

# 第19章 马尔可夫链蒙特卡罗法

1. 蒙特卡罗法是通过基于概率模型的抽样进行数值近似计算的方法，蒙特卡罗法可以用于概率分布的抽样、概率分布数学期望的估计、定积分的近似计算。

随机抽样是蒙特卡罗法的一种应用，有直接抽样法、接受拒绝抽样法等。接受拒绝法的基本想法是，找一个容易抽样的建议分布，其密度函数的数倍大于等于想要抽样的概率分布的密度函数。按照建议分布随机抽样得到样本，再按要抽样的概率分布与建议分布的倍数的比例随机决定接受或拒绝该样本，循环执行以上过程。

马尔可夫链蒙特卡罗法数学期望估计是蒙特卡罗法的另一种应用，按照概率分布 $p(x)$ 抽取随机变量 $x$ 的 $n$ 个独立样本，根据大数定律可知，当样本容量增大时，函数的样本均值以概率1收敛于函数的数学期望

$$\hat{f}_n \rightarrow E_{p(x)}[f(x)], \quad n \rightarrow \infty$$

计算样本均值 $\hat{f}_n$ ，作为数学期望 $E_{p(x)}[f(x)]$ 的估计值。

2. 马尔可夫链是具有马尔可夫性的随机过程 $P(X_t|X_0X_1\cdots X_{t-1}) = P(X_t|X_{t-1})$ ， $t = 1, 2, \cdots$ 通常考虑时间齐次马尔可夫链。有离散状态马尔可夫链和连续状态马尔可夫链，分别由概率转移矩阵 $P$ 和概率转移核 $p(x, y)$ 定义。

满足 $\pi = P\pi$ 或 $\pi(y) = \int p(x, y)\pi(x)dx$ 的状态分布称为马尔可夫链的平稳分布。

马尔可夫链有不可约性、非周期性、正常返等性质。一个马尔可夫链若是不可约、非周期、正常返的，则该马尔可夫链满足遍历定理。当时间趋于无穷时，马尔可夫链的状态分布趋近于平稳分布，函数的样本平均依概率收敛于该函数的数学期望。

$\lim_{t \rightarrow \infty} P(X_t = i | X_0 = j) = \pi_i, \quad i = 1, 2, \cdots; \quad j = 1, 2, \cdots$   $\hat{f}_t \rightarrow E_\pi[f(X)], \quad t \rightarrow \infty$  可逆马尔可夫链是满足遍历定理的充分条件。

3. 马尔可夫链蒙特卡罗法是以马尔可夫链为概率模型的蒙特卡罗积分方法，其基本想法如下：

- (1) 在随机变量 $x$ 的状态空间 $\chi$ 上构造一个满足遍历定理条件的马尔可夫链，其平稳分布为目标分布 $p(x)$ ；

- (2) 由状态空间的某一点 $X_0$ 出发，用所构造的马尔可夫链进行随机游走，产生样本序列 $X_1, X_2, \cdots, X_t, \cdots$ ；

- (3) 应用马尔可夫链遍历定理，确定正整数 $m$ 和 $n(m < n)$ ，得到样本集合 $\{x_{m+1}, x_{m+2}, \cdots, x_n\}$ ，进行函数 $f(x)$ 的均值（遍历均值）估计： $\hat{E}f = \frac{1}{n-m} \sum_{i=m+1}^n f(x_i)$

4. Metropolis-Hastings算法是最基本的马尔可夫链蒙特卡罗法。假设目标是对概率分布 $p(x)$ 进行抽样，构造建议分布 $q(x, x')$ ，定义接受分布 $\alpha(x, x')$ 进行随机游走，假设当前处于状态 $x$ ，按照建议分布 $q(x, x')$ 机抽样，按照概率 $\alpha(x, x')$ 接受抽样，转移到状态 $x'$ ，按照概率 $1 - \alpha(x, x')$ 拒绝抽样，停留在状态 $x$ ，持续以上操作，得到一系列样本。这样的随机游走是根据转移核为 $p(x, x') = q(x, x')\alpha(x, x')$ 的可逆马尔可夫链（满足遍历定理条件）进行的，其平稳分布就是要抽样的目标分布 $p(x)$ 。

5. 吉布斯抽样（Gibbs sampling）用于多元联合分布的抽样和估计吉布斯抽样是单分量 Metropolis-Hastings-算法的特殊情况。这时建议分布为满条件概率分布 $q(x, x') = p(x'_j | x_{-j})$  吉布斯抽样的基本做法是，从联合分布定义满条件概率分布，依次从满条件概率分布进行抽样，得到联合分布的

随机样本。假设多元联合概率分布为 $p(x) = p(x_1, x_2, \dots, x_k)$ ，吉布斯抽样从一个初始样本 $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_k^{(0)})^T$ 出发，不断进行迭代，每一次迭代得到联合分布的一个样本 $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_k^{(i)})^T$ ，在第 $i$ 次迭代中，依次对第 $j$ 个变量按照满条件概率分布随机抽样 $p(x_j | x_1^{(i)}, \dots, x_{j-1}^{(i)}, x_{j+1}^{(i-1)}, \dots, x_k^{(i-1)})$ ,  $j = 1, 2, \dots, k$ ，得到 $x_j^{(i)}$ 最终得到样本序列 $\{x^{(0)}, x^{(1)}, \dots, x^{(n)}\}$ 。

---

蒙特卡洛法 (Monte Carlo method)，也称为统计模拟方法 (statistical simulation method)，是从概率模型的随机抽样进行近似数值计

算的方法。马尔可夫链蒙特卡罗法 (Markov Chain Monte Carlo, MCMC)，则是以马尔可夫链 (Markov chain) 为概率模型的蒙特卡洛法。

马尔可夫链蒙特卡罗法构建一个马尔可夫链，使其平稳分布就是要进行抽样的分布，首先基于该马尔可夫链进行随机游走，产生样本的序列，

之后使用该平稳分布的样本进行近似数值计算。

Metropolis-Hastings算法是最基本的马尔可夫链蒙特卡罗法，Metropolis等人在1953年提出原始的算法，Hastings在1970年对之加以推广，

形成了现在的形式。吉布斯抽样(Gibbs sampling)是更简单、使用更广泛的马尔可夫链蒙特卡罗法，1984年由S. Geman和D. Geman提出。

马尔可夫链蒙特卡罗法被应用于概率分布的估计、定积分的近似计算、最优化问题的近似求解等问题，特别是被应用于统计学习中概率模型的学习

与推理，是重要的统计学习计算方法。

一般的蒙特卡罗法有直接抽样法、接受-拒绝抽样法、重要性抽样法等。

接受-拒绝抽样法、重要性抽样法适合于概率密度函数复杂（如密度函数含有多个变量，各变量相互不独立，密度函数形式复杂），不能直接抽样的情况。

## 19.1.2 数学期望估计

一般的蒙特卡罗法，如直接抽样法、接受-拒绝抽样法、重要性抽样法，也可以用于数学期望估计 (estimation Of mathematical expectation)。

假设有随机变量 $x$ ，取值 $x \in X$ ，其概率密度函数为 $p(x)$ ， $f(x)$ 为定义在 $X$ 上的函数，目标是求函数 $f(x)$ 关于密度函数 $p(x)$ 的数学期望 $E_{p(x)}[f(x)]$ 。

针对这个问题，蒙特卡罗法按照概率分布 $p(x)$ 独立地抽取 $n$ 个样本 $x_1, x_2, \dots, x_n$ ，比如用以上的抽样方法，之后计算函

数 $f(x)$ 的样本均值 $\hat{f}_n$

$$\hat{f}_n = \frac{1}{n} \sum_{i=1}^n f(x_i)$$

作为数学期望 $E_{p(x)}[f(x)]$ 近似值。

根据大数定律可知，当样本容量增大时，样本均值以概率1收敛于数学期望：

$$\hat{f}_n \rightarrow E_{p(x)}[f(x)], n \rightarrow \infty$$

这样就得到了数学期望的近似计算方法：

$$E_{p(x)}[f(x)] \approx \frac{1}{n} \sum_{i=1}^n f(x_i)$$

## 马尔可夫链

考虑一个随机变量的序列  $X = X_0, X_1, \dots, X(t), \dots$  这里  $X_t$ ，表示时刻  $t$  的随机变量， $t = 0, 1, 2, \dots$

每个随机变量  $X_t (t = 0, 1, 2, \dots)$  的取值集合相同，称为状态空间，表示为  $S$ 。随机变量可以是离散的，也可以是连续的。

以上随机变量的序列构成随机过程（stochastic process）。

假设在时刻 0 的随机变量  $X_0$  遵循概率分布  $P(X_0) = \pi$ ，称为初始状态分布。在某个时刻  $t \geq 1$  的随机变量  $X_t$  与前

一个时刻的随机变量  $X_{t-1}$  之间有条件分布  $P(X_t | X_{t-1})$  如果  $X_t$  只依赖于  $X_{t-1}$ ，而不依赖于过去的随机变量

$X_0, X_1, \dots, X_{t-2}$  这一性质称为马尔可夫性，即

$$P(X_t | X_0, X_1, \dots, X_{t-1}) = P(X_t | X_{t-1}), t = 1, 2, \dots$$

具有马尔可夫性的随机序列  $X = X_0, X_1, \dots, X(t), \dots$  称为马尔可夫链，或马尔可夫过程（Markov process）。条件概率分布

$P(X_t | X_{t-1})$  称为马尔可夫链的转移概率分布。转移概率分布决定了马尔可夫链的特性。

### 平稳分布

设有马尔可夫链  $X = X_0, X_1, \dots, X(t), \dots$ ，其状态空间为  $S$ ，转移概率矩阵为  $P = (p_{ij})$ ，如果存在状态空间  $S$  上的一个分布

$$\pi = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \end{bmatrix}$$

使得

$$\pi = P\pi$$

则称  $\pi$  为马尔可夫链  $X = X_0, X_1, \dots, X(t), \dots$  的平稳分布。

直观上，如果马尔可夫链的平稳分布存在，那么以该平稳分布作为初始分布，面向未来进行随机状态转移，之后任何一个时刻的状态分布都是该平稳分布。

#### 引理19. 1

给定一个马尔可夫链  $X = X_0, X_1, \dots, X(t), \dots$ ，状态空间为  $S$ ，转移概率矩阵为  $P = (p_{ij})$ ，则分布  $\pi = (\pi_1, \pi_2, \dots)^T$  为  $X$  的平稳分布的充要条件是  $\pi = (\pi_1, \pi_2, \dots)^T$  是下列方程组的解：

$$x_i = \sum_j p_{ij} x_j, i = 1, 2, \dots$$

$$x_i \geq 0, i = 1, 2, \dots$$

$$\sum_i x_i = 1$$

## 吉布斯采样

输入：目标概率分布的密度函数 $p(x)$ , 函数 $f(x)$ ;

输出： $p(x)$ 的随机样本  $x_{m+1}, x_{m+2}, \dots, x_n$ , 函数样本均值  $f_{mn}$ ;

参数：收敛步数 $m$ , 迭代步数  $n$ .

1. 初始化。给出初始样本  $x^0 = (x_1^0, x_2^0, \dots, x_k^0)^T$ .
2. 对 $i$ 循环执行
 

设第 $i-1$ 次迭代结束前的样本为 $x^{i-1} = (x_1^{i-1}, x_2^{i-1}, \dots, x_k^{i-1})^T$ , 则第 $i$ 次迭代进行如下几步操作:

  - (1)由满条件分布  $p(x_1|x_2^{i-1}, \dots, x_k^{i-1})$  抽取  $x_1^i$
  - ...
  - (j)由满条件分布  $p(x_j|x_1^i, \dots, x_{j-1}^i, x_{j+1}^{i-1}, \dots, x_k^{i-1})$  抽取  $x_j^i$
  - (k)由满条件分布  $p(x_k|x_1^i, \dots, x_k^i)$  抽取  $x_k^i$

得到第  $i$  次迭代值  $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_k^{(i)})^T$ .

3. 得到样本集合

$$\{x^{(m+1)}, x^{(m+2)}, \dots, x^{(n)}\}$$

4. 计算

$$f_{mn} = \frac{1}{n-m} \sum_{i=m+1}^n f(x^{(i)})$$

---

## 网络资源:

LDA-math-MCMC 和 Gibbs Sampling: <https://cosx.org/2013/01/lda-math-mcmc-and-gibbs-sampling>

MCMC蒙特卡罗方法: <https://www.cnblogs.com/pinard/p/6625739.html>

```
import random
import math
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

transfer_matrix = np.array([[0.6, 0.2, 0.2], [0.3, 0.4, 0.3], [0, 0.3, 0.7]],
                             dtype='float32')
start_matrix = np.array([[0.5, 0.3, 0.2]], dtype='float32')

value1 = []
value2 = []
value3 = []
for i in range(30):
    start_matrix = np.dot(start_matrix, transfer_matrix)
```

```

value1.append(start_matrix[0][0])
value2.append(start_matrix[0][1])
value3.append(start_matrix[0][2])
print(start_matrix)

```

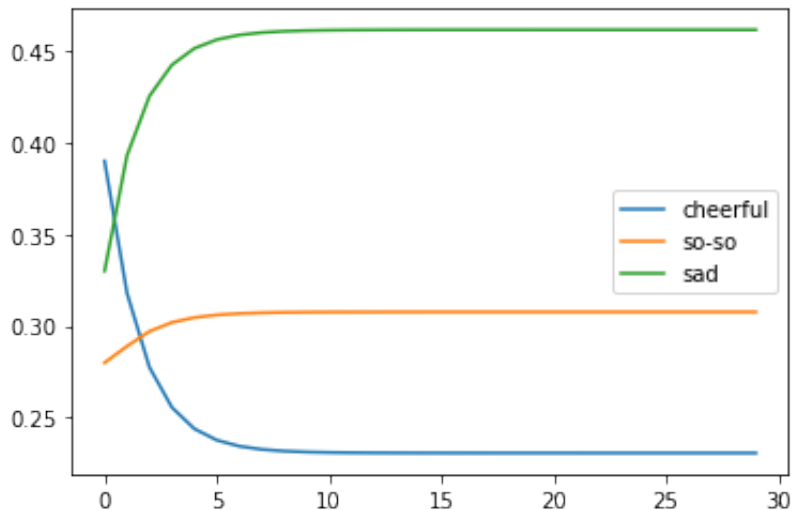
```
[[0.23076935 0.30769244 0.46153864]]
```

#进行可视化

```

x = np.arange(30)
plt.plot(x,value1,label='cheerful')
plt.plot(x,value2,label='so-so')
plt.plot(x,value3,label='sad')
plt.legend()
plt.show()

```



可以发现，从10轮左右开始，我们的状态概率分布就不变了，一直保持在 [0.23076934,0.30769244,0.4615386]

<https://zhuanlan.zhihu.com/p/37121528>

## M-H采样python实现

<https://zhuanlan.zhihu.com/p/37121528>

假设目标平稳分布是一个均值3，标准差2的正态分布，而选择的马尔可夫链状态转移矩阵  $Q(i, j)$  的条件转移概率是以  $i$  为均值,方差1的正态分布在位置  $j$  的值。

```

from scipy.stats import norm

def norm_dist_prob(theta):
    y = norm.pdf(theta, loc=3, scale=2)
    return y

```

```

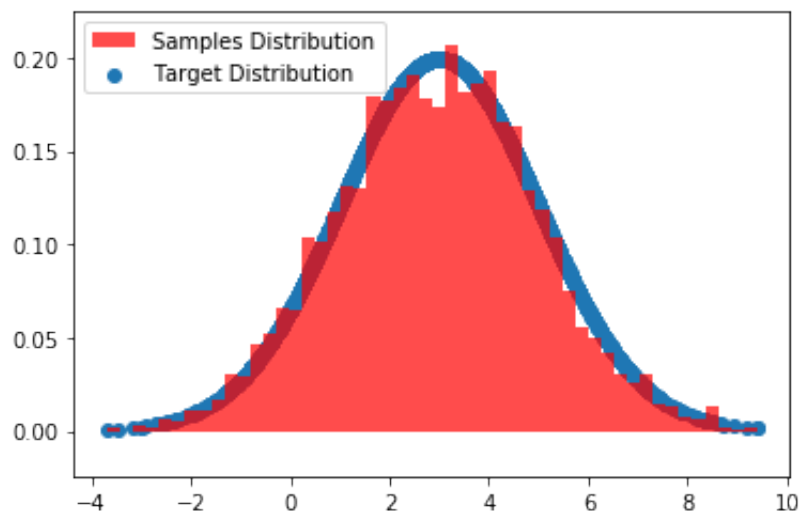
T = 5000
pi = [0 for i in range(T)]
sigma = 1
t = 0
while t < T - 1:
    t = t + 1
    pi_star = norm.rvs(loc=pi[t - 1], scale=sigma, size=1,
                      random_state=None) #状态转移进行随机抽样

    alpha = min(
        1, (norm_dist_prob(pi_star[0]) / norm_dist_prob(pi[t - 1]))) #alpha值

    u = random.uniform(0, 1)
    if u < alpha:
        pi[t] = pi_star[0]
    else:
        pi[t] = pi[t - 1]

plt.scatter(pi, norm.pdf(pi, loc=3, scale=2), label='Target Distribution')
num_bins = 50
plt.hist(pi,
         num_bins,
         density=1,
         facecolor='red',
         alpha=0.7,
         label='Samples Distribution')
plt.legend()
plt.show()

```



## 二维Gibbs采样实例python实现

假设我们要采样的是一个二维正态分布  $N(\mu, \Sigma)$ ，其中： $\mu = (\mu_1, \mu_2) = (5, -1)$ ， $\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 & \rho\sigma_2 \\ \rho\sigma_1 & \sigma_1^2 & \rho \\ \rho\sigma_2 & \rho & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 4 \end{pmatrix}$ ;

而采样过程中的需要的状态转移条件分布为：

$$P(x_1|x_2) = N(\mu_1 + \rho\sigma_1/\sigma_2(x_2 - \mu_2), (1 - \rho^2)\sigma_1^2)$$

$$P(x_2|x_1) = N(\mu_2 + \rho\sigma_2/\sigma_1(x_1 - \mu_1), (1 - \rho^2)\sigma_2^2)$$

```

from mpl_toolkits.mplot3d import Axes3D
from scipy.stats import multivariate_normal

samplesource = multivariate_normal(mean=[5,-1], cov=[[1,0.5],[0.5,2]])

def p_ygivenx(x, m1, m2, s1, s2):
    return (random.normalvariate(m2 + rho * s2 / s1 * (x - m1), math.sqrt(1 -
rho ** 2) * s2))

def p_xgiveny(y, m1, m2, s1, s2):
    return (random.normalvariate(m1 + rho * s1 / s2 * (y - m2), math.sqrt(1 -
rho ** 2) * s1))

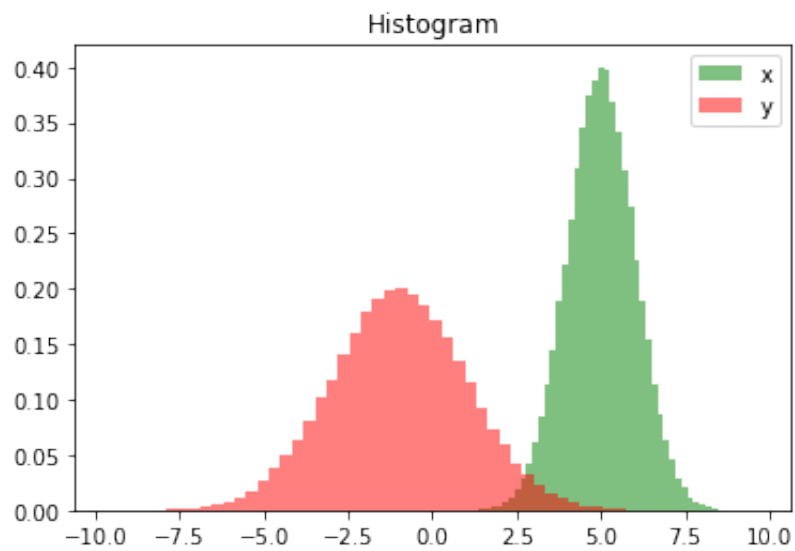
N = 5000
K = 20
x_res = []
y_res = []
z_res = []
m1 = 5
m2 = -1
s1 = 1
s2 = 2

rho = 0.5
y = m2

for i in range(N):
    for j in range(K):
        x = p_xgiveny(y, m1, m2, s1, s2)    #y给定得到x的采样
        y = p_ygivenx(x, m1, m2, s1, s2)    #x给定得到y的采样
        z = samplesource.pdf([x,y])
        x_res.append(x)
        y_res.append(y)
        z_res.append(z)

num_bins = 50
plt.hist(x_res, num_bins,density=1, facecolor='green', alpha=0.5,label='x')
plt.hist(y_res, num_bins, density=1, facecolor='red', alpha=0.5,label='y')
plt.title('Histogram')
plt.legend()
plt.show()

```



---

本章代码来源: <https://github.com/hktxt/Learn-Statistical-Learning-Method>

本文代码更新地址: <https://github.com/fengdu78/lihang-code>

中文注释制作: 机器学习初学者公众号: ID:ai-start-com

配置环境: python 3.5+

代码全部测试通过。