1}^{n} q(x_j^*)} = \sum_{i=1}^{n} \omega(x_i^*) I(x, x_i^*),$$

where $I(x, x_i^*)$ denotes the indicator function which satisfies $I(x, x_i^*) = 1$ when $x \geq x_i^*$ and $I(x, x_i^*) = 0$ otherwise.

$P(X = x_i^*)$ is approximated as $\omega(x_i^*)$.

See Smith and Gelfand (1992) and Bernardo and Smith (1994) for the importance resampling procedure.

As mentioned in Section 11.7.1, for rejection sampling, $f(x)$ may be a kernel of the target density, or equivalently, $f(x)$ may be proportional to the target density.

Similarly, the same situation holds in the case of importance resampling.

That is, $f(x)$ may be proportional to the target density for importance resampling, too.

To obtain a random draws from $f(x)$, importance resampling requires n random draws from the sampling density $f_*(x)$, but rejection sampling needs $(1 + N_R)$ random draws from the sampling density $f_*(x)$.

For importance resampling, when we have n different random draws from the sampling density, we pick up one of them with the corresponding probability weight.

The importance resampling procedure computationally takes a lot of time, because we have to compute all the probability weights Ω_j, $j = 1, 2, \cdots, n$, in advance even when we want only one random draw.

11.7.3 Metropolis-Hastings Algorithm (メトロポリスーハスティングス・アルゴリズム)

This section is based on Geweke and Tanizaki (2003), where three sampling distributions are compared with respect to precision of the random draws from the target density $f(x)$.

The **Metropolis-Hastings algorithm** is also one of the sampling methods to generate random draws from any target density $f(x)$, utilizing sampling density $f_*(x)$, even in the case where it is not easy to generate random draws from the target density.

Let us define the acceptance probability by:

$$\omega(x_{i-1}, x^*) = \min\left(\frac{q(x^*)}{q(x_{i-1})}, 1\right) = \min\left(\frac{f(x^*)/f_*(x^*)}{f(x_{i-1})/f_*(x_{i-1})}, 1\right),$$

where $q(\cdot)$ is defined as equation (1).

By the Metropolis-Hastings algorithm, a random draw from $f(x)$ is generated in the following way:

(i) Take the initial value of x as x_{-M}.

(ii) Generate x^* from $f_*(x)$ and compute $\omega(x_{i-1}, x^*)$ given x_{i-1}.

(iii) Set $x_i = x^*$ with probability $\omega(x_{i-1}, x^*)$ and $x_i = x_{i-1}$ otherwise.

(iv) Repeat Steps (ii) and (iii) for $i = -M+1, -M+2, \cdots, 1$.

In the above algorithm, x_1 is taken as a random draw from $f(x)$.

When we want more random draws (say, N), we replace Step (iv) by Step (iv)', which is represented as follows:

(iv)' Repeat Steps (ii) and (iii) for $i = -M+1, -M+2, \cdots, N$.

When we implement Step (iv)', we can obtain a series of random draws x_{-M}, x_{-M+1}, $\cdots$, x_0, x_1, x_2, $\cdots$, x_N, where x_{-M}, x_{-M+1}, $\cdots$, x_0 are discarded from further consideration.

The last N random draws are taken as the random draws generated from the target density $f(x)$.

Thus, N denotes the number of random draws.

M is sometimes called the **burn-in period**.

We can justify the above algorithm given by Steps (i) – (iv) as follows.

The proof is very similar to the case of rejection sampling in Section 11.7.1.

We show that x_i is the random draw generated from the target density $f(x)$ under the assumption x_{i-1} is generated from $f(x)$.

Let U be the uniform random variable between zero and one, X be the random variable which has the density function $f(x)$ and x^* be the realization (i.e., the random draw) generated from the sampling density $f_*(x)$. Consider the probability $P\Big(X \le x|U \le \omega(x_{i-1}, x^*)\Big)$, which should be the cumulative distribution of X, i.e., $F(x)$.

The probability $P\Big(X \le x|U \le \omega(x_{i-1}, x^*)\Big)$ is rewritten as follows:

$$P\Big(X \le x|U \le \omega(x_{i-1}, x^*)\Big) = \frac{P\Big(X \le x, U \le \omega(x_{i-1}, x^*)\Big)}{P\Big(U \le \omega(x_{i-1}, x^*)\Big)},$$

where the numerator is represented as:

$$\begin{aligned} P\Big(X \le x, U \le \omega(x_{i-1}, x^*)\Big) &= \int_{-\infty}^{x} \int_{0}^{\omega(x_{i-1},t)} f_{u,*}(u, t)\,\mathrm{d}u\,\mathrm{d}t \\ &= \int_{-\infty}^{x} \int_{0}^{\omega(x_{i-1},t)} f_u(u) f_*(t)\,\mathrm{d}u\,\mathrm{d}t = \int_{-\infty}^{x} \Big(\int_{0}^{\omega(x_{i-1},t)} f_u(u)\,\mathrm{d}u\Big) f_*(t)\,\mathrm{d}t \\ &= \int_{-\infty}^{x} \Big(\int_{0}^{\omega(x_{i-1},t)} \mathrm{d}u\Big) f_*(t)\,\mathrm{d}t = \int_{-\infty}^{x} \Big[u\Big]_{0}^{\omega(x_{i-1},t)} f_*(t)\,\mathrm{d}t \end{aligned}$$

$$= \int_{-\infty}^{x} \omega(x_{i-1}, t) f_*(t) \, \mathrm{d}t = \int_{-\infty}^{x} \frac{f_*(x_{i-1}) f(t)}{f(x_{i-1})} \, \mathrm{d}t = \frac{f_*(x_{i-1})}{f(x_{i-1})} F(x)$$

and the denominator is given by:

$$P\Big(U \leq \omega(x_{i-1}, x^*)\Big) = P\Big(X \leq \infty, U \leq \omega(x_{i-1}, x^*)\Big) = \frac{f_*(x_{i-1})}{f(x_{i-1})} F(\infty) = \frac{f_*(x_{i-1})}{f(x_{i-1})}.$$

The density function of U is given by $f_u(u) = 1$ for $0 < u < 1$.

Let X^* be the random variable which has the density function $f_*(x)$.

In the numerator, $f_{u,*}(u, x)$ denotes the joint density of random variables U and X^*.

Because the random draws of U and X^* are independently generated, we have $f_{u,*}(u, x) = f_u(u) f_*(x) = f_*(x)$.

Thus, the first four equalities are derived.

Substituting the numerator and denominator shown above, we have the following equality:

$$P\Big(X \leq x | U \leq \omega(x_{i-1}, x^*)\Big) = F(x).$$

Thus, the x^* which satisfies $u \leq \omega(x_{i-1}, x^*)$ indicates a random draw from $f(x)$.

We set $x_i = x_{i-1}$ if $u \leq \omega(x_{i-1}, x^*)$ is not satisfied. x_{i-1} is already assumed to be a random draw from $f(x)$.

Therefore, it is shown that x_i is a random draw from $f(x)$.

See Gentle (1998) for the discussion above.

As in the case of rejection sampling and importance resampling, note that $f(x)$ may be a kernel of the target density, or equivalently, $f(x)$ may be proportional to the target density.

The same algorithm as Steps (i) – (iv) can be applied to the case where $f(x)$ is proportional to the target density, because $f(x^*)$ is divided by $f(x_{i-1})$ in $\omega(x_{i-1}, x^*)$.

As a general formulation of the sampling density, instead of $f_*(x)$, we may take the sampling density as the following form: $f_*(x|x_{i-1})$, where a candidate random draw x^* depends on the $(i-1)$th random draw, i.e., x_{i-1}.

For choice of the sampling density $f_*(x|x_{i-1})$, Chib and Greenberg (1995) pointed out as follows.

$f_*(x|x_{i-1})$ should be chosen so that the chain travels over the support of $f(x)$, which implies that $f_*(x|_{i-1})$ should not have too large variance and too small variance, compared with $f(x)$.

See, for example, Smith and Roberts (1993), Bernardo and Smith (1994), O'Hagan (1994), Tierney (1994), Geweke (1996), Gamerman (1997), Robert and Casella (1999) and so on for the Metropolis-Hastings algorithm.

Therefore, for precision of the random draws, the Metropolis-Hastings algorithm gives us the worst random number of the three sampling methods. i.e., rejection sampling in Section 11.7.1, importance resampling in

Section 11.7.2 and the Metropolis-Hastings algorithm in this section.

Based on Steps (i) – (iii) and (iv)', under some conditions the basic result of the Metropolis-Hastings algorithm is as follows:

$$\frac{1}{N}\sum_{i=1}^{N} g(x_i) \longrightarrow \mathrm{E}\Big(g(x)\Big) = \int g(x)f(x)\,\mathrm{d}x, \qquad \text{as } N \longrightarrow \infty,$$

where $g(\cdot)$ is a function, which is representatively taken as $g(x) = x$ for mean and $g(x) = (x-\overline{x})^2$ for variance. $\overline{x}$ denotes $\overline{x} = (1/N)\sum_{i=1}^{N} x_i$.

Thus, it is shown that $(1/N)\sum_{i=1}^{N} g(x_i)$ is a consistent estimate of $\mathrm{E}\Big(g(x)\Big)$, even though x_1, x_2, $\cdots$, x_N are mutually correlated.

Furthermore, the sampling density has to satisfy the following conditions:

(i) we can quickly and easily generate random draws from the sampling density and

(ii) the sampling density should be distributed with the same range as the target density.

See, for example, Geweke (1992) and Mengersen, Robert and Guihenneuc-Jouyaux (1999) for the MCMC convergence diagnostics.

Since the random draws based on the Metropolis-Hastings algorithm heavily depend on choice of the sampling density, we can see that the Metropolis-Hastings algorithm has the problem of specifying the sampling density, which is the crucial criticism.

Several generic choices of the sampling density are discussed by Tierney (1994) and Chib and Greenberg (1995).

3.4.1.1 Sampling Density I (Independence Chain)

For the sampling density, we have started with $f_*(x)$ in this section.

Thus, one possibility of the sampling density is given by: $f_*(x|x_{i-1}) = f_*(x)$, where $f_*(\cdot)$ does not depend on x_{i-1}.

This sampling density is called the **independence chain**.

For example, it is possible to take $f_*(x) = N(\mu_*, \sigma_*^2)$, where μ_* and σ_*^2 are the hyper-parameters.

Or, when x lies on a certain interval, say (a, b), we can choose the uniform distribution $f_*(x) = 1/(b - a)$ for the sampling density.

3.4.1.2 Sampling Density II (Random Walk Chain)

We may take the sampling density called the **random walk chain**, i.e., $f_*(x|x_{i-1}) = f_*(x - x_{i-1})$.

Representatively, we can take the sampling density as $f_*(x|x_{i-1}) = N(x_{i-1}, \sigma_*^2)$, where σ_*^2 denotes the hyperparameter.

Based on the random walk chain, we have a series of the random draws which follow the random walk process.

3.4.1.3 Sampling Density III (Taylored Chain)

The alternative sampling distribution is based on approximation of the log-kernel (see Geweke and Tanizaki (1999, 2001, 2003)), which is a substantial extension of the **Taylored chain** discussed in Chib, Greenberg and Winkelmann (1998).

Let $p(x) = \log(f(x))$, where $f(x)$ may denote the kernel which corresponds to the target density.

Approximating the log-kernel $p(x)$ around x_{i-1} by the second order Taylor series expansion, $p(x)$ is represented as:

$$p(x) \approx p(x_{i-1}) + p'(x_{i-1})(x - x_{i-1}) + \frac{1}{2}p''(x_{i-1})(x - x_{i-1})^2, \tag{2}$$

where $p'(\cdot)$ and $p''(\cdot)$ denote the first- and second-derivatives.

Depending on the values of $p'(x)$ and $p''(x)$, we have the three cases: $p''(x) < 0$, $p''(x) \geq 0$ and $p'(x) = 0$.

Geweke and Tanizaki (2003) suggested introducing ϵ into the Taylored chain discussed in Geweke and Tanizaki (1999, 2001).

$\boldsymbol{p''(x_{i-1}) < 0}$: Equation (2) is rewritten by:

$$p(x) \approx p(x_{i-1}) - \frac{1}{2}\Big(\frac{1}{-1/p''(x_{i-1})}\Big)\Big(x - (x_{i-1} - \frac{p'(x_{i-1})}{p''(x_{i-1})})\Big)^2 + r(x_{i-1}),$$

where $r(x_{i-1})$ is an appropriate function of x_{i-1}.

Since $p''(x_{i-1})$ is negative, the second term in the right-hand side is equivalent to the exponential part of the normal density.

Therefore, $f_*(x|x_{i-1})$ is taken as $N(\mu_*, \sigma_*^2)$, where $\mu_* = x_{i-1} - p'(x_{i-1})/p''(x_{i-1})$ and $\sigma_*^2 = -1/p''(x_{i-1})$.

11.7.4 Gibbs Sampling

The sampling methods introduced in Sections 11.7.1 – 11.7.3 can be applied to the cases of both univariate and multivariate distributions.

The Gibbs sampler in this section is the random number generation method in the multivariate cases.

The Gibbs sampler shows how to generate random draws from the unconditional densities under the situation that we can generate random draws from two conditional densities.

Geman and Geman (1984), Tanner and Wong (1987), Gelfand, Hills, Racine-Poon and Smith (1990), Gelfand and Smith (1990), Carlin and Polson (1991), Zeger and Karim (1991), Casella and George (1992), Gamerman (1997) and so on developed the Gibbs sampling theory.

Carlin, Polson and Stoffer (1992), Carter and Kohn (1994, 1996) and Geweke and Tanizaki (1999, 2001) applied the Gibbs sampler to the nonlinear and/or non-Gaussian state-space models.

There are numerous other applications of the Gibbs sampler.

The Gibbs sampling theory is concisely described as follows.

We can deal with more than two random variables, but we consider two random variables X and Y in order to make things easier.

Two conditional density functions, $f_{x|y}(x|y)$ and $f_{y|x}(y|x)$, are assumed to be known, which denote the conditional distribution function of X given Y and that of Y given X, respectively.

Suppose that we can easily generate random draws of X from $f_{x|y}(x|y)$ and those of Y from $f_{y|x}(y|x)$.

However, consider the case where it is not easy to generate random draws from the joint density of X and Y, denoted by $f_{xy}(x, y)$.

In order to have the random draws of (X, Y) from the joint density $f_{xy}(x, y)$, we take the following procedure:

(i) Take the initial value of X as x_{-M}.

(ii) Given x_{i-1}, generate a random draw of Y, i.e., y_i, from $f(y|x_{i-1})$.

(iii) Given y_i, generate a random draw of X, i.e., x_i, from $f(x|y_i)$.

(iv) Repeat the procedure for $i = -M+1, -M+2, \cdots, 1$.

From the convergence theory of the Gibbs sampler, as M goes to infinity, we can regard x_1 and y_1 as random draws from $f_{xy}(x, y)$, which is a joint density function of X and Y.

M denotes the **burn-in period**, and the first M random draws, (x_i, y_i) for $i = -M+1, -M+2, \cdots, 0$, are excluded from further consideration.

When we want N random draws from $f_{xy}(x, y)$, Step (iv) should be replaced by Step (iv)', which is as follows.

(iv)' Repeat the procedure for $i = -M+1, -M+2, \cdots, N$.

As in the Metropolis-Hastings algorithm, the algorithm shown in Steps (i) – (iii) and (iv)' is formulated as follows:

$$f_i(u) = \int f^*(u|v) f_{i-1}(v)\,\mathrm{d}v.$$

For convergence of the Gibbs sampler, we need to have the invariant distribution $f(u)$ which satisfies $f_i(u) =$

$f_{i-1}(u) = f(u)$. If we have the reversibility condition shown in equation (**??**), i.e.,

$$f^*(v|u)f(u) = f^*(u|v)f(v),$$

the random draws based on the Gibbs sampler converge to those from the invariant distribution, which implies that there exists the invariant distribution $f(u)$.

Therefore, in the Gibbs sampling algorithm, we have to find the transition distribution, i.e., $f^*(u|v)$.

Here, we consider that both u and v are bivariate vectors.

That is, $f^*(u|v)$ and $f_i(u)$ denote the bivariate distributions. x_i and y_i are generated from $f_i(u)$ through $f^*(u|v)$, given $f_{i-1}(v)$.

Note that $u = (u_1, u_2) = (x_i, y_i)$ is taken while $v = (v_1, v_2) = (x_{i-1}, y_{i-1})$ is set.

The transition distribution in the Gibbs sampler is taken as:

$$f^*(u|v) = f_{y|x}(u_2|u_1)f_{x|y}(u_1|v_2)$$

Thus, we can choose $f^*(u|v)$ as shown above.

Then, as i goes to infinity, (x_i, y_i) tends in distribution to a random vector whose joint density is $f_{xy}(x, y)$.

See, for example, Geman and Geman (1984) and Smith and Roberts (1993).

Furthermore, under the condition that there exists the invariant distribution, the basic result of the Gibbs sampler is as follows:

$$\frac{1}{N}\sum_{i=1}^{N} g(x_i, y_i) \longrightarrow \mathrm{E}\big(g(x,y)\big) = \iint g(x,y) f_{xy}(x,y)\,\mathrm{d}x\,\mathrm{d}y, \quad \text{as } N \longrightarrow \infty,$$

where $g(\cdot,\cdot)$ is a function.

The Gibbs sampler is a powerful tool in a Bayesian framework.

Based on the conditional densities, we can generate random draws from the joint density.

Remark 1: We have considered the bivariate case, but it is easily extended to the multivariate cases.

That is, it is possible to take multi-dimensional vectors for x and y.

Taking an example, as for the tri-variate random vector (X, Y, Z), if we generate the ith random draws from $f_{x|yz}(x|y_{i-1}, z_{i-1})$, $f_{y|xz}(y|x_i, z_{i-1})$ and $f_{z|xy}(z|x_i, y_i)$, sequentially, we can obtain the random draws from $f_{xyz}(x, y, z)$.

Remark 2: Let X, Y and Z be the random variables.

Take an example of the case where X is highly correlated with Y.

If we generate random draws from $f_{x|yz}(x|y,z)$, $f_{y|xz}(y|x,z)$ and $f_{z|xy}(z|x,y)$, it is known that convergence of the Gibbs sampler is slow.

In this case, without separating X and Y, random number generation from $f(x,y|z)$ and $f(z|x,y)$ yields better random draws from the joint density $f(x,y,z)$.

References

Carlin, B.P. and Polson, N.G., 1991, "Inference for Nonconjugate Bayesian Models Using the Gibbs Sampler," *Canadian Journal of Statistics*, Vol.19, pp.399 – 405.

Carlin, B.P., Polson, N.G. and Stoffer, D.S., 1992, "A Monte Carlo Approach to Nonnormal and Nonlinear State Space Modeling," *Journal of the American Statistical Association*, Vol.87, No.418, pp.493 – 500.

Carter, C.K. and Kohn, R., 1994, "On Gibbs Sampling for State Space Models," *Biometrika*, Vol.81, No.3, pp.541 – 553.

Carter, C.K. and Kohn, R., 1996, "Markov Chain Monte Carlo in Conditionally Gaussian State Space Models," *Biometrika*, Vol.83, No.3, pp.589 – 601.

Casella, G. and George, E.I., 1992, "Explaining the Gibbs Sampler," *The American Statistician*, Vol.46, pp.167 – 174.

Chib, S. and Greenberg, E., 1995, "Understanding the Metropolis-Hastings Algorithm," *The American Statistician*, Vol.49, No.4, pp.327 – 335.

Gamerman, D., 1997, *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, Chapman & Hall.

Gelfand, A.E. and Smith, A.F.M., 1990, "Sampling-Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, Vol.85, No.410, pp.398 – 409.

Liu, J.S., 1996, "Metropolized Independent Sampling with Comparisons to Rejection Sampling and Importance Sampling," *Statistics and Computing*, Vol.6, pp.113 – 119.

Mengersen, K.L., Robert, C.P. and Guihenneuc-Jouyaux, C., 1999, "MCMC Convergence Diagnostics: A Reviewww," in *Bayesian Statistics, Vol.6*, edited by Bernardo, J.M., Berger, J.O., Dawid, A.P. and Smith, A.F.M., pp.514 – 440 (with discussion), Oxford University Press.

Smith, A.F.M. and Gelfand, A.E., 1992, "Bayesian Statistics without Tears: A Sampling-Resampling Perspective," *The American Statistician*, Vol.46, No.2, pp.84 – 88.

Smith, A.F.M. and Roberts, G.O., 1993, “Bayesian Computation via Gibbs Sampler and Related Markov Chain Monte Carlo Methods,” *Journal of the Royal Statistical Society*, Ser.B, Vol.55, No.1, pp.3 – 23.

Tanner, M.A. and Wong, W.H., 1987, “The Calculation of Posterior Distributions by Data Augmentation,” *Journal of the American Statistical Association*, Vol.82, No.398, pp.528 – 550 (with discussion).

Tierney, L., 1994, “Markov Chains for Exploring Posterior Distributions,” *The Annals of Statistics*, Vol.22, No.4, pp.1701 – 1762.