马尔科夫蒙特卡洛详解

> 本文由 [简悦 SimpRead](http://ksria.com/simpread/) 转码， 原文地址 [zhuanlan.zhihu.com](https://zhuanlan.zhihu.com/p/250146007)
>
> **参考：**
>
> [MCMC(三)MCMC 采样和 M-H 采样 - 刘建平 Pinard - 博客园]
>
> [MCMC(四)Gibbs采样 - 刘建平 Pinard - 博客园]

说明：MCMC 主要克服的是高维空间里采样效率过低的问题

> MCMC 算法的一般流程是：先给定目标分布完成采样过程，若目标分布是一维的，就用 M-H 采样方法；若目标分布是多维的，就用 Gibbs 采样方法。采样结束之后，蒙特卡罗方法来用样本集模拟求和，求出目标变量 (期望等) 的统计值作为估计值。这套思路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。

蒙特卡罗方法

1.1 蒙特卡罗是什么?

> 赌城！
> 蒙特卡洛是摩纳哥公国的一座城市，位于欧洲地中海之滨、法国的东南方，属于一个版图很小的国家摩纳哥公国，世人称之为 “赌博之国”、“袖珍之国”、“邮票小国”。
> 蒙特卡洛的赌业，海洋博物馆的奇观，格蕾丝王妃的下嫁，都为这个小国增添了许多传奇色彩，作为世界上人口最密集的一个国度，摩纳哥在仅有 1.95 平方千米的国土上聚集了 3.3 万的人口，可谓地窄人稠。但相对于法国，摩纳哥的地域实在是微乎其微，这个国中之国就像一小滴不慎滴在法国版图内的墨汁，小得不大会引起人去注意它的存在。

> 蒙特卡罗方法于 20 世纪 40 年代美国在第二次世界大战中研制原子弹的 “曼哈顿计划” 时首先提出，为保密选择用赌城摩纳哥的蒙特卡洛作为代号，因而得名。

看到这里，你可能似乎已经意识到，这个方法一定和**赌博**，**概率分布**，**近似数值计算**有着千丝万缕的联系。

事实的确如此，首先我想引用一段李航老师在《统计学习方法》中关于 MCMC 的介绍：

> 蒙特卡罗法 (Monte Carlo method)，也称为统计模拟方法 (statistical simulation method)，**是通过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 (Markovchain) 为**概率模型**的**蒙特卡罗法**。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的数值计算**。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。

相信读完这一段大多数人都依然懵逼，我们一个概念一个概念地介绍，对于任何一种方法，我们会先阐明它在生活中的用途，让读者有个总体的认识，再去推导它的数学原理，让它彻底地为你所用。

1.2 从蒙特卡洛方法说起

生活中的例子：

- **1. 蒙特卡罗估计圆周率 π 的值**

> 这个问题来自小学课本。

绘制一个单位长度的正方形，并绘制四分之一圆。在正方形上随机采样尽可能多的点，放上 3000 次，就可以得到如下这张图：

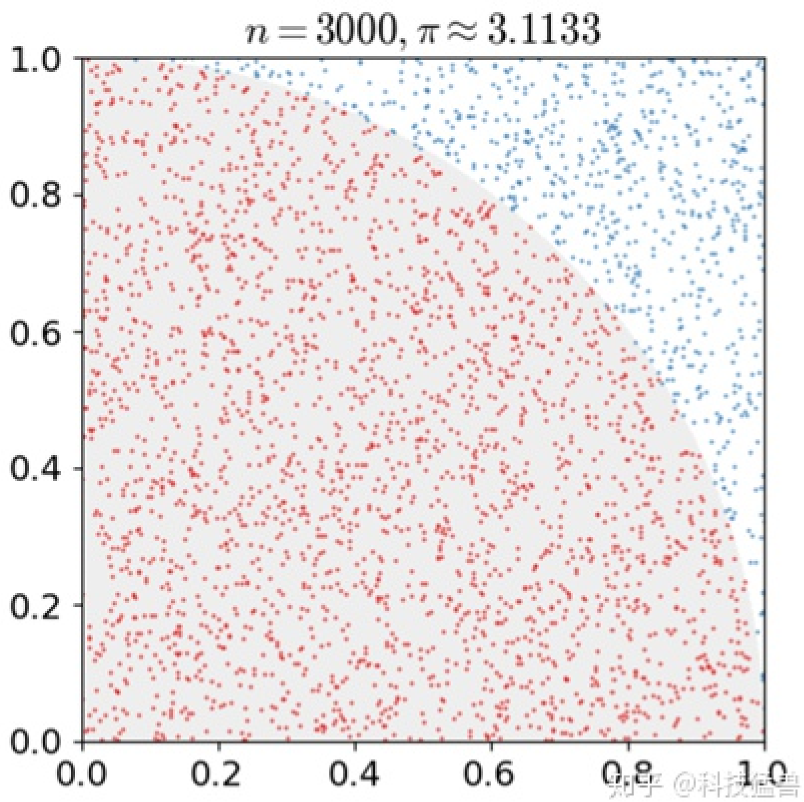


Fig. 1 蒙特卡罗估计圆周率

洒完这些点以后，圆周率的估计值就很明显了：

$$\pi = \frac{4N}{3000}$$

其中 N 是一个数据点放在圆内部的次数。

- **2. 蒙特卡罗估计任意积分的值**

> 这个问题来自中学课本。

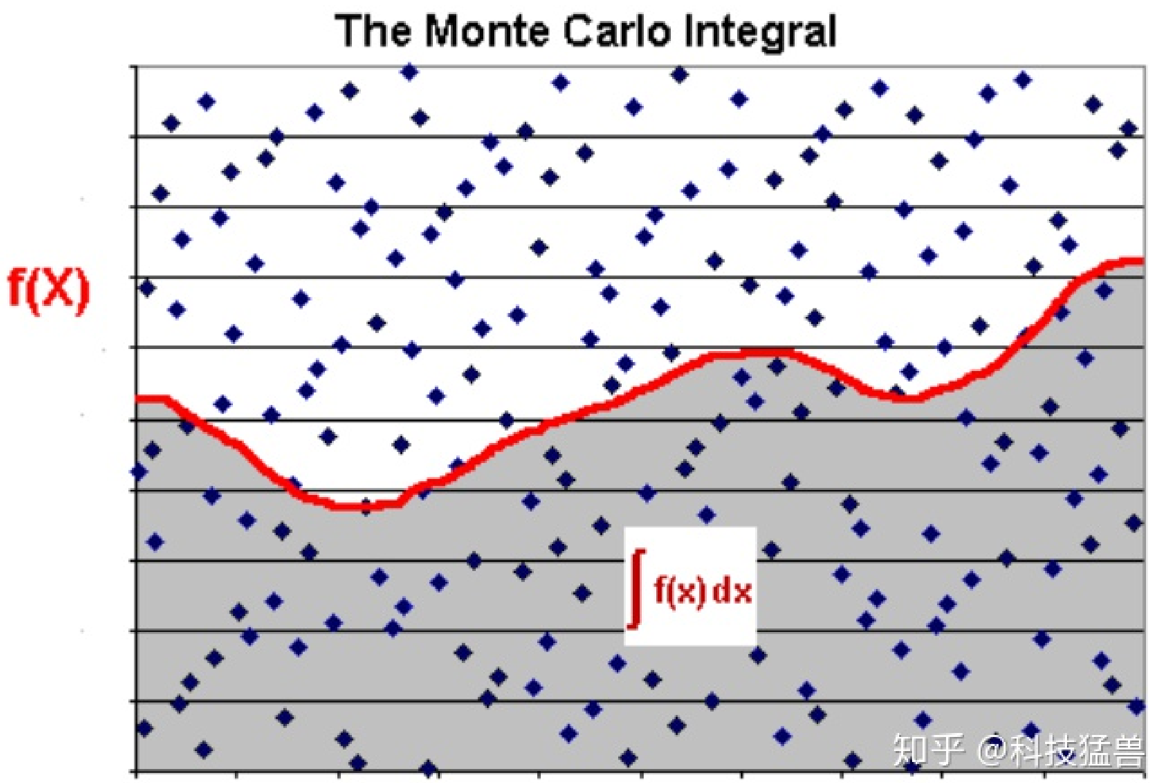


Fig. 2 特卡罗估计任意积分的值

在该矩形内部，产生大量随机点，可以计算出有多少点落在阴影部分的区域，判断条件为$y_i < f(y_i)$，这个比重就是所要求的积分值，即：

$$\frac{N_{阴影}}{N_{total}} = \frac{S_{积分}}{S_{矩形}}$$

- **3. 蒙特卡罗求解三门问题**

> 这个问题来自中学课本。

三门问题 (Monty Hall problem) 大致出自美国的电视游戏节目 Let's Make a Deal。问题名字来自该节目的主持人蒙提 · 霍尔(Monty Hall)。参赛者会看见三扇关闭了的门，其中一扇的后面有一辆汽车，选中后面有车的那扇门可赢得该汽车，另外两扇门后面则各藏有一只山羊。当参赛者选定了一扇门，但未去开启它的时候，节目主持人开启剩下两扇门的其中一扇，露出其中一只山羊。主持人其后会问参赛者要不要换另一扇仍然关上的门。问题是：换另一扇门会否增加参赛者赢得汽车的机率吗？如果严格按照上述的条件，即主持人清楚地知道，自己打开的那扇门后是羊，那么答案是会。不换门的话，赢得汽车的几率是 1/3。换门的话，赢得汽车的几率是 2/3。

在三门问题中，用 0、1、2 分别代表三扇门的编号，在 [0,2] 之间随机生成一个整数代表奖品所在门的编号 prize，再次在 [0,2] 之间随机生成一个整数代表参赛者所选择的门的编号 guess。用变量 change 代表游戏中的换门 (ture) 与不换门(false)：

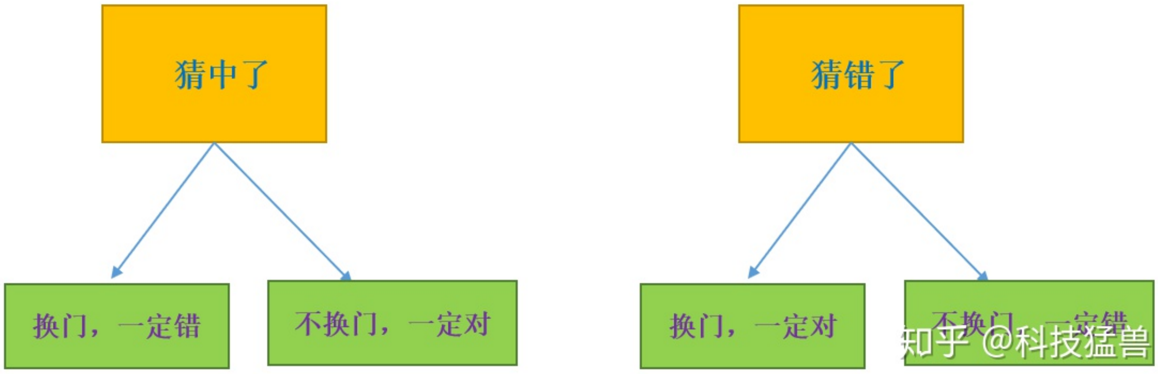


Fig. 3 蒙特卡罗求解三门问题

这样大量重复模拟这个过程（10000 次或者 100000 次）就可以得到换门猜中的概率和不换门猜中的概率。

使用 python 编程实现 (程序太简单就省了，节约版面)，结果为：

```
玩1000000次,每一次都换门:
中奖率为:
0.667383


玩1000000次,每一次都不换门:
中奖率为:
0.333867
```

发现了吗？蒙特卡罗方法告诉我们，换门的中奖率竟是不换门的 2 倍。Why?

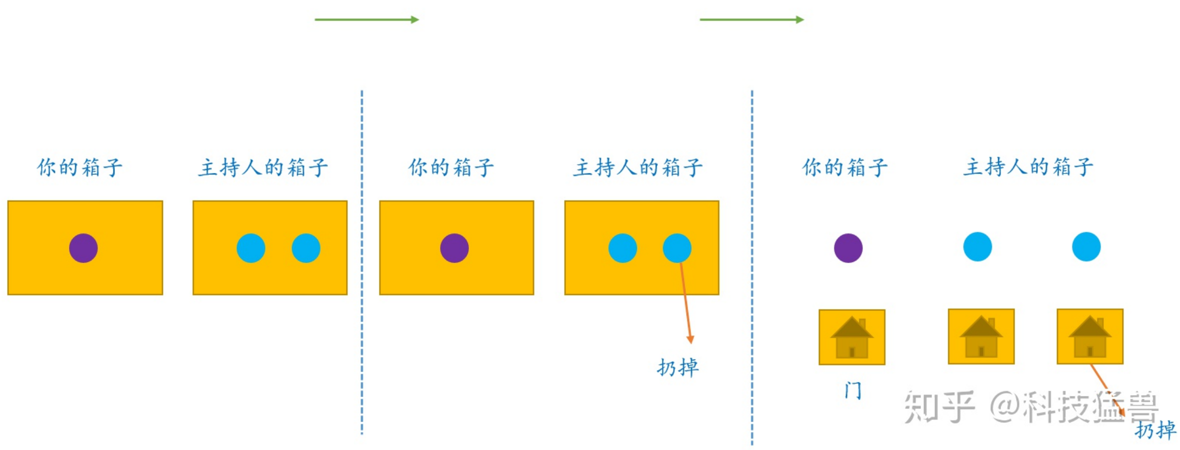


Fig. 4 蒙特卡罗求解三门问题理解

下面这个例子就能够让你理解这个问题：

比如说主持人和你做游戏，你有一个箱子，里面有 1 个球；主持人一个箱子，里面有 2 个球。他知道每个球的颜色，但你啥也不知道。但是 3 个球里面只有 1 个紫色的球，2 个蓝色的球，谁手里面有紫色的球，谁就获得大奖。

> 主持人说：你要和我换箱子吗？

当然换，我箱子里只有 1 个球，中奖率 $\frac{1}{3}$ ，他箱子里有 2 个球，中奖率 $\frac{2}{3}$ ，换的中奖率是不换的 2 倍。

这是这个游戏的结论。现在情况变了：

> 主持人从他的箱子里扔了一个蓝色的球之后说：你要和我换箱子吗？

当然换，我箱子里只有 1 个球，中奖率 $\frac{1}{3}$ ，他箱子里有 2 个球，中奖率 $\frac{2}{3}$ 。扔了一个蓝色的，中奖率没变还是 $\frac{2}{3}$ ，换的中奖率是不换的 2 倍。

这是这个游戏的结论。现在情况又变了：

> 没有箱子了，3 个球之前摆了 3 扇门，只有一扇门后面是紫色的球，你只有 1 扇门，主持人有 2 扇，现在，主持人排除了一扇后面是蓝色球的门，再问你：你要和我换门吗？

这种情况和上一种一模一样，只不过去掉了箱子的概念，换成了门而已，当然这不是重要的，你也可以换成铁门，木门，等等。

所以还是换，我只有 1 个门，中奖率 $\frac{1}{3}$ ，他有 2 个门，中奖率 $\frac{2}{3}$ 。扔了一个没用的门，中奖率没变还是 $\frac{2}{3}$ ，换的中奖率是不换的 2 倍。

Over！有点跑偏了，怎么说到三门问题上去了？？？本文想表达的只是蒙特卡罗方法可以帮助我们求解三门问题。

- **4. 蒙特卡罗估计净利润**

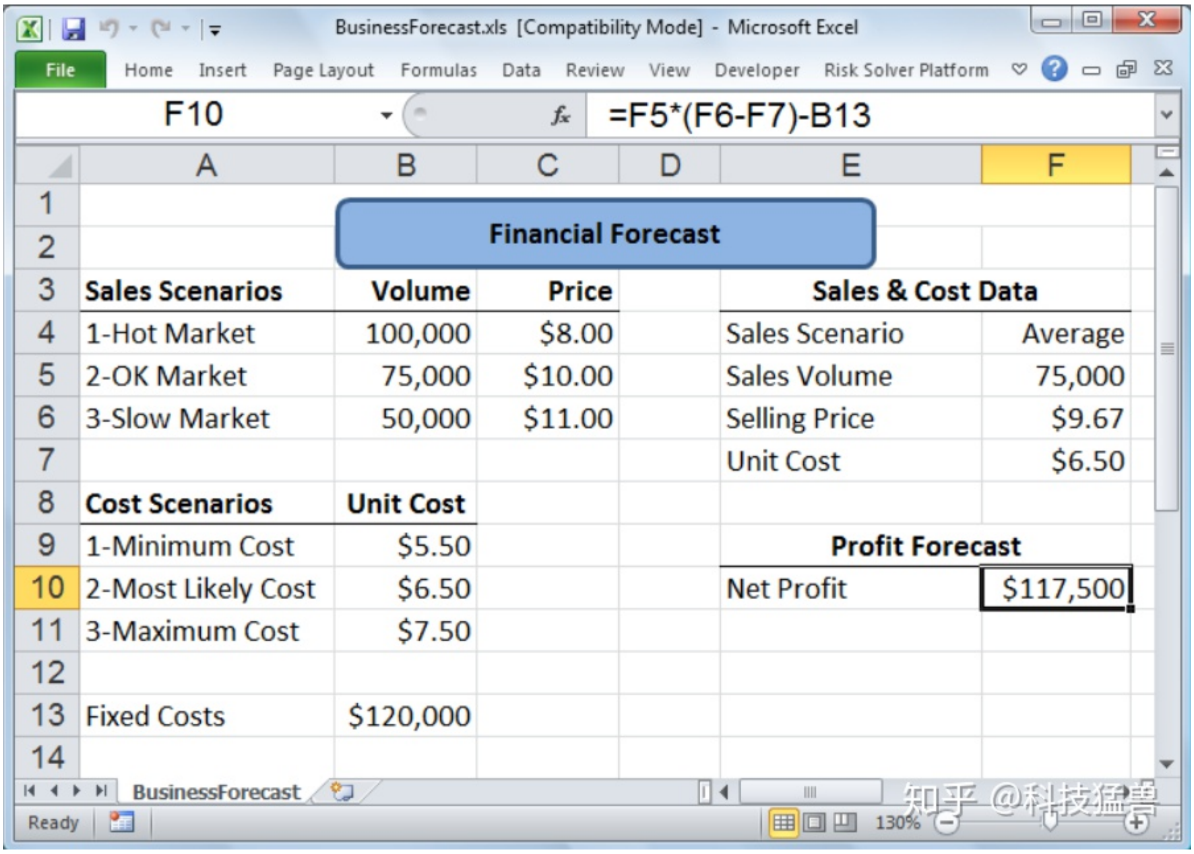
> 这个问题来自小学课本。

证券市场有时交易活跃，有时交易冷清。下面是你对市场的预测。

- 如果交易 Slow，你会以平均价 11 元，卖出 5 万股。
- 如果交易 Hot，你会以平均价 8 元，卖出 10 万股。
- 如果交易 Ok，你会以平均价 10 元，卖出 7.5 万股。
- 固定成本 12 万。

已知你的成本在每股 5.5 元到 7.5 元之间，平均是 6.5 元。请问接下来的交易，你的净利润会是多少？

取 1000 个随机样本，每个样本有两个数值：一个是证券的成本 (5.5 元到 7.5 元之间的均匀分布)，另一个是当前市场状态 (Slow、Hot、Ok，各有 $\frac{1}{3}$ 可能)。



	A	B	C	D	E	F
1		Financial Forecast				
2						
3	Sales Scenarios	Volume	Price		Sales & Cost Data	
4	1-Hot Market	100,000	$8.00		Sales Scenario	Average
5	2-OK Market	75,000	$10.00		Sales Volume	75,000
6	3-Slow Market	50,000	$11.00		Selling Price	$9.67
7					Unit Cost	$6.50
8	Cost Scenarios	Unit Cost				
9	1-Minimum Cost	$5.50			Profit Forecast	
10	2-Most Likely Cost	$6.50			Net Profit	$117,500
11	3-Maximum Cost	$7.50				
12						
13	Fixed Costs	$120,000				
14						

Fig. 5 蒙特卡罗估计净利润

模拟计算得到，平均净利润为 92, 427 美元。

计算方法是：1000 次抽样，每次按均匀分布随机选定一个成本值、同时按 1/3 的概率选中一个市场情境，然后拿这两个参数可计算出净利润、平均利润，千次加总并取均值就是结果。

$$(4.5 * 5 + 1.5 * 10 + 3.5 * 7.5)/3 - 12 = 9.25(万)$$

> **问：这些例子的共同点是什么？**

答：难度都不超过中学课本 (~)。最重要的是，它们都是通过**大量随机样本**，去**了解一个系统**，进而**得到所要计算的值**。

正是由于它的这种特性，所以被称为**统计模拟方法 (statistical simulation method)**，是通过从概率模型的随机抽样进行近似数值计算的方法。

其实随机算法分为两类：蒙特卡罗方法和拉斯维加斯方法，蒙特卡罗方法指的是算法的时间复杂度固定，然而结果有一定几率失败，采样越多结果越好。拉斯维加斯方法指的是算法一定成功，然而运行时间是概率的。

> **问：你有没有总结出蒙特卡罗方法的使用场景？**

答：有。当所求解的问题是**某种随机事件出现的概率**，或者是**某个随机变量的期望值**时，通过某种 "实验"(或者说 "计算机实验" 的方法)，以事件出现的频率作为随机事件的概率 (落在圆内的概率等)，或者得到这个随机变量的某些数字特征 (积分值，净利润等)，并将其作为问题的解。

> 估计概率和估计期望主要的统计学原理为大数定律；

你也许忽然间明白了，蒙特卡罗方法应该这么用：

比如说我要求某个**参量**，直接求解遇到了困难，那我就构造一个合适的**概率模型**，对这个模型进行大量的采样和统计实验，使它的**某些统计参量**正好是**待求问题的解**，那么，只需要把这个参量的值统计出来，那么问题的解就得到了估计值。

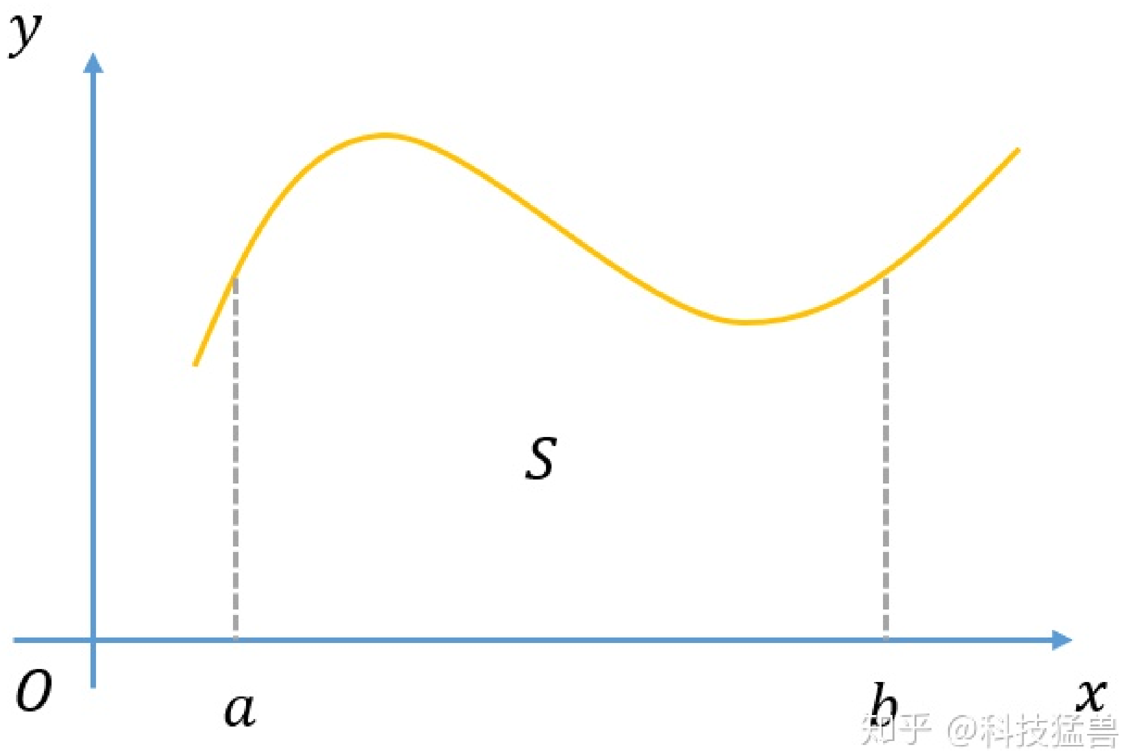
1.3 随机抽样

1.3.1 介绍

通过上面的几个例子我们发现：蒙特卡罗法要解决的问题是：**假设概率分布的定义已知**，通过抽样获得概率分布的随机样本，并**通过得到的随机样本对概率分布的特征进行分析**。比如，从样本得到经验分布，从而估计总体分布；或者从样本计算出样本均值，从而估计总体期望。所以蒙特卡罗法的核心是**随机抽样 (random sampling)**。

可是，随机抽样 (random sampling) 的方法从来都不是一成不变的，在下面的这个例子里面，我会阐明马尔科夫方法的一般形式：

求解积分：$S=\int_a^b f(x)dx$



如果很难求出 $f(x)$ 的具体表达式，那么我们就需要用到蒙特卡洛方法。你可以如前文所述在二维空间中洒 1000 个点，看有多少落在积分区域的内部。也可以换种方式理解这个做法：

$$S=\int_a^b f(x)dx=\int_a^b p(x)\frac{f(x)}{p(x)}dx=E_{x\sim p(x)}[\frac{f(x)}{p(x)}]$$

注意我们对这个积分表达式进行了一些 trick，使它变成了一个对 $\frac{f(x)}{p(x)}$ 的期望，这个期望服从的分布是 $p(x)$ 。注意这个分布 $p(x)$ 可以是任何一种概率分布。

下面如果想估计这个期望值，就需要按照概率分布 $p(x)$ 独立地抽取 n 个样本 $x_1,x_2,...,x_n$ 。

注意，这里不是胡乱抽取 n 个样本，而是按照概率分布 $p(x)$ 独立地抽取 n 个样本。 $p(x)$ 不同，抽取样本的方式当然也不会相同。

大数定律告诉我们，当 $n \rightarrow +\infty$ 时， $\frac{1}{n}\sum_{i=1}^{n}\frac{f(x_i)}{p(x_i)} \rightarrow E_{x\sim p(x)}[\frac{f(x)}{p(x)}]$

这句话的意思是说：我按照概率分布 $p(x)$ 独立地抽取 n 个样本，只要把 $\frac{1}{n}\sum_{i=1}^{n}\frac{f(x_i)}{p(x_i)}$ 算出来， $E_{x\sim p(x)}[\frac{f(x)}{p(x)}]$ 也就得到了。

一个特殊的情况是当按均匀分布抽取 n 个样本时，即 $p(x) = \frac{1}{b-a}$：

$$\frac{1}{n}\sum_{i=1}^{n}\frac{f(x_i)}{p(x_i)} = \frac{b-a}{n}\sum_{i=1}^{n}f(x_i)$$

这就是大一高数里面讲的积分的估计方法：

> 把积分区域等分为 n 份 (**均匀抽样**)，每一份取一个点 x_i ，计算出 $f(x_i)$ 取均值作为这一段积分的函数的均值，最后乘以积分区间的长度 $(b-a)$ 即可。

也就是蒙特卡罗方法的一个特例而已。

说了这么半天，它的基本思想就能用一个公式表达：

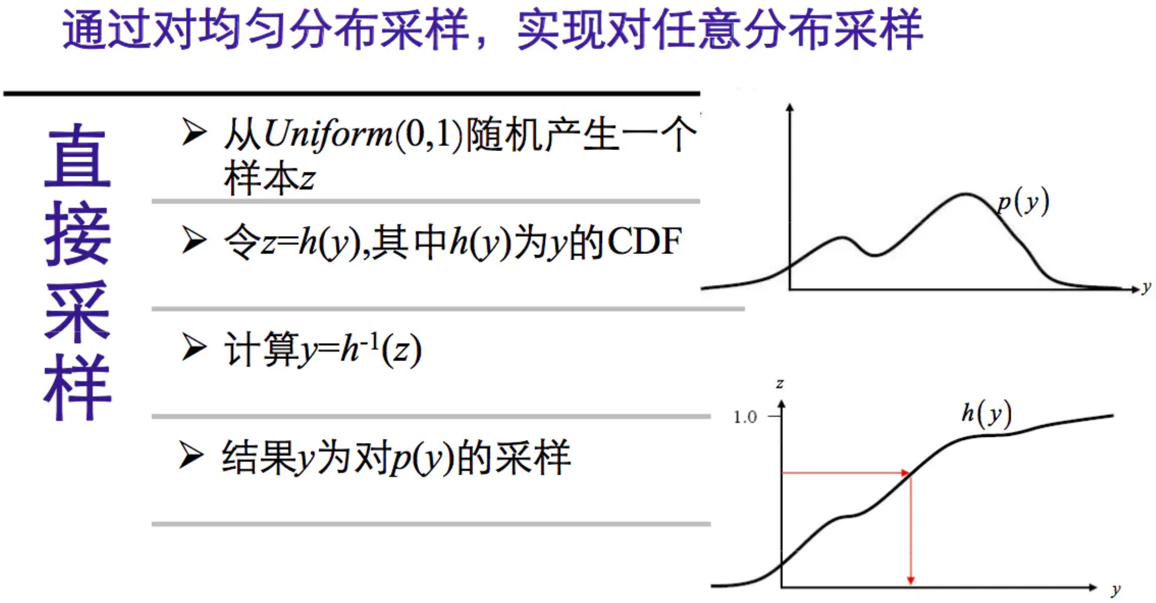
$\frac{1}{n}\sum_{i=1}^{n}\frac{f(x_i)}{p(x_i)} \rightarrow E_{x\sim p(x)}[\frac{f(x)}{p(x)}]$ ，注意 n 个点**要按照** $p(x)$ **采样**，且 n 越大，越精确。

现在的问题转到了如何按照 $p(x)$ 采样若干个样本上来。但是，**按照** $p(x)$ **采样绝非易事**，因为有些时候我们根本不知道 $p(x)$ 是什么，或者有时候是一个很复杂的表达式，计算机没法直接抽样。

1.3.2 随机抽样方法3：直接采样

直接采样的思想是，通过对均匀分布采样，实现对任意分布的采样。因为均匀分布采样好猜，我们想要的分布采样不好采，那就采取一定的策略通过简单采取求复杂采样。

假设y服从某项分布$p(y)$，其累积分布函数CDF为$h(y)$，有样本$z \sim Uniform(0,1)$，我们令$z = h(y)$，即 $y = h^{-1}(z)$，结果y即为对分布$p(y)$的采样。



举个例子：

例2：对指数分布采样

$$y \sim Exponential(1/\lambda)$$

$$h(y) = \int_{-\infty}^{y} p(y)dy = 1 - \int_{y}^{\infty} p(y)dy$$

$$= 1 - \int_{y}^{\infty} \lambda \exp(-\lambda y)dy$$

$$= 1 - \exp(-\lambda y)$$

$$z \sim Uniform(0,1)$$

$$z = h(y) = 1 - \exp(-\lambda y)$$

$$\therefore y = h^{-1}(z) = -\lambda^{-1}\ln(1-z) \sim Exponential(1/\lambda)$$

问题：如果$h(y)$不能确定怎么办？如果$h(y)$无法解析求逆怎么办？

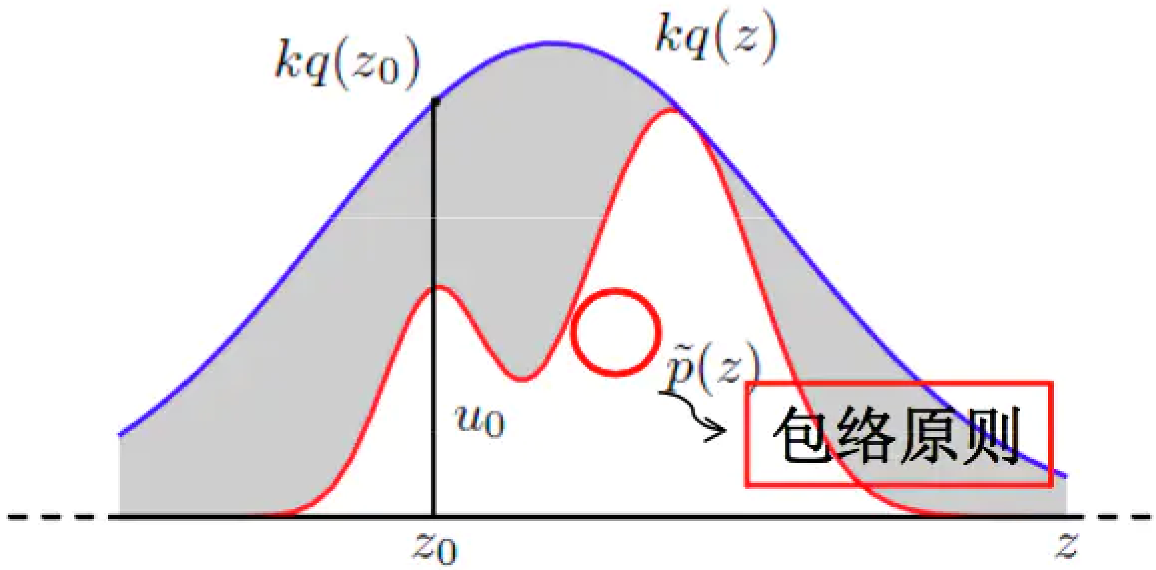
直接采样有一个问题，就是上图下方的那两个问号。

1.3.3 随机抽样方法 2：拒绝 - 接受采样

> **这种方法适用于$p(z)$极度复杂不规则的情况。**

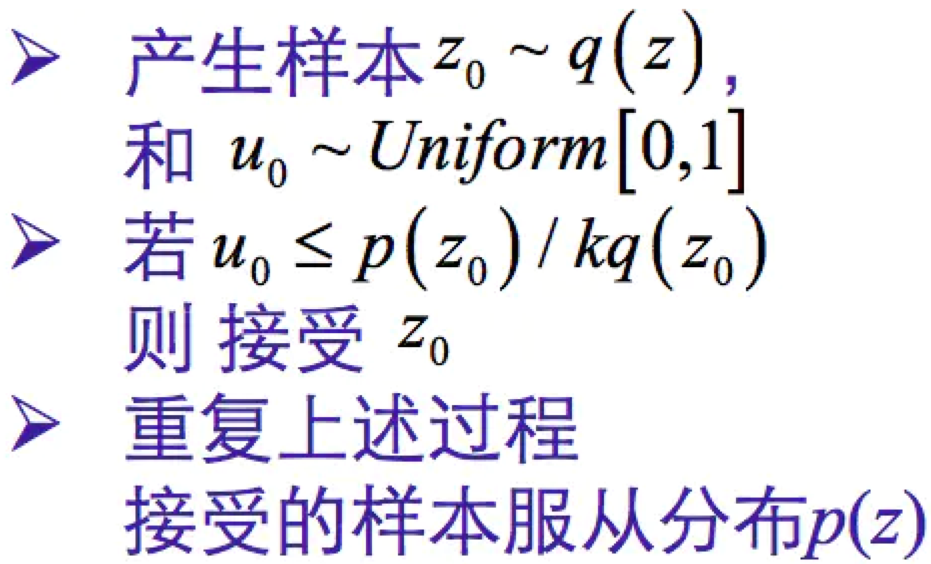
接受-拒绝采样的思想是，对于分布$p(z)$，很难通过直接采样进行采样，但是我们可以找一个可以直接采样分布$q(z)$作为媒介，这个媒介的专业术语为建议分布(proposal distribution)，可以通过直接采样或其他采样进行采样，那么我们怎么可以通过$q(z)$采样得到$p(z)$的采样呢？

先看下图：



红色的是$p(z)$, 蓝色的是$q(z)$，我们对$q(z)$乘一个参数k，让k能正好包住$p(z)$，那么对于每一个从$q(z)$得到的样本z_0，我们有一定的概率接受它，概率的大小就是$p(z_0)/kq(z_0)$。很容易就能看出来，在$p(z)$和$kq(z)$相切的地方的采样，接受率就是1。

那么有人问了，接受率能计算出来，但是我们对于一个样本z_0，到底怎么判断是接受还是不接受啊？我们有$u \sim Uniform[0,1]$，对于每一个样本z_0，我们一个u_0，如果$u_0 \leq p(z_0)/kq(z_0) = \alpha$($\alpha$为接受率)，我们就接受，否则就拒绝。重复此过程，得到的样本就服从分布$p(z)$。



当然，$q(z)$的选取要有一定规则：$q(z)$与$p(z)$外形要相近，$q(z)$采样方便。

现在有一个问题，k怎么求？

其实也很简单，看图有 $kq(z) \geq p(z)$，那么$k \geq p(z)/q(z)$，我们求$p(z)/q(z)$的最大值，即为k。

举个例子，对截断正态分布的接受-拒绝采样。

截断正态分布的意思就是对于正态分布$N(a,b)$，x属于[0, 4]，其在[0, 4]上的积分为1，而不是在负无穷到正无穷的积分为1。截断正态分布不是正态分布，所以，我们知道截断正态分布的概率密度函数。

维基上对截断正态分布的定义：

Suppose $X \sim N(\mu, \sigma^2)$ has a normal distribution and lies within the interval $X \in (a, b),\ -\infty \le a < b \le \infty$. Then X conditional on $a < X < b$ has a truncated normal distribution.

Its probability density function, f, for $a \le x \le b$, is given by

$$f(x; \mu, \sigma, a, b) = \frac{\frac{1}{\sigma}\phi(\frac{x-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}$$

and by $f = 0$ otherwise.

Here,

$$\phi(\xi) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\xi^2\right)$$

is the probability density function of the standard normal distribution and $\Phi(\cdot)$ is its cumulative distribution function. There is an understanding that if $b = \infty$, then $\Phi\left(\frac{b-\mu}{\sigma}\right) = 1$, and similarly, if $a = -\infty$, then $\Phi\left(\frac{a-\mu}{\sigma}\right) = 0$.

分子的小fai是标准正态分布的概率密度函数，分母上的Φ是标准正态分布的累积分布函数。

我们有$p(z)$服从$N(1,1)$，$I(0 <= x <= 4)$，令$q(z)$ $U[0,4]$，根据上图的公式，$p(z)$就有了，$q(z) = 1/4$，所以$k = \max(p(z)/q(z))$，在$z = 1$（均值）的时候$p(z)/q(z)$取最大，所以得到k:

$$k = \frac{4\phi(1-1)}{\Phi(4-1) - \Phi(0-1)} = 1.8997$$

这个例子比较巧，分母没有z，我们可以直接判断在$z = 1$时，$p(z)/q(z)$取最大，如果$p(z), q(z)$都有z，那么要通过求导的方式求k了。

1.3.4 随机抽样方法 3：重采样技术 reparameterization trick

> **这个方法在 VAE 中经常使用，可以参考我之前的 Blog，这种方法适用于 $p(x)$ 是常见的连续分布，比如正态分布，t 分布，F 分布，Beta 分布，Gamma 分布等。**

在 VAE 中使用重采样技术是为了能让网络能够完成反向传播，具体是这样子:

现在要从 $N(\mu, \sigma^2)$ 中采样一个 Z，相当于从 $N(0,1)$ 中采样一个 ε，然后让

$$Z = \mu + \varepsilon \times \sigma$$

于是，我们将从 $N(\mu, \sigma^2)$ 采样变成了从 $N(0,1)$ 中采样，然后通过参数变换得到从 $N(\mu, \sigma^2)$ 中采样的结果。这样一来，“采样” 这个操作就不用参与梯度下降了，改为采样的结果参与，使得整个模型可训练了。

> **重参数技巧**还可以这样来理解:
>
> 比如我有 $\dfrac{\partial L}{\partial Z}$，可是 Z 是采样得到的，后向传播无法继续往前面求导了。
>
> 现在使用了**重参数技巧**以后，$Z = \mu + \varepsilon \times \sigma$。$\dfrac{\partial L}{\partial \sigma^2} = \dfrac{\partial L}{\partial Z}\dfrac{\partial Z}{\partial \sigma^2} = \dfrac{\partial L}{\partial Z}\varepsilon$
>
> 这样就可以正常使用反向传播了。

所以说重采样技术的思想就是把**复杂分布的采样**转化为**简单分布的采样**，再通过一些**变换**把采样结果变回去。

再比如我要从**二维正态分布**中采样得到相互独立的 $X, Y \sim N(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2) = N(0,0,1,1)$，我就可以先从**均匀分布**中采样两个随机变量 $U_1, U_2 \sim U(0,1)$，再通过下面的变换得到 X, Y：

$$X = cos(2\pi U_1)\sqrt{-2lnU_2}$$

$$Y = sin(2\pi U_1)\sqrt{-2lnU_2}$$

这个变换的专业术语叫做 Box-Muller 变换，它的证明如下：

> 证：假设相互独立的 $X, Y \sim N(0,0,1,1)$，则：
>
> X 的概率密度：$p(X) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$，$Y$ 的概率密度：$p(Y) = \frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}$
>
> 因为相互独立，所以联合概率密度为：$p(X,Y) = \frac{1}{2\pi}e^{-\frac{x^2+y^2}{2}}$
>
> 使用二重积分的经典套路，将 X、Y 作坐标变换，使：
>
> $$X = Rcos\theta$$
> $$Y = Rsin\theta$$
>
> 得到：$p(R,\theta) = \frac{1}{2\pi}e^{-\frac{r^2}{2}}$
>
> 而且：
>
> $$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\frac{1}{2\pi}e^{-\frac{x^2+y^2}{2}}dXdY = \int_0^{2\pi}\int_0^{+\infty}\frac{1}{2\pi}e^{-\frac{r^2}{2}}rdrd\theta = 1$$
>
> **算到这里要明确我们的目标是什么？**
>
> **答：**应该是反推出来随机变量 $U_1, U_2 \sim U(0,1)$。先看看 R 和 θ 的概率分布和概率密度吧，根据概率论知识有：
>
> $$F_R(r) = \int_0^{2\pi}\int_0^{r}\frac{1}{2\pi}e^{-\frac{r^2}{2}}rdrd\theta = 1 - e^{-\frac{r^2}{2}}$$
>
> $$F_\theta(\varphi) = \int_0^{\varphi}\int_0^{+\infty}\frac{1}{2\pi}e^{-\frac{r^2}{2}}rdrd\theta = \frac{\varphi}{2\pi}$$
>
> 一眼就看出来 $\theta \sim U(0, 2\pi)$。所以 $\theta = 2\pi U_1 = 2\pi U_2$。
>
> 接下来设 $Z = 1 - e^{-\frac{R^2}{2}}$，不知道 Z 服从什么分布哦。
>
> $$P(Z<z) = P(1-e^{-\frac{R^2}{2}} < z) = P(R < \sqrt{-2ln(1-z)}) = F_R(\sqrt{-2ln(1-z)}) = z$$
>
> 所以 $Z = 1 - e^{-\frac{R^2}{2}} \sim U(0,1)$，现在知道了 Z 服从均匀分布。
>
> 所以此时有 $1 - e^{-\frac{R^2}{2}} \sim U(0,1)$，即 $e^{-\frac{R^2}{2}} \sim U(0,1)$，即有：
>
> $$R = \sqrt{-2lnU_1} = \sqrt{-2lnU_2}$$
>
> 所以：
>
> $$X = Rcos\theta = cos(2\pi U_1)\sqrt{-2lnU_2}$$
> $$Y = Rsin\theta = sin(2\pi U_1)\sqrt{-2lnU_2}$$
>
> Box-Muller 变换得证。

至此，我们讲完了蒙特卡洛方法，这个方法非常强大和灵活，也很容易实现。对于许多问题来说，它往往是最简单的计算方法，有时甚至是唯一可行的方法。

从上面可以看出，要想将蒙特卡罗方法作为一个通用的采样模拟求和的方法，必须解决如何方便得到各种**复杂概率分布**的对应的**采样样本集**的问题。而马尔科夫链就能帮助你找到这些**复杂概率分布**的对应的**采样样本集。**

马尔科夫链

2.1 马尔科夫的初步介绍

说明：

下文中涉及的转移概率矩阵与我们常用的不太一样，为转置后的概率矩阵，同样的状态分布向量也应该是一个行向量

在蒙特卡罗方法中，我们采集大量的样本，构造一个合适的**概率模型**，对这个模型进行大量的采样和统计实验，使它的**某些统计参量**正好是**待求问题的解。**但是，我们需要大量采样，虽然我们有**拒绝 - 接受采样**和**重采样技术**，但是依旧面临采样困难的问题。巧了，马尔科夫链可以帮我们解决这个难题。

首先我们看一些基本的定义：

> **定义 (马尔可夫链)：**考虑一个随机变量的序列 $X = \{X_0, X_1, \ldots, X_t, \ldots\}$ 这里 X_t 表示时刻 t 的随机变量，$t = 0, 1, 2, \ldots$。每个随机变量 X_t 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)**。
> 假设初始时刻的随机变量 X_0 遵循概率分布 $P(X_0) = \pi_0$，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 $\{X_0, X_1, \ldots, X_{t-2}\}$，这一性质称为马尔可夫性，即：
> $P(X_t|X_{t-1}, X_{t-2}, \ldots, X_0) = P(X_t|X_{t-1}), t = 1, 2, \ldots$
> 具有马尔可夫性的随机序列 $X = \{X_0, X_1, \ldots, X_t, \ldots\}$ 称为马尔可夫链 (Markov chain)，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。

如果这个**条件概率分布**与**具体的时刻** t 是无关的，则称这个马尔科夫链为时间齐次的马尔可夫链 (time homogenous Markov chain)。

定义：转移概率矩阵：

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$$

其中 $p_{ij} = P(X_t = i|X_{t-1} = j)$。

定义：马尔科夫链在 t 时刻的概率分布称为 t 时刻的状态分布：

$$\pi(t) = \begin{bmatrix} \pi_1(t) \\ \pi_2(t) \\ \pi_3(t) \end{bmatrix}$$

其中 $\pi_i(t) = P(X_t = i), i = 1, 2, \ldots$。

特别地，马尔可夫链的初始状态分布可以表示为：

$$\pi(0) = \begin{bmatrix} \pi_1(0) \\ \pi_2(0) \\ \pi_3(0) \end{bmatrix}$$

通常初始分布 $\pi(0)$ 向量只有一个分量是 1，其余分量都是 0，表示马尔可夫链从一个具体状态开始。

有限离散状态的马尔可夫链可以由有向图表示。结点表示状态，边表示状态之间的转移，边上的数值表示转移概率。从一个初始状态出发，根据有向边上定义的概率在状态之间随机跳转 (或随机转移)，就可以产生**状态的序列**。马尔可夫链实际上是刻画**随时间在状态之间转移**的模型，假设未来的转移状态只依赖于现在的状态，而与过去的状态无关。

下面通过一个简单的例子给出马尔可夫链的直观解释：如下图王者荣耀玩家选择英雄的职业的转变，转移概率矩阵为：

$$P = \begin{bmatrix} 0.5 & 0.5 & 0.25 \\ 0.25 & 0 & 0.25 \\ 0.25 & 0.5 & 0.5 \end{bmatrix}$$

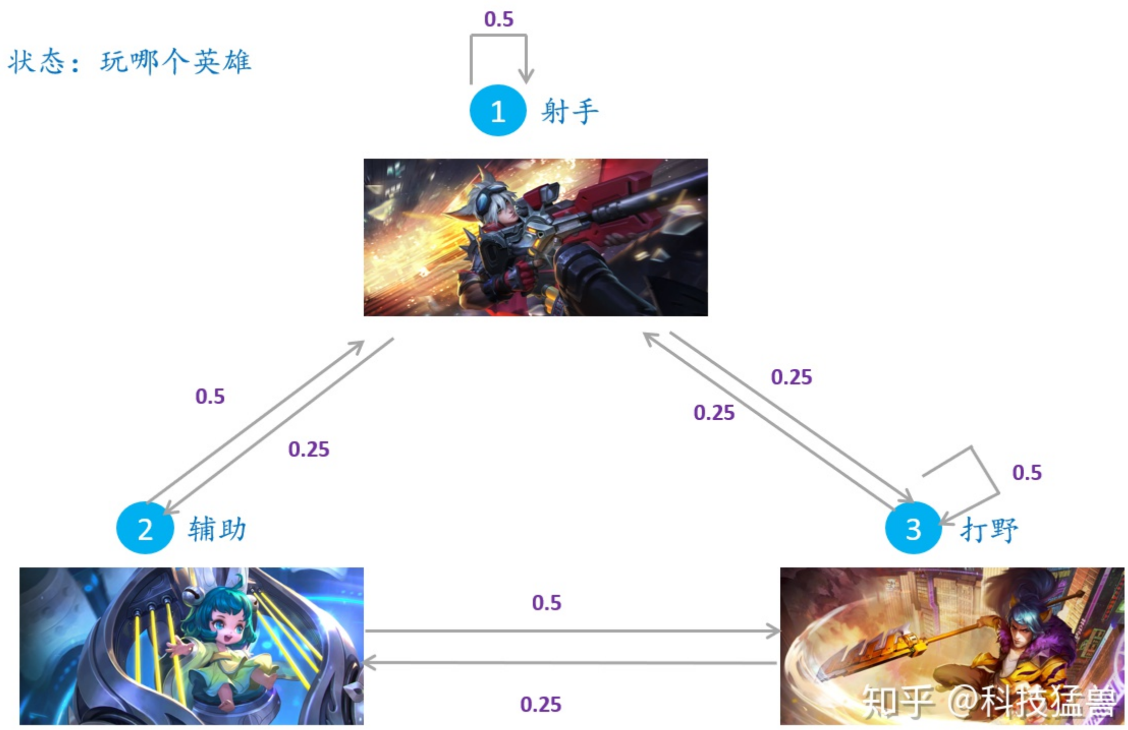
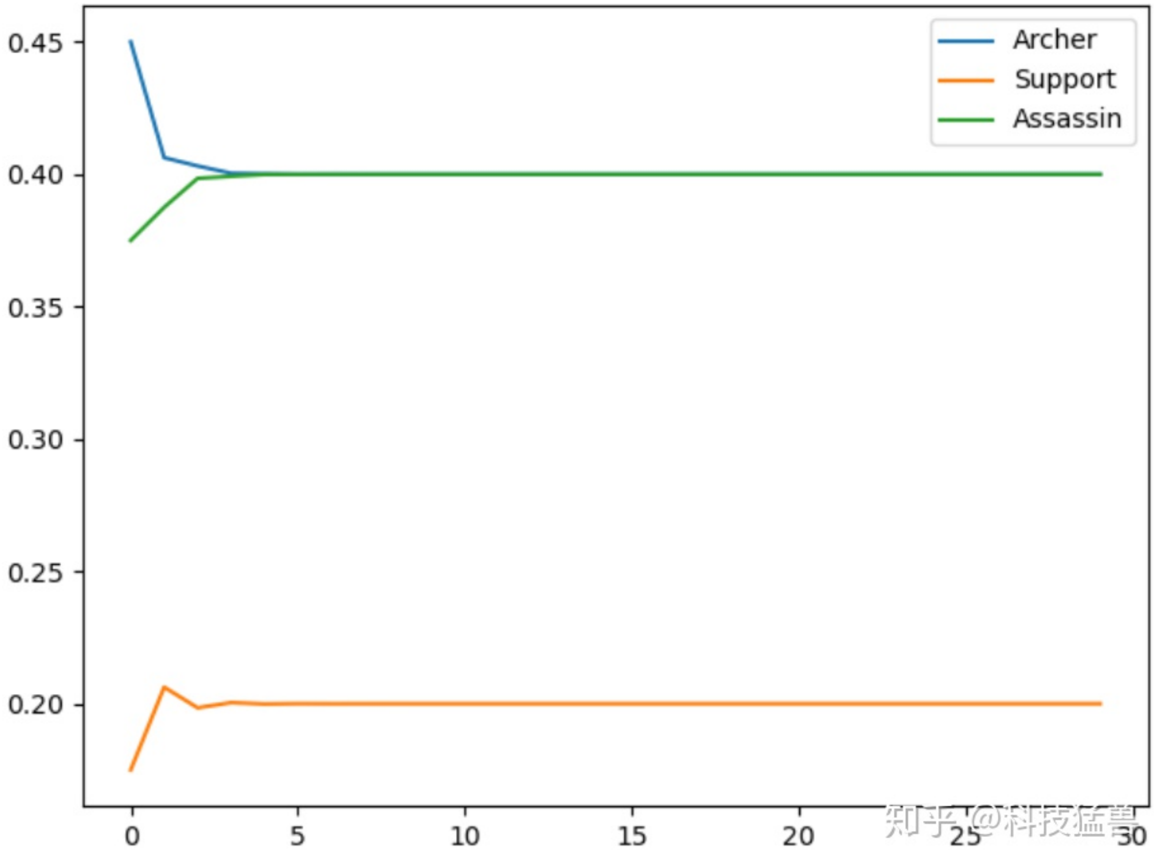


Fig. 7 王者荣耀玩家选择英雄的职业的转变

我们试图用程序去模拟这个状态的变化情况，任意假设一个初始状态：设初始的三个概率分别是 $[0.5, 0.3, 0.2]$，即t_0时刻，50% 概率选择射手，30% 概率选择辅助，20% 概率选择打野，将此代入转移概率，我们一直计算到t_{100}看看是什么情况：

```
import numpy as np
import matplotlib.pyplot as plt
transfer_matrix = np.array([[0.5,0.5,0.25],[0.25,0,0.25],
[0.25,0.5,0.5]],dtype='float32')
start_matrix = np.array([[0.5],[0.3],[0.2]],dtype='float32')
```

```python

value1 = []
value2 = []
value3 = []
for i in range(30):
    start_matrix = np.dot(transfer_matrix, start_matrix)
    value1.append(start_matrix[0][0])
    value2.append(start_matrix[1][0])
    value3.append(start_matrix[2][0])
print(start_matrix)

x = np.arange(30)
plt.plot(x,value1,label='Archer')
plt.plot(x,value2,label='Support')
plt.plot(x,value3,label='Assassin')
plt.legend()
plt.show()
```



可以发现，从 5 轮左右开始，我们的状态概率分布就不变了，一直保持在：

```
[[0.4       ]
 [0.19999999]
 [0.39999998]]
```

从这个实验我们得出了一个结论：这个玩家如果一直把王者荣耀玩下去，他最后选择射手，辅助，打野的概率会趋近于：$[0.4, 0.2, 0.4]$ ，很有意思的结论。

- **问：是不是所有马尔科夫链都有平稳分布？**

答：不一定，必须满足下面的定理：

> **定理**：给定一个马尔科夫链 $X=\{X_0,X_1,\ldots,X_t,\ldots\}$，$t$ 时刻的状态分布：$\pi=(\pi_1,\pi_2,\ldots)$ 是 X 的平稳分布的条件是 $\pi=(\pi_1,\pi_2,\ldots)$ 是下列方程组的解：
>
> $$x_i=\sum_j p_{ij}x_j, i=1,2,\ldots$$
>
> $$x_i\geq 0, i=1,2,\ldots$$
>
> $$\sum_i x_i=1$$
>
> **证**：
>
> 必要性：假设 $\pi=(\pi_1,\pi_2,\ldots)$ 是平稳分布，则显然满足后 2 式，根据平稳分布的性质，$\pi_i=\sum_j p_{ij}\pi_j, i=1,2,\ldots$，满足 1 式，得证。
>
> 充分性：假设 $\pi=(\pi_1,\pi_2,\ldots)$ 为X_t的分布，则：
>
> $$P(X_t=i)=\pi_i=\sum_j p_{ij}\pi_j=\sum_j p_{ij}P(X_{t-1}=j), i=1,2$$
>
> 所以$\pi=(\pi_1,\pi_2,\ldots)$ 也为X_{t-1}的分布，又因为对任意的 t 成立，所以 $\pi=(\pi_1,\pi_2,\ldots)$ 是平稳分布。

2.2 马尔科夫链的性质

- **不可约**

一个不可约的马尔可夫链，从任意状态出发，当经过充分长时间后，可以到达任意状态。 数学语言是：

> 定义：给定一个马尔科夫链 $X=\{X_0,X_1,\ldots,X_t,\ldots\}$，对于任意的状态 $i,j\in S$，如果存在一个时刻 t 满足：$P(X_t=i|X_0=j)>0$，也就是说，时刻 0 从状态j 出发，时刻 t 到达状态 i 的概率大于 0，则称此马尔可夫链 X 是不可约的 (irreducible)，否则称马尔可夫链是可约的 (reducible)。

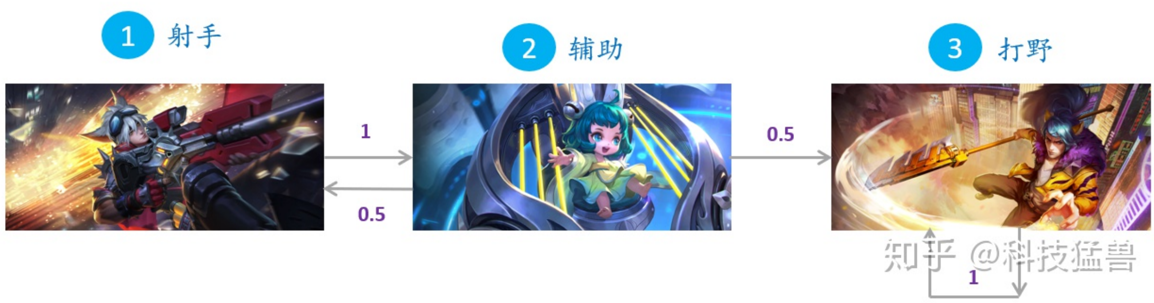


Fig. 8 可约的马尔科夫链

- **非周期**

> 定义：给定一个马尔科夫链 $X=\{X_0,X_1,\ldots,X_t,\ldots\}$，对于任意的状态 $i\in S$，如果时刻 0 从状态 i 出发，t 时刻返回状态的所有时间长 $\{t:P(X_t=i|X_0=i)>0\}$ 的最大公约数是 1，则称此马尔可夫链 X 是非周期的，否则称马尔可夫链是周期的。

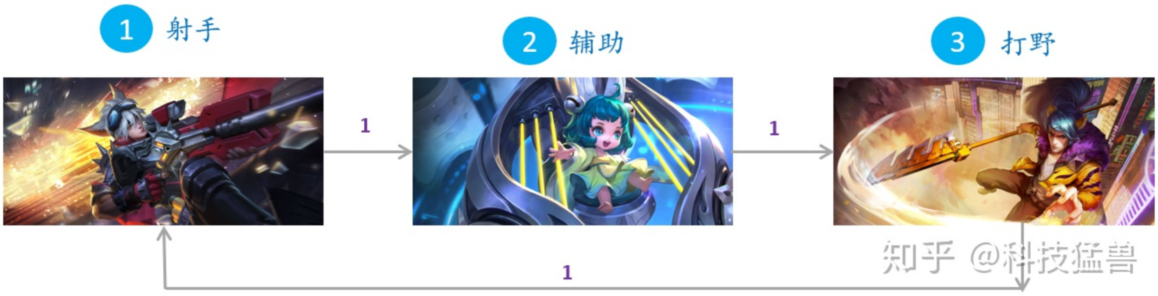


FIg. 9 周期的马尔科夫链

- **定理：不可约且非周期的有限状态马尔可夫链，有唯一平稳分布存在。**

定义：首达时间：

$T_{ij}=min\{n:n\geq 1,X_0=i,X_n=j\}$ 表示从状态 i 出发首次到达状态 j 的时间。若状态 i 出发永远不能到达状态 j，则 $T_{ij}=+\infty$。

定义：首达概率：

$$f_{ij}^{(n)}=P(X_n=j,X_m\neq j,m=1,2,\ldots,n-1|X_0=i)$$

$f_{ij}^{(n)}$ 为从状态 i 出发经过 n 步首次到达状态 j 的概率。

$f_{ij}^{(+\infty)}$ 为从状态 i 出发永远不能到达状态 j 的概率。

定义：从状态 i 出发经过有限步首次到达状态 j 的概率：

$$f_{ij}=\sum_{n=1}^{+\infty}f_{ij}^{(n)}=P(T_{ij}<+\infty)$$

定义：

状态 i 为常返态： $f_{ii}=1$，即有限步一定能回来。

状态 i 为非常返态： $f_{ii}<1$，即有限步可能回不来。

定义：平均返回时间：

$$\mu_i=\sum_{n=1}^{+\infty}nf_{ii}^{(n)}$$

- **正常返和零常返**

> **首先 i 得是常返态，且若 $\mu_i<+\infty$，称状态 i 为正常返。若 $\mu_i=+\infty$，称状态 i 为零常返。**

- **遍历态**

> i 既是正常返又是非周期，就是遍历态。

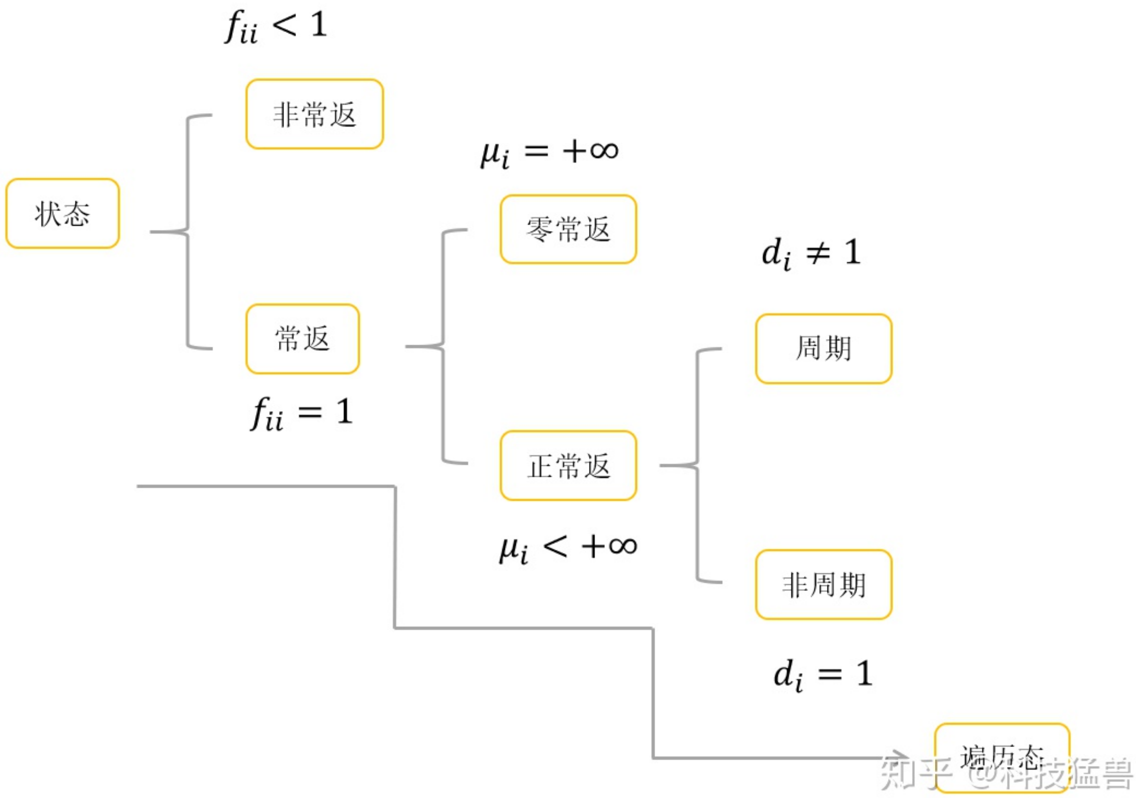
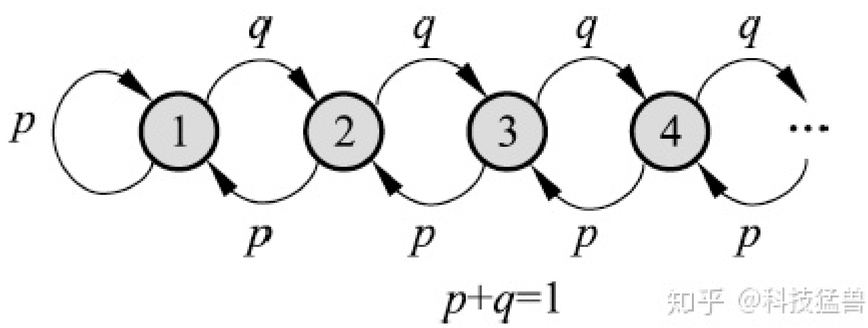


Fig. 10 各种定义汇总

直观上，一个正常返的马尔可夫链，其中任意一个状态，从其他任意一个状态出发，当时间趋于无穷时，首次转移到这个状态的概率不为 0。(从任意一个状态出发，走了能回来)



清华大学随机过程期

末考试题：这个马尔科夫链是正常返的吗?

如上图所示的马尔科夫链，当 $p > q$ 时是正常返的，当 $p < q$ 时不是正常返的。

> 证：转移概率矩阵：
>
> $$P = \begin{bmatrix} p & p & 0 & 0 & \dots \\ q & 0 & p & 0 & \dots \\ 0 & q & 0 & p & \dots \\ 0 & 0 & q & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$
>
> 若达到了平稳分布，则 $P\pi = \pi$，代入 P 化简，且注意 $p + q = 1$。
>
> $$\pi_2 = \frac{q}{p}\pi_1, \pi_3 = \frac{q}{p}\pi_2, \pi_4 = \frac{q}{p}\pi_3, \dots$$
>
> $$\pi_1 + \pi_2 + \pi_3 + \dots = 1$$
>
> 当 $p > q$ 时，平稳分布是：$\pi_i = (\frac{q}{p})^i(\frac{p-q}{p}), i = 1, 2, \dots$

> 当时间趋于无穷时，转移到任何一个状态的概率不为 0，马尔可夫链是正常返的。
> 当 $p < q$ 时，不存在平稳分布，马尔可夫链不是正常返的。
> 你看，清华的期末考试也不过如此~

- **遍历定理：不可约、非周期且正常返的马尔可夫链，有唯一平稳分布存在，并且转移概率的极限分布是马尔可夫链的平稳分布。**

$$\lim_{t\to+\infty} P(X_t = i|X_0 = j) = \pi_i, i = 1, 2, \ldots; j = 1, 2, \ldots$$

你会发现，不可约、非周期且正常返的马尔可夫链，它的转移概率矩阵在 $t \to +\infty$ 时竟然是：

$$\underset{n\to\infty}{lim} P^n = \begin{bmatrix} \pi(1) & \pi(1) & \cdots & \pi(1) & \cdots \\ \pi(2) & \pi(2) & \cdots & \pi(2) & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \pi(j) & \pi(j) & \cdots & \pi(j) & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix}$$

> **今后，你如果再遇到某个题目说"一个满足遍历定理的马尔科夫链，…"你就应该立刻意识到这个马尔科夫链只有一个平稳分布，而且它就是转移概率的极限分布。且随机游走的起始点并不影响得到的结果，即从不同的起始点出发，都会收敛到同一平稳分布。**注意这里很重要，一会要考。

说了这么多概念，用大白话做个总结吧：

- 不可约：每个状态都能去到。**(遍历别人)**
- 非周期：返回时间公约数是 1。**(不能周期性遍历，保证遍历的公平性)**
- 正常返：离开此状态有限步一定能回来。迟早会回来。**(遍历自己)**
- 零常返：离开此状态能回来，但需要无穷多步。
- 非常返：离开此状态有限步不一定回得来。
- 遍历定理：不可约，非周期，正常返 $\longrightarrow$ 有唯一的平稳分布。
- **可逆马尔可夫链**

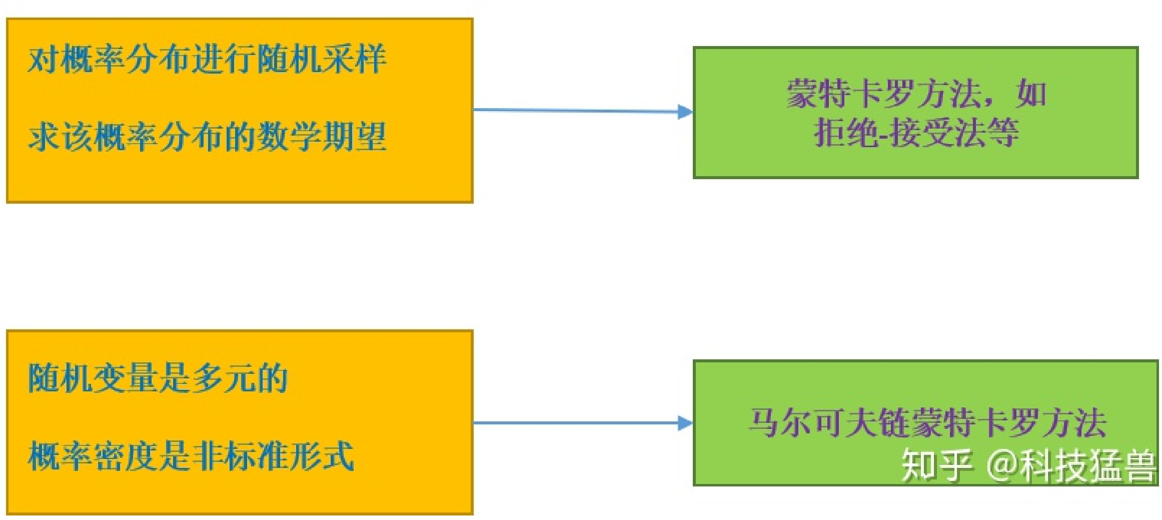
> 定义：给定一个马尔科夫链 $X = \{X_0, X_1, \ldots, X_t, \ldots\}$，如果有状态分布 $(\pi_1, \pi_2, \pi_3, \ldots)$。对于任意的状态 $i, j \in S$，对任意一个时刻 t 满足：
> $P(X_t = i|X_{t-1} = j)\pi_j = P(X_t = j|X_{t-1} = i)\pi_i, i = 1, 2, \ldots$
> 或简写为：
> $p_{ij}\pi_j = p_{ji}\pi_i$
> 则称此马尔可夫链 X 为可逆马尔可夫链 (reversible Markov chain)，上式称为
> **细致平衡方程 (detailed balance equation)。**
> 直观上，如果有可逆的马尔可夫链，那么以该马尔可夫链的**平稳分布**作为**初始分布**，进行随机状态转移，无论是面向未来还是面向过去，**任何一个时刻**的状态分布都是该**平稳分布**。概率分布π是状态转移矩阵P的平稳分布。
> **定理：**满足细致平衡方程的状态分布 π 就是该马尔可夫链的平稳分布 $P\pi = \pi$。
> **证：**
> $$(P\pi)_i = \sum_j p_{ij}\pi_j = \sum_j p_{ji}\pi_i = \pi_i \sum_j p_{ji} = \pi_i, i = 1, 2, \ldots$$

以上就是关于蒙特卡罗方法和马尔科夫链你分别需要掌握的知识，说了这么久还没有进入正题。本文是为了让你打好基础，那从下一篇文章开始，我们会讲解什么是马尔科夫链蒙特卡罗方法以及它的具体细节。

马尔科夫蒙特卡洛的结合

3.1 介绍

以上内容的核心思想可以用下图概括：



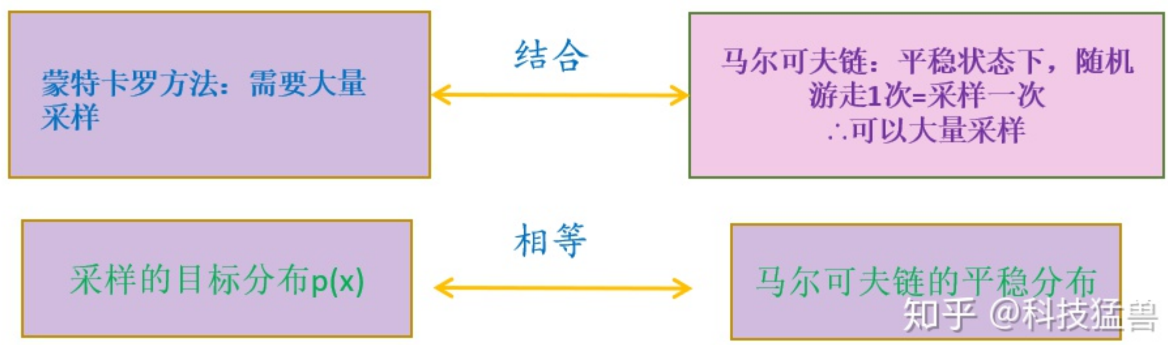
一般的采样问题，以及期望求解，数值近似问题，蒙特卡罗方法都能很好地解决；但遇到多元变量的随机分布以及复杂的概率密度时，仅仅使用蒙特卡罗方法就会显得捉襟见肘，这时就需要这篇文章要讲的马尔可夫链蒙特卡罗法来解决这个问题了。我们先从一维的讲起：

在开始之前首先统一下定义：

我们用符号 $\pi_i = \pi(i) = \lim_{t \to +\infty} P(X_t = i)$ 代表一个概率，即马尔科夫链达到平稳分布的时候，状态位于第 i 个状态的概率。

马尔科夫链和蒙特卡罗方法是如何结合在一起的?

一张图解释清楚：



还记得上篇文章中提到的遍历定理吗？如果你忘记了这些定义，请打开上面的链接再复习一遍并点赞：

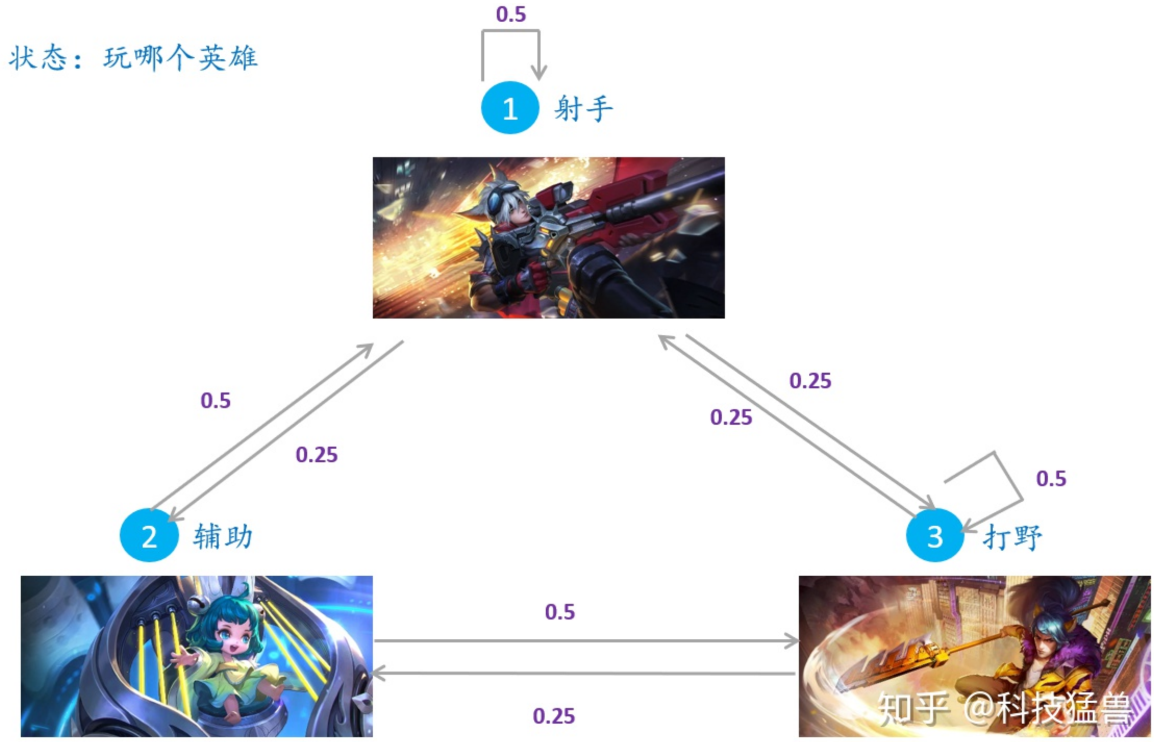
- **遍历定理：不可约、非周期且正常返的马尔可夫链，有唯一平稳分布存在，并且转移概率的极限分布是马尔可夫链的平稳分布。**

$$\lim_{t \to +\infty} P(X_t = i | X_0 = j) = \pi_i, i = 1, 2, \ldots; j = 1, 2, \ldots$$

当时让你记了一句话：**今后，你如果再遇到某个题目说“一个满足遍历定理的马尔科链，...”你就应该立刻意识到这个马尔科夫链只有一个平稳分布，而且它就是转移概率的极限分布。且随机游走的起始点并不影响得到的结果，即从不同的起始点出发，都会收敛到同一平稳分布。有一个成语叫殊途同归，形容的就是这件事。**

所以，我先定义一个满足遍历定理的马尔可夫链$X=\{X_0,X_1,\dots,X_t,\dots\}$，每一个状态就代表我在**王者荣耀**里面玩哪个英雄，比如初始状态 $X_0=1$ 就是玩射手，初始状态 $X_0=3$ 就是玩打野等等。

现在这个马尔科夫链因为满足遍历定理，所以有个平稳分布，代表的就是当 $t\to+\infty$ 时，我选择哪一个英雄的概率分布。



现在重复一下要解决的问题：

1. 从一个目标分布 $p(x)$ 中进行抽样；
2. 求出 $f(x)$ 的数学期望$E_{x\sim p(x)}[f(x)]$，那我就可以假设：

王者荣耀马尔可夫链的平稳分布 = 目标分布 $p(x)$

所以，一个惊人的结论诞生了：

在王者荣耀这个马尔科夫链上游走 1 次，对应的状态就相当于是从目标分布 $p(x)$ 中进行抽样 1 个样本。换句话说，比如王者荣耀平稳分布是 $[0.5,0.2,0.3]$ ，目标分布 $p(x)$也应该是 $[0.5,0.2,0.3]$，那在马尔科夫链上游走 1 次，比如说这把选了辅助，就相当于是在$p(x)$中采样了 x_2 。

所以，每个时刻在这个马尔可夫链上进行随机游走一次，就可以得到一个样本。根据遍历定理，当时间趋于无穷时，样本的分布趋近平稳分布，样本的函数均值趋近函数的数学期望。

所以，当时间足够长时 (时刻大于某个正整数 m)，在之后的时间 (时刻小于等于某个正整数 $n,n>m$) 里随机游走得到的样本集合 $\{x_{m+1},\dots,x_n\}$ 就是目标概率分布的抽样结果，得到的函数均值 (遍历均值) 就是要计算的数学期望值：

$$\hat{E}f = \frac{1}{n-m} \sum_{i=m+1}^{n} f(x_i)$$

到时刻 m 为止的时间段称为燃烧期。

马尔可夫链蒙特卡罗法中得到的样本序列，相邻的样本点是相关的，而不是独立的。因此，在**需要独立样本**时，可以在该样本序列中再次进行随机抽样，比如每隔一段时间取一次样本，将这样得到的子样本集合作为独立样本集合。马尔可夫链蒙特卡罗法比接受 - 拒绝法更容易实现，因为只需要定义马尔可夫链，而不需要定义建议分布。一般来说马尔可夫链蒙特卡罗法比接受 - 拒绝法效率更高，没有大量被拒绝的样本，虽然燃烧期的样本也要抛弃。

最大的问题是：给了我目标分布 $p(x)$，对应的王者荣耀马尔可夫链怎么构造？假设收敛的步数为 m，即迭代了 m 步之后收敛，那m是多少？迭代步数 n又是多少？

常用的马尔可夫链蒙特卡罗法有 Metropolis-Hastings 算法、吉布斯抽样。

马尔可夫链蒙特卡罗方法的基本步骤是：

1. 构造一个王者荣耀马尔可夫链，使其平稳分布为目标分布 $p(x)$ 。
2. 从某个初始状态 x_0 出发，比如第一把选射手，用构造的马尔可夫链随机游走，产生样本序列 $x_0, x_1, ..., x_t, ...$
3. 应用遍历定理，确定正整数 m 和 n，得到平稳分布的样本集合：$\{x_{m+1}, x_{m+2}, ..., x_n\}$，求得函数 $f(x)$ 的均值：

$$\hat{E}f = \frac{1}{n-m} \sum_{i=m+1}^{n} f(x_i)$$

3.2 如何构造一个王者荣耀马尔可夫链？

比如说现在已知的目标分布 $p(x)$是 $[0.5, 0.2, 0.3]$，想构造一个马尔可夫链，使它的平稳分布也是 $[0.5, 0.2, 0.3]$，关键还是要求出状态转移矩阵：

$$\begin{bmatrix} 0.5 \\ 0.2 \\ 0.3 \end{bmatrix} = P \begin{bmatrix} 0.5 \\ 0.2 \\ 0.3 \end{bmatrix}$$

在上篇文章中提到过这样一个定理：

定理：满足细致平衡方程的状态分布π就是该马尔可夫链的平稳分布$P\pi = \pi$。

所以，我只需要找到**可以使状态分布π满足细致平衡方程的矩阵 P 即可**，即满足：

$$\pi_i P_{j,i} = \pi_j P_{i,j}$$

上面这个细致平衡条件(Detailed Balance Condition)说的是任意两个状态之间转移是等可能的。需要注意的是这是一个充分条件，而**不是必要条件**，也就是说存在具有平稳分布的马尔科夫链不满足此细致平衡条件。

下面证明此条件的充分性：

因为：

$$\sum_i \pi_i P_{ij} = \sum_i \pi_j P_{ji} = \pi_j \sum_i P_{ji} = \pi_j$$

所以：

$$\pi P = \pi$$

注意上式的证明摘自原文，其中P_{ij}的定义等价于上文说到的P_{ji}。

仅仅从细致平衡方程还是很难找到合适的矩阵 P。比如我们的目标平稳分布是 π，随机找一个马尔科夫链状态转移矩阵 Q，它是很难满足细致平衡方程，即：

$$\pi_i Q_{j,i} \neq \pi_j Q_{i,j}$$

那么如何使这个等式满足呢？下面我们来看 MCMC 采样如何解决这个问题。

3.3 马尔科夫链蒙特卡罗方法总论

3.3.1 MCMC算法流程概述

构造一个 α_{ij} 和 α_{ji}，使上式强制取等号，即：

$$\pi_i Q_{j,i} \alpha_{j,i} = \pi_j Q_{i,j} \alpha_{i,j}$$

要使上式恒成立，只需要取：

$$\begin{aligned} \alpha_{j,i} &= \pi_j Q_{i,j} \\ \alpha_{i,j} &= \pi_i Q_{j,i} \end{aligned}$$

所以，马尔可夫链的状态转移矩阵就呼之欲出了：

$$\begin{aligned} P_{j,i} &= Q_{j,i} \alpha_{j,i} \\ P_{i,j} &= Q_{i,j} \alpha_{i,j} \end{aligned}$$

咦，状态转移矩阵 Q 是我们胡乱设的，α 值是可以根据 Q 和目标分布 $p(x)$ 算出来的，然后，要构造的满足细致平衡方程的矩阵P竟然被我们求出来了！！！

$\alpha_{i,j}$ 的专业术语叫做接受率。取值在 $[0, 1]$ 之间，可以理解为一个概率值。

状态转移矩阵 Q的平稳分布专业术语叫做建议分布 (proposal distribution)。

我们回顾一下在一顿数学公式之后，问题的演变过程：

从目标分布 $p(x)$ 中采样

$\longrightarrow$ 在马尔科夫链 (状态转移矩阵为 P) 中随机游走，当达到平稳分布时，每个时刻随机游走一次，就可以得到一个样本。

$\longrightarrow$ 在马尔科夫链 (状态转移矩阵为 Q) 中随机游走，当达到平稳分布时，每个时刻随机游走一次，就可以得到一个样本，这时，以一定的接受率获得，和上篇文章中的拒绝 - 接受采样极其类似，以一个常见的马尔科夫链状态转移矩阵 Q 通过一定的接受 - 拒绝概率得到目标转移矩阵 P。

好了，现在我们来总结下 MCMC 的采样过程。

1）输入我们任意选定的马尔科夫链状态转移矩阵 Q，平稳分布$\pi(x)$，设定状态转移次数阈值 n_1，需要的样本个数 n_2

2）从任意简单概率分布采样得到初始状态值 x_0

3）for $t=0$ to n_1+n_2-1:

a) 从条件概率分布 $Q(x|x_t)$中采样得到样本 x_*

b) 从均匀分布采样 $u \sim uniform[0,1]$

c) 如果 $u<\alpha(x_t,x_*)=\pi(x_*)Q(x_*,x_t)$，则接受转移 $x_t \to x_*$，即 $x_{t+1}=x_*$

d) 否则不接受转移，即 $x_{t+1}=x_t$

样本集 $(x_{n_1},x_{n_{1+1}},\ldots,x_{n_1+n_2-1})$即为我们需要的平稳分布对应的样本集。

Algorithm 5 MCMC 采样算法

1: 初始化马氏链初始状态$X_0=x_0$

2: 对$t=0,1,2,\cdots$，循环以下过程进行采样

- 第t个时刻马氏链状态为$X_t=x_t$，采样$y\sim q(x|x_t)$
- 从均匀分布采样$u\sim Uniform[0,1]$
- 如果$u<\alpha(x_t,y)=p(y)q(x_t|y)$ 则接受转移$x_t\to y$，即$X_{t+1}=y$
- 否则不接受转移，即$X_{t+1}=x_t$

3.3.2 Metropolis-Hastings 采样算法

> 主要改进了接收率α，使其不至于总是很小导致总是很难接受样本

M-H 采样是 Metropolis-Hastings 采样的简称，这个算法首先由 Metropolis 提出，被 Hastings 改进，因此被称之为 Metropolis-Hastings 采样或 M-H 采样

M-H 采样解决了我们上一节 MCMC 采样接受率过低的问题。

我们回到 MCMC 采样的细致平稳条件：

$$\pi(i)Q(i,j)\alpha(i,j)=\pi(j)Q(j,i)\alpha(j,i) \tag{1}$$

我们采样效率低的原因是$\alpha(i,j)$太小了，比如为 0.1，而$\alpha(j,i)$为 0.2。即：

$$\pi(i)Q(i,j)\times 0.1=\pi(j)Q(j,i)\times 0.2 \tag{2}$$

这时我们可以看到，如果两边同时扩大五倍，接受率提高到了 0.5，但是细致平稳条件却仍然是满足的，因此核心公式还是没有变，即：

$$\pi(i)Q(i,j)\times 0.5=\pi(j)Q(j,i)\times 1 \tag{3}$$

这样我们的接受率可以做如下改进，即：

$$\alpha(i,j)=min\left\{\frac{\pi(j)Q(j,i)}{\pi(i)Q(i,j)},1\right\} \tag{4}$$

通过这个微小的改造，我们就得到了可以在实际应用中使用的 M-H 采样算法过程如下：

- 1）输入我们任意选定的马尔科夫链状态转移矩阵 Q，平稳分布$\pi(x)$，设定状态转移次数阈值 $n1$，需要的样本个数 $n2$
- 2）从任意简单概率分布采样得到初始状态值 $x0$
- 3）for $t = 0$ to $n_1 + n_2 - 1$:
  - a) 从条件概率分布 $Q(x|x_t)$中采样得到样本 x_*
  - b) 从均匀分布采样 $u \sim uniform[0,1]$
  - c) 如果 $u < \alpha(x_t, x_*) = min\left\{\frac{\pi(j)Q(j,i)}{\pi(i)Q(i,j)}, 1\right\}$, 则接受转移 $x_t \to x_*$，即 $x_{t+1} = x_*$
  - d) 否则不接受转移，即 $x_{t+1} = x_t$

样本集 $(x_{n_1}, x_{n_1+1}, \ldots, x_{n_1+n_2-1})$即为我们需要的平稳分布对应的样本集。

很多时候，我们选择的马尔科夫链状态转移矩阵 Q如果是对称的，即满足 $Q(i,j) = Q(j,i)$, 这时我们的接受率可以进一步简化为：

$$\alpha(i,j) = min\left\{\frac{\pi(j)}{\pi(i)}, 1\right\} \tag{5}$$

Algorithm 6 Metropolis-Hastings 采样算法

1: 初始化马氏链初始状态$X_0 = x_0$
2: 对$t = 0, 1, 2, \cdots$，循环以下过程进行采样

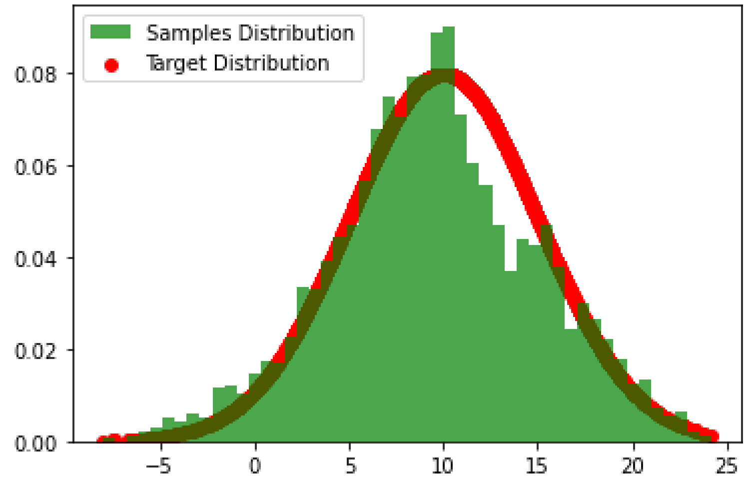
- 第t个时刻马氏链状态为$X_t = x_t$，采样$y \sim q(x|x_t)$
- 从均匀分布采样$u \sim Uniform[0,1]$
- 如果$u < \alpha(x_t, y) = \min\left\{\frac{p(y)q(x_t|y)}{p(x_t)p(y|x_t)}, 1\right\}$ 则接受转移$x_t \to y$，即$X_{t+1} = y$
- 否则不接受转移，即$X_{t+1} = x_t$

M-H 采样的 Python 实现

- 例子1：假设目标平稳分布是一个均值 10，标准差 5 的正态分布，而选择的马尔可夫链状态转移矩阵 $Q_{j,i}$ 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。

```
import random
from scipy.stats import norm    # scipy.stats专门用来生成指定分布
import matplotlib.pyplot as plt    # 绘图用的

"""
    stats连续型随机变量的公共方法
    rvs：产生服从指定分布的随机数, random variates of given type
    pdf：概率密度函数
    cdf：累计分布函数
    sf：残存函数（1-CDF）
    ppf：分位点函数（CDF的逆）
    isf：逆残存函数（sf的逆）
    fit：对一组随机取样进行拟合，最大似然估计方法找出最适合取样数据的概率密度函数系数。
    *离散分布的简单方法大多数与连续分布很类似，但是pdf被更换为密度函数pmf。

    常见分布 stats.
    beta：beta分布
```

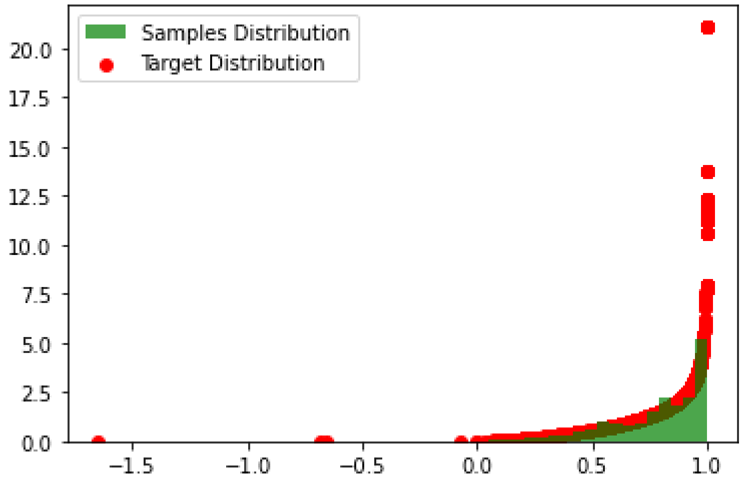
```
    f: F分布
    gamma: gam分布
    poisson: 泊松分布
    hypergeom: 超几何分布
    lognorm: 对数正态分布
    binom: 二项分布
    uniform: 均匀分布
    chi2: 卡方分布
    cauchy: 柯西分布
    laplace: 拉普拉斯分布
    rayleigh: 瑞利分布
    t: 学生T分布
    norm: 正态分布
    expon: 指数分布
"""

def norm_dist_prob(x):
    y = norm.pdf(x, loc=10, scale=5)  # mu=10,std=5的正态分布在x处的概率密度
    return y

T = 5000  # 采样5000次
pi = [0 for i in range(T)]   # 存储获取的样本，初始状态为0
sigma = 1   # 随机游走的幅度
t = 0
while t < T-1:
    t += 1

    ## Step 1: 随机游走
    # 以上一状态为中心随机游走得到下一个状态,size设定生成的数组形状与大小
    pi_new = norm.rvs(loc=pi[t-1], scale= sigma, size=1)
    # 计算接受率，状态转移矩阵为对称的
    alpha = min(1, (norm_dist_prob(pi_new[0]) / norm_dist_prob(pi[t-1])))

    ## Step 2: 判断游走的合理性
    u = random.uniform(0,1)
    pi[t] = pi_star[0] if u < alpha else pi[t-1]

plt.scatter(pi, norm.pdf(pi, loc=10, scale=5), label='Target Distribution', c='red')
num_bins = 50   # 直方图有50个块块
plt.hist(pi, num_bins, density=1, facecolor='green', alpha=0.7, label='Samples Distribution')
plt.legend()
plt.show()
```



- 例子2：再假设目标平稳分布是一个 $a=2.37, b=0.627$ 的 β 分布，而选择的马尔可夫链状态转移矩阵 $Q_{j,i}$ 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。

$$f(x,a,b)=\frac{\Gamma(a+b)x^{a-1}(1-x)^{b-1}}{\Gamma(a)\Gamma(b)}$$


```python
import random
from scipy.stats import beta, norm
import matplotlib.pyplot as plt

def beta_dist_prob(x):
    a = 2.37
    b = 0.627
    y = beta(a, b).pdf(x)
    return y

T = 5000
pi = [0 for i in range(T)]
sigma = 1
t = 0
while t < T-1:
    t += 1

    pi_new = norm.rvs(loc=pi[t-1], scale=sigma, size=1)
    alpha = min(1, beta_dist_prob(pi_new[0]) / beta_dist_prob(pi[t-1]))

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

plt.scatter(pi, beta_dist_prob(pi), label="Target Distribution", c='red')
num_bins = 50
plt.hist(pi, num_bins, density=1, facecolor='green', alpha=0.7, label='Samples Distribution')
plt.legend()
plt.show()
```



从输出的图中可以看到采样值的分布与真实的分布之间的关系如下，采样集还是比较拟合对应分布的。

M-H采样总结

M-H采样完整解决了使用蒙特卡罗方法需要的任意概率分布样本集的问题，因此在实际生产环境得到了广泛的应用。

但是在大数据时代，M-H采样面临着两大难题：

- 1) 我们的数据特征非常的多，M-H采样由于接受率计算式$\frac{\pi(j)Q(j,i)}{\pi(i)Q(i,j)}$的存在，在高维时需要的计算时间非常的可观，算法效率很低。同时$\alpha(i,j)$一般小于1，有时候辛苦计算出来却被拒绝了。能不能做到不拒绝转移呢?
- 2) 由于特征维度大，很多时候我们甚至很难求出目标的各特征维度联合分布，但是可以方便求出各个特征之间的条件概率分布。这时候我们能不能只有各维度之间条件概率分布的情况下方便的采样呢?

3.3.3 吉布斯抽样

> 相较于M-H采样，吉布斯抽样在多维分布且分布复杂但条件分布比较容易时表现得更好。总的来说是从时间复杂度的角度解决问题。

M-H 采样有两个缺点：一是需要计算接受率，在高维时计算量大。并且由于接受率的原因导致算法收敛时间变长。二是有些高维数据，特征的条件概率分布好求，但是特征的联合分布不好求。因此需要一个好的方法来改进 M-H 采样，这就是我们下面讲到的 Gibbs 采样。

M-H 采样由于接受率计算式$min\{\frac{\pi(j)Q(j,i)}{\pi(i)Q(i,j)},1\}$的存在，在高维时需要的计算时间非常的可观，算法效率仍然很低。而且，很多时候我们甚至很难求出目标的各特征维度联合分布，但是可以方便求出各个特征之间的条件概率分布。所以我们希望对条件概率分布进行抽样，得到样本的序列。

用大白话说，吉布斯抽样解决的问题和 M-H 方法**解决的问题是一致的**，都是从给定一个已知的目标分布$p(x)$中进行采样，并估计某个函数的期望值，区别只不过是此时，$p(x)$是一个多维的随机分布，$p(x)$的联合分布复杂，难以采样，但条件分布较容易，这样吉布斯抽样效果更好。

> 其基本做法是，**从联合概率分布定义条件概率分布，依次对条件概率分布进行抽样，得到样本的序列。可以证明这样的抽样过程是在一个马尔可夫链上的随机游走，每一个样本对应着马尔可夫链的状态，平稳分布就是目标的联合分布。**整体成为一个马尔可夫链蒙特卡罗法，燃烧期之后的样本就是联合分布的随机样本。

在前文中，我们讲到了细致平稳条件：如果非周期马尔科夫链的状态转移矩阵 P和概率分布$\pi(x)$对于所有的 i,j满足：

$$\pi(i)P(i,j) = \pi(j)P(j,i) \tag{6}$$

则称概率分布$\pi(x)$是状态转移矩阵 P的平稳分布。

在 M-H 采样中我们通过引入接受率使细致平稳条件满足。现在我们换一个思路。

从二维的数据分布开始，假设$\pi(x_1, x_2)$是一个二维联合数据分布，观察第一个特征维度相同的两个点$A(x_1^{(1)}, x_2^{(1)})$和$B(x_1^{(1)}, x_2^{(2)})$，容易发现下面两式成立：

$$\pi(x_1^{(1)}, x_2^{(1)})\pi(x_2^{(2)}|x_1^{(1)}) = \pi(x_1^{(1)})\pi(x_2^{(1)}|x_1^{(1)})\pi(x_2^{(2)}|x_1^{(1)})$$

$$\pi(x_1^{(1)}, x_2^{(2)})\pi(x_2^{(1)}|x_1^{(1)}) = \pi(x_1^{(1)})\pi(x_2^{(2)}|x_1^{(1)})\pi(x_2^{(1)}|x_1^{(1)})$$

由于两式的右边相等，因此我们有：

$$\pi(x_1^{(1)}, x_2^{(1)})\pi(x_2^{(2)}|x_1^{(1)}) = \pi(x_1^{(1)}, x_2^{(2)})\pi(x_2^{(1)}|x_1^{(1)})$$

也就是：

$$\pi(A)\pi(x_2^{(2)}|x_1^{(1)}) = \pi(B)\pi(x_2^{(1)}|x_1^{(1)})$$

观察上式再观察细致平稳条件的公式，我们发现在$x_1 = x_1^{(1)}$这条直线上，如果用条件概率分布$\pi(x_2|x_1^{(1)})$作为马尔科夫链的状态转移概率，则任意两个点之间的转移满足细致平稳条件！这真是一个开心的发现，同样的道理，在在$x_2 = x_2^{(1)}$这条直线上，如果用条件概率分布$\pi(x_1|x_2^{(1)})$作为马尔科夫链的状态转移概率，则任意两个点之间的转移也满足细致平稳条件。那是因为假如有一点$C(x_1^{(2)}, x_2^{(1)})$, 我们可以得到：

$$\pi(A)\pi(x_1^{(2)}|x_2^{(1)}) = \pi(C)\pi(x_1^{(1)}|x_2^{(1)})$$

基于上面的发现，我们可以这样构造分布$\pi(x_1, x_2)$的马尔可夫链对应的状态转移矩阵P：

$$P(A \to B) = \pi(x_2^{(B)}|x_1^{(1)}) \;\; if \; x_1^{(A)} = x_1^{(B)} = x_1^{(1)}$$

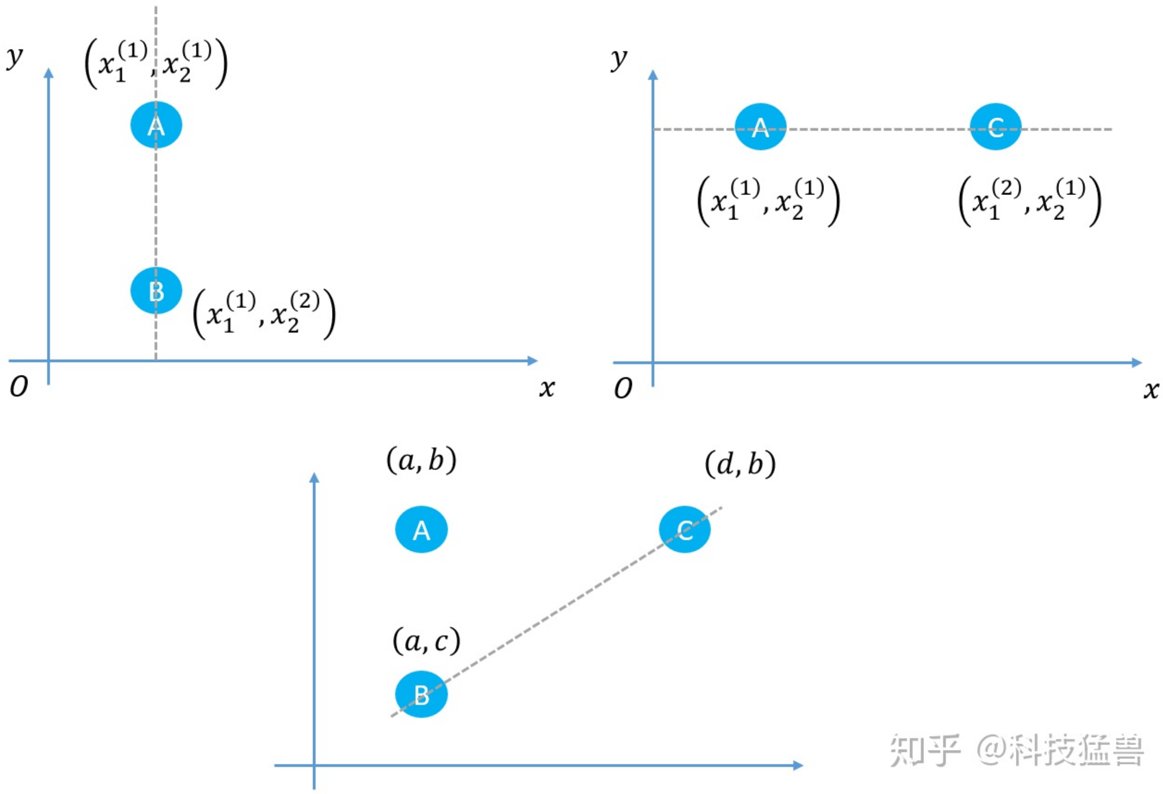
$$P(A \to C) = \pi(x_1^{(C)}|x_2^{(1)}) \;\; if \; x_2^{(A)} = x_2^{(C)} = x_2^{(1)}$$

$$P(A \to D) = 0 \;\; else$$

有了上面这个状态转移矩阵，我们很容易验证平面上的任意两点E, F，满足细致平稳条件：

$$\pi(E)P(E \to F) = \pi(F)P(F \to E)$$

接下来问题还是从二维分布 $p(x)$ 上进行采样，接下来这 3 个图代表以上所述状态转移的过程：



如下图所示为二维的状态空间，ABC 代表 3 个不同的状态。

对于第 1 个图：

$$\pi(x_1^{(1)},x_2^{(1)})\pi(x_2^{(2)}|x_1^{(1)})=\pi(x_1^{(1)})\pi(x_2^{(1)}|x_1^{(1)})\pi(x_2^{(2)}|x_1^{(1)})$$

$$\pi(x_1^{(1)},x_2^{(2)})\pi(x_2^{(1)}|x_1^{(1)})=\pi(x_1^{(1)})\pi(x_2^{(2)}|x_1^{(1)})\pi(x_2^{(1)}|x_1^{(1)})$$

发现上面 2 个式子右端是一样的（故意写成这样的，方便后面推导）。所以：

$$\pi(x_1^{(1)},x_2^{(1)})\pi(x_2^{(2)}|x_1^{(1)})=\pi(x_1^{(1)},x_2^{(2)})\pi(x_2^{(1)}|x_1^{(1)})$$

可以写成：

$$\pi(A)\pi(x_2^{(2)}|x_1^{(1)})=\pi(B)\pi(x_2^{(1)}|x_1^{(1)})$$

这个式子好像和细致平衡方程有点像，此时 A 和 B 是在一条直线：$x_1=x_1^{(1)}$ 上的。A 点在上方，B 点在下方，可以看做是一维分布，所以上式可以**看作是**：

$$\pi(\text{上})\pi(\text{上}\rightarrow\text{下})=\pi(\text{下})\pi(\text{下}\rightarrow\text{上})$$

就是细致平衡方程，所以 $\pi(x_2|x_1^{(1)})$ 就是我们要找的马尔科夫链的状态转移概率。

对于第 2 个图：

$$\pi(x_1^{(1)},x_2^{(1)})\pi(x_1^{(2)}|x_2^{(1)})=\pi(x_2^{(1)})\pi(x_1^{(1)}|x_2^{(1)})\pi(x_1^{(2)}|x_2^{(1)})$$

$$\pi(x_1^{(2)},x_2^{(1)})\pi(x_1^{(1)}|x_2^{(1)})=\pi(x_2^{(1)})\pi(x_1^{(2)}|x_2^{(1)})\pi(x_1^{(1)}|x_2^{(1)})$$

发现上面 2 个式子右端是一样的。所以：

$$\pi(x_1^{(1)},x_2^{(1)})\pi(x_1^{(2)}|x_2^{(1)})=\pi(x_1^{(2)},x_2^{(1)})\pi(x_1^{(1)}|x_2^{(1)})$$

可以写成：

$$\pi(A)\pi(x_1^{(2)}|x_2^{(1)}) = \pi(C)\pi(x_1^{(1)}|x_2^{(1)})$$

这个式子好像和细致平衡方程有点像，此时 A 和 C 是在一条直线： $x_2 = x_2^{(1)}$ 上的。A 点在左侧，B 点在右侧，可以看做是一维分布，所以上式可以**看作是**：

$$\pi(左)\pi(左 \rightarrow 右) = \pi(右)\pi(右 \rightarrow 左)$$

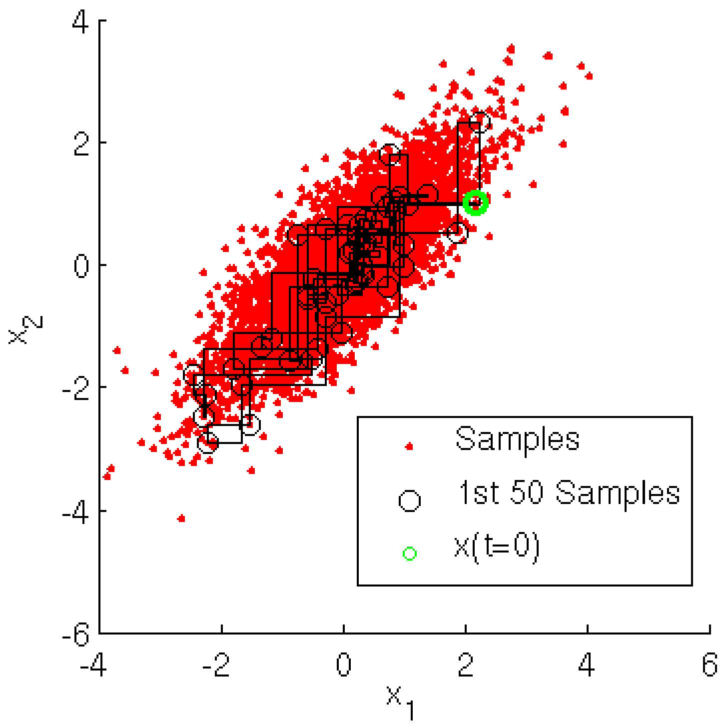
就是细致平衡方程，所以 $\pi(x_1|x_2^{(1)})$ 就是我们要找的马尔科夫链的状态转移概率。

对于第 3 个图：

把上 2 图的结论整合在一起可以得到最终的结论：

$$\pi(左下)\pi(左下 \rightarrow 右上) = \pi(右上)\pi(右上 \rightarrow 左下)$$

就是细致平衡方程，也就是说，我们在不同的维度上分别将 $\pi(x_2|x_1^{(1)})$ 和 $\pi(x_1|x_2^{(1)})$ 作为马尔科夫链的状态转移概率，就能实现最终的效果，即：**马尔科夫链达到平稳后的一次随机游走等同于高维分布的一次采样。**这就是 Gibbs 采样。



二维 Gibbs 采样的算法流程为：

1）输入平稳分布$\pi(x_1, x_2)$，设定状态转移次数阈值n_1，需要的样本个数n_2

2）随机初始化初始状态值$x_1^{(0)}$和$x_2^{(0)}$

3）for $t = 0$ to $n_1 + n_2 - 1$:

a) 从条件概率分布$P(x_2|x_1^{(t)})$中采样得到样本x_2^{t+1}

b) 从条件概率分布$P(x_1|x_2^{(t+1)})$中采样得到样本x_1^{t+1}

样本集$\{(x_1^{(n_1)}, x_2^{(n_1)}), (x_1^{(n_1+1)}, x_2^{(n_1+1)}), \ldots, (x_1^{(n_1+n_2-1)}, x_2^{(n_1+n_2-1)})\}$即为我们需要的平稳分布对应的样本集。

整个采样过程中，我们通过轮换坐标轴，采样的过程为：

$$(x_1^{(1)}, x_2^{(1)}) \rightarrow (x_1^{(1)}, x_2^{(2)}) \rightarrow (x_1^{(2)}, x_2^{(2)}) \rightarrow \ldots \rightarrow (x_1^{(n_1+n_2-1)}, x_2^{(n_1+n_2-1)})$$

Algorithm 7 二维Gibbs Sampling 算法

1: 随机初始化$X_0 = x_0 Y_0 = y_0$

2: 对$t = 0, 1, 2, \cdots$ 循环采样

1. $y_{t+1} \sim p(y|x_t)$

2. $x_{t+1} \sim p(x|y_{t+1})$

上面的这个算法推广到多维的时候也是成立的。比如一个 n 维的概率分布$\pi(x_1, x_2, \ldots x_n)$，我们可以通过在 n 个坐标轴上轮换采样，来得到新的样本。对于轮换到的任意一个坐标轴x_i上的转移，马尔科夫链的状态转移概率为$P(x_i|x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$，即固定$n-1$个坐标轴，在某一个坐标轴上移动。

具体的算法过程如下：

1）输入平稳分布$\pi(x_1, x_2, \ldots, x_n)$或者对应的所有特征的条件概率分布，设定状态转移次数阈值n_1，需要的样本个数n_2

2）随机初始化初始状态值$(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$

3）for $t = 0$ to $n_1 + n_2 - 1$:

a) 从条件概率分布$P(x_1|x_2^{(t)}, x_3^{(t)}, \ldots, x_n^{(t)})$中采样得到样本$x_1^{t+1}$

b) 从条件概率分布$P(x_2|x_1^{(t+1)}, x_3^{(t)}, x_4^{(t)}, \ldots, x_n^{(t)})$中采样得到样本$x_2^{t+1}$

c)...

d) 从条件概率分布$P(x_j|x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{j-1}^{(t+1)}, x_{j+1}^{(t)} \ldots, x_n^{(t)})$中采样得到样本$x_j^{t+1}$

e)...

f) 从条件概率分布$P(x_n|x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{n-1}^{(t+1)})$中采样得到样本$x_n^{t+1}$

样本集$\{(x_1^{(n_1)}, x_2^{(n_1)}, \ldots, x_n^{(n_1)}), \ldots, (x_1^{(n_1+n_2-1)}, x_2^{(n_1+n_2-1)}, \ldots, x_n^{(n_1+n_2-1)})\}$即为我们需要的平稳分布对应的样本集。

Algorithm 8 n维Gibbs Sampling 算法

1: 随机初始化$\{x_i : i = 1, \cdots, n\}$

2: 对$t = 0, 1, 2, \cdots$ 循环采样

1. $x_1^{(t+1)} \sim p(x_1|x_2^{(t)}, x_3^{(t)}, \cdots, x_n^{(t)})$
2. $x_2^{(t+1)} \sim p(x_2|x_1^{(t+1)}, x_3^{(t)}, \cdots, x_n^{(t)})$
3. $\cdots$
4. $x_j^{(t+1)} \sim p(x_j|x_1^{(t+1)}, \cdots, x_{j-1}^{(t+1)}, x_{j+1}^{(t)}, \cdots, x_n^{(t)})$
5. $\cdots$
6. $x_n^{(t+1)} \sim p(x_n|x_1^{(t+1)}, x_2^{t}, \cdots, x_{n-1}^{(t+1)})$

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：$\mu = (\mu_1, \mu_2) = (5, -1)$，$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}$；

首先需要求得：采样过程中的需要的状态转移条件分布：

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

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

证：

$$f(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} exp\left\{-\frac{1}{2(1-\rho^2)}\left[(\frac{x_1-\mu_1}{\sigma_1})^2 - 2r(\frac{x_1-\mu_1}{\sigma_1})(\frac{x_2-\mu_2}{\sigma_2}) + (\frac{x_2-\mu_2}{\sigma_2})^2\right]\right\}$$

$$f(x_2) = \frac{1}{\sqrt{2\pi}\sigma_2} exp\left\{-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right\}$$

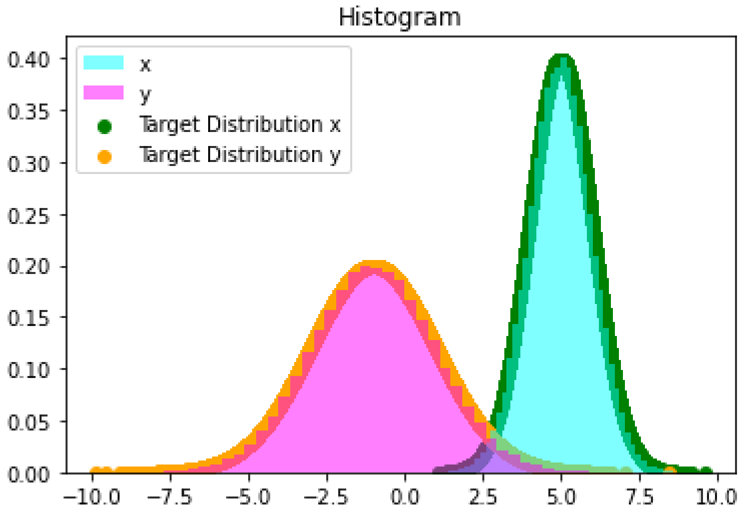
所以：

$$f(x_1|x_2) = \frac{1}{\sqrt{2\pi}\sigma_1\sqrt{1-\rho^2}} exp\left\{-\frac{1}{2(1-\rho^2)\sigma_1^2}\left[x_1 - \mu_1 - \frac{\sigma_1}{\sigma_2}\rho(x_2-\mu_2)\right]^2\right\}$$

得证。

```python
import random
import math
from scipy.stats import norm
import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d import Axes3D  # 绘制三维图
from scipy.stats import multivariate_normal  # 多维正态
```

```python

def p_x2givenx1(x1, m1, m2, s1, s2):
    return random.normalvariate(m2 + rho * s2 / s1 * (x1 - m1), math.sqrt(1 - rho**2)*s2)

def p_x1givenx2(x2, m1, m2, s1, s2):
    return random.normalvariate(m1 + rho * s1 / s2 * (x2 - m2), math.sqrt(1 - rho**2)*s1)

N = 5000
K = 20
x1_list = []
x2_list = []
z_list = []
m1 = 5
m2 = -1
s1 = 1
s2 = 2

rho = 0.5
x2 = m2    # 初值为均值

# 用来计算二元正态的概率密度
samplesource = multivariate_normal(mean=[m1,m2], cov=[[s1**2,rho*s1*s2],[rho*s1*s2,s2**2]])

for i in range(N):
    for j in range(K):
        x1 = p_x1givenx2(x2, m1, m2, s1, s2) #x2给定得到x1的采样
        x2 = p_x2givenx1(x1, m1, m2, s1, s2) #x1给定得到x2的采样
        z = samplesource.pdf([x1, x2])
        x1_list.append(x1)
        x2_list.append(x2)
        z_list.append(z)

num_bins = 50
plt.scatter(x1_list, norm.pdf(x1_list, loc=m1, scale=s1),label='Target Distribution x1', c= 'green')
plt.scatter(x2_list, norm.pdf(x2_list, loc=m2, scale=s2),label='Target Distribution x2', c= 'orange')
plt.hist(x1_list, num_bins, density=1, facecolor='Cyan', alpha=0.5,label='x1')
plt.hist(x2_list, num_bins, density=1, facecolor='magenta', alpha=0.5,label='x2')
plt.title('Histogram')
plt.legend()
plt.show()
```



然后我们看看样本集生成的二维正态分布，代码如下：

```
fig = plt.figure()    # 生成图像句柄
ax = Axes3D(fig, rect=[0,0,1,1], elev=30, azim=20)    # 生成坐标句柄
ax.scatter(x1_list, x2_list,z_list, marker='o', c='#00CED1')
plt.show()
```