10 MCMC

In general, $\hat{\mu}_{MC} \xrightarrow{a.s.} \int h(x) f_X(x)\,dx = \mu$

$\Leftrightarrow \hat{\mu}_{MC} = \frac{1}{n}\sum_{i=1}^{n} h(x_i) \approx E(h(x))$ as $n \to \infty$

$V(\hat{\mu}_{MC}) = V\left[\frac{1}{n}\sum_{i=1}^{n} h(x_i)\right] = \frac{1}{n^2}\sum_{i=1}^{n} V[h(x_i)] = \frac{1}{n}V[h(x)] = \frac{1}{n}E[h(x)-\mu]^2$

$\Rightarrow \hat{V}(\hat{\mu}_{MC}) = \frac{1}{n^2}\sum_{i=1}^{n}[h(x_i)-\hat{\mu}_{MC}]^2 \approx V[h(x)]$ as $n \to \infty$.

where $x_1, \dots x_n$ are generated from $f_X(x)$ by:

(1) Transformation sampling

(2) Acceptance-Rejection sampling

(3) Importance sampling

(4) Gibbs sampler

(5) Metropolis-Hastings algorithm.

} also known as Markov Chain MC

(1) Transformation Sampling

method: $F_X^{-1}(u) \sim F_X$

steps:
1. Generate $U = u$ from Unif(0, 1).
2. Compute $x = F^{-1}(u)$
3. Deliver $X = x$.

examples:

Distribution	Method
Uniform	See [171, 198, 334, 455, 456, 468]. For $X \sim$ Unif(a, b); draw $U \sim$ Unif(0, 1); then let $X = a + (b-a)U$.
Normal(μ, σ^2) and lognormal(μ, σ^2)	Draw $U_1, U_2 \sim$ i.i.d. Unif(0, 1); then $X_1 = \mu + \sigma\sqrt{-2\log U_1}\cos\{2\pi U_2\}$ and $X_2 = \mu + \sigma\sqrt{-2\log U_1}\sin\{2\pi U_2\}$ are independent $N(\mu, \sigma^2)$. If $X \sim N(\mu, \sigma^2)$ then $\exp\{X\} \sim$ lognormal(μ, σ^2).
Multivariate $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Generate standard multivariate normal vector, $\mathbf{Y}$, coordinatewise; then $\mathbf{X} = \boldsymbol{\Sigma}^{-1/2}\mathbf{Y} + \boldsymbol{\mu}$.
Cauchy(α, β)	Draw $U \sim$ Unif(0, 1); then $X = \alpha + \beta\tan\{\pi(U - 1/2)\}$.
Exponential(λ)	Draw $U \sim$ Unif(0, 1); then $X = -(\log U)/\lambda$.
Poisson(λ)	Draw $U_1, U_2, \dots \sim$ i.i.d. Unif(0, 1); then $X = j - 1$, where j is the lowest index for which $\prod_{i=1}^{j} U_i < e^{-\lambda}$.
Gamma(r, λ)	See Example 6.1, references, or for integer r, $X = -\frac{1}{\lambda}\sum_{i=1}^{r}\log U_i$ for $U_1, \dots, U_r \sim$ i.i.d. Unif(0, 1).
Chi-square (df$= k$)	Draw $Y_1, \dots, Y_k \sim$ i.i.d. $N(0, 1)$, then $X = \sum_{i=1}^{k} Y_i^2$; or draw $X \sim$ Gamma$(k/2, 1/2)$. (Y_i: N(0,1))
Student's t (df$= k$) and $F_{k,m}$ distribution	Draw $Y \sim N(0, 1)$, $Z \sim \chi^2_k$, $W \sim \chi^2_m$ independently, then $X = Y/\sqrt{Z/k}$ has the t distribution and $F = (Z/k)/(W/m)$ has the F distribution. (Y: N(0,1); Z: χ^2_k; Z: χ^2_k; W: χ^2_m)
Beta(a, b)	Draw $Y \sim$ Gamma$(a, 1)$ and $Z \sim$ Gamma$(b, 1)$ independently; then $X = Y/(Y + Z)$. (Y: $\gamma(a,1)$; Z: $\gamma(b,1)$)
Bernoulli(p) and Binomial(n, p)	Draw $U \sim$ Unif(0, 1); then $X = 1_{\{U<p\}}$ is Bernoulli(p). The sum of n independent Bernoulli(p) draws has a Binomial(n, p) distribution. $\rightarrow \sum_{i=1}^{n} I\{U_i < p\} = X$ is Bi(n, p)
Negative Binomial(r, p)	Draw $U_1, \dots, U_r \sim$ i.i.d. Unif(0, 1); then $X = \sum_{i=1}^{r}\lfloor(\log U_i)/\log\{1-p\}\rfloor$, and $\lfloor\cdot\rfloor$ means greatest integer.
Multinomial$(1, (p_1, \dots, p_k))$	Partition [0, 1] into k segments so the ith segment has length p_i. Draw $U \sim$ Unif(0, 1); then let X equal the index of the segment into which U falls. Tally such draws for Multinomial$(n, (p_1, \dots, p_k))$. $\rightarrow$ U_i: $\boxed{P_1 \mid P_2 \mid \cdots \mid P_k}$; $X_1 = \sum_{i=1}^{n} I\{U_i < P_1\}$ ⋮ $X_k = \sum_{i=1}^{n} I\{U_i < P_k\}$
Dirichlet$(\alpha_1, \dots, \alpha_k)$	Draw independent $Y_i \sim$ Gamma$(\alpha_i, 1)$ for $i = 1, \dots, k$; then $\mathbf{X}^T = \left(Y_1/\sum_{i=1}^{k} Y_i, \dots, Y_k/\sum_{i=1}^{k} Y_i\right)$. ($Y_i$: $\gamma(\alpha_i, 1)$)

* doesn't always have close form of F^{-1}

$\Downarrow$

(2) Acceptance-Rejection sampling.

2.1 rejection sampling I

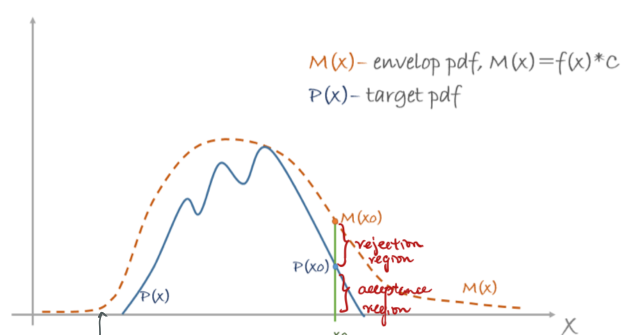
algorithm:

1. Generate $X = x$ from $f(x)$
2. Generate $Y = y$ from $U(0,1)$
3. If $y \le \frac{p(x)}{M(x)}$ then deliver $z = x$; otherwise reject

repeat until we have a distribution for z.

note:

- $M(x) \ge p(x)$ for all x that is in the support set
- $f(x)$ is a pdf that has the form $f(x) = C^{-1}M(x)$, where $C = \int_x M(x)\,dx$; (i.e. induced by $M(x)$) it should be chosen such that it's easy to draw from
- $M(x)$ shouldn't be too far away from $p(x)$. Ideally,
- hard part is choosing $M(x)$ such it has $f(x)$ in its form and it's close to $p(x)$.



$P(x_0)/M(x_0) < 1$ by definition

From all the generated values in the small neighbourhood of x_0, we will keep only $P(x_0)/M(x_0)$ (proportion) of them.

The rule $y \le P(x_0)/M(x_0)$ ensures this. ($y \sim U[0,1]$)

example: derive an A-R sampling algorithm for generating random numbers from Beta(a, b)

(i) $a \geq 1$ and $b \geq 1$.

$$p(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} I_{(0<x<1)}$$

Since $\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} I_{(0<x<1)} \leq \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} I_{(0<x<1)}$, $M(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} I_{(0<x<1)}$

Then the pdf induced by $M(x)$ is $f(x) = I_{(0<x<1)}$ i.e. $U(0,1)$

So $\frac{p(x)}{M(x)} = x^{a-1}(1-x)^{b-1}$

$\Rightarrow$ AR algorithm would be:

1. generate $X=x$ from $U(0,1)$
2. generate $Y=y$ from $U(0,1)$
3. If $y \leq x^{a-1}(1-x)^{b-1}$ then deliver $z=x$; otherwise reject

(ii) $a<1$ and $b<1$.

$$M(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\left[2^{1-b}x^{a-1} I_{(0<x\leq\frac{1}{2})} + 2^{1-a}(1-x)^{b-1} I_{(\frac{1}{2}<x<1)}\right]$$

So the induced pdf is

$$f(x) = \frac{ab}{a+b}\left[2^a x^{a-1} I_{(0<x\leq\frac{1}{2})} + 2^b(1-x)^{b-1} I_{(\frac{1}{2}<x<1)}\right]$$

acceptance probability and efficiency:

- efficiency of AR sampling: let $h(x)$ be the rejection rule

$$\tau = P(Y \leq h(x)) = \int_{-\infty}^{\infty}\int_{-\infty}^{h(x)} f(x,y)\,dx\,dy$$

$\Rightarrow$ for rejection sampling I, $\tau = P(\underbrace{Y}_{\text{uniform}(0,1)} \leq \frac{p(x)}{M(x)}) = \int_{-\infty}^{\infty}\int_{-\infty}^{p(x)/M(x)} 1 \cdot f(x)\,dx\,dy = \int_{-\infty}^{\infty} \frac{p(x)}{M(x)} f(x)\,dx = \frac{1}{c}\underbrace{\int_{-\infty}^{\infty} p(x)\,dx}_{=1} = \frac{1}{c}$ ($f(x) = c^{-1}M(x)$)

then number of trials to achieve the acceptance of $z=x$ is:

$\Rightarrow$ for rejection sampling II, $\tau = P(U \leq h(x)) = \int_{-\infty}^{\infty} h(x)g(x)\,dx = \frac{1}{L}$

$$P(T=t) = P(Y>h(x))^{t-1}P(Y \leq h(x)) = \tau(1-\tau)^{t-1} \sim Geo(\tau)$$

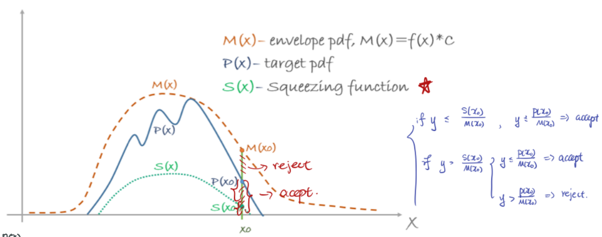
$\Rightarrow T \sim Geo(\tau) \Rightarrow E(T) = \tau^{-1}$ and $V(T) = \tau^{-2}(1-\tau)$

$\Rightarrow$ for rejection sampling I, $E(T) = c$. i.e. for rejection sampling I, expected # draws until acceptance is c, or $Pr(\text{acceptance}) = \frac{1}{c}$

2.2 Squeezed Rejection Sampling

algorithm:

1. Generate $X=x$ from $f(x)$
2. Generate $Y=y$ from $U(0,1)$
3. If $y \leq \frac{s(x)}{M(x)}$, then deliver $z=x$
4. Otherwise, compare y with $\frac{p(x)}{M(x)}$. If $y \leq \frac{p(x)}{M(x)}$, deliver $z=x$; otherwise reject



note:

- $s(x)$ is known as squeezing function such that $s(x) \leq p(x)$ on the support of $p(x)$.
- squeezed rejection sampling does not change the overall acceptance efficiency $\tau = \frac{1}{c}$.

2.3 Adaptive Rejection Sampling. (an automatic envelope generation strategy for squeezed rejection sampling for a continuous, differentiable, log-concave density)

Suppose we have initial grid T_k; envelope $e_k(x)$ and the squeezer $s_k(x)$, and the pdf $f_k(x)$ induced from $e_k(x)$ generated by:

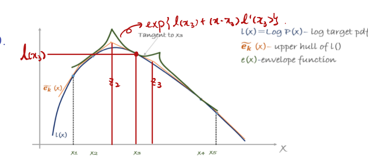
(1) log target function: $l(x) = \log p(x)$

(2) 1st derivative of $l(x)$: $l'(x) = \frac{d}{dx}l(x)$ at k points, $x_1, x_2, \ldots x_k \Rightarrow T_k = \{x_1, \ldots x_k\}$ ($l'(x_1) > 0$, $l'(x_k) < 0$.)

(3) upper hull of l: $\tilde{e}_k(X) = l(x_i) + (x-x_i)l'(x_i)$ for $x \in [z_{i-1}, z_i]$, $i = 1, 2, \ldots k-1$

(piecewise linear upper hull of l formed by the tangents to l at each point in T_k)

where $z_i = \frac{l(x_{i+1}) - l(x_i) - x_{i+1}l'(x_{i+1}) + x_i l'(x_i)}{l'(x_i) - l'(x_{i+1})}$, is the intersect of tangents of x_i and x_{i+1}

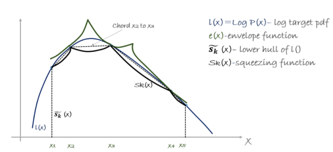


(4) upper hull of p: $e_k(X) = e^{\tilde{e}_k(X)}$ $\Rightarrow$ envelope.

(exponential of the above)

(5) lower hull of l: $\hat{S}_k(X) = \frac{(x_{i+1}-x)l(x_i) + (x-x_i)l(x_{i+1})}{x_{i+1}-x_i}$ for $x \in [x_i, x_{i+1}]$

(piecewise linear lower hull of l formed by the chords between adjacent points in T_k)

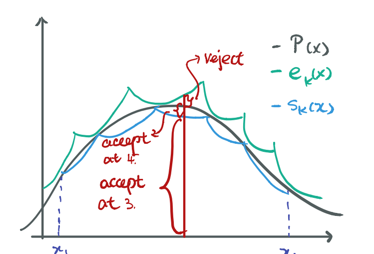


(6) lower hull of p: $S_k(X) = e^{\tilde{S}_k(X)}$ $\Rightarrow$ squeezer

(exponential of the above)

algorithm:

1. Generate $X=x$ from $f_k(x)$ (induced from $e_k(x)$)
2. Generate $Y=y$ from $U(0,1)$
3. If $y \leq \frac{S_k(x)}{e_k(x)}$, deliver $z=x$ and go to 6.
4. If $y \leq \frac{p(x)}{e_k(x)}$, deliver $z=x$ and update T_k by including x to get $T_{k+1} = \{x_1, \ldots x_{k+1}\}$ and update $e_k(x)$, $S_k(x)$ and $f_k(x)$ to $e_{k+1}(x)$, $S_{k+1}(x)$ and $f_{k+1}(x)$ accordingly. Then go to 6.
5. If $y \geq \frac{p(x)}{e_k(x)}$, reject x.
6. Repeat with updated T_{k+1}, $e_{k+1}(x)$, $S_{k+1}(x)$ and $f_{k+1}(x)$ if any to generate new samples from $p(x)$

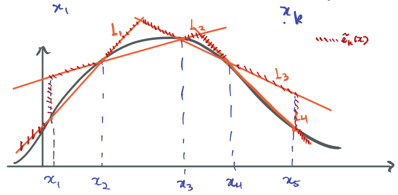


note:

- Adaptive Rejection algorithm still works when $p(x) = \tilde{c}\tilde{p}(x)$ with $\tilde{c}$ being an intractable constant.
- when $l(x)$ is not differentiable, but $p(x)$ is log-concave, $e_k(X) = e^{\tilde{e}_k(X)}$ can be:

$$\tilde{e}_k(X) = \begin{cases} \min\{L_{i-1}(X), L_{i+1}(X)\} & \text{for } x \in [x_i, x_{i+1}] \\ L_1(X) & \text{for } x < x_1 \\ L_{k-1}(X) & \text{for } x > x_k \end{cases}$$

where $L_i(x)$ is the straight line function connecting $(x_i, l(x_i))$ and $(x_{i+1}, l(x_{i+1}))$ for $i = 1, \ldots k-1$



2.4 fundamental theorem of A-R sampling.

let $f(x)$ and $g(x)$ be 2 pdf's and $h(x)$ be a given real-valued function.

algorithm:

1. Generate $X=x$ from $f(x)$
2. Generate $Y=y$ from $g(y)$
3. If $y \le h(x)$ then deliver $z=x$; otherwise reject

Then pdf of z is $p(z) = \dfrac{f(z)G(h(z))}{\int_{-\infty}^{\infty} f(s)G(h(s))ds}$. $\rightarrow$ x, Y, z can be multivariate

* AR requires large sample size, large variance and bad estimation of tails

(3) Importance Sampling

method: simulate the events of interest with higher frequency to improve the accuracy of their estimation. $\Rightarrow$ avoid oversampling tails

equations: $\mu = E[h(x)] = \dfrac{\int h(x)f(x)dx}{\int f(x)dx_{=1}} = \dfrac{\int h(x)\frac{f(x)}{g(x)}g(x)dx}{\int \frac{f(x)}{g(x)}g(x)dx}$ $x \sim g(x)$

$\Rightarrow \hat{\mu}_{IS} = \frac{1}{n}\sum_{i=1}^{n} h(x_i) = \dfrac{\sum_i h(x_i)\frac{f(x_i)}{g(x_i)}}{\sum_i \frac{f(x_i)}{g(x_i)}}$ where $X \sim g(x)$ with its support covering that of $h(x)f(x)$ and $f(x)$

$\hookrightarrow$ important sampling function or envelope

$\Rightarrow \hat{\mu}_{IS} = \sum_i h(x_i)w(x_i)$ where $w(x_i) = \dfrac{f(x_i)/g(x_i)}{\sum_j f(x_j)/g(x_j)}$

note:
- $w(x_i)$ is also known as standardized weights; if we use unstandardized weights (i.e. $\tilde{w}_i = f(x_i)/g(x_i)$), it might not be feasible to compute $\tilde{w}(x_i)$ when $f(x)$ is only up to a proportionality constant.
- $f(x)$ should be easier to draw samples from than $g(x)$
- If $g(x)$ is chosen such that the weights $\tilde{w}(x)$ or $w(x)$ are largely uniform over those influential x points (events of interest), and do not have large values over insignificant points, the resultant estimators will have smaller variance and converge to $\hat{\mu}_{MC}$ more quickly.

examples:

(1) Calculate $E(h(x))$ on $[0,10]$ where $h(x) = \exp\{-2|x-5|\}$ and $X \sim U(0,10)$

$E(h(x)) = \int_0^{10} \exp\{-2|x-5|\}\frac{1}{10}dx$

$\Rightarrow \hat{\mu}_{MC} = \frac{1}{n}\sum_{i=1}^{n} \exp\{-2|x_i-5|\}$ where $X_i \sim U(0,1)$

let $X \sim N(5,1)$, i.e. $g(x) = \frac{1}{\sqrt{2\pi}}\exp\{-(x-5)^2/2\}$

$\Rightarrow \hat{\mu}_{IS} = \frac{1}{n}\sum_{i=1}^{n} \exp\{-2|x_i-5|\} \dfrac{\frac{1}{10}}{\frac{1}{\sqrt{2\pi}}\exp\{-(x_i-5)^2/2\}}$

$= \frac{1}{n}\sum_{i=1}^{n} \frac{\sqrt{2\pi}}{10}\exp\{(x_i-5)^2/2 - 2|x_i-5|\}$ where $X_i \sim N(5,1)$

(2) Estimate $\Phi(-5.5)$ using IS

let $X \sim \exp(1) + 5.5$, i.e. $g(x) = e^{x-5.5}$ for $x \ge 5.5$

$\Phi_{IS}(-5.5) = \frac{1}{n}\sum_{i=1}^{n} I(x_i \ge 5.5) = \frac{1}{n}\sum_{i=1}^{n}\frac{f(x_i)}{g(x_i)} = \frac{1}{n}\sum_{i=1}^{n}\dfrac{e^{-x_i^2/2+x_i-5.5}}{\sqrt{2\pi}}$ where $X \sim \exp(1)+5.5$ and $x \ge 5.5$

algorithm: (also known as sampling importance resampling or SIR)

1. Sample $Y_1, \dots Y_m$ from $g(x)$
2. Calculate the standardized importance weights $w(Y_1), \dots w(Y_m)$
3. Resample $X_1, \dots X_n$ from $Y_1, \dots Y_m$ with replacement with probabilities $w(Y_1), \dots w(Y_m)$

Then the independent samples generated by SIR converges to $f(x)$ as $m \to \infty$ if $w(x_i) = \dfrac{f_0(x_i)/g(x_i)}{\sum_j f_0(x_j)/g(x_j)}$

note: alternatives: $\begin{cases} \text{adaptive importance, bridge, path sampling is for poor importance sampling envelopes.} \\ \text{sequential importance sampling (SIS) is for high-dimensional envelopes} \end{cases}$

(4) Gibbs Sampler

method: a random vector generation method that does not require the complete information of the target multivariate pdf; instead it requires only the information of a set of the associated conditional distribution.

4.1 Gibbs Sampler

algorithm: suppose $U = (U_1, \dots U_k)$ has a joint pdf $f(u)$, and $f(u_k | u_{-k}) = f(u_k | u_1, \dots u_{k-1}, u_{k+1}, \dots u_k)$

1. Arbitrarily generate/assign an initial vector $u^{(0)} = (u_1^{(0)}, \dots u_k^{(0)})$ from the support of $f(u)$.
2. Generate $u_1^{(1)}$ from $f(u_1 | u_2 = u_2^{(0)}, \dots u_k = u_k^{(0)})$

   $u_2^{(1)}$ from $f(u_2 | u_1 = u_1^{(1)}, u_3 = u_3^{(0)}, \dots u_k = u_k^{(0)})$

   $\vdots$

   $u_k^{(1)}$ from $f(u_k | u_1 = u_1^{(1)}, u_2 = u_2^{(1)}, \dots\dots u_{k-1} = u_{k-1}^{(1)})$

   Then $u^{(1)} = (u_1^{(1)}, u_2^{(1)}, \dots u_k^{(1)})$

3. For $i = 1, 2, \dots$

Generate $u_1^{(j)}$ from $f(u_1 | u_2 = u_2^{(j-1)}, \dots u_k = u_k^{(j-1)})$

$u_2^{(j)}$ from $f(u_2 | u_1 = u_1^{(j)}, u_3 = u_3^{(j-1)}, \dots u_k = u_k^{(j-1)})$

$\vdots$

$u_k^{(j)}$ from $f(u_k | u_1 = u_1^{(j)}, u_2 = u_2^{(j)}, \dots\dots u_{k-1} = u_{k-1}^{(j)})$

Then $u^{(j)} = (u_1^{(j)}, u_2^{(j)}, \dots u_k^{(j)})$

(i.e. the conditioning is always based on the latest values)

Then using theory of Markov Chain, $u^{(j)} \to f(u)$ as $j \to \infty$

note: • we can also use burn-in sequence to remove the initial J values, i.e. $u^{(1)}, u^{(2)}, \dots u^{(J)}$ and only use $u^{(J+1)}, u^{(J+2)}, \dots u^{(m+J)}$ because $u^{(J+1)}, u^{(J+2)}, \dots u^{(J+m)}$ are not independent and thus constitute a markov chain.

4.2 Alternatives in Gibbs Sampler family.

(1) Random scan Gibbs sampling: the order of updating the different components of $X^{(t)}$ is random.

(2) Blocking: update joint marginal density: e.g. $X_2^{(t+1)}, X_3^{(t+1)} \sim f(x_2, x_3 | X_1^{(t+1)}, X_4^{(t)})$

(3) Hybrid Gibbs sampling: different Gibbs for each dimension.

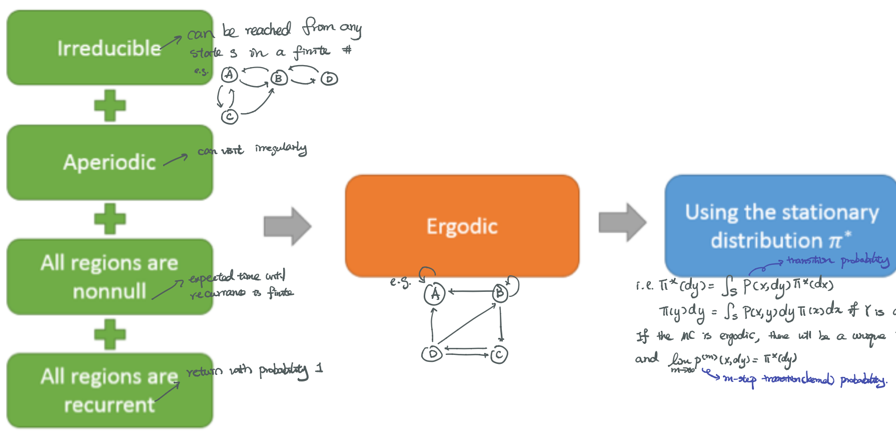
(5) Metropolis-Hastings Monte Carlo

5.1 Ânc based on Markov Chain.

method: According Ergodic Theorem, if $X^{(1)}, \dots X^{(n)}, \dots$ are realisations from an ergodic Markov Chain with stationary distribution π^*, then:

(1) $X^{(n)}$ converges in distribution to π^* as $n \to \infty$;

(2) for any function h, $\frac{1}{n}\sum_{t=1}^{n} h(x^{(t)}) \to E_{\pi^*}[h(X)]$ almost surely as $n \to \infty$



i.e. $\pi^*(dy) = \int_S P(x, dy)\pi^*(dx)$

$\pi(y)dy = \int_S P(x,y)dy\, \pi(x)dx$ if X is continuous

If the MC is ergodic, there will be a unique π^*

and $\lim_{m\to\infty} P^{(m)}(x, dy) = \pi^*(dy)$ m-step transition(kernel) probability.

⇒ but how do we know if P has a stationary distribution. ⇒ detailed balance (sufficient but not necessary condition)

it says $\pi(x)p(x,y) = \pi(y)p(y,x)$ for all $x, y \in S$ ⇒ i.e. the MC is reversible
(transition density)

⇒ but how do we find P that leads to detailed balance ⇒ we can just find a convenient distribution and fix it later.

5.2 Metropolis-Hasting algorithm.

let $q(x,y)$ be a proposal transition pdf. such as $\exp(\cdot)$ (it's called Metropolis algorithm if $q(x,y)$ is symmetric, i.e. $q(x,y) = q(y,x) \Rightarrow \alpha(x,y) = \min\{\frac{f(y)}{f(x)}, 1\}$)

$\alpha(x,y) = \frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}$ be the acceptance probability. (also probability of moving from x to y)
(target pdf. e^x, e^y)

$P_{MH}(x,y) = q(x,y)\alpha(x,y)$

Since we want $\pi(x)P_{MH}(x,y) = \pi(y)P_{MH}(y,x) \Rightarrow \pi(x)q(x,y)\alpha(x,y) = \pi(y)q(y,x)\alpha(y,x) \Rightarrow \frac{\alpha(x,y)}{\alpha(y,x)} = \frac{\pi(y)q(y,x)}{\pi(x)q(x,y)} = \frac{f(y)q(y,x)}{f(x)q(x,y)}$

So the solution for $\alpha(x,y)$ is $\min\{\frac{f(y)q(y,x)}{f(x)q(x,y)}, 1\} = \begin{cases} \text{if } \pi(x)q(x,y) < \pi(y)q(y,x) \Rightarrow \text{tend to move from } y \text{ to } x \Rightarrow \text{need to balance it} \Rightarrow \alpha(y,x) < 1 \text{ and } \alpha(x,y) = 1 \\ \text{if } \pi(x)q(x,y) > \pi(y)q(y,x) \Rightarrow \text{tend to move from } x \text{ to } y \Rightarrow \text{need to balance it} \Rightarrow \alpha(y,x) = 1 \text{ and } \alpha(x,y) < 1 \end{cases}$

algorithm:

1. initial value $x^{(0)}$
2. For $j = 0, 1, 2, \dots N$

   2.1 Generate y from $q(x^{(j)}, y)$ and u from $U(0,1)$

   2.2 If $u \le \alpha(x^{(j)}, y)$, set $x^{(j+1)} = y$; else set $x^{(j+1)} = x^{(j)}$
3. Return $\{x^{(0)}, x^{(1)}, \dots x^{(N)}\}$

Then the pdf of $x^{(j)}$ converges to $\pi(x)$ as $j \to \infty$

(6) Don't know where these belong to

6.1 Rao-Blackellised $\hat{\mu}_{MC}$

Rao-Blackwell Theorem: $\hat{\mu}_{MC} = \frac{1}{n}\sum_{i=1}^{n} h(X_i)$ can be improved by conditioning on sufficient statistic

$\Rightarrow \hat{\mu}_{RB} = \frac{1}{n}\sum_{i=1}^{n} h(X_i|S) = \underline{\frac{1}{n}\sum_{i=1}^{n} h^*(S_i)}$ where $S_i = S(X_i)$ e.g. $E[X_1] = E[E(X_1|x_2, x_3)] \approx \frac{1}{n}\sum_i \int_x x f(x|x_{2i}, x_{3i})\,dx$

only depends on S_i

$\Rightarrow V(\hat{\mu}_{RB}) \leq V(\hat{\mu}_{MC})$

6.2 Monitoring MCMC convergence

Burn-in period: required for Markov Chain to let samples become stationary. i.e. remove $u^{(1)}, u^{(2)}, \dots u^{(J)}$ and only use $u^{(J+1)}, u^{(J+2)}, \dots u^{(m+J)}$

Sub-sampling: reduce the dependence among the samples to be used.

M: no final conclusion but some popular methods (R package coda), start with k MCs with different initial values

Two types of convergence: stationary distribution and ergodic average.

(7) Case study

1. (X, P, N) where $N \sim Po(\lambda)$, $P \sim Beta(a, b)$ and $(X|P, N) \sim bi(N, p)$

- Gibbs

Suppose $N \sim Poi(16)$, $P \sim Beta(2,4)$, $X|P, N \sim bi(N, p)$.

first find $f(x, P, N) = f(x|P,N) f(N) f(p)$

$$= \binom{N}{x} p^x (1-p)^{N-x} \cdot \frac{\Gamma(2+4)}{\Gamma(2)\Gamma(4)} p^{2-1} (1-p)^{4-1} \cdot \frac{16^N}{N!} e^{-16}$$

$$= 20\binom{N}{x} \frac{16^N}{N!} e^{-16} p^{x+1} (1-p)^{N+3-x}$$

$$\Rightarrow f(p|x, N) = \frac{f(x,P,N)}{f(x,N)} = \frac{20e^{-16}\binom{N}{x}\frac{16^N}{N!}p^{x+1}(1-p)^{N+3-x}}{\int_0^1 20e^{-16}\binom{N}{x}\frac{16^N}{N!}p^{x+1}(1-p)^{N+3-x}dp} = \frac{p^{x+1}(1-p)^{N+3-x}}{\int_0^1 p^{x+1}(1-p)^{N+3-x}dp} = \frac{\frac{\Gamma(N+6)}{\Gamma(x+2)\Gamma(N+4-x)}p^{x+1}(1-p)^{N+3-x}}{\int_0^1 \frac{\Gamma(x+2+N+4-x)}{\Gamma(x+2)\Gamma(N+4-x)}p^{x+1}(1-p)^{N+3-x}dp} \Rightarrow P|x,N \sim Beta(x+2, N+4-x)$$

$$\Rightarrow f(N|p, x) = \frac{f(x,P,N)}{f(P,x)} = \frac{20e^{-16}\binom{N}{x}\frac{16^N}{N!}p^{x+1}(1-p)^{N+3-x}}{\sum_{n=x}^{\infty} 20e^{-16}\binom{n}{x}\frac{16^n}{n!}p^{x+1}(1-p)^{n+3-x}} = \frac{\frac{16^N}{N!}(1-p)^N}{\sum_{n=x}^{\infty}\binom{n}{x}\frac{16^n}{n!}(1-p)^n} = \frac{[16(1-p)]^{N-x}}{(N-x)!}e^{-16(1-p)} \Rightarrow N-x|p,x \sim Po(16(1-p))$$

$\Rightarrow$ 1. Initialise $(x^{(0)}, p^{(0)}, N^{(0)}) = (8, 0.5, 16)$ (the mean)

2. For $j = 1, 2, \dots,$

   generate $x^{(j)}$ from $Bi(N^{(j-1)}, p^{(j-1)})$

   generate $p^{(j)}$ from $Beta(x^{(j)}+2, N^{(j-1)}+4-x^{(j)})$

   generate $N^{(j)}$ from $Poi(16(1-p^{(j)})) + x^{(j)}$

- Rao-Blackwell

$\hat{\mu}_{RB} = \frac{1}{n}\sum_{j=1}^{n} E(X|P_j, N_j) = \frac{1}{n}\sum_{j=1}^{n} N_j P_j$ $\quad\quad$ $\hat{\mu}_{MC} = \frac{1}{n}\sum_{j=1}^{n} X_j$

$\hat{f}_{RB}(x) = \frac{1}{n}\sum_{j=1}^{n} f(x|P_j, N_j) = \frac{1}{n}\sum_{j=1}^{n}\binom{N_j}{x}P_j^x(1-P_j)^{N_j-x}$ $\quad\quad$ $\hat{f}_{MC}(x) = \frac{1}{n}\sum_{j=1}^{n} I(X_j = x)$

2. Beta(2, 8).

- AR.

1. Generate $X = x$ from $U(0,1)$
2. Generate $Y = y$ from $U(0,1)$
3. If $y \leq x(1-x)^7$ then deliver $z = x$; otherwise reject

$P(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} I_{(0<x<1)}$

$M(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} I_{(0<x<1)}$

pdf induced by $M(x)$ in Unif(0,1)

- MH using Unif(0,1) as proposal pdf.

1. initial value $x^{(0)}$
2. For $j = 0, 1, 2, \dots N$

   2.1 Generate y from $\overset{\text{proposal pdf.}}{U(0,1)}$ and u from $U(0,1)$

   2.2 If $u \leq \alpha(x^{(j)}, y)$, set $x^{(j+1)} = y$; else set $x^{(j+1)} = x^{(j)}$ where $\alpha = \dfrac{\overset{\pi(y)}{Beta(y; 2,8)} \cdot \overset{q(y,x)}{1}}{\underset{\pi(x)}{Beta(x^{(j)}; 2,8)} \cdot \underset{q(x,y)}{1}}$
3. Return $\{x^{(0)}, x^{(1)}, \dots x^{(N)}\}$