

Toy Notes of MCMC

Renhe W.

Contents

1	MH 算法	2
1.1	MH 算法基本步骤	2
1.2	为什么这样能够工作?	2
1.3	Note	3
1.3.1	为什么设置的是 $A(x_t, x') = \min\left(1, \frac{p(x')q(x_t x')}{p(x_t)q(x' x_t)}\right)$	3
1.3.2	为什么引入均匀分布?	3
2	Gibbs sampling	4
3	Example	5
3.1	AR(1) 模型示例	5
A	R code	6

List of Figures

List of Tables

1 MH 算法

1.1 MH 算法基本步骤

1. 初始化: 选择一个初始状态 x_0 ,
2. 对于每一步 $t = 1, 2, \dots, T$
 - 建议步骤: 从建议分布 $q(x' | x_t)$ 中抽取一个候选状态 x'
 - 接受步骤: 以以下的接受概率 $A(x_t, x')$ 接受候选状态:

$$A(x_t, x') = \min \left(1, \frac{p(x') q(x_t | x')}{p(x_t) q(x' | x_t)} \right)$$

如果接受, 则 $x_{t+1} = x'$, 否则 $x_{t+1} = x_t$.

这里, $p(x)$ 是我们想要采样的目标分布, $q(x' | x_t)$ 是给定当前状态 x_t 时, 提议一个新状态 x' 的建议分布.

1.2 为什么这样能够工作?

关键在于保证马尔可夫链的平稳分布 (stationary distribution) 是我们要抽样的目标分布。平稳分布 $\pi(x)$ 是一个分布, 对其而言, 如果从该分布中抽取一个样本并应用转移核, 新的样本仍然服从 $\pi(x)$ 。数学上我们可以表达为:

$$\pi(x') = \sum_x \pi(x) P(x' | x)$$

这里, $P(x' | x)$ 是从状态 x 到状态 x' 的转移概率。MH 算法通过精心设计的接受准则确保了其转移核满足**细致平稳条件**, 也就是说, 对于任意的状态 x 和 x' , 以下等式成立:

$$\pi(x) P(x' | x) = \pi(x') P(x | x')$$

其中 $P(x' | x)$ 是总的从状态 x 到状态 x' 的转移概率, 包括了提议和接受两个步骤, 可以写作:

$$P(x' | x) = q(x' | x) A(x, x')$$

通过 MH 算法的接受准则, 我们可以验证细致平稳条件确实成立:

$$\pi(x) q(x' | x) A(x, x') = \pi(x') q(x | x') A(x', x)$$

由于 $A(x, x') = \min\left(1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right)$ 和 $A(x', x) = \min\left(1, \frac{\pi(x)q(x'|x)}{\pi(x')q(x|x')}\right)$ ，我们可以看到这两边确实是相等的，从而确保了平稳分布 $\pi(x)$ 就是我们要采样的分布 $p(x)$ 。

1.3 Note

1.3.1 为什么设置的是 $A(x_t, x') = \min\left(1, \frac{p(x')q(x_t|x')}{p(x_t)q(x'|x_t)}\right)$

Metropolis-Hastings (MH) 算法中的接受概率

$$A(x_t, x') = \min\left(1, \frac{p(x')q(x_t|x')}{p(x_t)q(x'|x_t)}\right)$$

是一个精心设计的准则，旨在确保生成的样本 x 的分布最终收敛到目标分布 $p(x)$ 。这里的 x_t 是当前状态， x' 是建议的下一个状态， $q(x_t|x')$ 是从状态 x' 到状态 x_t 的转移概率， $p(x)$ 是我们想要采样的分布。

理解这个接受概率的一个简单方法是考虑比率

$$\frac{p(x')q(x_t|x')}{p(x_t)q(x'|x_t)}$$

分子 $p(x')q(x_t|x')$ ：这部分表示我们建议从状态 x' 移动到 x_t 并且 x' 本身的概率。实际上，这部分衡量了建议的状态 x' 到当前状态 x_t 的“前进”概率。

分母 $p(x_t)q(x'|x_t)$ ：这部分表示我们建议从状态 x_t 移动到状态 x' 并且 x_t 本身的概率。这部分衡量了当前状态 x_t 到建议状态 x' 的“前进”概率。

这个比率的直观意义是一个“平衡”：我们想要平衡从当前状态到建议状态的前进概率和反方向的前进概率。

- 当这个比率大于 1 时，我们总是接受建议的状态，因为这意味着建议的状态比当前状态更可能来自目标分布。
- 当这个比率小于 1 时，我们只有一定的概率接受建议的状态，这个概率正比于比率的大小。

1.3.2 为什么引入均匀分布？

均匀分布在 MH 算法中的角色体现在决定是否接受建议状态的步骤。具体来说，即使建议状态的接受概率 $A(x_t, x')$ 小于 1，我们仍然有可能接受它，这样做可以防止算法过早地陷入局部最优解并增加探索性。我们通常通过以下方式使用均匀分布：

1. 从均匀分布 $U(0, 1)$ 中抽取一个随机数 u 。

2. 如果 $u \leq A(x_t, x')$ ，接受建议状态，否则保持当前状态不变。

使用均匀分布的这一步增加了算法的随机性，并允许它有可能接受一个在目标分布下不太可能的状态，从而增加了算法的探索能力。

2 Gibbs sampling

Gibbs 抽样是一种特殊的 Metropolis-Hastings (MH) 算法，其中提议分布是条件分布，而接受概率始终为 1。这意味着 Gibbs 抽样总是接受新提议的样本。

Gibbs sampling

Gibbs 抽样是一种在高维分布上进行抽样的方法。对于一个 d 维分布 $p(x_1, x_2, \dots, x_d)$ ，Gibbs 抽样在每一步依次对每一个变量 x_i 进行抽样，条件于其他所有变量的当前值：

$$x_i^{(t+1)} \sim p\left(x_i \mid x_1^{(t+1)}, x_2^{(t+1)}, \dots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \dots, x_d^{(t)}\right),$$

这里， (t) 是迭代的步数。

Gibbs 抽样作为 MH 算法的特例：

为了理解 Gibbs 抽样是如何成为 MH 算法的一个特例的，我们需要考虑 MH 算法的两个主要步骤：建议 (proposal) 和接受 (acceptance)。

- 在 MH 算法中：
 - 建议步骤：从建议分布 $q(x' \mid x)$ 中抽样一个候选样本。
 - 接受步骤：以一定的接受概率 $A(x, x')$ 接受这个样本。
- 在 Gibbs 抽样中：
 - 建议步骤：从完全条件分布中抽样。
 - 接受步骤：总是接受从条件分布中抽样出的样本。

如果我们将 Gibbs 抽样的建议步骤视为 MH 算法中的一个特殊情况，其中建议分布是条件分布，那么 Gibbs 抽样的接受概率可以计算为：

$$A(x, x') = \min\left(1, \frac{p(x')q(x \mid x')}{p(x)q(x' \mid x)}\right)$$

由于在 Gibbs 抽样中 $q(x' | x) = p(x'_i | x_{-i})$ ，其中 x_{-i} 表示除了 x_i 之外的所有变量，我们可以得出：

$$A(x, x') = \min \left(1, \frac{p(x'_i | x_{-i}) p(x_{-i})}{p(x_i | x_{-i}) p(x_i)} \right)$$

由于 $x'_{-i} = x_{-i}$ ，我们可以简化这个比例为：

$$A(x, x') = \min \left(1, \frac{p(x'_i | x_{-i})}{p(x_i | x_{-i})} \right)$$

但是由于 x'_i 是直接来自分布 $p(x'_i | x_{-i})$ 中抽样出来的，我们知道这个比例总是等于 1，因此接受概率总是 1。

这就解释了为什么说 Gibbs 抽样是 MH 算法的一个特例：Gibbs 抽样总是接受新抽样出的样本，因此它可以被看作是 MH 算法中建议分布总是等于条件分布，接受概率总是 1 的一个特殊情况。

3 Example

3.1 AR(1) 模型示例

考虑一个 AR(1) 模型：

$$y_t = \phi y_{t-1} + \epsilon_t$$

其中 $\epsilon_t \sim N(0, \sigma^2)$ 。

我们的目标：给定观测值 y_1, y_2, \dots, y_n ，使用 MCMC 估计 ϕ 。

步骤

1. 基于 AR(1) 模型定义似然函数 $p(y|\phi)$ 。
2. 为 ϕ 定义一个先验 $p(\phi)$ 。
3. 结合似然和先验得到后验 $p(\phi|y)$ 。
4. 使用 Metropolis-Hastings 算法（一种 MCMC 形式）从后验中抽取样本。

A R code

```
1  # Simulate some data
2  set.seed(123)
3  n <- 100
4  phi_true <- 0.5
5  sigma_true <- 1
6  y <- arima.sim(n=n, list(ar=phi_true, order=c(1,0,0), sd=
sigma_true))
7  plot(y,type='l')
8  # Initialize parameters for Gibbs sampler
9  niter <- 10000
10 phi_samples <- numeric(niter)
11 sigma2_samples <- numeric(niter)
12 phi_samples[1] <- 0 # starting value for phi
13 sigma2_samples[1] <- 1 # starting value for sigma^2
14
15 s2 <- 10 # variance for phi prior
16 alpha <- 0.01
17 beta <- 0.01
18
19 for (i in 2:niter) {
20   # Update phi using Metropolis-Hastings
21   phi_candidate <- rnorm(1, phi_samples[i-1], 0.1)
22   acceptance_ratio <- exp((sum(dnorm(y[-1], mean=phi_
candidate*y[-n], sd=sqrt(sigma2_samples[i-1]), log=TRUE)) +
dnorm(phi_candidate, 0, sqrt(s2), log=TRUE)) - (sum(dnorm(y
[-1], mean=phi_samples[i-1]*y[-n], sd=sqrt(sigma2_samples[i
-1]), log=TRUE)) + dnorm(phi_samples[i-1], 0, sqrt(s2), log=
TRUE)))
23   if (runif(1) < acceptance_ratio) {
24     phi_samples[i] <- phi_candidate
25   } else {
26     phi_samples[i] <- phi_samples[i-1]
27   }
```

```
28
29     # Update sigma^2 using inverse gamma distribution
30     residuals <- y[-1] - phi_samples[i] * y[-n]
31     alpha_star <- alpha + n/2
32     beta_star <- beta + 0.5 * sum(residuals^2)
33     sigma2_samples[i] <- 1 / rgamma(1, alpha_star, beta_star)
34 }
35
36 mean(phi_samples[2000:niter])
37 mean(sigma2_samples[2000:niter])
38
39 # Plot results
40 plot(phi_samples, type="l")
41 plot(sigma2_samples, type="l")
42
```

Listing 1: A MCMC example: R code