

# 马氏链蒙特卡洛方法

黄政宇

北京大学北京国际数学研究中心  
北京大学国际机器学习研究中心



# 贝叶斯采样、推理

➤ 有未知归一化常数的目标分布

$$\rho^*(\theta) = \frac{1}{Z} e^{-\Phi_R(\theta)}$$

未知  $\nearrow$   $\longleftarrow$  已知

$$\Phi_R(\theta, y) = \frac{1}{2} \|\Sigma_\eta^{-\frac{1}{2}} (y - \mathcal{G}(\theta))\|^2 + \frac{1}{2} \|\Sigma_0^{-\frac{1}{2}} (\theta - r_0)\|^2$$

- 计算目标分布的期望、协方差等
- 计算目标函数的期望  $\mathbb{E}[f] = \int f(\theta) \rho^*(\theta) d\theta$
- 生成服从目标分布的样本  $\{\theta_j\} \sim \rho^*(\theta)$



# 马氏链蒙特卡洛方法

## ➤ 马氏链蒙特卡洛(Markov Chain Monte Carlo)方法

暴力网格搜索：假设  $N_\theta = 2$ ,

$$\theta_{i,j} = \left[-L + \frac{2(i-1)}{N-1}L, -L + \frac{2(j-1)}{N-1}L\right] \quad Z = \sum \rho(\theta_{i,j})$$

随机游走： $\theta_n \rightarrow \theta_{n+1}$  概率分布  $\psi_I(\theta_n, \theta_{n+1})$

不变分布是目标分布

$$\rho^*(\theta) = \int \psi_I(\theta', \theta) \rho^*(\theta') d\theta'$$

遍历性质

$$\{\theta_n\}_{n \geq N} \sim \rho^*(\theta)$$



# 马氏链蒙特卡洛方法

## ➤ 马氏链

马氏核： $p(\cdot, \cdot) : R^{N_\theta \times N_\theta} \rightarrow R$

$$- (\cdot, \cdot) \geq 0$$

$$- \int p(\theta, \theta') d\theta' = 1$$

不变分布  $\rho^*$ ：

$$\int \rho^*(\theta') p(\theta', \theta) d\theta' = \rho^*(\theta)$$

马氏链： $\{\theta_n\} \quad n \in Z^+$

$$\theta_0 \sim \rho_0(\theta)$$

$$\theta_{n+1} \sim p(\theta_n, \theta)$$

当  $\rho_0 = \rho^*$ , 我们有  $\theta_n \sim \rho^*$



# 马氏链蒙特卡洛方法

## 马氏链采样

给定  $f: R^{N\theta} \rightarrow R$ ,  $Var_{\rho} [f] = 1$ , 构造不变分布是  $\rho$  的马氏链  $\{\theta_n\}$ , 我们定义

$$\rho_{\text{MCMC}}^N(f) = \frac{1}{N} \sum_{n=1}^N f(\theta_n)$$

我们有

$$\mathbb{E}^{\theta_0 \sim \rho} \left[ \rho_{\text{MCMC}}^N(f) - \mathbb{E}_{\rho}[f] \right] = 0$$

$$\mathbb{E}^{\theta_0 \sim \rho} \left[ \left( \rho_{\text{MCMC}}^N(f) - \mathbb{E}_{\rho}[f] \right)^2 \right] = \frac{\tau_N^2(f)}{N}$$

其中

$$\tau_N^2(f) = 1 + 2 \sum_{n=1}^N \frac{N-n}{N} \text{Cov} [f(\theta^0), f(\theta^n)]$$



# Metropolis-Hastings 算法

## ➤ 马氏链

马氏核：

$$\int p(\theta, \theta') d\theta' = 1$$

不变分布  $\rho^*$ ：

$$\int \rho^*(\theta') p(\theta', \theta) d\theta' = \rho^*(\theta)$$

$$\rho^*(\theta) = \int \rho^*(\theta) p(\theta, \theta') d\theta'$$

## ➤ 细致平衡

$$\rho^*(\theta') p(\theta', \theta) = \rho^*(\theta) p(\theta, \theta')$$



# Metropolis-Hastings 算法

➤ 细致平衡

$$\rho^*(\theta')p(\theta', \theta) = \rho^*(\theta)p(\theta, \theta')$$

$$\frac{\rho^*(\theta')}{\rho^*(\theta)} = \frac{p(\theta, \theta')}{p(\theta', \theta)}$$

➤ Metropolis-Hastings 算法

提议核(proposal) :

$$q(\cdot, \cdot) : R^{N_\theta \times N_\theta} \rightarrow R^+ \quad \int q(\theta, \theta') d\theta' = 1$$

接收概率(acceptance-rejection) :

不需要归一化常数

$$a(\theta, \theta') = \min \left\{ \frac{\rho^*(\theta')q(\theta', \theta)}{\rho^*(\theta)q(\theta, \theta')}, 1 \right\}$$



# Metropolis-Hastings 算法

## ➤ Metropolis-Hastings 算法

提议核： $q(\cdot, \cdot) : R^{N_\theta \times N_\theta} \rightarrow R^+$

修正： $a(\theta, \theta') = \min \left\{ \frac{\rho^*(\theta')q(\theta', \theta)}{\rho^*(\theta)q(\theta, \theta')}, 1 \right\}$

给定  $\theta_n$

- 采样  $\theta' \sim q(\theta_n, \theta')$

- 计算  $a(\theta_n, \theta')$

$\theta_{n+1} = \begin{cases} \theta' & \text{以 } a(\theta_n, \theta') \text{ 的概率} \\ \theta_n & \text{以 } 1 - a(\theta_n, \theta') \text{ 的概率} \end{cases}$

$$p_{\text{MH}}(\theta, \theta') = a(\theta, \theta')q(\theta, \theta')$$

$$p_{\text{MH}}(\theta, \theta) = q(\theta, \theta) + \int (1 - a(\theta, \theta'))q(\theta, \theta')d\theta \delta(\theta)$$





# Metropolis-Hastings 算法

## ➤ Metropolis-Hastings 算法

提议核： $q(\cdot, \cdot) : R^{N_\theta \times N_\theta} \rightarrow R^+$

修正： $a(\theta, \theta') = \min \left\{ \frac{\rho^*(\theta')q(\theta', \theta)}{\rho^*(\theta)q(\theta, \theta')} , 1 \right\}$

## ➤ 随机游走的Metropolis-Hastings 算法

$$q(\theta, \theta') = \mathcal{N}(\theta'; \theta, \delta^2 I)$$

$$q(\theta, \theta') = q(\theta', \theta) \quad a(\theta, \theta') = \min \left\{ \frac{\rho^*(\theta')}{\rho^*(\theta)} , 1 \right\}$$

希望  $a(\theta, \theta')$  尽量大，以更大概率接收、移动



# Metropolis-Hastings 算法

## ➤ Metropolis-Hastings 算法

提议核： $q(\cdot, \cdot) : R^{N_\theta \times N_\theta} \rightarrow R^+$

修正： $a(\theta, \theta') = \min \left\{ \frac{\rho^*(\theta')q(\theta', \theta)}{\rho^*(\theta)q(\theta, \theta')} , 1 \right\}$

## ➤ 重要性采样的Metropolis-Hastings 算法

$q(\theta, \theta') = f(\theta')$

$a(\theta, \theta') = \min \left\{ \frac{\rho^*(\theta')f(\theta)}{\rho^*(\theta)f(\theta')} , 1 \right\}$

目标  $f(\theta') \sim \rho^*(\theta')$ ，比如： $f = \rho_{\text{prior}}$



# Metropolis-Hastings 算法

## ➤ Metropolis-Hastings 算法

提议核： $q(\cdot, \cdot) : R^{N_\theta \times N_\theta} \rightarrow R^+$

修正： $a(\theta, \theta') = \min \left\{ \frac{\rho^*(\theta')q(\theta', \theta)}{\rho^*(\theta)q(\theta, \theta')}, 1 \right\}$

## ➤ Metropolis-Adjusted Langevin 算法

$\rho^*(\theta) \propto e^{-\Phi_R(\theta)}$

梯度下降方法： $\theta \rightarrow \theta - \epsilon \nabla_\theta \Phi_R(\theta)$

$\Phi_R(\theta - \epsilon \nabla_\theta \Phi_R(\theta)) < \Phi_R(\theta) \quad \rho^*(\theta - \epsilon \nabla_\theta \Phi_R(\theta)) > \rho^*(\theta)$

$q(\theta, \theta') = \mathcal{N}(\theta'; \theta - \epsilon \nabla_\theta \Phi_R(\theta), \delta^2 I)$

$a(\theta, \theta') = \min \left\{ \frac{\rho^*(\theta') \mathcal{N}(\theta; \theta' - \epsilon \nabla_{\theta'} \Phi_R(\theta'), \delta^2 I)}{\rho^*(\theta) \mathcal{N}(\theta'; \theta - \epsilon \nabla_\theta \Phi_R(\theta), \delta^2 I)}, 1 \right\}$



# 马氏链蒙特卡洛方法

## ➤ 有限状态空间

考虑有限状态空间  $S = \{1, 2, \dots, d\}$  上的马氏链  
马氏核对应了转移概率矩阵  $P \in R^{d \times d}$

$$P(i, j) \geq 0$$

$$\sum_j P(i, j) = 1$$

不变分布对应向量  $\rho^* \in R^d$ ，满足

$$\rho^{*T} P = \rho^{*T}$$

对于马氏链  $\theta_0 \sim \rho_0^T$ ，我们有

$$\theta_n \sim \rho_n^T = \rho_0^T P^n$$

$$\rho_{n+1}^T = \rho_n^T P$$



# 马氏链收敛性

## 有限状态马氏链的收敛性

对于有限状态空间  $S = \{1, 2, \dots, d\}$  上的马氏链，假设转移概率矩阵  $P \in R^{d \times d}$  满足

$$\min_{(i,j) \in S \times S} P(i,j) \geq \frac{\epsilon}{d}$$

那么存在唯一不变分布  $\rho \in R^d$ ，满足

$$\rho^T P = \rho^T$$

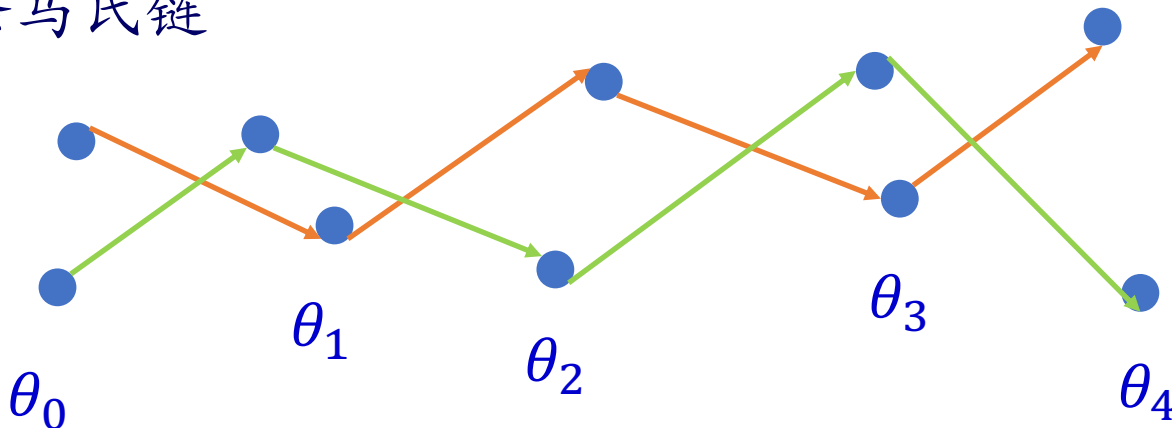
对于马氏链  $\{\theta_n\}$ ， $\theta_n \sim \rho_n$ ，我们有

$$\mathcal{D}_{TV}(\rho_n, \rho) \leq (1 - \epsilon)^n$$



# 马氏链蒙特卡洛方法

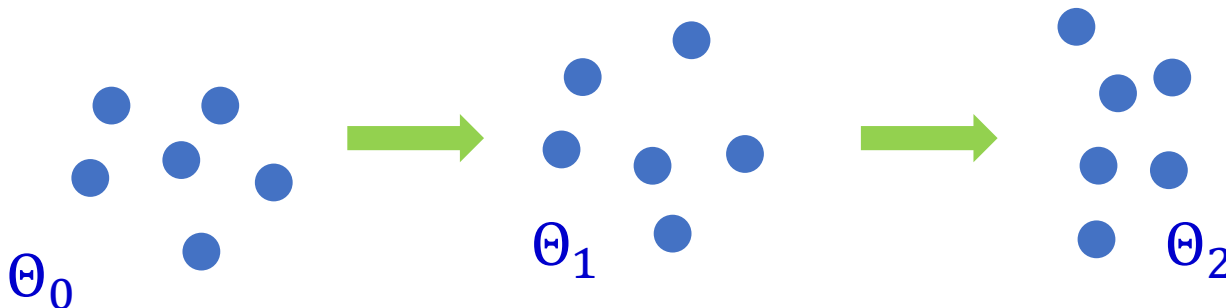
## ➤ 两条马氏链



## ➤ 交互粒子系统

$$\Theta_n = [\theta_n^1; \theta_n^2; \dots; \theta_n^J] \in R^{N_\theta} \otimes R^{N_\theta} \dots \otimes R^{N_\theta}$$

$$P^* = \rho^* \otimes \rho^* \dots \otimes \rho^*$$





# 马氏链蒙特卡洛方法

## ▶ 仿射不变马尔可夫链蒙特卡洛集合采样器

<https://emcee.readthedocs.io/en/stable/index.html>

### emcee

**emcee** is an MIT licensed pure-Python implementation of Goodman & Weare's [Affine Invariant Markov chain Monte Carlo \(MCMC\) Ensemble sampler](#) and these pages will show you how to use it.

This documentation won't teach you too much about MCMC but there are a lot of resources available for that (try [this one](#)). We also [published a paper](#) explaining the emcee algorithm and implementation in detail.

emcee has been used in quite a few projects in the astrophysical literature and it is being actively developed on [GitHub](#).

GitHub [dfm/emcee](#) Tests [passing](#) license [MIT](#) arXiv [1202.3665](#) coverage [96%](#)

### Basic Usage

If you wanted to draw samples from a 5 dimensional Gaussian, you would do something like:

```
import numpy as np
import emcee

def log_prob(x, ivar):
    return -0.5 * np.sum(ivar * x ** 2)

ndim, nwalkers = 5, 100
ivar = 1. / np.random.rand(ndim)
p0 = np.random.randn(nwalkers, ndim)

sampler = emcee.EnsembleSampler(nwalkers, ndim, log_prob, args=[ivar])
sampler.run_mcmc(p0, 10000)
```