2.8.2 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.

If we can generate the random daws from $f(x)$, denoted by x_i for $i = 1, 2, \cdots, n$, the expectation of a function $g(X)$ is approximately given by:

$$\mathrm{E}(g(X)) = \int g(x)f(x)\mathrm{d}x \approx \frac{1}{n}\sum_i g(x_i),$$

which implies that we choose $g(x_i)$ with probability $1/n$.

However, if it is not easy to generate the random daws from $f(x)$ and instead we can easily generate random draws from $f_*(x)$, denoted by x_i^* for $i = 1, 2, \cdots, n$, the expectation of a function $g(X)$ is given by:

$$\mathrm{E}(g(X)) = \int g(x)f(x)\mathrm{d}x = \int g(x)q(x)f_*(x)\mathrm{d}x \approx \frac{1}{n}\sum_i g(x_i^*)q(x_i^*).$$

which implies that we choose $g(x_i^*)$ with probability $q(x_i^*)/n$

Let x_i^* be the ith random draw of x generated from $f_*(x)$.
Taking into account $\dfrac{1}{n}\sum\limits_i q(x_i^*) \longrightarrow 1$, i.e., $\sum\limits_i q(x_i^*) \longrightarrow n$, 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). Note that n is replaced by $\sum_{j=1}^n q(x_j^*)$.

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) Choose one of x_j^*, $j = 1, 2, \cdots, n$, with probability $\omega(x_j^*)$, $j = 1, 2, \cdots, n$.

In other words, generate a uniform random draw u between zero and one and take $x = x_j^*$ when $\Omega_{j-1} \leq u < \Omega_j$, 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.

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.

Example of Importance Resampling: Beta Distribution $B(\alpha,\ \beta)$:

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

for $\alpha > 0$ and $\beta > 0$.

$f_*(x)$ is taken as the uniform distribution between zero and one:

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

$$q(x) = \frac{f(x)}{f_*(x)} = \frac{1}{B(\alpha,\ \beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

$\omega(x_i)$ is given by:

$$\omega(x_i) = \frac{q(x_i)}{\sum_{j=1}^{n} q(x_j)} = \frac{x_i^{\alpha-1}(1-x_i)^{\beta-1}}{\sum_{j=1}^{n} x_j^{\alpha-1}(1-x_j)^{\beta-1}}$$

From computational viewpoint, the numerator of $\omega(x_i)$ is computed as:

$$\exp\Big((\alpha-1)\log(x_i) + (\beta-1)\log(1-x_i)\Big)$$

is recommended.

B(a,b) Distribution

```
#include <math.h>
#include <stdio.h>

  int ix=1,iy=1;

void main(){

  float  a,b;
  int    i,j,n;
  double urnd(void);
  double x0[100001],q[100001],w[100001];
  double x,u,x1=0.0,x2=0.0;

  for(i=1;i<=10000;i++) urnd();

  scanf("%f%f%d",&a,&b,&n);

  w[0]=0.0;
```

```
for(i=1;i<=n;i++){
  x0[i]=urnd();
  w[i]=w[i-1]+exp( (a-1.)*log(x0[i])+(b-1.)*log(1.-x0[i]) );
}
for(i=1;i<=n;i++) w[i]/=w[n];
for(i=1;i<=n;i++){
  u=urnd();
  for(j=1;j<=n;j++){
    if( (w[j-1] <= u) && (u < w[j]) ){
      x=x0[j];
      goto LABEL;
    }
  }
  LABEL:
  x1+=x/((double)n);
  x2+=x*x/((double)n);
}

printf("# of Random Draws = %5d\n",n);
printf("Parameters = (%7.1f,%7.1f)\n",a,b);
```

```
  printf("Mean     = %10.5lf, which should be close to %10.5f\n"
         ,x1,a/(a+b));
  printf("Variance = %10.5lf, which should be close to %10.5f\n"
         ,x2-x1*x1,a*b/((a+b)*(a+b)*(a+b+1.)));

}
/* ------------------------------------------------ */
double urnd(void)
{
   int    kx,ky;
   double rn;
/*
   Input:
     ix, iy:  Seeds
   Output:
     rn: Uniform Random Draw U(0,1)
*/
   kx=ix/53668;
   ix=40014*(ix-kx*53668)-kx*12211;

```

```
59:     ky=iy/52774;
60:     iy=40692*(iy-ky*52774)-ky*3791;
61:
62:     rn=(float)(ix-iy)/2147483563.;
63:     rn-=(int)rn;
64:     if( rn<0.) rn++;
65:
66:     return rn;
67: }
```

In Lines 18 – 23, $\Omega_i = \sum_{j=1}^{i} \omega(x_j^*)$ or equivalently $\Omega_i = \Omega_{i-1} + \omega(x_i^*)$ with $\Omega_0 = 0$ is computed as `w[i]`.

In Lines 26 – 30, we search where `u` in Line 25 is.

2.8.3 Metropolis-Hastings Algorithm (メトロポリス－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:

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

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 2.8.1.

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 $P(X \leq x | U \leq \omega(x_{i-1}, x^*))$, which should be the cumulative distribution of X, i.e., $F(x)$.

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

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

where the numerator is represented as:

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

and the denominator is given by:

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

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:

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

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 2.8.1, importance resampling in Section 2.8.2 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:

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

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 $\mathrm{E}(g(x))$, 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 hyper-parameter.

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 $\sigma_*^2 = -1/p''(x_{i-1})$.