MCMC 讲义

@NWU

强喆

zhe.qng@nwu.edu.cn

2019 年 4 月 25 日

目录

1 马尔可夫链

- 马尔科夫链蒙特卡洛模拟 (MCMC) 成功的关键, 并不是马尔科夫性，而是近似分布在模拟的每一步都会有所改进，并在某个意义下收敛到目标分布。
- 而马尔科夫性对于收敛到目标分布是有帮助的。
- 马尔科夫链的模拟一般用于以下两种情况：
  1. 不可能
  2. 计算上没有有效的方法

  直接从参数的后验分布 $p(\theta|y)$ 直接采样 θ;
- 相反，用迭代的方式采样，在每一步迭代中的分布都离 $p(\theta|y)$ 更加接近。
- 在得到全部的模拟后，我们也应该
  1. 检查模拟序列的收敛性
  2. 构造模拟序列关于相关样本额有效数目的表达式。

1.1 马尔可夫链的基本概念

定义 1.1.1 (马尔可夫链). 称随机变量序列是马尔可夫链,如果这个序列满足:给定 $X_1, \cdots, X_n$ 时 X_{n+1} 的条件分布只依赖于 X_n，也就是：

$$P(X_{n+1}|X_1, X_2, \cdots, X_n) = P(X_{n+1}|X_n)$$

由定义可以看出，马尔可夫链是由以下两个条件完全确定的：

- 初始分布；
- 转移概率分布：$P(X_{n+1}|X_n)$, 这里记作 $K(X_n, X_{n+1})$

定义 1.1.2 (平稳马尔可夫链). 进一步,如果这个马尔可夫链对于任意的 i, k,有 $(X_{n+1}, \cdots, X_{n+k})$ 的分布不依赖于 n，则称这个链是平稳的马尔可夫链。

定义 1.1.3 (平稳性). 对于 $\sigma-$ 有限的测度 π，称它关于转移核 $K(\cdot,\cdot)$ 是不变的，如果:

$$\pi(B)=\int_{\mathcal{X}}K(x,B)\pi(dx),\quad \forall B\in\mathcal{B}(X).$$

这样，如果 $X_0\sim\pi$，那么 $X_n\sim\pi,\ \forall n$，因而把不变分布也称为平衡分布或者平稳分布。

1.2 马尔可夫链的可逆性和细致平衡条件

定义 1.2.1 (可逆性). 称马尔可夫链是可逆的，如果向前和向后的转移律相同，也就是

$$P(X_{i+1},X_{i+2},\cdots,X_{i+k})=P(X_{i+k},\cdots,X_{i+2},X_{i+1}),\quad \forall i,k$$

当可逆性只考虑在前后两步转移时，称为细致平衡。

定义 1.2.2 (细致平衡条件). 称马尔可夫链的转移核 K 满足细致平衡条件 (detailed balance)，如果存在函数 π 满足:

$$K(y,x)\pi(y)=K(x,y)\pi(x),\forall x,y\in\mathcal{X}$$

下面的定理给出细致平衡条件、可逆性和平稳分布之间的关系。

定理 1.2.3. 如果马尔可夫链的转移核 K 满足细致平衡条件，则

(i) 密度 π 就是平稳分布的密度;

(ii) 链是可逆的。

证明. (i) 对样本空间 $\mathcal{X}$ 中的可测集 B，由细致平衡条件可知:

$$\begin{aligned}\int_{\mathcal{X}}K(y,B)\pi(y)dy&=\int_{\mathcal{X}}\int_{B}K(y,x)\pi(y)dxdy\\&=\int_{\mathcal{X}}\int_{B}K(x,y)\pi(x)dxdy\\&=\int_{B}\pi(x)dx\end{aligned}$$

因而，不变分布存在，其密度就是 π。

(ii) 如果链满足细节平衡条件，那么对任意的 i,k, 有

$$\begin{aligned}&\pi(X_{i+1})K(X_{i+1},X_{i+2})=\pi(X_{i+2})K(X_{i+2},X_{i+1})\\&\Longrightarrow\pi(X_{i+1})K(X_{i+1},X_{i+2})K(X_{i+2},X_{i+3})\\&=\pi(X_{i+2})K(X_{i+2},X_{i+3})K(X_{i+2},X_{i+1})\\&=\pi(X_{i+3})K(X_{i+3},X_{i+2})K(X_{i+2},X_{i+1})\\&\vdots\\&\Longrightarrow\pi(X_{i+1})K(X_{i+1},X_{i+2})\cdots K(X_{i+k-1},X_{i+k})\\&=\pi(X_{i+k})K(X_{i+k},X_{i+k-1})\cdots K(X_{i+2},X_{i+1})\end{aligned}$$

因而链是可逆的。

取上式中 $k=1$，即可得出链满足细致平衡条件。

通常在设计马尔可夫链的模拟算法时，用细致平衡条件约束关系设计算法非常简单，并且能够保证马尔可夫链的平稳分布存在，而平稳分布存在又是算法收敛的必要条件。这是因为，平稳马尔可夫链的平稳分布（不变分布）存在的充分不必要条件是，马尔可夫链的可逆性或者要求细致平衡条件成立，可以证明后二者是等价的。

1.3 马尔可夫链蒙特卡罗模拟的原理

MCMC 算法原理是用蒙特卡罗 (Monte Carlo,MC) 来模拟马尔可夫链。样本生成过程是：从 $X^{(t)}$ 根据马尔可夫转移核生成 $X^{(t+1)}$,算法结束就得到一系列的样本 $(X^{(0)}, X^{(1)}, \cdots, X^{(t)}, \cdots, X^{(T)})$。因而这些样本实际上形成了一条马尔可夫链。由马尔可夫链的遍历性可以得到, 对任何特定的函数 $h(x)$，有如下结论：

$$\frac{1}{T}\sum_{t=1}^{T} h(X^{(t)}) \to \int_{\mathcal{X}} h(x)\pi(x)dx$$

这意味这马尔可夫链的状态概率分布可以收敛到平稳分布。这一结果和经典的蒙特卡罗 (Monte Carlo, MC) 模拟是一致的：经典的蒙特卡罗模拟是利用独立同分布的随机变量 $X_1, X_2, \cdots$，服从相同的分布 $\pi(x)$，由强大数定律 (the strong law of large numbers, SLLN)，以及满足控制收敛定理的某个连续函数 $h(x)$:

$$\frac{1}{n}\sum_{i=1}^{n} h(X_i) \to_{a.s.} \int_{\mathcal{X}} h(x)\pi(x)dx$$

这样通过蒙特卡罗模拟马尔可夫链产生的 $(X^{(0)}, \cdots, X^{(t)}, \cdots)$ 理论上等同于用不变分布 π 产生的独立同分布 (independent identically distribution,iid) 的样本，这样马尔可夫链门特卡罗模拟产生的序列 $(X^{(t)})$ 就可以作为 i.i.d 的样本使用。
实际上，MCMC 产生的样本是非独立的，但只要 $cov(X^{(t)}, X^{(t+k)})$ 随着 k 的增大而减小，$(X^{(k)}, X^{(2*k)}, \cdots, X^{(t*k)}, \cdots)$ 就是拟独立的样本，因而可以近似地替代 i.i.d 的样本来使用。

1.4 作业

- 强大数定律、弱大数定律是什么
- 安装 Rstudio www.rstudio.com, 简单学习 r 语言
- 产生随机数的算法

2 接受拒绝算法

2.1 模拟的基本定理

定理 2.1.1. 模拟

$$X \sim f(x)$$

等价于模拟

$$(X,U) \sim \mathcal{U}\{(x,u) : 0 < u < f(x)\}.$$

解决办法是在一个更大的集合模拟 (X,U)，这样模拟更简单，一旦约束满足就选择这对样本。

- 假设

  $$\int_a^b f(x)dx = 1$$

  并且 f 的上界是 m。

- 按照如下的方式采样随机变量对 $(X,U) \sim \mathcal{U}(0 < u < m)$:

  –

  $$Y \sim \mathcal{U}(a,b)$$

  –

  $$U|Y = y \sim \mathcal{U}(0,m)$$

- 如果约束 $0 < u < f(y)$ 满足，则保留这对样本.

$$\begin{aligned} P(X \le x) &= P(Y \le x | U < f(Y)) \\ &= \frac{\int_a^x \int_0^{f(y)} dudy}{\int_a^b \int_0^{f(y)} dudy} = \int_a^x f(y)dy \end{aligned}$$

例子 2.1.2 (Beta 分布随机数的模拟)**.** 目标是生成 $X \sim \mathcal{B}](\alpha,\beta)$, 取 $Y \sim \mathcal{U}(0,1)$ and $U \sim \mathcal{U}(0,m)$. 这里 m 是 Beta density 的最大值. Beta 分布的参数是 $\alpha = 2.7$, $\beta = 6.3$, 画

出产生的 1000 个样本对 (Y,U). 对于在盒子 $[a,b]\times[0,m]$ 中一个给定的采样结果，它的接受概率是

$$P(Accept)=P(U<f(Y))=\frac{1}{m}\int_0^1\int_0^{f(y)}dudy=\frac{1}{m}$$

, 生成的样本结果如图1

R 代码

```
optimize(f=function(x){dbeta(x,2.7,6.3)},
         interval = c(0,1), maximum=TRUE)$objective
Nsim=2500
a = 2.7;b=6.3
M=2.67
u=runif(Nsim,max=M)
y=runif(Nsim)
x=y[u<dbeta(y,a,b)]
xu=u[u<dbeta(y,a,b)]
```

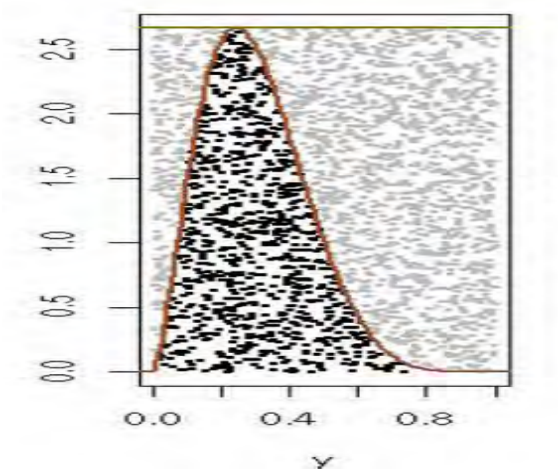

图 1: AR 算法生成 Beta 分布随机数

- 可以从另一个非盒子形状的更大的集合对上述公式进行一般化。这个更大的集合是

$$\mathcal{L}=\{(y,u):0<u<m(y)\}$$

实际上，不再取 $f(x)$ 的最大值最为上界，而是采用 $f(x)$ 的上界函数 $m(x)$，因而约束变为 $m(x)\geq f(x)$.

- 有效性意味着, 为了避免浪费 simulation,m 必须尽可能的靠近 f.
- $m(x)$ 有可能不是概率测度，因为 $m(x) \geq f(x)$.
  因而把 $m(x)$ 记为

$$m(x) = Mg(x) \quad where \quad \int_X m(x)dx = \int_X Mg(x)dx = M$$

只需要先采样 $Y \sim g$ and $U|Y = y \sim \mathcal{U}(0, Mg(y))$，然后只接受满足 $u < f(y)$ 的 y。
下面计算由这种算法生成的随机变量 X 的分布：

$$\begin{aligned} P(X \in \mathcal{A}) &= P(Y \in \mathcal{A}|U < f(Y)) \\ &= \frac{\int_{\mathcal{A}} \int_0^{f(y)} \frac{du}{Mg(y)} g(y)dy}{\int_{\mathcal{X}} \int_0^{f(y)} \frac{du}{Mg(y)} g(y)dy} = \int_{\mathcal{A}} f(y)dy \end{aligned}$$

这样对于任意的可测集 $\mathcal{A}$, 被接受的随机变量 X 确实是服从 f 的分布.

例子 2.1.3. 用 AR 算法采样 $X \sim f(x)$, 这里

$$f(x) \propto \exp(-x^2/2)(\sin(6x)^2 + 3\cos(x)^2 \sin(4x)^2 + 1)$$

上界函数使用正态密度函数

$$g(x) = \exp(-x^2/2)/\sqrt{2\pi}$$

2.2 拒绝—接受算法

接受-拒绝算法步骤

step 1. 产生 $X \sim g, U \sim \mathcal{U}(0, 1)$

step 2. 接受 $Y = X$，如果 $U \leq f(X)/Mg(X)$;

step 3. 否则返回 1.

注意为了让 f/g 保持有界, g 的尾部要比 f 细. 因而用 AR 算法，用正态分布 g 模拟 Cauchy 分布 f 是不可能的，但反过来是可以的。

Remark:

- 一个好的建议分布 $g(x)$ 应该大致上正比于 $f(x)$。在这种情况下，以一个合适的 M，每一次采样的接受概率都是 1.

- 当 g 不是正比于 p 时, 上界 M 一定要设置得很大，这样在 step1 中几乎所有的 draw 都会在 step2 中被拒绝。
- 拒绝采样的一个优点是 self-monitoring—如果这个方法不是很有效，很少的模拟结果会被接受.
- 拒绝采样用于从标准的一维分布进行某种快速采样;
- 拒绝采样也可以用于一般的截断高维分布。

例子 2.2.1 (从双指数分布生成正态分布). 考虑用双指数分布 $\mathcal{L}(\alpha)$ 生成正态分布 $\mathcal{N}(0,1)$, 密度是

$$g(x|\alpha) = (\alpha/2)\exp(-\alpha|x|).$$

由于

$$\frac{f(x)}{g(x|\alpha)} \leq \sqrt{2/\pi}\alpha^{-1}e^{\alpha^2/2}$$

而这个上界当 $\alpha = 1$ 取得最小值. 算法的接受概率是 $\sqrt{\pi/2e} = 0.76$, 这说明为了产生正态分布, 这个 AR 算法需要平均 $1/0.76 \approx 1.3$ 均匀分布随机数。

例子 2.2.2 (Gamma A-R). 用 $\mathcal{G}a(a,b)$ 产生 $\mathcal{G}a(\alpha,\beta)$, 这里 a,b 都是整数. $a = [\alpha](\alpha \geq 1)$, 设 $\beta = 1$. 比值 f/g 是 $b^{-a}x^{\alpha-a}\exp\{-(1-b)x\}$, 差一个归一化的常数, 界是:

$$M = b^{-a}\left(\frac{\alpha-a}{(1-b)e}\right)^{\alpha-a}, \quad b < 1$$

由于 $b^{-a}(1-b)^{\alpha-a}$ 的最大值是在 $b = a/\alpha$ 取得的, 因而生成 $\mathcal{G}a(\alpha,1)$ 的最优 b 是 $\mathcal{G}a(a,a/\alpha)$.

例子 2.2.3 (截断正态分布). 截断正态分布，约束是 $x \geq \underline{\mu}$，它的密度正比于

$$e^{-(x-\mu)^2/2\sigma^2}I_{x\geq\underline{\mu}}$$

$\underline{\mu}$ 是比 μ 大的界.

1. 一般的方法: 模拟 $X \sim \mathcal{N}(\mu,\sigma^2)$, 只接受比 $\underline{\mu}$ 大的 X. 这个算法需要平均 $\frac{1}{\Phi(\frac{\mu-\underline{\mu}}{\sigma})}$ 次模拟.
2. 考虑 $\mu = 0, \sigma = 1$. 平移的指数分布 $\mathcal{E}xp(\alpha,\underline{\mu})$, 密度是

   $$g_\alpha(z) = \alpha e^{-\alpha(z-\underline{\mu})}I_{z\geq\underline{\mu}}$$

   比值 $f/g_\alpha(z) = e^{-\alpha(z-\underline{\mu})}e^{-z^2/2}$, 它的界是:

   $$\begin{cases} \frac{1}{\alpha}\exp(\alpha^2/2 - \alpha\underline{\mu}) & if\ \alpha > \underline{\mu} \\ \frac{1}{\alpha}\exp(-\underline{\mu}^2/2) & otherwise \end{cases}$$

第一个表达式被下式最小化：

$$\alpha^\star = \underline{\mu} + \frac{1}{2}\sqrt{\underline{\mu}^2 + 4}$$

第二个表达式被 $\alpha' = \underline{\mu}$ 最小化. 最优的 α 是 $\alpha^\star$.

2.3 从截断正态分布生成随机数

2.3.1 逆变换法

定义 2.3.1. 对于 $\mathrm{F}\mathbb{R}$ 上的某个非降函数, F 的广义逆, F^- 定义为：

$$F^-(u) = \inf\{x : f(x) \geq u\}.$$

引理 2.3.2. 如果随机变量 $U \sim \mathcal{U}(0,1)$, 则随机变量 $F^-(U)$ 的分布是 F.

Proof. 对于所有的 $u \in [0,1]$ 和所有的 $x \in F^-([0,1])$, 广义逆满足:

$$\{u : F^- \leq x\} = \{u : F(x) \geq u\}$$

因此,

$$P(F^-(U) \leq x) = P(F(x) \geq U) = F(x)$$

□

例子 2.3.3 (指数族随机数的生成). 如果随机变量 $X \sim \mathcal{E}xp(1)$, 则 $F(x) = 1 - e^{-x}$, 在等式 $u = 1 - e^{-x}$ 中反解 x，得到 $x = -\log(1-u)$. 因此，如果 $U \sim \mathcal{U}(0,1)$, 则随机变量 $X = -\log U$ 服从指数分布.

2.3.2 用逆变换法从截断正态分布生成随机数

R 语言程序：

```
for(i in 2:nsim){
            temp=runif(n-m,min=pnorm(a,mean=that[i-1],sd=1),
          max=1)
          zbar[i]=mean(qnorm(temp,mean=that[i-1],sd=1))
          that[i]=rnorm(1,mean=(m/n)*xbar+(1-m/n)*zbar,
          sd=sqrt(1/n))}
```

因为截断正态分布是：

$$F(x) = \int_a^x \frac{\varphi(x)}{1-\Phi(a)}dx = \frac{\Phi(x) - \Phi(a)}{1-\Phi(a)}$$

则

$$\begin{aligned} & F(x) = u \\ \Leftrightarrow & \Phi(x) = u(1-\Phi(a)) + \Phi(a) \\ \Leftrightarrow & x = \Phi^{-1}(\underbrace{u(1-\Phi(a)) + \Phi(a)}_{runif(n-m, min=pnorm(a, mean=that[i-1], sd=1), max=1)}) = F^{-1}(u) \end{aligned}$$

2.3.3 用切片采样的方法生成截断正态分布随机数

见练习5.3.5

2.4 Kiss 生成器

定义 2.4.1 (周期). 生成器的周期 T_0, 是使得 $u_{i+T} = u_i$, $\forall i$ 的最小的整数 T，也就是 s.t. D^T 等于恒同函数.

普通的随机数生成器的局限：

- 具有形式 $X_{n+1} = f(X_n)$ 的生成器的周期不超过 M+1(对于 C 语言 float 类型, 2^{32})
- 为了克服这个界，生成器必须同时利用几个 X_n^i 的序列；
- 或者除了 X_n 以外还需要包括进来 $X_{n-1}, X_{n-2}, \cdots$
- 或者必须要利用其他的查找表的方式。

Kiss 生成器能够克服以上缺点，它同时利用两种生成技术：共轭生成器，*congruential generation* + 平移生成器，*shift register generation*

定义 2.4.2 (共轭生成器，congruential generator). 在 $\{0, 1, \cdots, M\}$ 区间的 共轭生成器 *congruential generator* 定义为如下的函数:

$$D(x) = (ax + b) mod(M + 1)$$

定义 2.4.3 (平移生成器，shift register generator). 对于给定的 $k \times k$ 的矩阵 T, 它的元素全部都是 0 或者 1, 对应的 平移生成器 *shift register generator*由以下这个变换给出：

$$x_{n+1} = Tx_n$$

这里 x_n 被表示称一个二元的坐标向量 e_{ni}，也就是说

$$x_n = \sum_{i=0}^{k-1} e_{ni} 2^i$$

这里 e_{ni} 是 0 或者 1.

Kiss 生成器是由以下矩阵构成：左移矩阵和右移矩阵

$$T_L = \begin{pmatrix} 1 & 1 & & \\ & \ddots & & 0 \\ & & \ddots & 1 \\ & 0 & & 1 \end{pmatrix} T_R = \begin{pmatrix} 1 & & & \\ 1 & \ddots & 0 & \\ & & \ddots & \\ & 0 & 1 & 1 \end{pmatrix} \tag{2.1}$$

例如

$$R(e_1, \cdots, e_k)^T = (0, e_1, \cdots, e_{k-1})^T$$
$$L(e_1, \cdots, e_k)^T = (e_2, e_3, \cdots, e_k, 0)^T$$

$T_R = (I + R),\ \ T_L = (I + L).$

Kiss 生成器包括一个共轭生成器：

$$I_{n+1} = (69069 \times I_n + 23606797)(mod\ 2^{32})$$

和两个平移生成器：

$$J_{n+1} = (I + L^{15})(I + R^{17})J_n(mod\ 2^{32})$$
$$K_{n+1} = (I + R^{13})(I + L^{18})K_n(mod\ 2^{31})$$

最后把这两个生成器合起来生成：

$$X_{n+1} = (I_{n+1} + J_{n+1} + K_{n+1})(mod\ 2^{32})$$

Kiss 的 C 语言代码

```
long int kiss (i,j,k)
unsigned long *i,*j,*k
{
*j = *j^ (*j<<17);
*k = (*k ^ (*k<<18))&0X7FFFFFFF;
return ((*i = 69069 * (*i) + 23606797) +
(*j ^= (*j>> 15)) + (*k ^= (*k >>13)) );
 }
```

Kiss 的周期是 2^{95}! 这几乎是 $(2^{32})^3$。

3 重要性采样

- 背景：假设我们对 $E(h(\theta)|y)$ 感兴趣, 但又不能直接从 θ from $p(\theta|y)$ 生成随机样本.
- 如果 $g(\theta)$ 是一个我们可以从中生成随机样本的概率密度.
- 则可以把这个积分写成

$$E(h(\theta)|y) = \int h(\theta)p(\theta|y)d\theta = \int [h(\theta)\frac{p(\theta|y)}{g(\theta)}]g(\theta)d\theta \tag{3.1}$$

  这里 $w(\theta) \triangleq \frac{p(\theta|y)}{g(\theta)}$ 被称为 importance ratios.

重要性采样算法如下:

- 1. sample $\theta \sim g(\theta)$
  2. calculate

$$\frac{1}{S}\sum_{s=1}^{S} h(\theta^s)w(\theta^s)$$

精确而有效的重要性采样估计

- 目标：希望避免某些非常大但又极少出现的重要性权重.
- 方法：检查所采样分布的重要性权重 (最大的 importance ratio 的对数直方图) 来发现可能存在的问题：
  - 如果最大的 importance ratio 相比与平均值非常大，则估计会很差。
  - 作为对比，不需要担心小的 importance ratio 的行为，因为他们对等式3.1的影响非常小。
- 如果权重的方差有限，则有效样本可以被如下的估计逼近:

$$S_{eff} = \frac{1}{\sum_{s=1}^{S}(w(\theta^s))^2}$$

重要性重采样，Importance resampling/sampling-importance resampling, SIR
一旦有了 S 个采样结果 $\{\theta^1, \cdots, \theta^S\} \sim g(\theta)$, 则 $k < S$ 个 draws 可以如下方式得到：

- 从集合 $\{\theta^1, \cdots, \theta^S\}$ 以 $w(\theta^s)$ 的概率进行采样;
- 用同样的程序采样第二个值，但不要包含集合中已经存在的样本;
- 进行 k-2 次不放回重复采样.

为什么进行不放回采样?

- 如果重要性权重 importance weights 较小, 放回和不放回采样给出类似的结果.
- 考虑一个差的情况，大的权重非常小，小的权重很多。放回采样将会重复地去除很多相同的 θ; 作为对比，不放回采样将会产生介于初始密度和目标密度中间某个希望出现的情况。

4 Metropolis Hastings 算法

4.1 一般的 Metropolis Hastings 算法

一般的 Metropolis Hastings 算法步骤

step 1. 选一个初始的 $\theta^0 \sim p(\theta^0)$, 使得 $p(\theta^0|y) > 0$;

step 2. 对于 $t = 1, 2, \cdots$

(a) 提出建议 $\theta^\star \sim q(\theta^\star|\theta^{t-1})$,$q(\theta_a|\theta_b)$ 不对称;

(b) 计算 $\rho = \frac{p(\theta^\star|y)/q(\theta^\star|\theta^{t-1})}{p(\theta^{t-1})/q(\theta^{t-1}|\theta^\star)}$;

(c) 设

$$\theta^t = \begin{cases} \theta^\star & \text{以概率 } \min(\rho, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

收敛性

定理 4.1.1. 记 $X^{(t)}$ 是由以上算法生成的马尔科夫链. 对于每一个条件分布 q，它的支撑集必须包含 supp(f)

(a) 这个马尔科夫链满足细节平衡，且其中细节平衡中的条件分布就是 f;

(b) f 是这个马尔科夫链的平稳分布.

Proof. (a) 转移核是:

$$K(x,y) = \underbrace{\rho(x,y)q(y|x)}_{\text{accept proposal}} + (\underbrace{1 - \int \rho(x,y)q(y|x)dy}_{\text{reject proposal and didn't provide proposal}})\delta_x(y)$$

现在验证细节平衡公式:

$$f(x)K(x,y) = f(y)K(y,x)$$

这个公式可以被划分为两部分：

$$f(x)\rho(x,y)q(y|x) = f(y)\rho(y,x)q(x|y)$$

和

$$f(x)(1-\int \rho(x,y)q(y|x)dy)\delta_x(y) = f(y)(1-\int \rho(y,x)q(x|y)dx)\delta_y(x)$$

由于

$$\rho(x,y) = \min\left\{1, \frac{f(y)q(x|y)}{f(x)q(y|x)}\right\},$$

证毕。

(b) 由于满足细节平衡，所以，对任意的 B,

$$\begin{aligned}\int_X K(y,B)f(y)dy &= \int_X\int_B K(y,x)f(y)dxdy \\ &= \int_X\int_B K(x,y)f(x)dxdy \\ &= \int_B f(x)dx = f(B)\end{aligned}$$

□

定理 4.1.2. 假设 Metrololis-Hastings 算法生成的马尔科夫链是 f-irreducible. 那么

(i) 如果 $h \in L^1(f)$, 则

$$lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} h(X^{(t)}) = \int h(x)f(x)dx \quad a.e.f.$$

(ii) 另外，如果 $X^{(t)}$ 是非周期的，则

$$\lim_{n\to\infty} \| \int K^n(x,\cdot)\mu(dx) - f\|_{TV} = 0$$

对于每一个初始分布 μ 均成立。这里 $K^n(x,\cdot)$ 表示 n 步转移核.

例子 4.1.3. 用 MH 算法产生服从 $f = Be(2.7, 6.3)$ 的随机数, 建议分布 $q(y|x)$ 是 $U[0,1]$, 也就是说不依赖于前一时刻的值. 对应的 R 代码如下:

```
a = 2.7;b=6.3;c=2.669    #initial values
Nsim=5000
X =rep(runif(1),Nsim) #initialize the chains
for (i in 2:Nsim){
  Y=runif(1)
```

```
  rho=dbeta(Y,a,b)/dbeta(X[i-1],a,b)
  X[i]=X[i-1] + (Y - X[i-1])*(runif(1)<rho)
}
```

注释

- 理想的 M-H 跳是 $q(\theta^\star|\theta) = p(\theta^\star|y)$，就是目标分布,$\forall\theta^\star$, 这时 $\rho \equiv 1$，但这是没有意义的;
- 好的 M-H 的性质
  1. $\forall\theta$, $q(\theta^\star|\theta)$ 易于采样
  2. ρ 容易计算
  3. 每次跳在参数空间距离合适
  4. 不要拒绝太频繁

4.2 Metropolis 算法

Metropolis 算法是一种带有 A/C 规则的随机游走算法。

Metropolis 算法

step 1. 选一个初始的 $\theta^0 \sim p(\theta^0)$, 使得 $p(\theta^0|y) > 0$;

step 2. 对于 $t = 1, 2, \cdots$

(a) 提出建议 $\theta^\star \sim q(\theta^\star|\theta^{t-1})$, 对称性要求 $q(\theta_a|\theta_b) = q(\theta_b|\theta_a)$, $\forall\theta_a, \theta_b, t$;

(b) 计算 $r = \frac{p(\theta^\star|y)}{p(\theta^{t-1}|y)}$;

(c) 设

$$\theta^t = \begin{cases} \theta^\star & \text{以概率 } \min(r,1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

转移分布

$$T(\theta^t|\theta^{t-1}) = \delta_{\theta^{t-1}}(\theta^t) + q(\theta^t|\theta^{t-1})\min(r,1)$$

是一个 $\theta^t = \theta^{t-1}$ 的单点估计量和眺分布 $q(\theta^t|\theta^{t-1})$ 的加权，权重为接受概率。

算法要求具有一下能力

(a) r 可以计算

(b) 可以从 $q(\theta^\star|\theta)$ 采样 θ

(c) step c 需要产生一个均匀分布随机数。

例子 4.2.1 (二元正态分布).

- $p(\theta|y) = \mathcal{N}(\theta|0, 0.8 * I)$
- $q(\theta^\star|\theta^{t-1}) = \mathcal{N}(\theta^\star|\theta^{t-1}, I)$
- $r = \frac{\mathcal{N}(\theta^\star|0,0.8I)}{\mathcal{N}(\theta^{t-1}|0,0.8I)}$

用 R 来实现:

```
  library('mvtnorm')
  #metropolis algorithm
  t1 <- -2.5
  t2 <- 2.5
  #' Number of iterations.
  M <- 5000

  #' Metropolis sampling
  tt <- matrix(rep(0, 2*M), ncol = 2)
  tt[1,] <- c(t1, t2)
  for(i in 2:M){
    Y = mvrnorm(1,tt[i-1,],diag(2))
    rho = dmvnorm(Y,c(0,0),diag(2)*0.8)
    /dmvnorm(tt[i-1,],c(0,0),diag(2)*0.8)
    tt[i,] = tt[i-1,] + (Y - tt[i-1,])*(runif(1)<rho)
  }
```

下面画出迭代的过程。

先载入库，并设置路径:

```
library(ggplot2)
theme_set(theme_minimal())
library(tidyr)
library(gganimate)
```

```
library(ggforce)
library(MASS)
library(rprojroot)
library(rstan)
setwd('/media/zheqng/Seagate␣Backup␣Plus␣Drive/zheqng@nwu/文档/teaching␣lectures/2019

root<-has_dirname("BDA_R_demos-master")$make_fix_file()
```

输入目标分布的参数，并做出 100000 个服从目标分布的样本点，便于在后面画出它的椭圆高概率区域：

```
#'␣Parameters␣of␣a␣normal␣distribution␣used␣as␣a␣toy␣target␣distribution
y1 <- 0
y2 <- 0
r <- 0.8
S <- diag(2)
S[1, 2] <- r
S[2, 1] <- r
#'␣Metropolis␣proposal␣distribution␣scale
sp <- 0.3
```

画出样本的前 100 次迭代的收敛过程：

```
df100 <- data.frame(id=rep(1,100),
                    iter=1:100,
                    th1 = tt[1:100, 1],
                    th2 = tt[1:100, 2],
                    th1l = c(tt[1, 1], tt[1:(100-1), 1]),
                    th2l = c(tt[1, 2], tt[1:(100-1), 2]))
#'␣Sample␣from␣the␣toy␣distribution␣to␣visualize␣90%␣HPD
#'␣interval␣with␣ggplot's stat_ellipse()
dft <- data.frame(mvrnorm(100000, c(0, 0), S))
# labels and frame indices for the plot
labs1 <- c('Draws', 'Steps␣of␣the␣sampler', '90%␣HPD')
p1 <- ggplot() +
geom_jitter(data = df100, width=0.05, height=0.05,
            aes(th1, th2, color ='1'), alpha=0.3) +
geom_segment(data = df100, aes(x = th1, xend = th1l, color = '2',
```

```
                                    y = th2, yend = th2l)) +
stat_ellipse(data = dft, aes(x = X1, y = X2, color = '3'), level = 0.9) +
coord_cartesian(xlim = c(-4, 4), ylim = c(-4, 4)) +
labs(x = 'theta1', y = 'theta2') +
scale_color_manual(values = c('red', 'forestgreen','blue'), labels = labs1)
guides(color = guide_legend(override.aes = list(
  shape = c(16, NA, NA), linetype = c(0, 1, 1)))) +
theme(legend.position = 'bottom', legend.title = element_blank())

#' The following generates a gif animation
#' of the steps of the sampler (might take 10 seconds).
#+ Metropolis (1)
animate(p1 +
        transition_reveal(id=id, along=iter) +
        shadow_trail(0.01))

#' Plot the final frame
p1
```

删去前 50 次做热身，并取前 5000 次记做 dfs:

```
#' Take the first 5000 observations after warmup of 50
s <- 5000
warm <- 500
dfs <- data.frame(th1 = tt[(warm+1):s, 1], th2 = tt[(warm+1):s, 2])
```

画出热身后的前 1000 次迭代:

```
#' show 1000 draws after the warm-up
labs2 <- c('Draws', '90% HPD')
ggplot() +
geom_point(data = dfs[1:1000,],
           aes(th1, th2, color = '1'), alpha = 0.3) +
stat_ellipse(data = dft, aes(x = X1, y = X2, color = '2'), level = 0.9) +
coord_cartesian(xlim = c(-4, 4), ylim = c(-4, 4)) +
labs(x = 'theta1', y = 'theta2') +
scale_color_manual(values = c('steelblue', 'blue'), labels = labs2) +
guides(color = guide_legend(override.aes = list(
```

```
  shape = c(16, NA), linetype = c(0, 1), alpha = c(1, 1)))) +
theme(legend.position = 'bottom', legend.title = element_blank())
```

画出前 4500 次迭代：

```
#' show 4500 draws after the warm-up
labs2 <- c('Draws', '90% HPD')
ggplot() +
geom_point(data = dfs,
           aes(th1, th2, color = '1'), alpha = 0.3) +
stat_ellipse(data = dft, aes(x = X1, y = X2, color = '2'), level = 0.9) +
coord_cartesian(xlim = c(-4, 4), ylim = c(-4, 4)) +
labs(x = 'theta1', y = 'theta2') +
scale_color_manual(values = c('steelblue', 'blue'), labels = labs2) +
guides(color = guide_legend(override.aes = list(
  shape = c(16, NA), linetype = c(0, 1), alpha = c(1, 1)))) +
theme(legend.position = 'bottom', legend.title = element_blank())
```

计算 n_eff:

```
#' ### Convergence diagnostics
samp <- tt
dim(samp) <- c(dim(tt),1)
samp <- aperm(samp, c(1, 3, 2))
res<-monitor(samp, probs = c(0.25, 0.5, 0.75), digits_summary = 2)
neff <- res[,'n_eff']
# both theta have owen neff, but for plotting these are so close to each
# other, so that single relative efficiency value is used
reff <- mean(neff/(s/2))
```

可视化这个马尔科夫链

```
#' ### Visual convergence diagnostics

#' Collapse the data frame with row numbers augmented
#' into key-value pairs for visualizing the chains
dfb <- dfs
sb <- s-warm
dfch <- within(dfb, iter <- 1:sb) %>% gather(grp, value, -iter)
```

```
#' Another data frame for visualizing the estimate of
#' the autocorrelation function
nlags <- 50
dfa <- sapply(dfb, function(x) acf(x, lag.max = nlags, plot = F)$acf) %>%
data.frame(iter = 0:(nlags)) %>% gather(grp, value, -iter)

#' A third data frame to visualize the cumulative averages
#' and the 95% intervals
dfca <- (cumsum(dfb) / (1:sb)) %>%
within({iter <- 1:sb
uppi <-  1.96/sqrt(1:sb)
upp <- 1.96/(sqrt(1:sb*reff))}) %>%
gather(grp, value, -iter)

#' Visualize the chains
ggplot(data = dfch) +
geom_line(aes(iter, value, color = grp)) +
labs(title = 'Trends') +
scale_color_discrete(labels = c('theta1','theta2')) +
theme(legend.position = 'bottom', legend.title = element_blank())
```

可视化协方差函数

```
#' Visualize the estimate of the autocorrelation function
ggplot(data = dfa) +
geom_line(aes(iter, value, color = grp)) +
geom_hline(aes(yintercept = 0)) +
labs(title = 'Autocorrelation function') +
scale_color_discrete(labels = c('theta1', 'theta2')) +
theme(legend.position = 'bottom', legend.title = element_blank())
```

可视化置信区间

```
#' Visualize the estimate of the Monte Carlo error estimates
# labels
labs3 <- c('theta1', 'theta2',
           '95% interval for MCMC error',
```

```
          '95% interval for independent MC')
ggplot() +
geom_line(data = dfca, aes(iter, value, color = grp, linetype = grp)) +
geom_line(aes(1:sb, -1.96/sqrt(1:sb*reff)), linetype = 2) +
geom_line(aes(1:sb, -1.96/sqrt(1:sb)), linetype = 3) +
geom_hline(aes(yintercept = 0)) +
coord_cartesian(ylim = c(-1.5, 1.5), xlim = c(0,4000)) +
labs(title = 'Cumulative averages') +
scale_color_manual(values = c('red','blue',rep('black', 2)), labels = labs3) +
scale_linetype_manual(values = c(1, 1, 2, 3), labels = labs3) +
theme(legend.position = 'bottom', legend.title = element_blank())

#' Same again with r=0.99
```

与优化的关系：

- 若这次跳使得后验分布增加，设 $\theta^t = \theta^\star$;
- 若这次跳使得后验分布减小，以概率 r 设 $\theta^t = \theta^\star$, 否则设 $\theta^t = \theta^{t-1}$

可以视为分段找 mode 的随机版本：若增大密度，接受这一步；若降低密度，有时也接受。

收敛性

收敛性同 M-H 算法的收敛性。

4.3 独立情形的 M-H 算法

独立情形的 M-H 算法

step 1. 选一个初始的 $\theta^0 \sim p(\theta^0)$, 使得 $p(\theta^0|y) > 0$;

step 2. 对于 $t = 1, 2, \cdots$

(a) 提出建议 $\theta^\star \sim q(\theta^\star)$;

(b) 计算 $r = \frac{p(\theta^\star|y)q(\theta^\star)}{p(\theta^{t-1}|y)q(\theta^{(t-1)})}$;

(c) 设

$$\theta^t = \begin{cases} \theta^\star & \text{以概率 } \min(r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

定理 4.3.1. 如果存在常数 M 使得

$$f(x) \leq Mq(x), \quad \forall x \in supp(f)$$

则算法产生一个一致遍历的马尔科夫链，收敛速度是几何收敛速度：

$$\|K^n(x,\cdot) - f\|_{TV} \leq 2(1-1/M)^n$$

定理 4.3.2. 如果

$$f(x) \leq Mq(x), \quad \forall x \in supp(f)$$

成立，则链平稳后，算法的接受概率的期望至少是 $1/M$。

例子 4.3.3. 考虑从正态分布 $\mathcal{N}(0,1)$ 作为建议分布，产生柯西分布随机数 $f = \mathcal{C}(0,1)$，对应的 R 程序如下：

```
Nsim = 10^4
X = c(rt(1,1))  # initialize the chain from the stationary
for (t in 2:Nsim){
  Y=rnorm(1) # candidate normal
  rho=dt(Y,1)*dnorm(X[t-1])/(dt(X[t-1],1)*dnorm(Y))
   X[t]=X[t-1] + (Y - X[t-1])*(runif(1)<rho)
}
```

但是当初值 $X^{(0)}$ 太大, 比如说是 12.788. 在这种情况下 dnorm(X[t-1]) 等于 0, 并且

```
  pnorm(12.78,log=T,low=F)/log(10)
   [1] -36.97455
```

这就意味着超过 12.78 的概率是 10^{-37}, 因而马尔科夫链在这 10^4 迭代中保持常数。另外，序列中的大值将会产生重尾，导致链很长一段保持常数。直方图中单独的尖峰代表这种事件的发生。

相反，用 t 分布作为独立的建议分布，自由度是 0.5，(也就是用 Y=rt(1,.5) 代替 Y = rnorm(1))。对应的 R 程序如下：

```
Nsim = 10^4

Z =12.788  # initialize the chain from the stationary
for (t in 2:Nsim){
  Y=rt(1,.5) # candidate normal
  rho=dt(Y,1)*dt(Z[t-1],.5)/(dt(Z[t-1],1)*dt(Y,.5))
   X[t]=X[t-1] + (Y - X[t-1])*(runif(1)<rho)
}
```

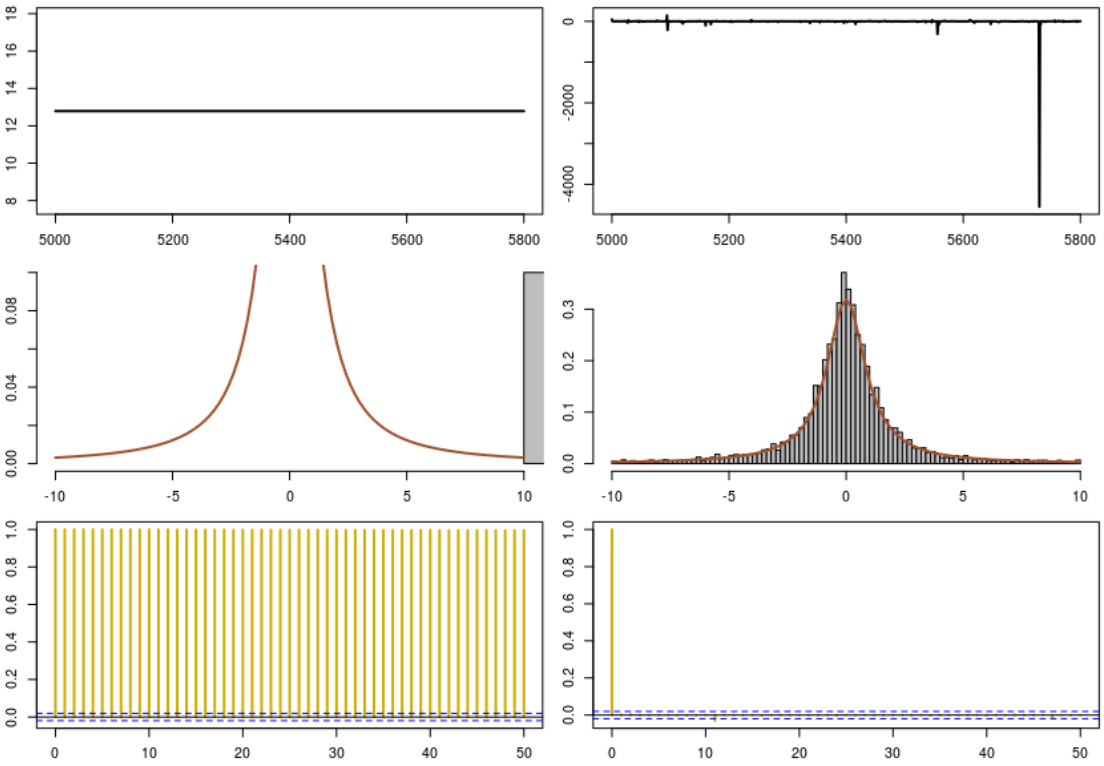

图 1: 用独立的 MH 算法采样柯西分布 $f = \mathcal{C}(0,1)$。左侧三幅图是用正态分布 Y = rnorm(1) 做建议分布，右侧三幅图用 t 分布 Y=rt(1,.5) 做建议分布。从上到下依次是目标参数的轨迹、目标参数的直方图、目标参数的自相关系数。

如图1

5 吉布斯采样 Gibbs sampling

5.1 两阶段的吉布斯采样

如果随机变量 X 和 Y 具有联合分布 $f(x,y)$, 两阶段的吉布斯采样根据如下的步骤生成一条马尔科夫链 (Markov Chain)(X_t, Y_t):

算法 5.1.1. 两阶段的吉布斯采样算法

对于 $t = 1, 2, \cdots,$

step 1. $Y_t \sim f_{Y|X}(\cdot|x_{t-1})$;

step 2. $X_t \sim f_{X|Y}(\cdot|y_t)$.

这里 $f_{Y|X}$ 和 $f_{X|Y}$ 是关于 f 的条件分布:

$$f_Y(y) = \int f(x,y)dx, \quad f_{X|Y}(x|y) = \frac{f(x,y)}{f_Y(y)}$$

注意不仅序列 (X_t, Y_t) 是一条马尔科夫链，并且每一个子链 (X_t) 和 (Y_t) 也分别是一条马尔科夫链。

Proof. • (X_t) 是一条马尔科夫链: 转移核的 density 是:

$$K(x, x^*) = \int f_{Y|X}(y|x) f_{X|Y}(x^*|y)dy$$

它确实只依赖最后一个 (X_t) 的值，与再之前的值无关.
另外，f_X 是对应的这个子链的平稳分布，因为

$$\begin{aligned} f_X(x') &= \int f_{X|Y}(x'|y) f_Y(y)dy \\ &= \int f_{X|Y}(x'|y) \int f_{Y|X}(y|x) f_X(x)dxdy \\ &= \int \left[\int f_{X|Y}(x'|y) f_{Y|X}(y|x)dy \right] f_X(x)dx. \\ &= \int K(x, x') f_X(x)dx \end{aligned}$$

- (X_t, Y_t) 是一条马尔科夫链: 转移核是:

$$K(x, y; x^*, y^*) = f_{Y|X}(y^*|x^*) f_{X|Y}(x^*|y)$$

它确实只依赖最后一个 (X_t, Y_t) 的值，与再之前的值无关.
另外, $f(x, y)$ 就是这个马尔科夫链的平稳分布，因为

$$\begin{aligned}
&\int f(x,y) K(x,y;x^*,y^*) dxdy \\
&= \int f(x,y) f_{Y|X}(y^*|x^*) f_{X|Y}(x^*|y) dxdy \\
&= \int [\int f(x,y)dx] f_{Y|X}(y^*|x^*) f_{X|Y}(x^*|y) dy \\
&= [\int f_Y(y) f_{X|Y}(x^*|y) dy] f_{Y|X}(y^*|x^*) \\
&= [\int f(x^*, y) dy] f_{Y|X}(y^*|x^*) \\
&= f_X(x^*) f_{Y|X}(y^*|x^*) \\
&= f(x^*, y^*)
\end{aligned}$$

□

例子 5.1.2 (二元正态分布的吉布斯采样)**.** 对于二元正态分布的特殊情况，

$$(X, Y) \sim \mathcal{N}_2 \left(0, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$$

对应的吉布斯采样器是: 给定 y_t，生成

$$\begin{aligned}
X_{t+1}|y_t &\sim \mathcal{N}(\rho y_t, 1-\rho^2) \\
Y_{t+1}|x_{t+1} &\sim \mathcal{N}(\rho x_{t+1}, 1-\rho^2)
\end{aligned}$$

Proof.

$$\begin{aligned}
f_{X|Y}(x|y) &= \frac{f(x,y)}{f_Y(y)} \\
f_Y(y) &= \int f(x,y) dx \\
f(x,y) &= \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left(-\frac{(x,y)\Sigma^{-1}(x,y)^T}{2}\right) \\
|\Sigma| &= det \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} = 1-\rho^2 \\
\Sigma^{-1} &= \frac{1}{1-\rho^2} \begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix} \\
f(x,y) &= \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{x^2+y^2-2\rho xy}{2(1-\rho^2)}\right)
\end{aligned}$$

因此，

$$\begin{aligned} f_Y(y) &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\infty}^{\infty} \exp\left(-\frac{x^2+y^2-2\rho xy}{2(1-\rho^2)}\right) dx \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) \end{aligned}$$

则

$$\begin{aligned} f_{X|Y}(x|y) &= \frac{f(x,y)}{f_Y(y)} \\ &= \frac{\frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{x^2+y^2-2\rho xy}{2(1-\rho^2)}\right)}{\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right)} \\ &= \frac{1}{\sqrt{2\pi}\sqrt{1-\rho^2}} \exp\left(-\frac{(x-\rho y)^2}{2(1-\rho^2)}\right) (\mathcal{N}(\rho y, 1-\rho^2)) \end{aligned}$$

□

注意对应的边缘马尔科夫链是 AR(1) 的时间序列：

$$X_{t+1} = \rho^2 X_t + \sigma\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1)$$

Proof.

$$k(x,x^*) = \int f(x^*|y) f(y|x) dy = \int \mathcal{N}(y\rho, 1-\rho^2)\mathcal{N}(x\rho, 1-\rho^2) dy = \mathcal{N}(x\rho, \sigma^2)$$

实际上，

$$\begin{aligned} k(x,x^*) &= \int \frac{1}{\sqrt{2\pi}\sqrt{1-\rho^2}} \exp(-\frac{(x^*-y\rho)^2}{2(1-\rho^2)}) \frac{1}{\sqrt{2\pi}\sqrt{1-\rho^2}} \exp(-\frac{(y-x\rho)^2}{2(1-\rho^2)}) dy \\ &= \frac{1}{\sqrt{2\pi}\sqrt{(1-\rho^4)}} \exp(-\frac{(x^*-\rho^2 x)^2}{2(1-\rho^4)}) (Note: \mathcal{N}(\rho^2 x, 1-\rho^4)) \end{aligned}$$

实际上，我们可以证明，如果 $Z = a\epsilon + b\eta,\ \epsilon, \eta \sim \mathcal{N}(0,1)$, 则 $Z \sim \mathcal{N}(0,\ )$.

$$\begin{aligned} P(Z \le z) &= \frac{1}{ab} \int_{-\infty}^{z} ds \int_{-\infty}^{\infty} dx p(a\epsilon = x, b\eta = s - a\epsilon) \\ &= \frac{1}{ab} \int_{-\infty}^{z} ds \int_{-\infty}^{\infty} dx f(\epsilon = \frac{x}{a}) f(\eta = \frac{s-x}{b}) \\ &= \frac{1}{ab} \int_{-\infty}^{z} ds \int_{-\infty}^{\infty} dx \frac{1}{\sqrt{2\pi}} \exp(-\frac{(\frac{x}{a})^2}{2}) \frac{1}{\sqrt{2\pi}} \exp(-\frac{(\frac{s-x}{b})^2}{2}) \\ &= \int_{-\infty}^{z} ds \frac{1}{\sqrt{2\pi}\sqrt{(a^2+b^2)}} \exp(-\frac{s^2}{2(a^2+b^2)}) \end{aligned}$$

因而,

$$f(Z=z)=\frac{1}{\sqrt{2\pi}\sqrt{(a^2+b^2)}}\exp(-\frac{s^2}{2(a^2+b^2)})$$

也就是, $Z\sim\mathcal{N}(0,a^2+b^2)$. 那么,

$$\begin{aligned}X_{t+1}&=\rho Y_t+\sqrt{1-\rho^2}\epsilon_{t+1}\\&=\rho(\rho X_t+\sqrt{1-\rho^2}\epsilon_t)+\sqrt{1-\rho^2}\epsilon_{t+1}\\&=\rho^2X_t+\sqrt{1-\rho^4}\epsilon,\quad(\epsilon\sim\mathcal{N}(0,1))\end{aligned}$$

□

这个链的平稳分布是 $\mathcal{N}(0,1)$.

Proof. 正态分布 $\mathcal{N}(\mu,\tau^2)$ 是 AR(1) 链 $X_{n+1}\sim\mathcal{N}(\theta x_{n-1},\sigma^2)$ 的平稳分布, 仅当

$$\mu=\theta\mu,\quad\tau^2=\tau^2\theta^2+\sigma^2$$

也就是，平稳的关系是:

$$\begin{aligned}&\int_B\frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{(x-\mu)^2}{2\tau^2}\right)dx\\&=\int_X dx\int_B dy\frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{(x-\mu)^2}{2\tau^2}\right)\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-\theta x)^2}{2\sigma^2}\right)\end{aligned}$$

利用 Fubini 定理, 交换积分顺序:

$$\begin{aligned}&\int_B\frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{(x-\mu)^2}{2\tau^2}\right)dx\\&=\int_B dy\int_X dx\frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{(x-\mu)^2}{2\tau^2}\right)\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-\theta x)^2}{2\sigma^2}\right)\end{aligned}$$

并把等式左边的变量符号 x 改成 y:

$$\begin{aligned}&\int_B\frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{(y-\mu)^2}{2\tau^2}\right)dy\\&=\int_B dy\int_X dx\frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{(x-\mu)^2}{2\tau^2}\right)\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-\theta x)^2}{2\sigma^2}\right)\end{aligned}$$

则，扔掉关于 y 的积分，变为:

$$\begin{aligned}&\frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{(y-\mu)^2}{2\tau^2}\right)dy\\&=\int_X dx\frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{(x-\mu)^2}{2\tau^2}\right)\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-\theta x)^2}{2\sigma^2}\right)\end{aligned}$$

删除相同的项 $\frac{1}{\sqrt{2\pi}\tau}$, 有:

$$\begin{aligned}
&\exp\left(-\frac{(y-\mu)^2}{2\tau^2}\right)dy \\
&= \int_X dx \exp\left(-\frac{(x-\mu)^2}{2\tau^2}\right)\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-\theta x)^2}{2\sigma^2}\right) \\
&= \underbrace{\int_X dx \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2}\left((\frac{1}{\tau^2}+\frac{\theta^2}{\sigma^2})(x-\frac{\frac{\mu}{\tau^2}+\frac{y\theta}{\sigma^2}}{(\frac{1}{\tau^2}+\frac{\theta^2}{\sigma^2})})^2\right)\right)}_{:=A} \\
&\times \exp\left(-\frac{(\frac{\mu}{\tau^2}+\frac{y\theta}{\sigma^2})^2}{\frac{1}{\tau^2}+\frac{\theta^2}{\sigma^2}}+(\frac{\mu^2}{\tau^2}+\frac{y^2}{\sigma^2})\right)
\end{aligned}$$

因为 $A=1$, 则有

$$(\frac{1}{\tau^2}+\frac{\theta^2}{\sigma^2}) = \frac{1}{\sigma^2}$$

也就是

$$\tau^2\sigma^2+\theta^2 = \tau^2$$

上述关系变成:

$$\begin{aligned}
&\exp\left(-\frac{(y-\mu)^2}{2\tau^2}\right)dy \\
&= \exp\left(-\frac{(\frac{\mu}{\tau^2}+\frac{y\theta}{\sigma^2})^2}{\frac{1}{\tau^2}+\frac{\theta^2}{\sigma^2}}+(\frac{\mu^2}{\tau^2}+\frac{y^2}{\sigma^2})\right) \\
&= \exp\left(-\frac{(y-\mu\theta)^2}{\tau^2}+C'\right)
\end{aligned}$$

因而有:

$$\mu\theta = \mu$$

则，平稳分布是 $\mathcal{N}(0, \frac{\sigma^2}{1-\theta^2})$. □

本例中, $\theta=\rho^2, \sigma^2 = 1-\rho^4$, 则平稳分布是 $\mathcal{N}(0, \frac{1-\rho^4}{1-\rho^4}) = \mathcal{N}(0,1)$.

5.2 缺失数据和隐变量模型

- 包含有辅助变量 z，完备数据似然 (complete-data likelihood) 或者叫完备模型 (complete -model):

  $$L^c(\theta|x,z) = f(x,z|\theta)$$

  对应与完备数据 (x,z) 的观测值. 这通常被称为去边缘化.

- 这个似然可以被表述为
$$g(x|\theta) = \int_{\mathcal{Z}} f(x, z|\theta)dz$$
- 并且，给定观察到的数据 x, 基于缺失数据 z 的条件分布 $k(z|\theta, x)$ 是
$$k(z|\theta, x) = \frac{f(x, z|\theta)}{g(x|\theta)}$$

5.3 切片采样，The slice sampler

5.3.1 基本定理

本节，我们想在 f 的子图上进行随机数生成:

$$\mathcal{R}(f) = \{(x, u) : 0 \leq u \leq f(x)\}$$

从 $\mathcal{R}(f)$ 上的某个点 (x, u) 开始，根据如下的条件分布沿着 u 轴的运动：

$$U|X = x \sim \mathcal{U}(\{u : u \leq f(x)\})$$

再根据如下的条件分布沿着 x 轴运动：

$$X|U' \sim \mathcal{U}(\{x : u' \leq f(x)\})$$

我们将不准确的称这个方法是 $2\mathcal{D}$ 切片采样 (slice sampler):

算法 5.3.1. $2\mathcal{D}$ **切片采样 (slice sampler):**
在迭代的第 t 步，模拟：

step 1. $u^{(t+1)} \sim \mathcal{U}_{[0, f(x^{(t)})]}$;

step 2. $x^{(t+1)} \sim \mathcal{U}_{A^{(t+1)}}$, 这里
$$A^{(t+1)} = \{x : f(x) \geq u^{(t+1)}\}$$

如果设 $f(x) = Cf_1(x)$ ，并且用 f_1 而不用 f(见练习5.3.2), 算法仍然有效。

练习 5.3.2. 根据 $2\mathcal{D}$ 切片采样算法，证明

(a) 算法生成的马尔科夫链的平稳分布是 $\{(x, u) : 0 \leq u \leq f(x)\}$ 上的均匀分布

(b) 证明如果在这个算法中用 f_1, 则 (a) 部分的结论仍然相同，其中 $f(x) = Cf_1(x)$ 。

现在我们想要证明这个算法生成一条马尔科夫链：

证明. 首先，如果 $x^{(t)} \sim f(x)$ 并且 $u^{(t+1)} \sim \mathcal{U}_{[0,f_1(x^{(t)})]}$，则

$$(x^{(t)}, u^{(t+1)}) \sim f(x)\frac{I_{[0,f_1(x)]}(u)}{f_1(x)} \propto I_{0\le u\le f_1(x)}$$

其次，如果 $x^{(t+1)} \sim \mathcal{U}_{A^{(t+1)}}$，则

$$(x^{(t)}, u^{(t+1)}, x^{(t+1)}) \sim f(x^{(t)})\frac{I_{[0,f_1(x^{(t)})]}(u^{(t+1)})}{f_1(x^{(t)})}\frac{I_{A^{(t+1)}}(x^{(t+1)})}{mes(A^{(t+1)})}$$

这里 $mes(A^{(t+1)})$ 表示集合 $A^{(t+1)}$ 的 Lebesgue 测度。这样

$$\begin{aligned}(u^{(t+1)}, x^{(t+1)}) &\sim C\int I_{0\le u\le f_1(x)}\frac{I_{f_1(x^{(t+1)})\ge u}}{mes(A^{(t+1)})}dx\\ &= CI_{0\le u\le f_1(x)}\int\frac{I_{u\le f_1(x)}}{mes(A^{(t+1)})}dx\\ &\propto I_{0\le u\le f_1(x)}\end{aligned}$$

这样子图 $\mathcal{R}(f)$ 上的均匀分布在这两步确实是平稳分布。

例子 5.3.3 (简单的切片采样). 考虑密度函数 $f(x) = \frac{1}{2}e^{-\sqrt{x}}$ for $x > 0$. 可以按照如下进行采样：

$$U|x \sim \mathcal{U}(0, \frac{1}{2}e^{-\sqrt{x}}),\quad X|u \sim \mathcal{U}(0, [\log(2u)]^2)$$

更进一步，还可以模拟练习5.3.4:

练习 5.3.4. 在练习5.3.2中，证明 $\mathbb{R}_+$ 上的密度函数 $\exp(-\sqrt{x})$ 对应的 cdf 可以用闭形式计算出来 (提示：做变量替换 $z = \sqrt{(x)}$, 对 $z\exp(-z)$ 进行分部积分)。

例子 5.3.5 (截断正态分布). 截断的正态分布 $N(-3, 1)$ 是限制在区间 $[0, 1]$ 上:

$$f(x) \propto f_1(x) = \exp\{-(x+3)^2/2\}\mathbb{I}_{[0,1]}(x)$$

5.4 回到吉布斯采样

- 切片采样可以表示为两阶段吉布斯采样的特例,
- 切片采样从 $f_X(x)$ 开始，创建一个联合密度函数 $f(x, u) = \mathbb{I}(0 < u < f_X(x))$.
- 相关的条件密度是

$$f_{X|U}(x|u) = \frac{\mathbb{I}(0 < u < f_X(x))}{\int\mathbb{I}(0 < u < f_X(x))dx},\quad f_{U|X}\frac{\mathbb{I}(0 < u < f_X(x))}{\int\mathbb{I}(0 < u < f_X(x))du}$$

- 因而，X 的序列也是一个马尔科夫链，转移核是

$$K(x,x') = \int f_{X|U}(x'|u) f_{U|X}(u|x) du$$

  平稳分布是 $f_X(x)$.

- 更进一步，可以从任意的边缘分布 $f_X(x)$ 诱导出一个吉布斯采样器，通过形式上创建一个联合分布，形式上是任意的。

- 从 $f_X(x)$, 可以取任意的条件密度函数 $g(y|x)$，并创建以下的吉布斯采样器：

$$f_{X|Y}(x|y) = \frac{g(y|x)f_X(x)}{\int g(y|x)f_X(x)dx}, \quad f_{Y|X}(y|x) = \frac{g(y|x)f_X(x)}{\int g(y|x)f_X(x)dy}$$

5.5 The Hammersley-Clifford Theorem

- 吉布斯采样最令人惊喜的特征是条件分布包含足够的信息、足以从联合分布生成样本.

  定理 5.5.1. 条件分布 $f_{Y|X}(y|x)$ and $f_{X|Y}(x|y)$ 相关的联合分布具有如下的联合密度

$$f(x,y) = \frac{f_{Y|X}(y|x)}{\int [\frac{f_{Y|X}(y|x)}{f_{X|Y}(x|y)}] dy}$$

- 通过和最大化问题进行比较，这个方法类似于在给定的基的方向上连续的最大化目标函数。

- 而优化方法众所周知的一点是，可能不会收敛到全局最大值，而很可能在局部的鞍点就终止了。

证明. 由于 $f(x,y) = f_{Y|X}(y|x)f_X(x) = f_{X|Y}(x|y)f_Y(y)$ 有

$$\int \frac{f_{Y|X}(y|x)}{f_{X|Y}(x|y)} dy = \int \frac{f_Y(y)}{f_X(x)} dy = \frac{1}{f_X(x)}$$

得证.

6 Monitoring Convergence to the Stationary Distribution

6.1 推断、评价收敛

从迭代的 simulation 中推断带来的问题

- 如果迭代不足，simulation 不足以代表 target 分布。即使 simulation 已经足够接近收敛，早期的迭代仍然影响总体的近似效果；
- 序列内的相关性: 收敛后，序列内的相关性影响有效样本数。

用三种方法来处理

(1) 设计 simulation 使得可以进行有效的 convergence monitor, 例如把初始 θ 在参数空间选得分散开

(2) 通过大致比较序列内和序列间的方差 (between and within simulated sequence)

(3) 若有效性仍旧不可接受，换算法。

具体来说

1. 去掉前一半的 simulation,warm-up
2. 对序列间的相关性，隔几个留一个样本
3. 从分散开的初始点多跑几条链
4. monitor 标量变化
5. monitoring convergence 的挑战:mixing and stationarity

a) 如图 11.3(a), 每条链单独看似乎均已稳定，但两条链并列说明还没收敛到共同的分布

b) 如图 11.3(b), 两条似乎存在一个共同的分布，但均不平稳。

这两个例子说明了当评价收敛时应考虑 between-sequence 和 within-sequence 的信息。

6. 把每条保存下来的链分成两半, 如图1

- 把每条链分成两半，检查最终全部的半链均已 mixed
- 假设 warm-up 部分已经被删去

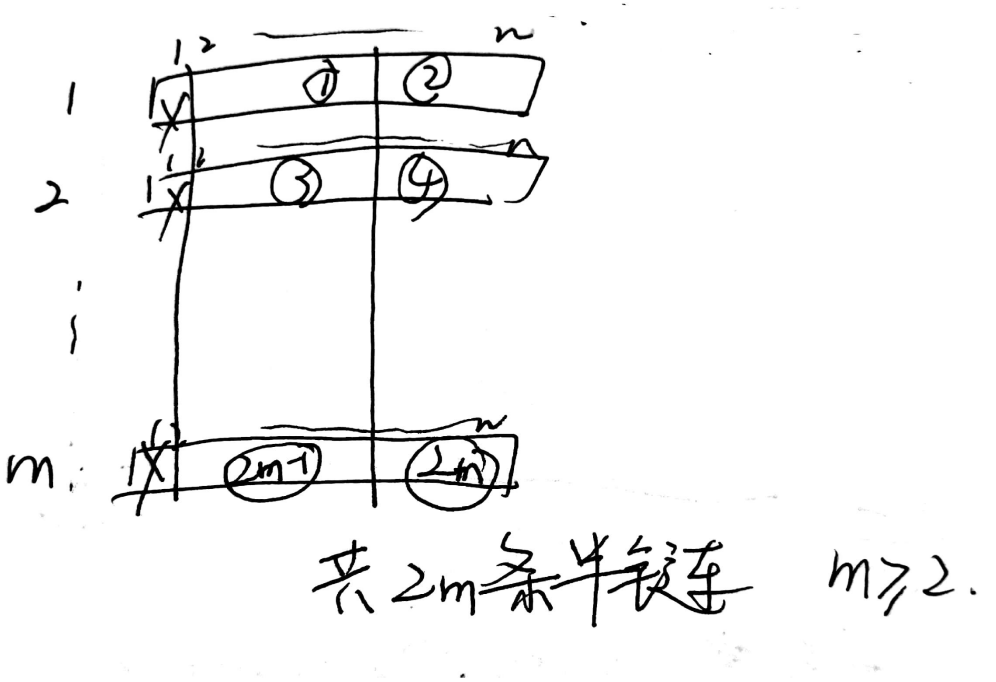

图 1: 共有 m($m \geq 2$) 条链，每条链有 n 个样本，删去 warm-up 后，每条链分为两半。共有 2m 条半链

7. 用序列内和序列间的方差 (Between-sequence, within-sequence) 评估 mixing:$\theta_{ij},\ i = 1, \cdots, n,\ j = 1, \cdots, m$

$$\text{Between-sequence } B = \frac{n}{m-1}\sum_{j=1}^{m}(\overline{\theta}_{\cdot,j} - \overline{\theta}_{\cdot\cdot})^2$$

$$\text{Within-sequence } W = \frac{1}{m}\sum_{j=1}^{m} s_j^2, \quad s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\theta_{ij} - \overline{\theta}_{\cdot j})^2$$

$$\overline{\theta}_{\cdot j} = \frac{1}{n}\sum_{i=1}^{n}\theta_{ij},\ \ \overline{\theta}_{\cdot\cdot} = \frac{1}{m}\sum_{j=1}^{m}\overline{\theta}_{\cdot j}$$

总方差定义为：

$$var(\theta|y) = \hat{var}^{+}(\theta|y) = \frac{n-1}{n}W + \frac{1}{n}B$$

这个统计量再平稳或者极限 $(n \to \infty)$ 的意义下是无偏的.

一般用伸缩变换后的方差：

$$\hat{R} = \sqrt{\frac{\hat{var}^{+}(\theta|y)}{W}}$$

当 $n \to \infty$ 时，降到 1.

若 $\hat{R}$ 较大，有理由相信，更多的迭代将改进推断.

例子 6.1.1 (二元正态分布). 以二元正态跳核

6.2 simulation draws 的有效数目

- $$\lim_{n\to\infty} mnvar(\overline{\theta}_{\cdot\cdot}) = (1 + 2\sum_{t=1}^{\infty}\rho_t)var(\theta|y)$$

  ρ_t 是 θ 在 lag t 处的自相关系数.

- $$n_{eff} = \frac{mn}{1 + 2\sum_{t=1}^{\infty}\rho_t}$$

- 下面估计 ρ_t

  定义

  $$V_t = \frac{1}{m(n-t)}\sum_{j=1}^{m}\sum_{i=t+1}^{n}(\theta_{ij} - \theta_{i-t,j})^2$$

  由 $E(\theta_i - \theta_{i-t})^2 = 2(1-\rho_t)var(\theta)$ 所以有

  $$\hat{\rho}_t = 1 - \frac{V_t}{2\hat{var}(\theta|y)}$$

  但 ρ_t 不能取到 ∞, 部分和一般从 lag 0 开始，直到连续两个 lag $\hat{\rho}_{2t} + \hat{\rho}_{2t+1} < 0$

  $$n_{eff} = \frac{mn}{1 + 2\sum_{t=1}^{T}\rho_t}$$

何时终止 simulation 计算 $\hat{R}$ 和 n_{eff}

- 如果 $\hat{R}$ 不是近似为 1，继续迭代; 一般以 1.1 为阈值
- $\hat{n}_{eff}$ 给出精度的意义：一般要保证 $\hat{n}_{eff} \geq 5m$，(例如，如果共有两条链，每条链至少有 10 个有效样本.)

6.3 Graphical diagnoses

例子 6.3.1 (Logit model using Monte Carlo EM and Metropolis-Hastings)**.** A simple random effect logit model processed in Booth and Hobert(1999) represents observations $y_{i,j}(i = 1, \cdots, n, j = 1, \cdots, m)$ as distributed conditionally on one covariate $x_{i,j}$ as a logit model:

$$P(y_{ij} = 1|x_{ij}, u_i, \beta) = \frac{\exp(\beta x_{ij} + u_i)}{1 + \exp(\beta x_{ij} + u_i)}$$

where $u_i \sim \mathcal{N}(0, \sigma^2)$ is an unobserved random effect. The vector of random effects $(U_1, \cdots, U_n)$ therefore corresponds to the missing data Z. When considering the function $Q(\theta'|\theta, x, y)$,

$$\begin{aligned} Q(\theta'|\theta, x, y) = &\sum_{i,j} y_{ij} \mathbb{E}[\beta' x_{ij} + U_i|\beta, \sigma, x, y] \\ &- \sum_{i,j} \mathbb{E}[\log[1 + \exp\{\beta' x_{ij} + U_i\}]|\beta, \sigma, x, y] \\ &- \sum_{i} \mathbb{E}[U_i^2|\beta, \sigma, x, y]/2\sigma'^2 - n \log \sigma' \end{aligned}$$

with $\theta = (\beta, \sigma)$. It is impossible to compute the expectations in U_i. The M-step would then almost straightforward for maximizing $Q(\theta'|\theta, x, y)$ in σ' leads to

$$\sigma'^2 = \frac{1}{n} \sum_i \mathbb{E}[U_i^2|\beta, \sigma, x, y]$$

while maximizing $Q(\theta'|\theta, x, y)$ in β' produces the fixed-point equation

$$\sum_{i,j} y_{ij} x_{ij} = \sum_{i,j} \mathbb{E}\left[\frac{\exp\{\beta' x_{ij} + U_i\}}{1 + \exp\{\beta' x_{ij} + U_i\}}|\beta, \sigma, x, y\right] x_{ij}$$

The alternative to EM is to simulate the $U_i's$ conditional on β, σ, x, y in order to replace the expectations above with Monte Carlo approximations.

$$\pi(u_i|\beta, \sigma, x, y) \propto \frac{\exp\{\sum_j y_{ij} u_i - u_i^2/2\sigma^2\}}{\prod_j [1 + \exp\{\beta x_{ij} + u_i\}]}$$

Opting for a standard random walk Metropolis-Hastings algorithm,we simulate both u_i, β from Normal distributions centered at the previous values of those parameters:

$$u_i^{(t)} \sim \mathcal{N}(u_i^{(t-1)}, \sigma^2), \quad \beta^{(t)} \sim \mathcal{N}(\beta^{(t-1)}, \tau)$$

The scale parameter σ can be simulated directly from an inverse gamma distribution:

$$\sigma \sim I\Gamma(1, 4 * \sum_i u_i^2/n)$$

6.4 Nonparametric tests of stationarity

Kolmogorov-Smirnov statistic

$$K = \frac{1}{M} \sup \left| \sum_{g=1}^{M} \mathbb{I}_{(0,\eta)}(x_1^{(gG)}) - \sum_{g=1}^{M} \mathbb{I}_{(0,\eta)}(x_2^{(gG)}) \right|$$

6.5 A missing Mass

To assess how much of the support of the target distribution has been explored by the chain via an evaluation of

$$\int_{\mathcal{A}} f(x)dx$$

If $\mathcal{A}$ denotes the support of the distribution of the chain. This is not necessarily easy, especially in large dimensions,but we can use the Riemann approximation method to assess it. When f is a one-dimensional density, the quantity

$$\sum_{t=1}^{T} [\theta^{(t+1)} - \theta^{(t)}] f(\theta^{(t)})$$

converges to 1, even when the $\theta^{(t)}$ are not generated from the density f.

例子 6.5.1 (Bimodal target)**.** Consider the density

$$f(x) = \frac{\exp(-x^2/2)}{\sqrt{2\pi}} \frac{4(x-0.3)^2 + 0.01}{4(1+(0.3)^2) + 0.01}$$

6.6 Geweke.diag function

Geweke.diag uitilize Spectral analysis. Geweke takes the first T_A and the last T_B observations from a sequence of length T to derive

$$\delta_A = \frac{1}{T_A} \sum_{t=1}^{T_A} h(x^{(t)}), \quad \delta_B = \frac{1}{T_B} \sum_{t=T-T_B+1}^{T} h(x^{(t)}),$$

and the estimates σ_A and σ_B based on both subsamples, respectively. The test statistic is then the asymptotically normal so-called Z-score

$$\sqrt{T}(\delta_A - \delta_B) / \sqrt{\frac{\sigma_A^2}{\tau_A} + \frac{\sigma_B^2}{\tau_B}},$$

with $T_A = \tau_A T, T_B = \tau_B T$, and $\tau_A + \tau_B < 1$. This is a t test which assess the equality of the means of the first and last parts of the Markov chain.

α	0.10	0.05	0.025	0.01	0.005	0.001
$c(\alpha)$	1.22	1.36	1.48	1.63	1.73	1.95

6.7 Kolmoforov-Smirnov statistic

The empirical distribution function F_n for n iid observations X_i is defined as

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{[-\infty,x]}(X_i)$$

The Kolmoforiv-Smirnov statistic for a given cumulative distribution function F(x) is

$$D_n = \sup_x |F_n(x) - F(x)|$$

- By the Glivenko-Cantelli theorem, D_n converges to 0 almost surely.
- Kolmogorov strengthened this result, by effectively providing the rate of this Convergence

$$\lim_n P(\sqrt{n}D_n \leq x) = 1 - 2\sum_{k=1}^{\infty}(-1)^{k-1}e^{-2k^2x^2}$$

- the goodness-of-fit test or the Kolmogorov-Smirnov test is constructed by using the critical values of the Kolmogorov distribution. The null hypothesis is rejected at level α if

$$\sqrt{n}D_n > K_\alpha$$

where K_α is found from

$$Pr(K \leq K_\alpha) = 1 - \alpha$$

6.8 Two-sample Kolmoforov-Smirnov test

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)|$$

where $F_{1,n}$ and $F_{2,m}$ are the empirical distribution functions of the first and the second sample respectively. The null hypothesis is rejected at level α if

$$D_{n,m} > c(\alpha)\sqrt{\frac{n+m}{nm}}$$

and in general by

$$c(\alpha) = \sqrt{-\frac{1}{2}\ln(\frac{\alpha}{2})}$$

6.9 summary–convergence diagnosing

- randogibs.R:

```
library(coda)
plot(mcmc(cbind(beta, sigma)))
browser()
cumuplot(mcmc(cbind(beta, sigma)))
list(beta = beta, sigma = sigma)
```

结果如图2 3

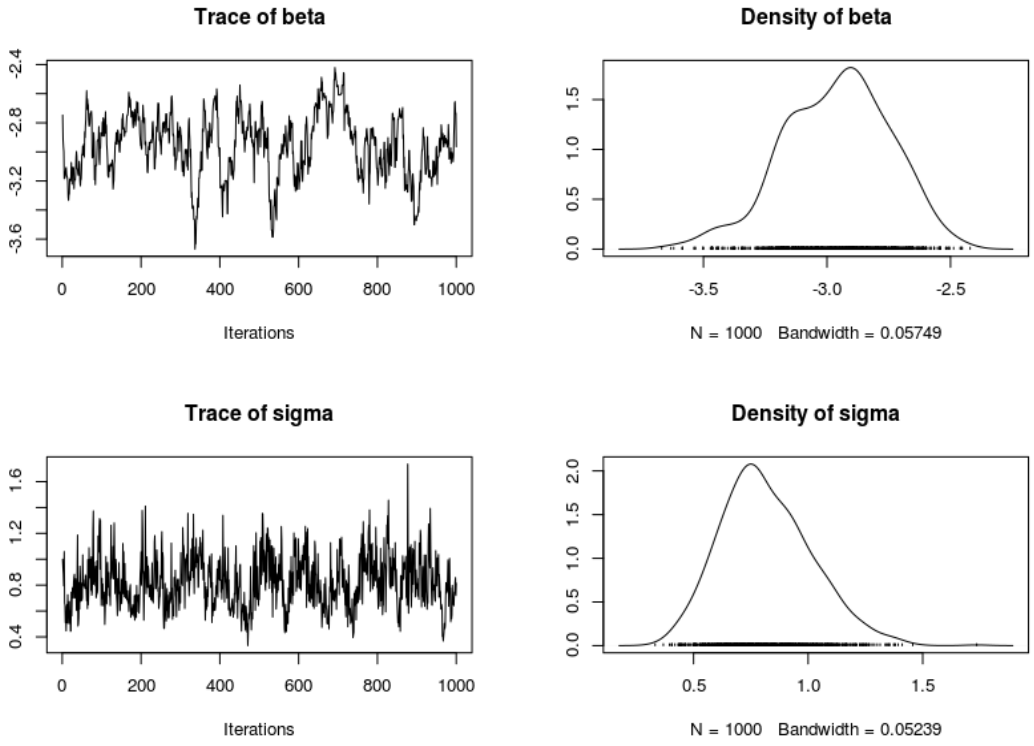

图 2: plot(mcmc)

- kscheck.R

```
#oldks
ks=NULL
```

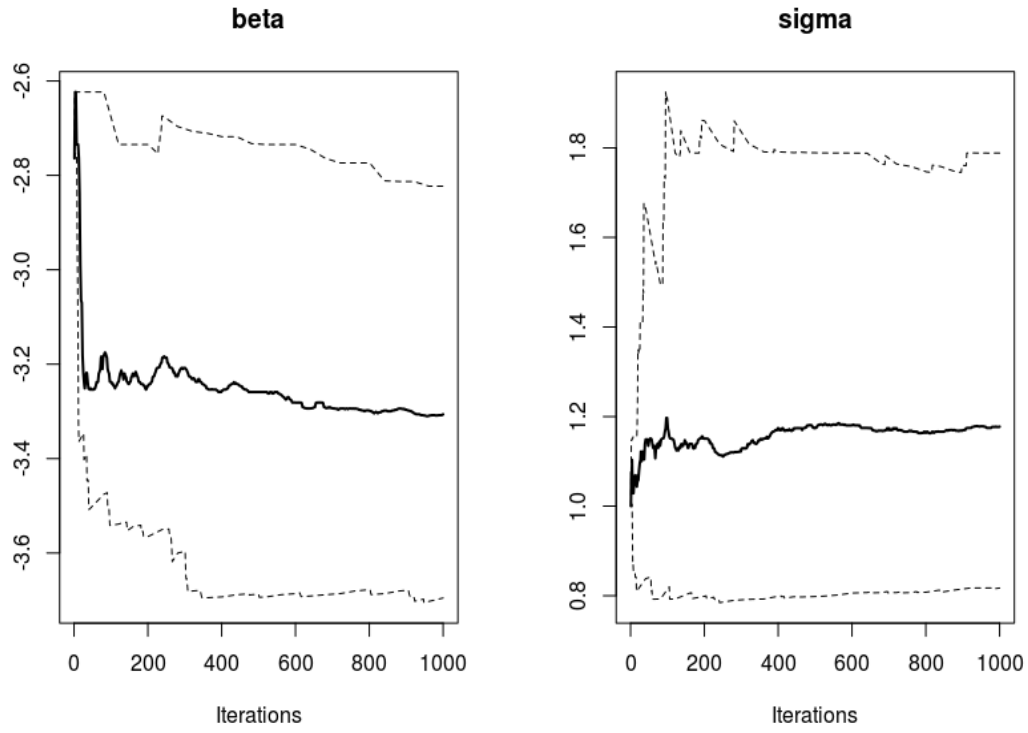

图 3: cumuplot

```
M=10

for (t in seq(T/10,T,le=100)){
    beta1=beta[1:(t/2)]
    beta2=beta[(t/2)+(1:(t/2))]
    beta1=beta1[seq(1,t/2,by=M)]
    beta2=beta2[seq(1,t/2,by=M)]
    ks=c(ks,ks.test(beta1,beta2)$p)
    }
```

$t = seq(1000, 10000, by = 90)$
for example, if $t = 1000$,
$\beta_1 = \beta(1:500), \beta_2 = \beta(501:1000)$,
$\beta_1 = \beta_1(seq(1, 500, by = 10))$,$\beta_2 = \beta_2(seq(1, 500, by = 10))$

```
oldbeta=beta[seq(1,T,by=M)]
olks=ks

#dual chain KS:
 #new ks
beta=beta[seq(1,T,by=M)]
ks=NULL
for (t in seq((T/(10*M)),(T/M),le=100))
     ks=c(ks,ks.test(beta[1:t],oldbeta[1:t])$p)
```

$\beta = \beta(seq(1, 10000, by = 10))$
$t = seq(100, 1000, by = 9)$
if $t = 100$,$beta(1:100), old\beta(1:100)$
if $t = 1000$,$\beta(1:1000), old\beta(1:1000)$

```
  #figure
par(mar=c(4,4,1,1),mfrow=c(2,1))
plot(seq(1,T,le=100),olks,pch=19,cex=.7,
xlab="Iterations",ylab="p-value")
plot(seq(1,T,le=100),ks,pch=19,cex=.7,
xlab="Iterations",ylab="p-value")
```

从图4中可以看出，第一种 ks 计算方法，看不出是否平稳；第二幅图是比较了两条不同的链。图中说明了需要更长时间达到平稳。迭代 4000 次左右两个样本有类似的经验 cdf，之后又探索了不同的空间。

```
heidel.diag(mcmc(beta))
geweke.diag(mcmc(beta))
```

在 sqar 中讲解。

- sqar.R

```
thn=jitter(xmc[seq(1,T,by=M)])
kst=NULL
for (m in seq(T/(10*M),T/M,le=100))]
 kst=c(kst,ks.test(thn[1:(m/2)],
 thn[(m/2)+(1:(m/2))])$p)
```

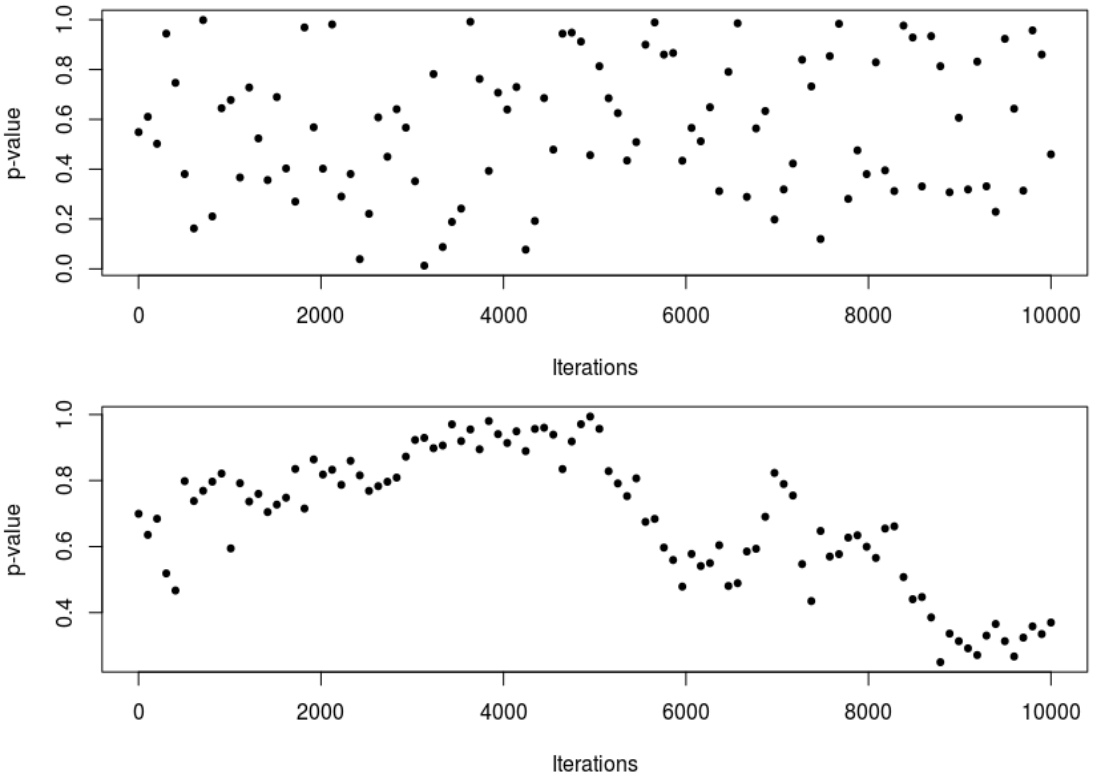
p-value
0.0
0.2
0.4
0.6
0.8
1.0
0
2000
4000
6000
8000
10000
Iterations
p-value
0.4
0.6
0.8
1.0
0
2000
4000
6000
8000
10000
Iterations

图 4: kscheck

kscheck 中第一种方法。

```
hist(xmc,pro=T,col="grey85",nclass=150,main="",
ylab="",xlab="")
ordin=apply(as.matrix(seq(min(xmc),max(xmc),le=200)),
1,eef)
lines(seq(min(xmc),max(xmc),le=200),
ordin*max(density(xmc)$y)/max(ordin),lwd=2,
col="gold4")
plot(seq(1,T,le=100),kst,pch=19,cex=.5,
xlab="Iterations",ylab="p-value")
```

画出直方图和分布，ks 统计量5。

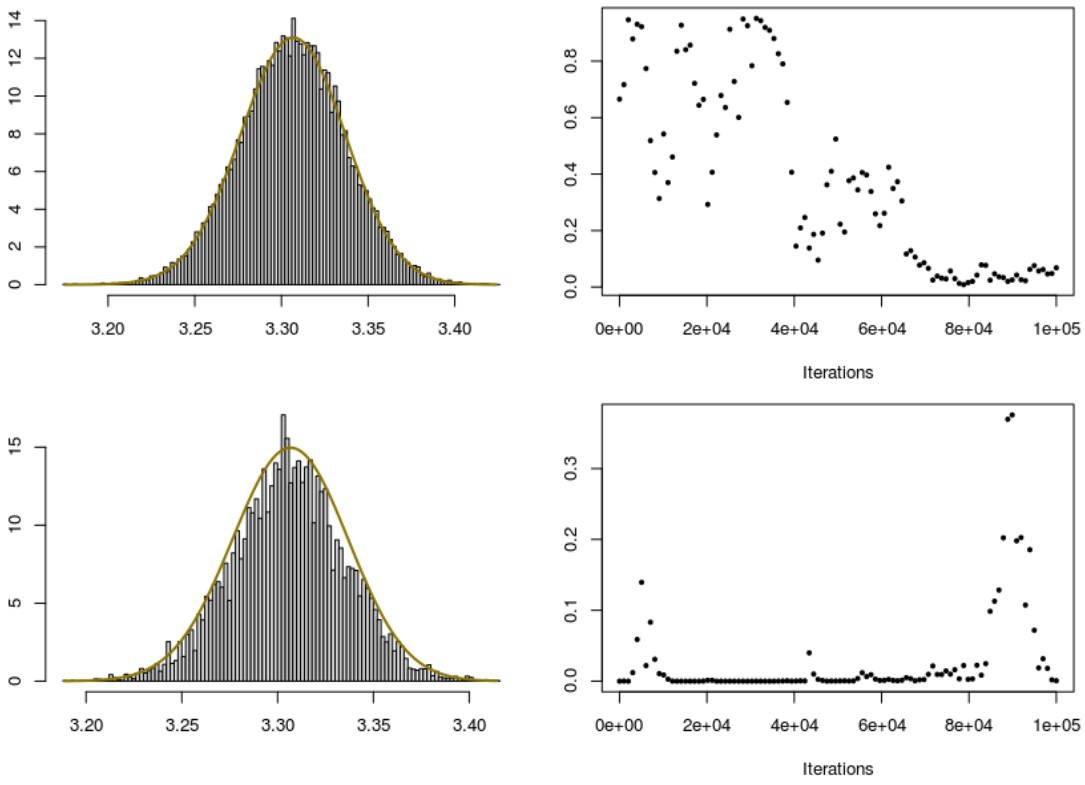

图 5: single chains

```
print(geweke.diag(mcmc(xmc)))
```

谱分析。统计量服从 $\mathcal{N}(0,1)$. 返回值是 p-value. 执行双边检验的假设检验。计算 $\Phi(|X| > p - value)$ 与 significance level 进行比较。如果返回负值，利用 geweke.diag 查看。

```
print(heidel.diag(mcmc(xmc)))
```

如图6

```
Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

  var1
0.9636


     Stationarity start     p-value
     test         iteration
var1 passed       1         0.934

     Halfwidth Mean Halfwidth
     test
var1 passed    3.31 0.001
```

图 6: single chains

- sqar.R,multiple chains:

```
par(mfrow=c(3,2),mar=c(4,2,1,1))
for (1 in 2:5)
  plot(smpl[,t],type="l",ylim=range(smpl),
  xlab="Iterations",ylab="",col=heat.colors(10)[6-t])

plot(smpl[,1],type="l",ylim=range(smpl),
xlab="Iterations",ylab="",col=heat.colors(10)[5])
for (t in 2:5) lines(smpl[,t],col=heat.colors(10)[6-t])
```

结果如图7

```
par(mfrow=c(3,2),mar=c(4,2,1,1))
for (t in 1:5)
plot(density(smpl[,t],n=1024),main="",ylab="",
xlab=paste("Bandwith",format(density(smpl)$b,
```

```
dig=3),sep="␣"),lwd=2)
plot(density(smpl,n=1024),main="",ylab="",
xlab=paste("Bandwith",format(density(smpl)$b,
dig=3),sep="␣"),lwd=2)
```

结果如图8

```
par(mfrow=c(1,2),mar=c(4,2,1,1))
plot(smpl[,1],type="l",ylim=range(smpl),xlab="Iterations",
ylab="",col=heat.colors(10)[5])
for (t in 2:5) lines(smpl[,t],col=heat.colors(10)[6-t])
plot(density(smpl,n=1024),main="",ylab="",
xlab=paste("Bandwith",format(density(smpl)$b,dig=3),sep="␣"),lwd=2)
```

结果如图9

```
plot(mcmc.list(mcmc(smpl[,1]),mcmc(smpl[,2]),
mcmc(smpl[,3]),mcmc(smpl[,4]),mcmc(smpl[,5])))
```

结果如图10

metropolis-hastings:

- hist:

```
ks.test(jitter(X),rbeta(5000,a,b))

par(mfrow=c(1,2),mar=c(2,2,1,1))
hist(X,nclass=150,col="grey",main="Metropolis-Hastings",fre=FALSE)
curve(dbeta(x,a,b),col="sienna",lwd=2,add=TRUE)
hist(rbeta(5000,a,b),nclass=150,col="grey",main="Direct␣Generation",fr
curve(dbeta(x,a,b),col="sienna",lwd=2,add=TRUE)
```

结果如图1112

- acf

```
par(mfrow=c(2,2),mar=c(4,4,2,2))
hist(X1,col="grey",nclas=125,freq=FALSE,xlab="",main="Accept-Reject",xli
curve(dgamma(x, a, rate=1),lwd=2,add=TRUE)
hist(X2[2500:nsim],nclas=125,col="grey",freq=FALSE,xlab="",main="Metrop
```

```
curve(dgamma(x, a, rate=1),lwd=2,add=TRUE)
acf(X1,lag.max=50,lwd=2,col="red")                    #Accept-Reject
acf(X2[2500:nsim],lag.max=50,lwd=2,col="blue")        #Metropolis-Hastings
```

结果如图13

- ```
  plot(cumsum(X<3)/(1:nsim),lwd=2,ty="l",ylim=c(.85,1),xlab="iterations",ylab="")
  lines(cumsum(Z<3)/(1:nsim),lwd=2,col="sienna")
  ```

结果如图]reffig:cumsum

- mass

```
#Corresponding sequence of masses
mass=0*(1:2000)
that=thet[1]
for (i in 2:2000){
 that=sort(c(that,thet[i]))
 mass[i]=sum((that[2:i]-that[1:(i-1)])*f(that[1:(i-1)]))
 }

# Plots
plot(mass,type="l",ylab="mass",col="sienna4",lwd=2)
par(new=T);plot(thet,pch=5,axes=F,cex=.3,col="steelblue2",ylab="")
```

结果如图15
```

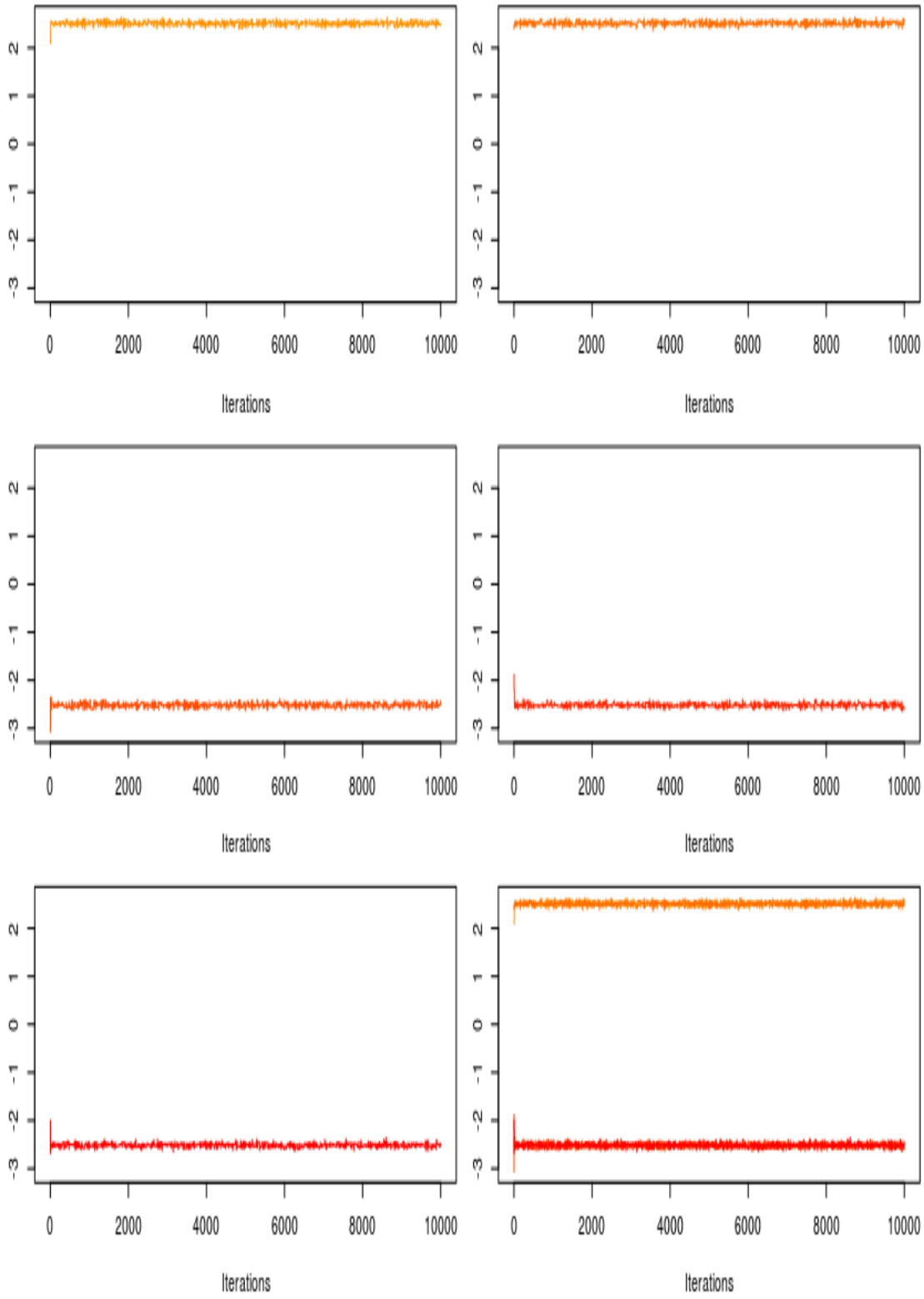

图 7: multiple chains

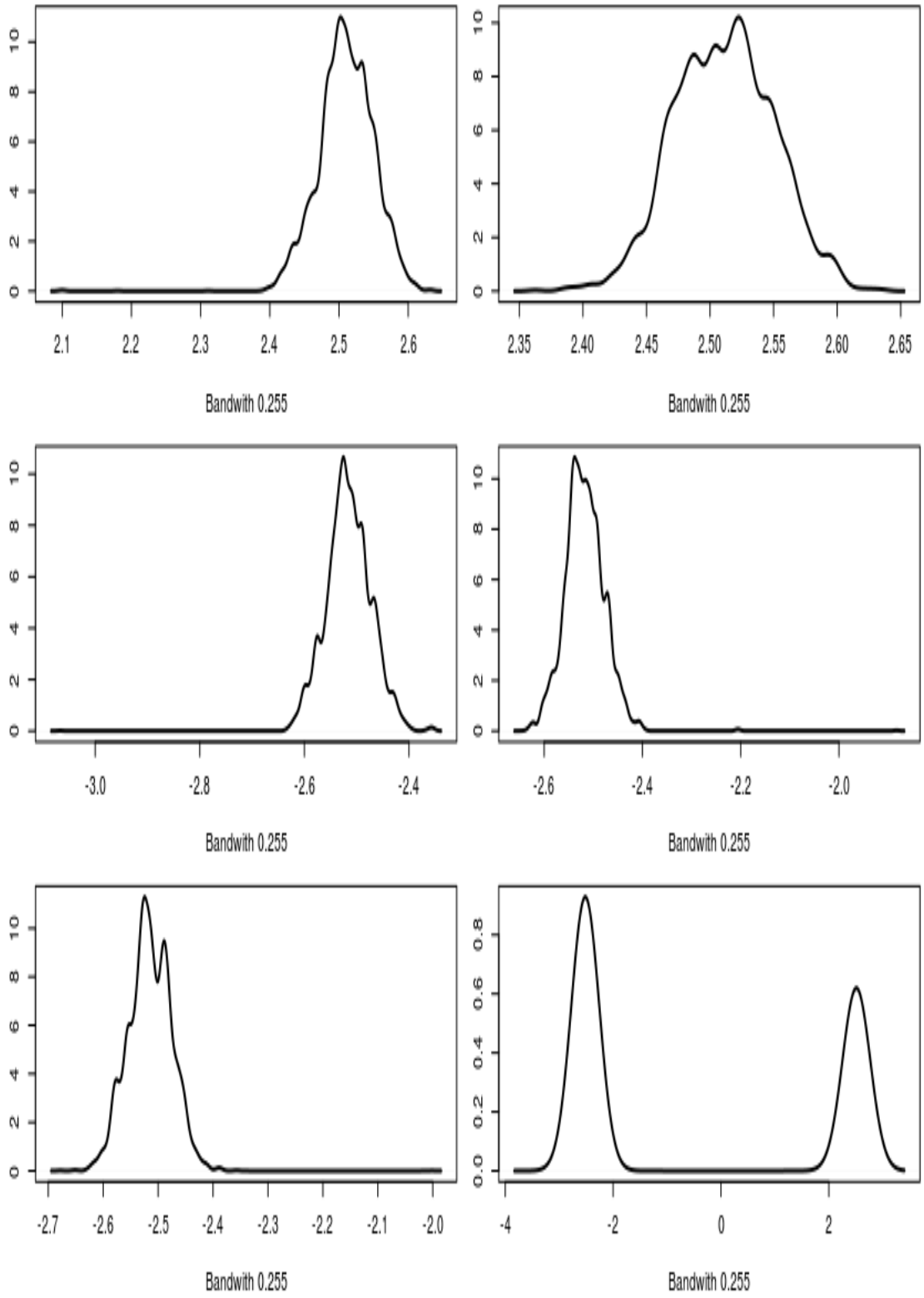

图 8: multiple chains

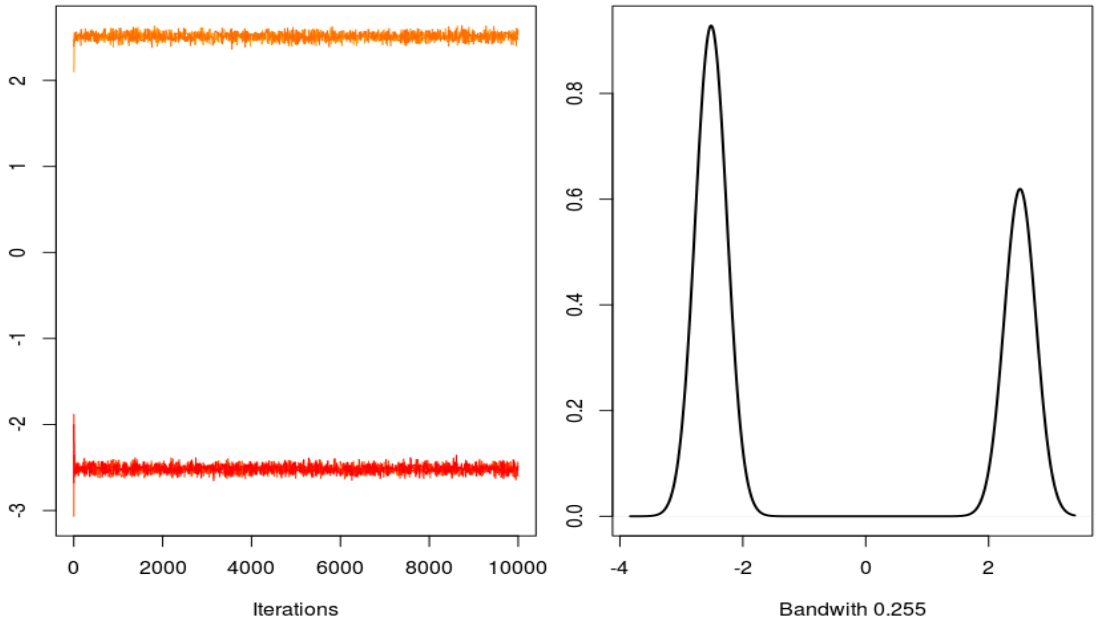

图 9: multiple chains

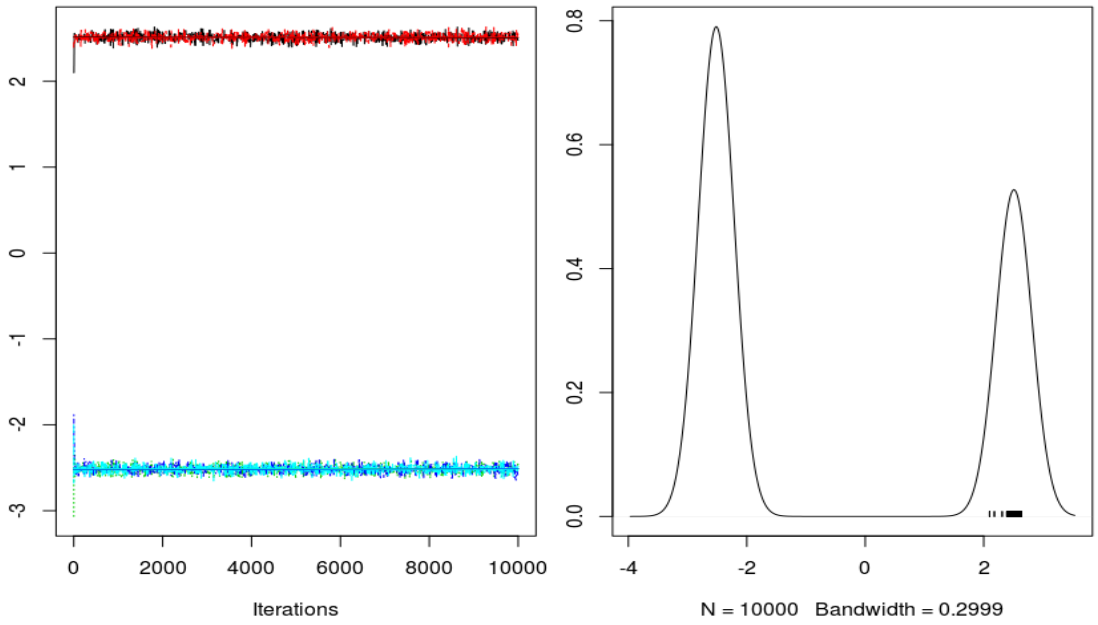

图 10: multiple chains

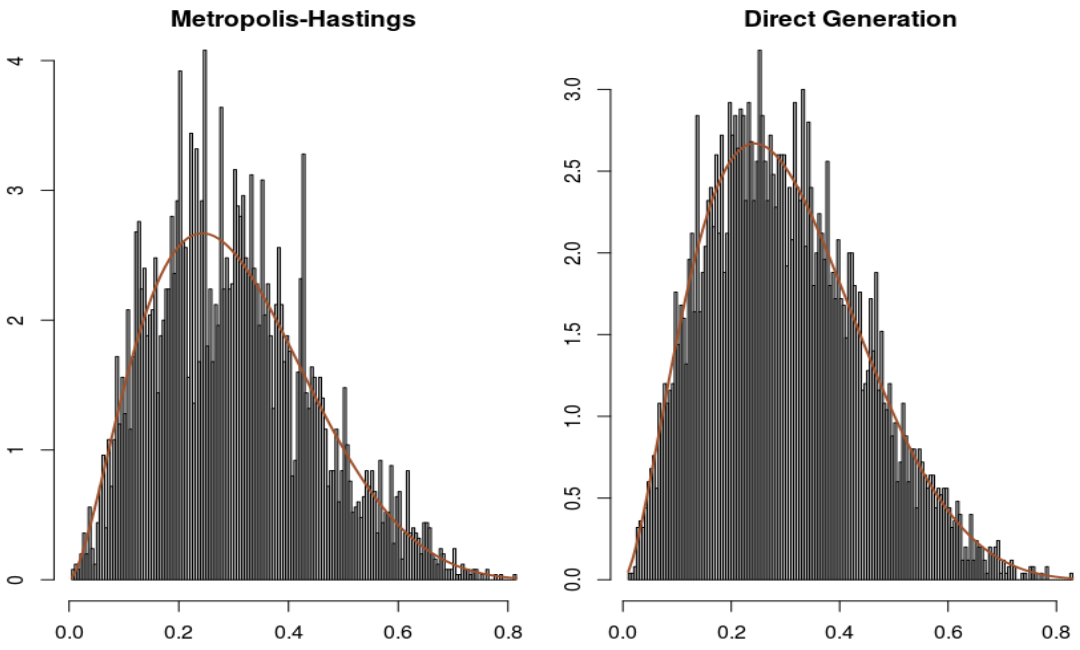

图 11: hist& curve

```
> ks.test(jitter(X),rbeta(5000,a,b))

        Two-sample Kolmogorov-Smirnov test

data:  jitter(X) and rbeta(5000, a, b)
D = 0.03, p-value = 0.02222
alternative hypothesis: two-sided
```

图 12: ks.test

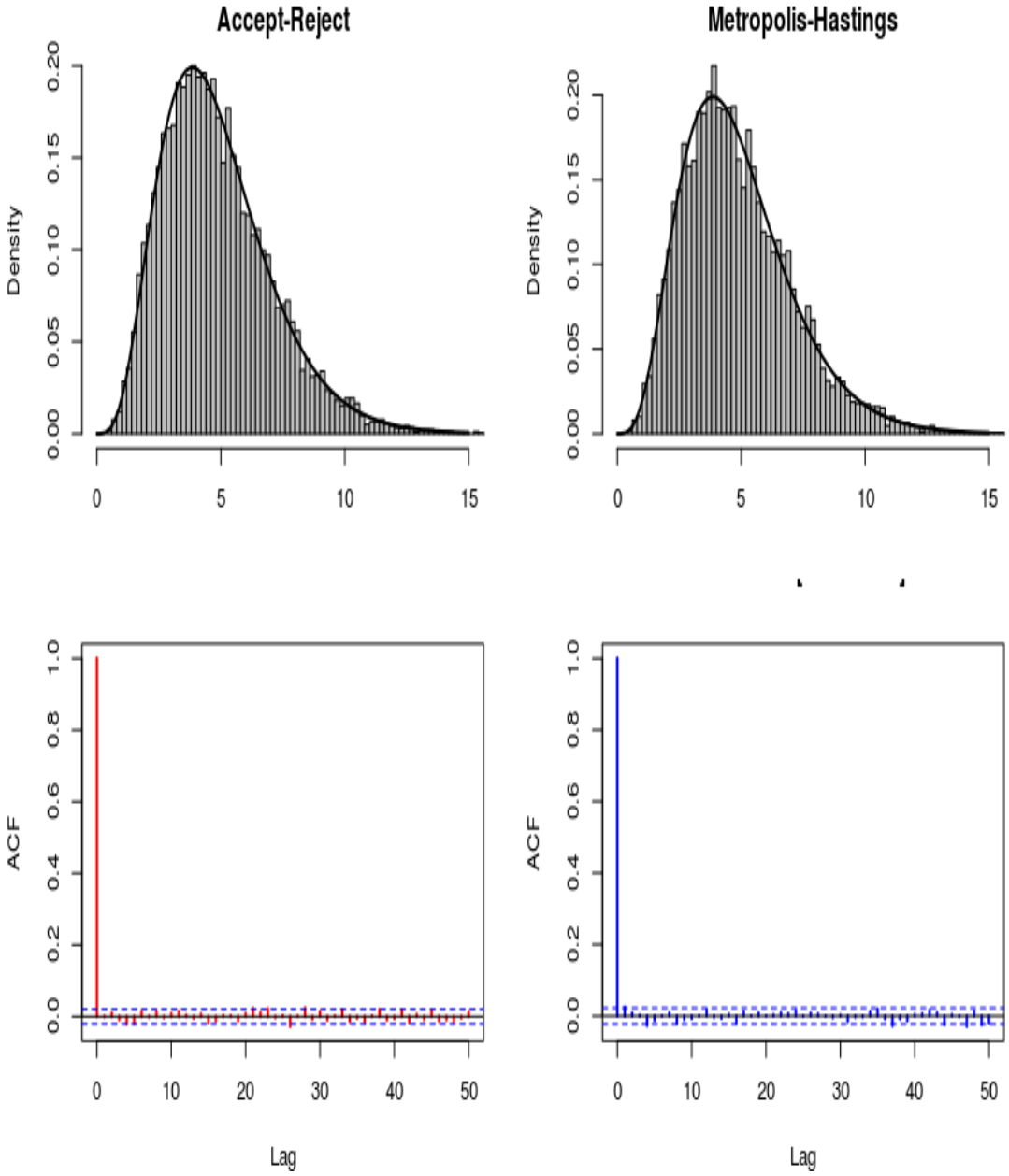
Accept-Reject
Metropolis-Hastings
Density
ACF
Lag
0.00
0.05
0.10
0.15
0.20
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
30
40
50

图 13: acf

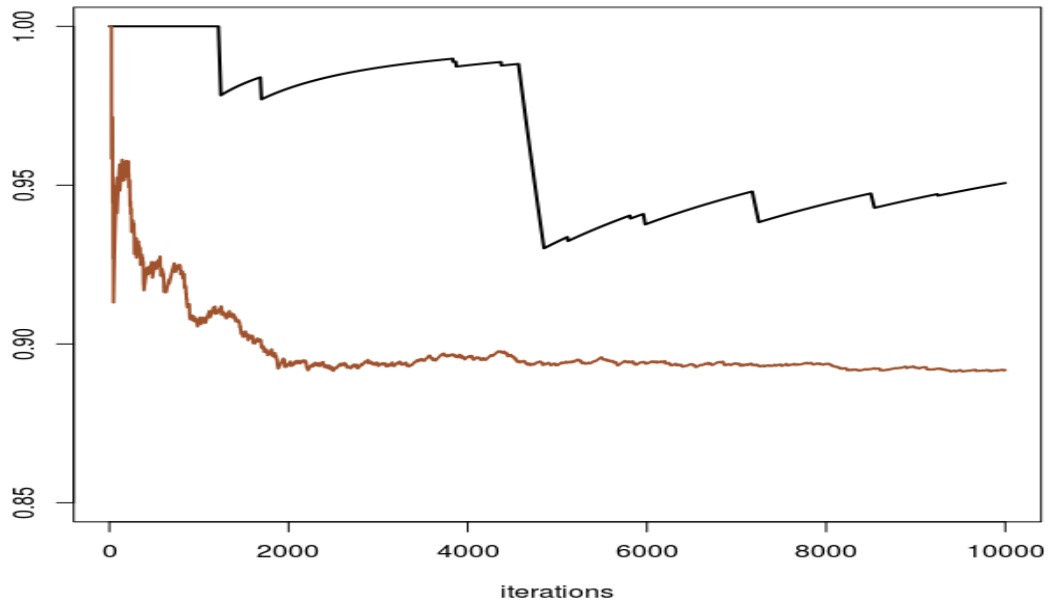

图 14: cumsum

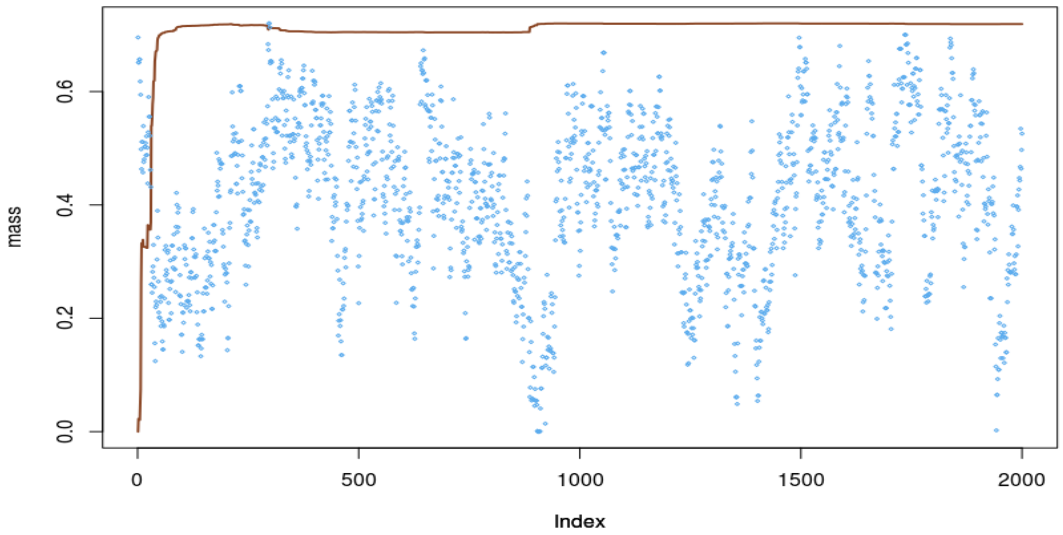

图 15: mass