

## 1. 什么是蒙特卡洛方法(Monte Carlo method)

蒙特卡罗方法也称统计模拟方法，是 1940 年代中期由于科学技术的发展和电子计算机的发明，而提出的一种以概率统计理论为指导的数值计算方法。是指使用随机数（或更常见的伪随机数）来解决很多计算问题的方法。

20 世纪 40 年代，在冯·诺伊曼，斯塔尼斯拉夫·乌拉姆和尼古拉斯·梅特罗波利斯在洛斯阿拉莫斯国家实验室为核武器计划工作时，发明了蒙特卡罗方法。因为乌拉姆的叔叔经常在摩纳哥的蒙特卡洛赌场输钱得名，而蒙特卡罗方法正是以概率为基础的方法。

与它对应的是确定性算法。

## 2. 蒙特卡洛方法的基本思想

通常蒙特卡罗方法可以粗略地分成两类：一类是所求解的问题本身具有内在的随机性，借助计算机的运算能力可以直接模拟这种随机的过程。例如在核物理研究中，分析中子在反应堆中的传输过程。中子与原子核作用受到量子力学规律的制约，人们只能知道它们相互作用发生的概率，却无法准确获得中子与原子核作用时的位置以及裂变产生的新中子的行进速率和方向。科学家依据其概率进行随机抽样得到裂变位置、速度和方向，这样模拟大量中子的行为后，经过统计就能获得中子传输的范围，作为反应堆设计的依据。

另一种类型是所求解问题可以转化为某种随机分布的特征数，比如随机事件出现的概率，或者随机变量的期望值。通过随机抽样的方法，以随机事件出现的频率估计其概率，或者以抽样的数字特征估算随机变量的数字特征，并将其作为问题的解。这种方法多用于求解复杂的多维积分问题。

## 3. 蒙特卡洛求定积分

蒙特卡洛方法的一个重要应用就是求定积分。来看下面的一个例子(参考文献 2)

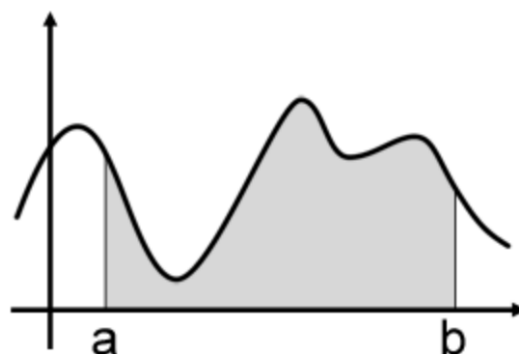
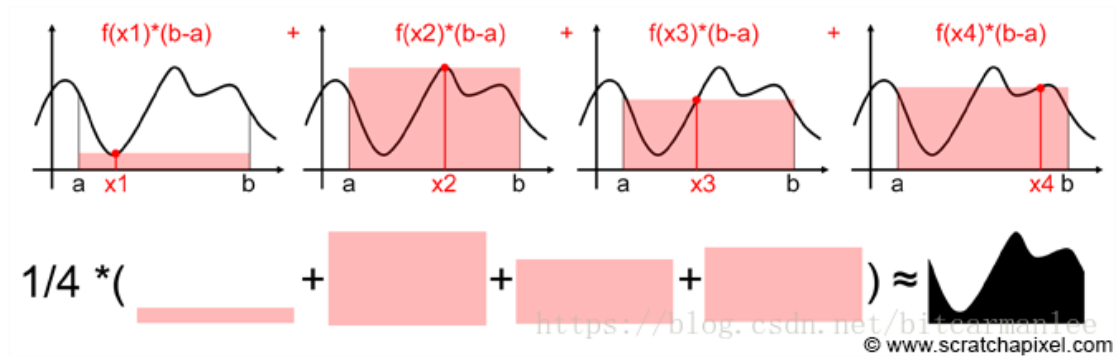


Figure 1: the integral over the domain  $[a,b]$  can be seen as the area under the curve.

当我们在 $[a,b]$ 之间随机取一点  $x$  时，它对应的函数值就是  $f(x)$ 。接下来我们就可以用  $f(x) * (b - a)$ 来粗略估计曲线下方的面积，也就是我们要求的积分值，当然这种估计（或近似）是非常粗略的。



在此图中，做了四次随机采样，得到了四个随机样本  $x_1, x_2, x_3, x_4$ ，并且得到了这四个样本的  $f(x_i)$  的值分别为  $f(x_1), f(x_2), f(x_3), f(x_4)$ 。对于这四个样本，每个样本能求一个近似的面积值，大小为  $f(x_i) * (b - a)$ 。为什么能这么干么？对照图下面那部分很容易理解，每个样本都是对原函数  $f$  的近似，所以我们认为  $f(x)$  的值一直都等于  $f(x_i)$ 。

按照图中的提示，求出上述面积的数学期望，就完成了蒙特卡洛积分。

如果用数学公式表达上述过程：

$$S = \frac{1}{4}(f(x_1)(b-a) + f(x_2)(b-a) + f(x_3)(b-a) + f(x_4)(b-a)) \\ = \frac{1}{4}(b-a)(f(x_1) + f(x_2) + f(x_3) + f(x_4)) = \frac{1}{4}(b-a) \sum f(x_i)$$

对于更一般的情况，假设要计算的积分如下：

$$I = \int_a^b g(x) dx$$

其中被积函数  $g(x)$  在  $[a,b]$  内可积。如果选择一个概率密度函数为  $f_X(x)$  的方式进行抽样，并且满足  $\int_a^b f_X(x) dx = 1$ ，那么令  $g^*(x) = g(x)f_X(x)$ ，原有的积分可以写成如下形式：

$$I = \int_a^b g^*(x) dx$$

$$I = \int_a^b g^*(x) f_X(x) dx$$

那么我们求积分的步骤应该是：

1. 产生服从分布律  $f_X(x)$  的随机变量  $X_i (i=1, 2, \dots, N)$

2. 计算均值

$$\bar{I} = \frac{1}{N} \sum_{i=1}^N g^*(X_i)$$

$$\bar{I} = \frac{1}{N} \sum_{i=1}^N g^*(X_i)$$

此时有  $\bar{I} \approx I$

当然实际应用中，我们最常用的还是取  $f_X(x)$  为均匀分布：

$$f_X(x) = \frac{1}{b-a}, a \leq x \leq b$$

$$f_X(x) = \frac{1}{b-a}, a \leq x \leq b$$

此时

$$g^*(x)=(b-a)g(x)$$

$$g^*(x)=(b-a)g(x)$$

代入积分表达式有:

$$I=(b-a)\int_a^b g(x)dx$$

$$I=(b-a)\int_a^b abg(x)dx$$

最后有

$$\overline{I} = b-aN \sum_{i=1}^N g(X_i)$$

$$\overline{I} = b-aN \sum_{i=1}^N g(X_i)$$

如果从直观上理解这个式子也非常简洁明了:

在 $[a,b]$ 区间上按均匀分布取  $N$  个随机样本  $X_i$ , 计算  $g(X_i)$  并取均值, 得到的相当于  $Y$  坐标值, 然后乘以  $(b-a)$  为  $X$  坐标长度, 得到的即为对应矩形的面积, 即积分值。

#### 4.蒙特卡洛方法 python 实例

首先看一个经典的用蒙特卡洛方法求  $\pi$  值。

```
import random
```

```
def calpai():
    n = 1000000
    r = 1.0
    a, b = (0.0, 0.0)
    x_neg, x_pos = a - r, a + r
    y_neg, y_pos = b - r, b + r

    count = 0
    for i in range(0, n):
        x = random.uniform(x_neg, x_pos)
        y = random.uniform(y_neg, y_pos)
        if x*x + y*y <= 1.0:
            count += 1

    print (count / float(n)) * 4
```

简单介绍下思路:

正方形内部有一个相切的圆, 它们的面积之比是  $\pi/4$ 。现在, 在这个正方形内部, 随机产生  $n$  个点, 计算它们与中心点的距离, 并且判断是否落在圆的内部。若这些点均匀分布, 则圆周率  $\pi=4 * count/n$ , 其中  $count$  表示落到圆内投点数  $n$ :表示总的投点数。

然后看一个求定积分的例子。

假设我们想求 $\int_0^1 x^2 dx$  的值。具体代码如下。

```
def integral():
    n = 1000000
    x_min, x_max = 0.0, 1.0
    y_min, y_max = 0.0, 1.0

    count = 0
    for i in range(0, n):
        x = random.uniform(x_min, x_max)
        y = random.uniform(y_min, y_max)
        # x*x > y, 表示该点位于曲线的下面。所求的积分值即为曲线下方的面积与正方形面积的比。
        if x*x > y:
            count += 1

    integral_value = count / float(n)
    print integral_value
```

代码运行的结果为：

0.332665

1

由此可见，与解析值的结果误差还是比较小的。

参考文献：

- 1.<https://zh.wikipedia.org/wiki/蒙地卡羅方法>
- 2.<http://www.scratchapixel.com/lessons/mathematics-physics-for-computer-graphics/monte-carlo-methods-in-practice/monte-carlo-integration>
- 3.<https://blog.csdn.net/baimafujinji/article/details/53869358>