MC Simulations

(1) <u>Random number:</u>

Computer doesn't know randomness. So the number generated by it randomly will be called pseudorandom numbers.

Due to finite arithematics of computer for generating $u_k \sim U[0,1]$: we consider $i_k \in \{0, 1, \cdots m-1\}$ and $u_k = i_k/m$.

$\Rightarrow$ Next, we only consider to generate random number $i_k \in \{0, 1, \cdots m-1\}$ uniformly.

<u>Def</u>: A Random number generator (RNG) is $(X, x_0, T, G, \{0, 1, \cdots m-1\})$. X is finite set. $x_0 \in X$ is seed. $T: X \to X$ is transition func. $G: X \to \{0, 1, \cdots m-1\}$ is output func. We let $x_n = T(x_{n-1})$, $i_n = G(x_n)$ recursively.

Rmk: i) i_l will be periodic. Note $\exists$ $x_k = x_l$. Then by recursion: $x_{k+1} = x_{l+1} \cdots \Rightarrow i_{k,+k} = i_{k,+l}$. $\forall k_1$

ii) Criteria of goodness of RNG:

a) Stat. uniformity: No computationally feasible test can distinguish (i_k) and truly random number

b) Speed. c) Period length

e.g. (linear congruential generators)

$X = \{0, 1, \cdots m-1\}$. $i_l = h(x_l) = x_l$. And $x_{l+1} \equiv (a x_l + c) \bmod m$.

Rmk: i) For $c \neq 0$. we generally require

a) $(a, c) = 1$. b) $4 \mid m \Rightarrow 4 \mid a-1$

c) $\forall p$. prime. $p \mid m \Rightarrow p \mid a-1$.

ii) Weakness: Note for truly random $(I_l, \cdots I_{l+d-1})$ will be uniformly distribute on $\{0, \cdots m-1\}^d$

But RNG $(i_k, \cdots i_{k+d-1})$ only falls

on a small hyperplane of $\{0, 1, \cdots m-1\}^d$

(2) <u>Random variables</u>:

Assume we can produce (u.k) uniform

random number by some perfect RNG.

① Inversion method: F^{-1} is quantile of

r.v. X. Then: $F^{-1}(u) \sim X$.

<u>Rmk</u>: i) Sometimes explicit inverse F^{-1}

don't exist. We can try the

numerical inverse.

ii) One goodness of it is that it

can transfer the structure

prop.: let $u^* = \max_{1 \le i \le n} \{u_i\}$. $\Rightarrow$

$F^{-1}(u^*) \sim \max_{1 \le i \le n} \{X_i\}$.

② Acceptance - Reject method:

Let $f: \mathbb{R}^d \to \mathbb{R}^{\ge 0}$ is density. We can sample efficiently. And we want to sample from another density $f : \mathbb{R}^d \to \mathbb{R}^{\geq 0}$.

<u>Assume</u>: $\exists c \geq 1$. s.t. $f(x) \leq c\,g(x)$. $\forall x \in \mathbb{R}^d$.

<u>Algo.</u>: Given RNG producing $X \overset{i.i.d.}{\sim} g$ and $U \sim U[0,1]$. Let U is indept of X.

i) Generate one X and one U.

ii) If $U \leq f(X)/c\,g(X)$ return X.

else go back to i).

<u>Rmk</u>: Note $\mathbb{P}(g(X)=0) = \int_{\{g(x)=0\}} g(x) = 0$

<u>Prop.</u> Let Y is outcome of Algo. above. $\Rightarrow$

$Y \sim f$. And the loop in the algo. has to be traversed c times averagely.

<u>Pf</u>: $\mathbb{P}(u \leq f(x)/(c\,g(x))) =$

$$\iint \mathbb{I}\{u \leq f(x)/c\,g(x)\}\, g(x)\, du\, dx = 1/c$$

<u>Rmk</u>: i) So we expect c is as small as possible.

ii) If $c < 1$. then $\mathbb{P}(\Rightarrow) > 1$. So it isn't proba. !

If $c = 1$. then X always $= Y$.

③ Ziggurate algo.:

It's inefficient if we feed with true r.v. to produce desired r.v. e.g. c is large. So next we construct an variant of acceptance - rejection method.

The idea is: sample from the density f $\Leftrightarrow$ sample a pt uniformly from the area between 0 and graph of f. Besides we can approxi. the area by rectangles.

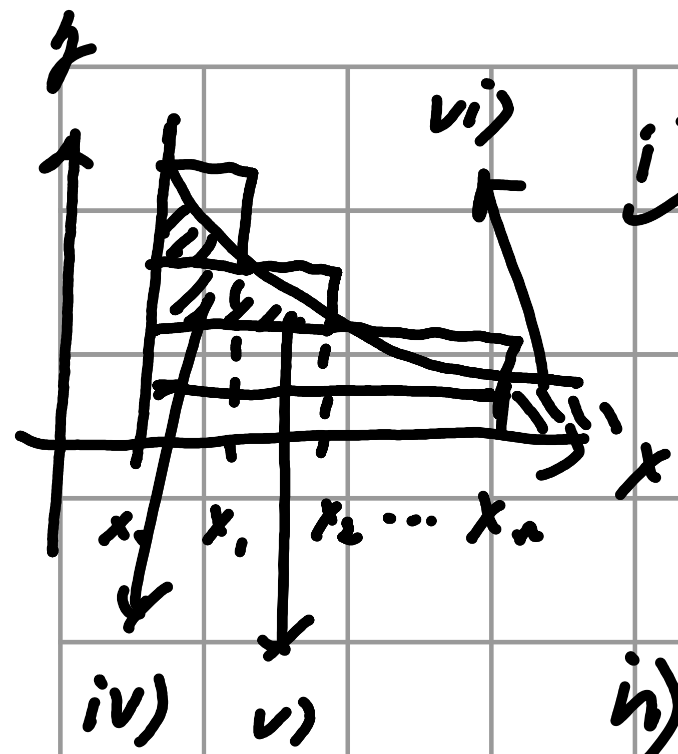
<u>Assume</u>: i) density $f: \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$ $\downarrow$ like exp.

ii) $0 = x_0 < x_1 < \cdots < x_n$. s.t. $y_i = f(x_i)$ and

$$x_i(y_{i-1} - y_i) = x_n y_n + \int_{x_n}^{\infty} f(x)\,dx =: V.$$

iii) We know dist. $X \mid X > x_n$.

<u>Algo.</u> (goal is sampling $X \sim f$).



i) generate $i \in \{1, \cdots n\}$ uniformly

(choose a rectangle randomly)

ii) $i = n \Rightarrow$ go to iv)

iii) generate $u_1 \sim U[0,1]$. Set $X = u_1 x_i$

iv) if $X < x_{i-1}$, return X.

(fall inside area of graph f)

v) otherwise generate $u_2 \sim U[0,1]$. Set

$y = y_i + u_2 (y_{i-1} - y_i)$. if $y < f(x)$, $\Rightarrow$

return X. else go back to i)

vi) generate $u_1 \sim U[0,1]$. Set $X = V u_1 / y_n$

vii) if $X < x_n$, return X. otherwise return

a sample from dist. $X \mid X > x_n$.

__Rmk__: Most stop at iv). It's fast.

Next, we introduce 2 methods to generate

$(X_1, X_2) \sim N(0, I_2)$:

④ Box-Muller method:

__Algo.__: i) generate $u_i \overset{i.i.d}{\sim} U[0,1]$, $i = 1, 2$.

ii) Set $\theta = 2\pi u_2$, $e = (-2\log u_1)^{\frac{1}{2}}$.

iii) Return $(X_1, X_2) = (e\cos\theta, e\sin\theta)$

Rmk: $\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = h(u_1, u_2) = \begin{pmatrix} (-2\log u_1)^{\frac{1}{2}}\cos 2\pi u_2 \\ (-2\log u_1)^{\frac{1}{2}}\sin 2\pi u_2 \end{pmatrix}$. So

density is $|dh^{-1}/du| = -e^{-\frac{1}{2}(u_1^2+u_2^2)}/2\pi$.

5) Polar method:

Alg. i) generate $u_i \overset{i.i.d}{\sim} U[-1,1]$, $i=1,2$.

ii) $S = u_1^2 + u_2^2$.

iii) if $S<1$, return $\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} u_1(-2\ln S/S)^{1/2} \\ u_2(-2\ln S/S)^{1/2} \end{pmatrix}$

else go back to i).

Rmk: It's more efficient than Box-Muller method since it doesn't need to compute trigo. func.

Rmk: To generate $N(\mu, \Sigma)$, let $\Sigma = AA^T$, then $\mu + AX \sim N(\mu, \Sigma)$.

(3) Monte Carlo:

By SLLN: $\lim_{m\to\infty} I_m(f,X) \overset{a.s.}{=} E(f(X)) =: I(f,X)$

where $I_m(f,X) = \frac{1}{m}\sum_{1}^{m} f(X_k)$. $X_k \overset{i.i.d}{\sim} X$.

Set: $\varepsilon_m = I(f,X) - I_m(f,X)$

Rmk: $\mathbb{E}(\varepsilon_m) = 0 \Rightarrow I_m$ is unbiased.

prop. let $\sigma = \sigma(f,X) < \infty$. std. var. of $f(X)$.

$\Rightarrow \mathbb{E}(\varepsilon_m(f,X)^2)^{\frac{1}{2}} = \sigma/\sqrt{m}$.

And $\sqrt{m}\,\varepsilon_m \sim AN(0,\sigma^2)$.

Rmk: i) Error ε_m is probabilistic.

ii) MSE has order $1/2$.

iii) The analysis should assume we know σ^2.

① Curse of dim:

Compare to the traditional method for d-dim r.v. $X \sim U[0,1]^d$. $I(f,X) = \int_{[0,1]^d} f(x)dx$

It base on grid $\{X_1, \cdots X_N\}^d$. size is N^d.

Given method of order k. error $\sim N^{-k}$.

But we have to evalute N^d pts. So:

its accuracy is $N^{-k/d}$.

~~Rmk~~ So it's unfeasible if k large.

But MC method has no such problem.

② Variance reduction:

The idea is to find r.v. Y and func. g

s.t. $\mathbb{E}(g(Y)) = \mathbb{E}(f(X))$. $Var(g(Y)) < Var(f(X))$.

i) Antithetic variates:

Assume we know a simple transf. $\tilde{X} \sim X$.

e.g. $B \sim N(0, t^2)$. $-B \sim B$

Def: $I_m^A(f, X) = m^{-1} \sum_1^m (f(X_i) + f(\tilde{X}_i))/2$.

Note computational cost of I_m^A won't exceed compute of I_{2m}. So hopefully we require:

$Var(\frac{1}{2}(f(X_i) + f(\tilde{X}_i)))/m < Var(f(X_i))/2m$.

$\Longleftrightarrow Cov(f(X), f(\tilde{X})) < 0$.

ii) Control Variates:

Assume we have r.v. Y and func. g. s.t.

$\mathbb{E}(g(Y)) = I(g, Y)$ is known.

$\Rightarrow \mathbb{E}(f(X) - \lambda(g(Y) - I(g, Y))) = I(f, X)$.

So: let $I_m^{c.v}(f, X) = m^{-1} \sum_1^m (f(X_i) - \lambda g(Y_i)) + \lambda I(g, Y)$

where $X_i, Y_i \overset{i.i.d}{\sim} X, Y$.

And $I_m^{c.v}$ cost at most twice cost of I_m. (When $X = Y$ it takes less)

We choose optimal $\lambda^* = Cov(f(X), g(Y)) / Var(g(Y))$ to reduce $Var(I_m^{c.v})$

Rmk: i) $Cov(g(Y), f(X)) \uparrow$. Var-reduced $\uparrow$.

ii) $Cov(g(Y), f(X))$ need to be known.

iii) Stratified sampling:

The idea is to partition $X = \sum_1^L A_\ell$. disjoint strata. And estimate:

$\mathbb{E}(f(X)) = \sum_1^L \mathbb{E}(f(X) | X \in A_\ell) p_\ell$. where $p_\ell = \mathbb{P}(X \in A_\ell)$.

Set $m_\ell = \#\{X_k \in A_\ell \mid k \le m\}$. $q_\ell = m_\ell / m$.

$$\Rightarrow I_m^{ST}(f, X) = \sum_1^L p_\ell \cdot \frac{1}{m_\ell} \sum_{X_k \in A_\ell} f(X_k)$$
$$= \frac{1}{m} \sum_1^L \frac{p_\ell}{q_\ell} \sum_{X_k \in A_\ell} f(X_k)$$

Rmk: Stratum can depend on another r.v. Z called stratifying r.v. And let $p_\ell = \mathbb{P}(Z \in A_\ell)$.

Next, we want to optimize the following variables: Z, $(A_\ell)_1^L$, $(m_\ell)_1^L$ and also know how to efficiently sample from $(X, Z) \mid Z \in A_\ell$.

Denote: $\mu_\ell = \mathbb{E}(f(X) \mid Z \in A_\ell)$.
$$\sigma_\ell^2 = \mathrm{Var}(f(X) \mid Z \in A_\ell).$$

For m large enough, $q_\ell \sim p_\ell$. Assume: proportional allocation $q_\ell = p_\ell$ holds.

So: $MSE(I_m^{ST}) = \mathrm{Var}(I_m^{ST}) = m^{-1} \sum_1^L p_\ell \sigma_\ell^2$.

$$MSE(I_m) = \mathrm{Var}(I_m) = m^{-1} \mathrm{Var}(f(X))$$
$$= m^{-1}\left(\sum_1^L p_\ell \mathbb{E}(f(X)^2 \mid Z \in A_\ell) - \left(\sum_1^L p_\ell \mu_\ell\right)^2\right)$$
$$= m^{-1}\left(\sum_1^L p_\ell (\sigma_\ell^2 + \mu_\ell^2) - \left(\sum_1^L p_\ell \mu_\ell\right)^2\right)$$

$\Rightarrow MSE(I_m) = MSE(I_m^{ST}) + m^{-1}(\sum p_i \mu_i^2 - (\sum p_i \mu_i)^2)$

With Cauchy inequi.: $\sum p_i \mu_i^2 \geq (\sum p_i \mu_i)^2$.

So stratified sampling can really reduce the variance / MSE.

Rmk: In general, $Var(I_m^{ST}) = m^{-1} \sum_1^L \frac{p_i^2}{q_i} \sigma_i^2$.

and the optimizer $q_i^* = p_i \sigma_i / \sum p_k \sigma_k$.

$\Rightarrow Var(I_m^{ST,*}) = m^{-1} (\sum_1^L p_i \sigma_i)^2$

Rmk: We should know σ_i here as well. $\Rightarrow$ simulate q_i^* first.

④ Importance sampling:

Given another r.v. $Y \sim$ density q. Note:

$I(f, X) = \int f p = \int f \frac{p}{q} q \, dx = I(f \frac{p}{q}, Y)$

$\Rightarrow$ Set $\tilde{I}_m(f, X) = m^{-1} \sum_1^m f(Y_i) p(Y_i) / q(Y_i)$

To choose q, i.e. reduce $Var(f(Y) \frac{p(Y)}{q(Y)})$, take q is proportional to fp. $\Rightarrow fp/q$ is flat so Var. ≈ 0. (In fact, we need to know $I(f) = \int f p$ to normalize q (most accurate))