



应用随机过程

张波 商豪 邓军

中国人民大学 统计学院

1/52



GoBack

FullScreen

Close

Quit



第11章 Markov链Monte Carlo方法

- § 11.1 计算积分的Monte Carlo方法
- § 11.2 Markov链Monte Carlo方法简介
- §11.3 Metropolis-Hastings算法
- §11.4 Gibbs抽样
- §11.5 贝叶斯MCMC估计方法

2/52



GoBack

FullScreen

Close

Quit



3/52

Monte Carlo方法也即随机模拟方法的别称，在应用统计学、金融学、经济计量学、计算物理学等多个领域有着广泛的应用。法国数学家蒲丰（Buffon）于1777年提出的利用投针实验问题，利用针与平行线相交的频率近似表示相交概率，从而求得圆周率，这被看成是Monte Carlo方法的起源。而由“计算机之父”和“博弈论之父”冯·诺依曼（John von Neumann）等制订的使用随机数处理确定性数学问题的Monte Carlo方法极大推动了数值分析的发展。冯·诺依曼使用摩洛哥的赌城Monte Carlo来命名这种方法。



GoBack

FullScreen

Close

Quit



Monte Carlo方法的基本原理是：当求解随机事件发生的概率或随机变量的数学期望时，通过设计的某种实验，得出某个特定事件发生的频率，使用这个频率来近似表示这一事件发生的概率，从而得到问题的数值解。在Monte Carlo方法中包含三个核心问题，分别为：构造概率过程、从已知概率分布中抽样、建立估计量。

4/52



GoBack

FullScreen

Close

Quit



5/52

通过构造独立同分布的随机数来计算积分的Monte Carlo方法，也称为静态Monte Carlo方法。在维数非常高的情况下，因为计算量太大，使用静态Monte Carlo方法处理速度太慢。故引入一种动态Monte Carlo，即通过构造Markov链的极限平稳分布，来模拟计算积分的方法，称为Markov链Monte Carlo方法，以下均简记为MCMC (Markov Chain Monte Carlo)。本章简要介绍计算积分的 Monte Carlo方法，以及Metropolis-Hastings算法，Gibbs抽样方法。



GoBack

FullScreen

Close

Quit



§11.1 计算积分的Monte Carlo方法

静态Monte Carlo方法是通过构造独立同分布的随机数来计算积分，有频率法与期望法两种，我们分别给出例子。其中频率法以经典的Buffon投针问题为例。

例 11.1.1 平面上画着一些平行线，线之间的距离均为 a ，向此平面随机地投一长度为 $l(l < a)$ 的针投掷到平面上，求针与任一平行线相交的概率。

解：以 Z 表示针的中点到最近一条平行线的距离， β 表示针与平行线的交角，则 (β, Z) 可以确定针所落的位置，且有 $0 \leq Z \leq \frac{a}{2}, 0 \leq \beta \leq \pi$ ，即针的所有可能位置可用 βZ 平面上的矩形来表示，将此矩形区域记为 G 。



当针与平行线相交时， Z 与 β 应满足关系 $Z \leq \frac{l}{2} \sin \beta$ ，即为图11-1(b)中的阴影部分，将此区域记为 R .

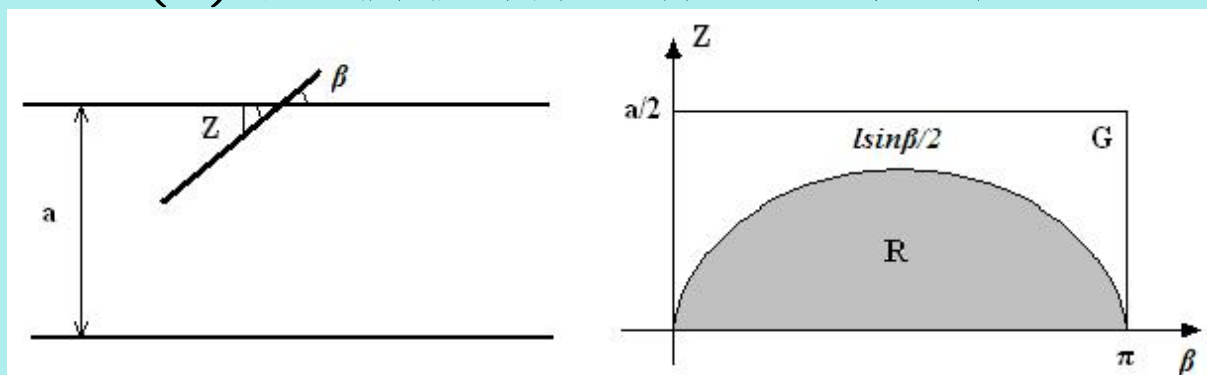


图11-1 Buffon投针图

故针与平行线相交的概率为：

$$p = \frac{L(R)}{L(G)} = \frac{\frac{1}{2} \int_0^\pi l \sin \beta d\beta}{\frac{1}{2} \pi a} = \frac{2l}{\pi a}$$



利用此例的答案来估算 π 的值，其方法是投针 N 次，记录针与线相交的次数 n ，再以频率值 $\frac{n}{N}$ 作为概率 p 的值，于是可得

$$\frac{n}{N} \approx \frac{2l}{\pi a}$$

即得

$$\pi \approx \frac{2lN}{an}$$

以下我们将使用 R 软件来实现 $Buffon$ 投针实验，其中两个关键的步骤为：

(1) 产生随机数： Z 在 $[0, \frac{a}{2}]$ 内随机取值， β 在 $[0, \pi]$ 内随机取值，也即首先需要生成服从 $[0, \frac{a}{2}]$ 上均匀分布的随机数，以及服从 $[0, \pi]$ 上均匀分布的随机数。



(2) 模拟投针实验：考察 n 次投针试验，若针与平行线相交，则表示试验成功，假设共成功 n 次，则 π 的估计值为 $\frac{2lN}{an}$ ，其中线之间的距离 a 及针的长度 $l(l < a)$ 均预先给定。

使用 R 软件来进行模拟，例如选取试验次数为 $N = 50000$ ，线之间的距离 $a = 1$ ，针的长度 $l = 0.75$ ，代码如下：

```

> touzhen < -function(N, l = 0.75, a = 1)
+{
+  n < -0; beta < -runif(N, 0, pi); z < -runif(N, 0, a/2)
+  for(i in 1 : N)
+  {
+    if(z[i] <= l * sin(beta[i])/2)
+      n < -n + 1
+  }
+  2 * l * N / (n * a)
+}
> touzhen(50000)

```



10/52



GoBack

FullScreen

Close

Quit



例 11.1.2 (随机投点法) 计算任一定义域为 $[0, 1]$ 的函数 $g(x)$ 在区间 $[0, 1]$ 上的积分 $\int_0^1 g(x)dx$.

解 假设随机变量 X 和 Y 均服从 $[0, 1]$ 上的均匀分布, 且相互独立, 则二维随机向量 (X, Y) 服从矩形区域 $[0, 1] \times [0, 1]$ 上的二维均匀分布, 联合概率密度为:

$$f(x, y) = \begin{cases} 1, & 0 < x < 1, 0 < y < 1 \\ 0, & \text{其他} \end{cases}$$

现用 B 表示事件 $\{w : Y \leq g(X)\}$ (g 为任一定义域为 $[0, 1]$ 的函数), 也即我们向矩形区域 $[0, 1] \times [0, 1]$ 随机投点, 其中点落在以 $[0, 1]$ 为底, 以函数 $g(x)$ 为曲边的曲边梯形内。

可计算事件 B 发生的概率为

$$P(B) = P\{Y \leq g(X)\} = \int_{Y \leq g(X)} f(x, y) dx dy$$



GoBack

FullScreen

Close

Quit



计算此二重积分可得

$$P(B) = \int_0^1 \left[\int_0^{g(x)} 1 dy \right] dx = \int_0^1 g(x) dx$$

也即积分 $\int_0^1 g(x) dx$ 的计算归结为考察上述事件 B 发生的概率。

由Bernoulli大数定律可知，可用重复试验中事件 B 发生的频率近似表示事件 B 发生的概率。

(1) 产生服从 $[0, 1]$ 上均匀分布的随机数。

(2) 模拟实验：考察 n 次投点实验，记录事件 $\{w : Y \leq g(X)\}$ 发生的频率，用来近似表示其发生的概率，即得到积分值 $\int_0^1 g(x) dx$ 。

例如，我们计算标准正态分布的密度函数在 $[0, 1]$ 上的积分 $\frac{1}{\sqrt{2\pi}} \int_0^1 e^{-\frac{x^2}{2}} dx$ ，使用R软件来实现这个试验过程，代码如下

下:

```
> mc1 < -function(n)
+{
+k < -0; x < -runif(n); y < -runif(n)
+for (i in 1:n)
+{
+if (y[i] < exp(-x[i]^2/2)/sqrt(2 * pi))
+k < -k + 1
+}
+k/n
+}
> mc1(10000)
```



13/52



GoBack

FullScreen

Close

Quit



另一方面，我们采用R软件中的内置函数integrate计算此积分

```
> integrate(function(x) exp(-x^2/2)/sqrt(2*pi), 0, 1)
0.3413447 with absolute error < 3.8e-15    (11.1.2)
```

两种方法计算出的积分值非常接近。



例 11.1.3 (平均值法) 计算定积分 $\int_0^1 g(x)dx$.

解 假设随机变量 X 服从 $[0, 1]$ 上的均匀分布, 由随机变量的函数的期望值的求法可知, $Y = g(X)$ 的期望值为

$$E[Y] = E[g(X)] = \int_0^1 g(x)dx$$

也即积分 $\int_0^1 g(x)dx$ 的计算归结为计算 $g(X)$ 的数学期望值。有辛钦大数定律知, 若 X_n 是独立同分布的随机变量序列, 则 $\frac{1}{n} \sum_{i=1}^n X_i$ 依概率收敛 $E[X_i]$, 故可用 $g(X)$ 的观察值的均值来估计 $g(X)$ 的期望值。

(1) 产生服从 $[0, 1]$ 上均匀分布的随机数 x_i .

(2) 对每个 x_i 计算 $g(x_i)$, 即可得到积分 $\int_0^1 g(x)dx$ 的估计值为 $\frac{1}{n} \sum_{i=1}^n g(x_i)$.



计算标准正态分布的密度函数在 $[0, 1]$ 上的积分 $\frac{1}{\sqrt{2\pi}} \int_0^1 e^{-\frac{x^2}{2}} dx$,
使用R软件来实现这个试验过程, 代码如下:

```
> mc3 <- function(n)
+{
+  x <- runif(n); f <- 1:n
+  for (i in 1:n)
+  f[i] <- exp(-x[i]^2/2)/sqrt(2*pi)
+  sum(f)/n
+}
> mc3(10000)
```

用平均值法计算出的结果与随机投点法以及直接计算定积分的值非常接近。



GoBack

FullScreen

Close

Quit

§11.2 Markov链Monte Carlo方法简介

在维数非常高的情况下，因为计算量太大，使用上节介绍的静态Monte Carlo方法处理速度太慢。动态Monte Carlo，即Markov链Monte Carlo方法的主要用于对维数非常高的随机向量取样。

MCMC方法首先应建立一个Markov链，使得其极限分布是平稳分布，则从目标分布中产生随机样本，就是从达到平稳状态的Markov链中产生样本路径。一个好的Markov链应该满足从任意位置出发能快速达到平稳分布这种性质。

MCMC方法的理论依据是几个极限定理：

首先，由第5章定理5.3.3可知，遍历的Markov链（即



不可约、正常返、非周期的Markov链) 的极限分布是平稳分布且是唯一的平稳分布;

考虑一个状态空间为 $S = \{1, 2, \dots, N\}$ 的Markov链 $\{X_n, n = 0, 1, 2, \dots\}$, 转移概率矩阵记为

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{pmatrix}$$

式中, $p_{ij}(i, j = 1, 2, \dots, N)$ 表示时齐的Markov链从状态*i*经过一步后转移到状态*j*的概率; 对于遍历的Markov链, 极限分布为

$$\pi_j = \lim_{n \rightarrow \infty} p_{ij}^n, \quad j \in S$$

π_j 表示经过长时间运行后系统处于状态*j*的时间的比例,



常令

$$\pi = (\pi_1, \pi_2, \dots, \pi_N)$$

则 $\pi_j (j = 1, 2, \dots, N)$ 是满足下列线性方程组的唯一解:

$$(\pi_1, \pi_2, \dots, \pi_N) = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{pmatrix} * (\pi_1, \pi_2, \dots, \pi_N)$$

$$\sum_{j=1}^N \pi_j = 1$$

或简记为:

$$\begin{cases} \pi = \pi P \\ \sum_{j=1}^N \pi_j = 1 \end{cases}$$



上述 $\pi_j (j = 1, 2, \dots, N)$ 被称为Markov链的平稳概率, 且针对任意 $j = 1, 2, \dots, N$, 均有 $\pi_j = \sum_{i=1}^N \pi_i p_{ij}$, 表示Markov链处于状态 j 的时间所占的比例等于Markov链从状态 i 转移到状态 j 所占的比例对 i 求和; $\sum_{j=1}^N \pi_j = 1$ 则表示Markov链处于状态 j 的时间所占的比例对 j 求和为1.

此定理说明了当此马尔科夫链运行足够长时间后分布 (Markov链的极限分布) 和初始分布 π_0 无关。

除了求解式 (11.2.1) 来获取平稳概率, 还有另外一种方法:



GoBack

FullScreen

Close

Quit



假设存在正数 $x_j (j = 1, 2, \dots, N)$, 使得

$$\begin{cases} x_i p_{ij} = x_j p_{ji} \\ \sum_{j=1}^N x_j = 1 \end{cases}$$

对所有状态 i 求和可得 $\sum_{i=1}^N x_i p_{ij} = x_j \sum_{i=1}^N p_{ji} = x_j$,
由式(11.2.1)的解的唯一性可知 $\pi_j = x_j$.

当对任意 $i \neq j$, 均有 $\pi_i p_{ij} = \pi_j p_{ji}$ 成立, 则称此Markov链是时间可逆的, 并称此方程为平衡方程。

这表示若用概率分布 $\{\pi_j, j = 1, 2, \dots, N\}$ 表示Markov链的初始分布, 则从任意时刻开始, 以相反方向考察系统的变化情况, 仍是一个转移概率为 p_{ij} 的Markov链。



其次，Markov链还有如下重要性质：

定理 11.2.1 （马尔可夫大数定律） 假设 $\{X_n, n = 0, 1, \dots\}$ 为一平稳分布为 π 的遍历的Markov链（其中每个 X_n 可以是多维的），则 X_n 依分布收敛到分布为 π 的随机变量 X ，且对任意函数 g ，当 $E_\pi[g(X)]$ 存在且 $n \rightarrow \infty$ 时，有

$$\overline{g_n} = \frac{1}{n} \sum_{i=1}^n g(X_i) \longrightarrow E_\pi[g(X)], \quad a.s.$$

定理11.2.1说明Markov链的实值函数的遍历均值几乎处处收敛到极限分布下的均值。

22/52



GoBack

FullScreen

Close

Quit



定理 11.2.2 若一个链是一致的几何遍历（转移速度以速度 $\lambda^t (0 < \lambda < 1)$ 收敛）的，并且 f 相对于平稳分布 π 是平方可积的，则有

$$\sqrt{(n)} \frac{\overline{f_n} - E_{\pi} f(x)}{\Gamma} \longrightarrow N(0, 1), \quad n \rightarrow \infty$$

定理11.2.2说明遍历均值做合适的变化依分布收敛到标准正态分布。

因此，我们可以建立一个以 π 为平稳分布的马氏链，则在运行此链足够长时间后，该马氏链就会达到平稳状态，此时马氏链的值就相当于从分布 π 中抽取的样本。



§11.3 Metropolis-Hastings算法

MCMC方法的重点在于构造合适的马氏链，因此算法的主要目的是对Markov链 $\{X_n, n = 0, 1, 2, \dots\}$ ，在给定一个 X_n 所处的状态下，产生下一步的状态 X_{n+1} 。本节我们介绍Hastings-Metropolis算法（简称M-H算法）来构造一个将某一个给定概率分布作为极限分布的Markov链。

假设我们感兴趣的目标分布为 $\{p_j, j \in S, \sum_{j \in S} p_j = 1\}$ 分布， $\mathbf{Q}$ 是状态空间为 $S = \{1, 2, \dots, N\}$ 的不可约Markov链的转移概率矩阵， $\mathbf{Q} = (q_{ij})(i, j = 1, 2, \dots, N)$ ；定义一个Markov链 $\{X_n, n = 0, 1, 2, \dots\}$ ，当 $X_n = i$ 时，生成一个随机变量使得其分布为：

$$P\{X = j\} = q_{ij}, \quad j = 1, 2, \dots, N$$



如果 $X = j$, 则 X_{n+1} 以概率 α_{ij} 转移到 j , 以概率 $1 - \alpha_{ij}$ 转移到 i . 如果取

$$\alpha_{ij} = \min\left\{\frac{\pi_j q_{ji}}{\pi_i q_{ij}}, 1\right\}$$

则 Markov 链 $\{X_n, n = 0, 1, 2, \dots\}$ 的转移概率矩阵为:

$$\begin{cases} p_{ij} = q_{ij}\alpha_{ij}, & \text{若 } j \neq i \\ p_{ii} = q_{ii} + \sum_{k \neq i} q_{ik}(1 - \alpha_{ik}) \end{cases}$$

且 $\pi_i q_{ij} \alpha_{ij} = \pi_j q_{ji} \alpha_{ji}$ (当 $j \neq i$), 亦即 $\pi_i p_{ij} = \pi_j q_{ji}$ (当 $j \neq i$),

(易知 $\alpha_{ij} = \frac{\pi_j q_{ji}}{\pi_i q_{ij}}$ 时, $\alpha_{ji} = 1$; $\alpha_{ij} = 1$ 时, $\alpha_{ji} = \frac{\pi_i q_{ij}}{\pi_j q_{ji}}$)



我们可得Metropolis-Hasting算法（简称为M-H算法），其关键步骤为：

（1）构造适合的建议分布 $g(\cdot|X_n)$ ，并产生一个服从建议分布的备选值 Y （其中建议分布为任意一个友好的分布，即知道如何从该分布中抽样。建议分布的选择要使得产生的马氏链的平稳分布为目标抽样分布 f ）。

（2）设定

$$X_{n+1} = \begin{cases} Y, & \text{以概率 } \alpha \\ X_n, & \text{以概率 } 1 - \alpha \end{cases}$$

式中，

$$\alpha = \alpha(X_n, Y) = \min\left\{\frac{f(Y)g(X_n|Y)}{f(X_n)g(Y|X_n)}, 1\right\}$$

在执行上述步骤（2）时，常采用的方法是产生服从均匀分布 $U(0,1)$ 的 U ，若 $U < \alpha$ ，则令 $X_{n+1} = Y$ ；否则



令 $X_{n+1} = X_n$.

将上述过程简要概括，即M-H抽样方法通过如下方式生产马氏链 $\{X_t, t = 0, 1, 2, \dots\}$:

(1) 构造合适的建议分布 $g(\cdot|X_t)$ ，并产生服从该分布的 X_0 .

(2) 从 $g(\cdot|X_t)$ 中产生 Y ，从 $U(0, 1)$ 中产生 U ，若

$$U \leq \frac{f(Y)g(X_t|Y)}{f(X_t)g(Y|X_t)}$$

则接受 Y 并令 $X_{t+1} = Y$ ，否则令 $X_{t+1} = X_t$

(3) 重复步骤 (2) 中的过程直至马氏链达到平稳状态。

过程中的接受概率为：

$$\alpha(X_t, Y) = \min\left\{1, \frac{f(Y)g(X_t|Y)}{f(X_t)g(Y|X_t)}\right\}$$



GoBack

FullScreen

Close

Quit



因此在计算过程中只要明确目标分布 f 与全概率密度函数正比于某个未知常数即可。

M-H算法是Metropolis抽样方法的推广。在Metropolis算法中，建议分布是对称的，即 $g(\cdot|X_t)$ 满足

$$g(X|Y) = g(Y|X)$$

因此接受概率为：

$$\alpha(X_t, Y) = \min\left\{1, \frac{f(Y)}{f(X_t)}\right\}$$



§11.4 Gibbs抽样

Gibbs抽样是M-H算法的一个特例，它是一种将高维问题转化为一维问题的方法。Gibbs抽样方法主要特点是分别逐一把每个分量进行抽取。在对每个分量进行抽取的时候，是对其他所有分量的条件分布进行抽样的。

以二维分布 (X_1, X_2) 为例说明Gibbs抽样的算法：

- (a) 令 $(x_1, x_2) = \mathbf{X}(t - 1)$;
- (b) 从 $f(x_1|x_2)$ 中产生候选点 $X_1^*(t)$ ，更新 $x_1 = X_1^*(t)$;
- (c) 从 $f(x_2|x_1)$ 中产生候选点 $X_2^*(t)$ ，更新 $x_2 = X_2^*(t)$;
- (d) 令 $\mathbf{X}(t) = (X_1^*(t), X_2^*(t))$

29/52



GoBack

FullScreen

Close

Quit



易推广至多维情况，令 $\mathbf{X} = (X_1, X_2, \dots, X_q)$ 是 q 维随机向量，定义相应的 $q - 1$ 维随机向量为：

$$\mathbf{X}_{-j} = (X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_q)$$

已知 $\mathbf{X}_{-j}$ 的情况下 X_j 的条件分布对应的概率密度函数为 $f(X_j|\mathbf{X}_{-j})$ ，称为满条件分布，一共有 q 个满条件分布，Gibbs抽样是从这 q 个满条件分布中产生候选点。 $\forall j = 1, 2, \dots, q$ ，从 $f(X_j|\mathbf{X}_{-j})$ 中产生候选点 $X_j^*(t)$ ，更新 $x_j = X_j^*(t)$ ；

从一元分布 $f(x_j|x_1(t), x_2(t), \dots, x_{j-1}(t), x_{j+1}(t), \dots, x_q(t-1))$ 中抽样较为简单，因为除了变量 x_j 外，其他变量均是常数。

30/52



GoBack

FullScreen

Close

Quit



例 11.4.1 使用Gibbs抽样方法产生服从二元正态分布的随机数。

解 设随机向量 (X, Y) 服从二维正态分布 $N(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho)$, 计算其边缘分布可知 X 服从一维正态分布 $N(\mu_1, \sigma_1^2)$, Y 服从一维正态分布 $N(\mu_2, \sigma_2^2)$; 然后计算给定 $Y = y$ 时 X 的条件概率密度函数为

$$f(x|y) = \frac{1}{\sqrt{2\pi}\sigma_1\sqrt{1-\rho^2}} \exp \left\{ -\frac{1}{2\sigma_1^2(1-\rho^2)} \left[x - \left(\mu_1 + \rho \frac{\sigma_1}{\sigma_2} (y - \mu_2) \right) \right]^2 \right\}$$

给定 $Y = y$ 时, X 仍然服从一维正态分布, 且有

$$E[X|Y = y] = \mu_1 + \rho \frac{\sigma_1}{\sigma_2} (y - \mu_2)$$
$$Var[X|Y = y] = \sigma_1^2 (1 - \rho^2)$$

同理可计算得给定 $X = x$ 时, Y 服从一维正态分布。



GoBack

FullScreen

Close

Quit



使用Gibbs算法的R代码为:

```
> N <- 6000      定义Markov链的长度
```

```
> drop <- 1000
```

```
> X <- matrix(0, N, 2)
```

```
> rho <- -0.65; mu1 <- -1; mu2 <- -2; sigma1 <- -1; sigma2 <- -0.8
```

```
> s1 <- sqrt(1 - rho^2) * sigma1; s2 <- sqrt(1 - rho^2) * sigma2
```

32/52



GoBack

FullScreen

Close

Quit

生成Markov链

```
> X[1,] <- -c(mu1, mu2)
> for(iin2 : N){
+ x2 <- -X[i - 1, 2]
+ m1 <- -mu1 + rho * sigma1/sigma2 * (x2 - mu2)
+ X[i, 1] <- -rnorm(1, m1, s1)
+ x1 <- -X[i, 1]
+ m2 <- -mu2 + rho * sigma2/sigma1 * (x1 - mu1)
+ X[i, 2] <- -rnorm(1, m2, s2)
+}
> b <- -drop + 1
> x <- -X[b : N,]
```



33/52



GoBack

FullScreen

Close

Quit



生成的Markov链的1000个观测被废弃掉，剩下的观测存入x中，可看出x的维数为5000行，2列。

```
> dim(x)
```

```
[1]50002
```

使用colMeans函数或apply函数计算x各列的样本均值，可与给出的 $\mu_1 = 1$, $\mu_2 = 2$ 进行对比；

```
> colMeans(x)
```

```
[1]0.96329442.0231725
```

```
> apply(x, 2, mean)
```

```
[1]0.96329442.0231725
```



使用cov函数计算x的样本协方差阵及相关系数阵

```
> cov(x)
      [, 1]      [, 2]
[1,] 0.9838777 -0.5144265
[2,] -0.5144265 0.6297255
> cor(x)
      [, 1]      [, 2]
[1,] 1.0000000 -0.6535476
[2,] -0.6535476 1.0000000
```

可与给出的 $\sigma_1 = 1$, $\sigma_2 = 0.8$, $\rho = -0.65$ 进行比较, 发现各个参数的样本估计值与真值是比较接近的。



使用plot函数绘制Gibbs抽样得到的二元分布的散点图:

```
> plot(x, main = "", cex = 0.5, xlab = bquote(X[1]),  
      ylab = bquote(X[2]), ylim = range(x[, 2]))
```

散点图也显示出二元正态分布所具有的球面对称性和负相关性特征。

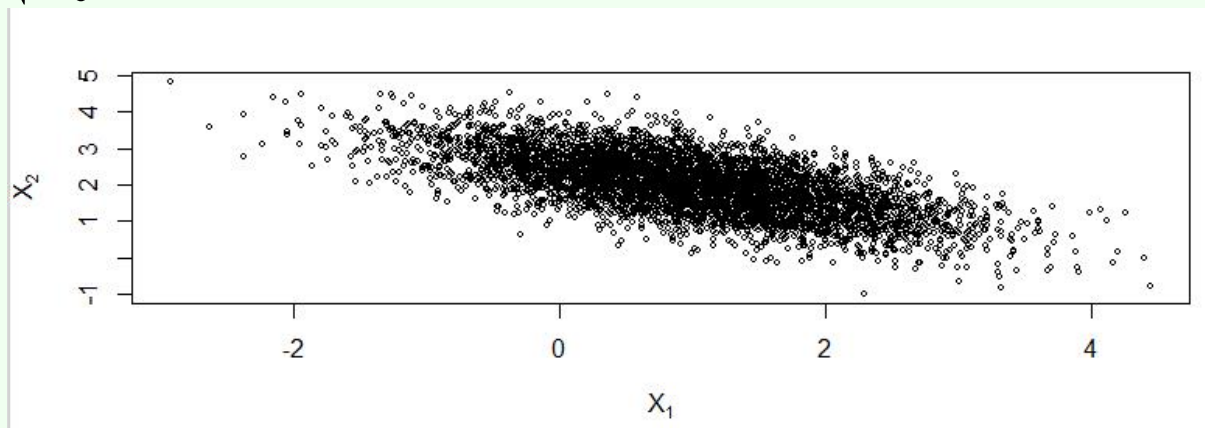


图11-2 Gibbs抽样得到的二元分布的散点图

§11.5 贝叶斯MCMC估计方法

从前面几节我们看到的MCMC方法最早由物理学家Metropolis 1953年提出，为解决物理中的高维复杂积分计算问题，该方法是通过一定策略产生平稳的Markov链，该Markov链的平稳分布是我们所感兴趣的分布，这样就可以利用该Markov链来进行参数推断，MCMC方法被称为20世纪的十大算法之一，给工程与实践科学带来了巨大影响并推动其向前发展，但是具有讽刺意义的是，强大的MCMC这个有力工具，直至上个世纪末才对统计学领域产生实质的影响，不过目前MCMC在统计应用、金融和贝叶斯计量经济学等领域已带来了广泛的应用。本节我们主要讨论MCMC方法在贝叶斯估计中的应用。



37/52

⏮ ⏭

⏪ ⏩

GoBack

FullScreen

Close

Quit



§11.5.1 贝叶斯MCMC估计

应用贝叶斯方法进行分析问题的时候，一个最主要的局限和困难就是得到其后验分布需要进行高维积分函数计算，通常高维积分是非常困难又消耗计算时间，目前已有一些巧妙的方法来避免直接进行高维积分计算，我们这里只讨论其中的MCMC方法。

简单来说，假设 $p(\mathbf{y}|\theta)$ 为抽样概率密度函数，其中 $\mathbf{y}$ 是一组观测向量， $\theta = (\theta_1, \dots, \theta_d)$ 是待估计的未知参数。我们这里采用贝叶斯方法进行参数估计，设未知参数的先验概率密度函数为 $\pi(\theta)$ ，则后验概率密度为

$$\pi(\theta|y) = \frac{p(\mathbf{y}|\theta)\pi(\theta)}{\int_{\Theta} p(\mathbf{y}|\theta)\pi(\theta)d\theta} p(\mathbf{y}|\theta)\pi(\theta) \quad (11.5.1)$$



GoBack

FullScreen

Close

Quit



但是在实际问题中，上述后验密度（11.5.1）通常是比较复杂未知的形式。所以，利用直接的分析方法或者数值积分的方法甚至是传统的蒙特卡洛方法来对后验分布进行分析，一般来说是不可行的。然而，这些困难可以使用MCMC方法来解决。

由马氏链的性质可知，在给定当前值和过去的链值下， $\theta^{(g+1)}$ 的分布只依赖 $\theta^{(g)}$ ，这样它可以用一步转移核来表示

$$K(\theta^{(g)}, \mathbb{A} | y) = P(\theta^{(g+1)} \in \mathbb{A} | y, \theta^{(g)})$$

一旦选取 $K(\theta, \mathbb{A} | y)$ ，就可以抽取 $\theta^{(i+1)} \sim K(\theta^{(i)}, \cdot | y)$ ，所有按照递归方法可得到序列 $\{\theta^{(i+1)}\}$ 。因为构造的马氏链收敛到目标分布，则除去前面足够大的 n_0 次迭代之后的 G 个值 $\{\theta^{(n_0+1)}, \theta^{(n_0+2)}, \dots, \theta^{(n_0+G)}\}$ 可以认为是来自 $\pi(\theta | y)$ 的一个抽样。只要 G 足够大，则样本 $\{\theta^{(n_0+1)}, \theta^{(n_0+2)}, \dots, \theta^{(n_0+G)}\}$



GoBack

FullScreen

Close

Quit



可以作为后验密度函数的一个代理 (surrogate)。所以可以用

$$\hat{h}(\theta) = G^{-1} \sum_{g=1}^G h(\theta^{(n_0+g)}) \quad (11.5.2)$$

来估计 $h(\theta)$ 的期望，用

$$\hat{\theta}_l = G^{-1} \sum_{g=1}^G \theta_l^{(n_0+g)} \quad (11.5.3)$$

来估计 θ 的第 l 个分量。



Metropolis-Hastings算法

M-H算法的主要思想是从一个友好而又简便的转移密度函数 $q(\theta^\dagger, \theta | \mathbf{y})$ 直接模拟产生 θ ，其中 $\theta^\dagger, \theta$ 是任意的两个点，然后修正转移密度函数，确保修正后的Markov链有着正确的极限分布（目标分布），这里我们称 $q(\theta^\dagger, \theta | \mathbf{y})$ 为参考生成密度或建议密度函数。

下面给出具体的M-H算法，令 $\theta^{(g)}$ 是当前值，下一期值 $\theta^{(g+1)}$ 由建议步和移动步这两步产生：

- Step1: 初始化 $\theta^{(0)}$;
- Step2: 在每一次迭代 $g(g = 0, 1, \dots, n_0 + G)$

建议步：从建议密度函数 $q(\theta^{(g)}, \theta | \mathbf{y})$ 产生建议值 $\theta^\dagger$ 并计算

$$\alpha(\theta^{(g)}, \theta^\dagger | \mathbf{y}) = \min \left\{ 1, \frac{\pi(\theta^\dagger | \mathbf{y}) q(\theta^\dagger, \theta^{(g)} | \mathbf{y})}{\pi(\theta^{(g)} | \mathbf{y}) q(\theta^{(g)}, \theta^\dagger | \mathbf{y})} \right\} \quad (11.5.4)$$



移动步：令

$$\theta^{(g+1)} = \begin{cases} \theta^\dagger, & \text{以概率 } \alpha(\theta^{(g)}, \theta^\dagger | y) \\ \theta^{(g)}, & \text{以概率 } 1 - \alpha(\theta^{(g)}, \theta^\dagger | y) \end{cases}$$

- **Step3:** 扔掉前面 n_0 次预选代数据并存储随后的 G 次抽样 $\{\theta^{(n_0+1)}, \theta^{(n_0+2)}, \dots, \theta^{(n_0+G)}\}$

在实际应用中，移密度函数 $\theta^\dagger, \theta | y$ 的选取是关键，下面是几种常见的选取方法。

1) Random walk M-H 建议步：给定当前值 θ ，建议值依 $\theta^\dagger = \theta + \varepsilon$ 抽取，其中 ε 取对称分布，比如 $q \sim N(0, \tau V)$ ，因此

$$\alpha(\theta, \theta^\dagger | y) = \min \left\{ 1, \frac{\pi(\theta^\dagger | y)}{\pi(\theta | y)} \right\}$$



2) Independence M-H 建议步: 令 $q(\theta, \theta^\dagger, \theta | \mathbf{y}) = q(\theta^\dagger, \theta | \mathbf{y})$, 这时

$$\alpha(\theta, \theta^\dagger | y) = \min \left\{ 1, \frac{\pi(\theta^\dagger | y) q(\theta | y)}{\pi(\theta | y) q(\theta^\dagger | y)} \right\}$$

当 θ 的维数比较大的时候, 在一个分块里面抽样的效率不理想, 这时有必要将 θ 分割成一些小的区块轮次抽取。为了解释这个思想, 我们假设 θ 被分割成两块: (θ_1, θ_2) (在很多问题中要根据问题本身来分割), 令 $q_1(\theta_1, \theta_1^\dagger | y, \theta_2)$, $q_2(\theta_2, \theta_2^\dagger | y, \theta_1)$ 表示建议密度。同样定义

$$\alpha_1(\theta_1, \theta_1^\dagger | y, \theta_2) = \min \left\{ 1, \frac{\pi(\theta_1^\dagger | y, \theta_2) q_1(\theta_1^\dagger, \theta_1 | y, \theta_2)}{\pi(\theta_1 | y, \theta_2) q_1(\theta_1, \theta_1^\dagger | y, \theta_2)} \right\} \quad (11.5.5)$$



$$\alpha_2(\theta_2, \theta_2^\dagger | y, \theta_1) = \min \left\{ 1, \frac{\pi(\theta_2^\dagger | y, \theta_1) q_2(\theta_2^\dagger, \theta_2 | y, \theta_1)}{\pi(\theta_2 | y, \theta_1) q_2(\theta_2, \theta_2^\dagger | y, \theta_1)} \right\} \quad (11.5.6)$$

条件后验密度 $\pi(\theta_1 | y, \theta_2)$, $\pi(\theta_2 | y, \theta_1)$ 被称为满条件分布, 由 **Bayes** 定理可知其与联合后验分布成比例, 比如 $\pi(\theta_1 | y, \theta_2) \propto \pi(\theta_1, \theta_2 | y)$, 因此在(11.5.5)和(11.5.6)中的转移概率可以用联合后验密度的核 $\pi(\theta_1, \theta_2 | y)$ 表示, 因为满条件分布的正则化常数被约去。



3) 多分块 M-H 算法:

Step1: 生成 $\theta_1^\dagger \sim q_1(\theta_1(g), \theta_1^\dagger | y, \theta_2^{(g)})$ 以概率 $\alpha_1(\theta_1^{(g)}, \theta_1^\dagger | y, \theta_2^{(g)})$ 更新 $\theta_1^{(g+1)}$.

Step2: 更新第一块的值, 生成 $\theta_2^\dagger \sim q_2(\theta_2(g+1), \theta_2^\dagger | y, \theta_1^{(g+1)})$ 以概率 $\alpha_2(\theta_2^{(g)}, \theta_2^\dagger | y, \theta_1^{(g+1)})$ 更新 $\theta_2^{(g+1)}$.

45/52



GoBack

FullScreen

Close

Quit



(2) Gibbs 抽样方法

在很多情形下每一个满条件分布都具有已知的形式并可以直接抽样，我们可以用它直接作为建议分布：

$$q_1(\theta_1(g), \theta_1^\dagger | y, \theta_2^{(g)}) = \pi(\theta_1^\dagger | y, \theta_2^{(g)})$$

$$q_2(\theta_2(g+1), \theta_2^\dagger | y, \theta_1^{(g+1)}) = \pi(\theta_2^\dagger | y, \theta_1^{(g+1)})$$

并且容易得到更新概率 $\alpha_1(\theta_1(g), \theta_1^\dagger | y, \theta_2^{(g)}) = 1$ ，这种特殊的分块M-H算法称之为Gibbs抽样算法。

Gibbs抽样算法：每一次迭代 $g(g = 1, 2, \dots, n_0 + G)$ ：

- 从 $\pi(\theta_1 | y, \theta_2^{(g)})$ 生成 $\theta_1^{(g+1)}$ ；
- 从 $\pi(\theta_2 | y, \theta_1^{(g+1)})$ 生成 $\theta_2^{(g+1)}$

返回 $\{\theta^{(n_0+1)}, \theta^{(n_0+2)}, \dots, \theta^{(n_0+G)}\}$ ，计算估计量

$$\hat{h}_G = G^{-1} \sum_{g=1}^G h(\theta^{(n_0+g)}).$$



(3) 混和抽样

在一些问题中，会遇到一个或多个满条件分布不能直接抽样，这时候需要采用M-H和Gibbs混合起来的抽样方法，对于具有可识别形式的满条件分布直接抽样，对于不可识别的满条件分布采取M-H方法。

47/52



GoBack

FullScreen

Close

Quit



§11.5.2 MCMC算法的收敛性质

使用MCMC算法进行推断依赖于MCMC算法的收敛性质，下面的两个重要结果来自于 Tierney(1994).

定理 11.5.1 遍历性大数定律

假设马氏链 $\{\theta^{(g)}\}$ 是不可约的并具有不变分布 $\pi(\theta|y)$, 则 $\pi(\theta|y)$ 是唯一的不变分布; 如果马氏链是不可约、非周期的, 则对任给的可测集 $\mathbb{A}$,

$$\|P(\theta^{(g)} \in \mathbb{A} | y, \theta^{(0)}) - \int_{\mathbb{A}} \pi(\theta|y)\| \longrightarrow 0, \quad a.s., \quad \text{当 } G \rightarrow \infty$$

(11.5.7)

如果马氏链是遍历的, 则对所有满足

$$\int_{\Theta} |h(\theta)| \pi(\theta|y) d\theta < \infty$$

的函数 $h(\theta)$ 和初始分布, 都有



$$\hat{h}_G \longrightarrow \int_{\Theta} |h(\theta)| \pi(\theta|y) d\theta, \quad a.s., \text{ 当 } G \rightarrow \infty \quad (11.5.8)$$

这里估计量 $\hat{h}_G := G^{-1} \sum_{g=1}^G h(\theta^{(g)})$

定理11.5.1给我们提供了MCMC推断的基本方法，给予我们利用转移核的模拟样本路径来得到后验矩、后验概率以及其它量的相合估计。注意到定理的条件是比较弱的，所以具有非常广泛的应用。



定理 11.5.2 遍历性中心极限定理

假设马氏链 $\{\theta^{(g)}\}$ 是一致遍历并具有不变分布 $\pi(\theta|y)$, 则对所有满足 $\int_{\Theta} |h(\theta)|^2 \pi(\theta|y) d\theta < \infty$ 的函数 $h(\theta)$ 和初始分布, 都有

$$\sqrt{G}(\hat{h}_G - E_{\pi}h) \longrightarrow N(0, \sigma_h^2) \quad (11.5.9)$$

式中

$$E_{\pi}h = \int_{\Theta} h(\theta) \pi(\theta|y) d\theta < \infty$$

$$\sigma_h^2 = Var_{\pi}(h(\theta_1^{(1)})) + 2 \sum_{j=2}^{\infty} Cov_{\pi}(h(\theta_1^{(1)}), h(\theta_1^{(j)}))$$



定理11.5.2为我们提供了计算 $E_{\pi}h$ 的MCMC参数估计估计量 $\hat{h}_G$ 的估计精度的思路，我们需要计算 $Var(\hat{h}_G) = \frac{\sigma_h^2}{G}$ 。 $Var(\hat{h}_G)$ 的平方根称为Numerical standard error 或者 Monte Carlo standard error (MCSE)。它可以通过Ripley(1987)的批量均值 (batch means) 法得到：

令 $Z_g = h(\theta^{(g)})(g = 1, 2, \dots, G)$ ，把数据 $\{Z_1, Z_2, \dots, Z_G\}$ 划分成互不重叠的长度为 m 的 k 个区间，均值为：

$$B_i = m^{-1}(Z_{(i-1)m+1} + \dots + Z_{im}), i = 1, 2, \dots, k$$

式中， m 的选取需要保证批量均值的一阶自相关系数小于0.05。记 $\bar{B} := \frac{1}{k} \sum_{i=1}^k B_i = \hat{h}_G$ ， $\bar{B}$ 的样本方差 $Var(\bar{B}) = \frac{1}{k(k-1)} \sum_{i=1}^k (B_i - \bar{B})^2$ ，当 k 和 m 都随 G 增加时，它是 $\frac{\sigma_h^2}{G}$ 的相合估计。



一旦我们得到了估计量 $Var(\hat{h}_G)$ ，我们就可以计算一个重要的统计量：非有效因子(Inefficiency Factor, Ineff):

$$Ineff(\hat{h}_G) = \frac{Var(\hat{h}_G)}{s^2/G} = \frac{\sigma_h^2/G}{s^2/G}$$

式中， s^2 是 $\{Z_g\}$ 的样本方差。我们也可以用：有效样本数(effective sample size, ESS)来解释Ineff的意义：

$$ESS(\hat{h}_G) = \frac{G}{Ineff(\hat{h}_G)}$$

关于Ineff的另外一个等价表示为：

$$Ineff(\hat{h}_G) = \sum_{g=-\infty}^{\infty} \rho_g = 1 + 2 \sum_{g=1}^{\infty} \rho_g \quad (11.5.10)$$

其中， ρ_g 是马氏链 $h(\theta)$ 的 g 阶自相关函数。