第 112 天：机器学习算法之蒙特卡洛

原创 轩辕御龙 Python技术 1月10日

大家听说过的算法，比如快速排序法、二分查找法，或是像梯度下降法、K 近邻算法，这些算法都有比较严格的逻辑要求，使用起来有些繁琐。

这里我们介绍一个很简单却又通常行之有效的算法：蒙特卡洛方法。严格来说，蒙特卡洛方法并不是特指某一种具体的算法，而是对遵循某种思想的算法的统称，应该是一“类”算法。

“在试验不变的条件下，重复试验多次，随机事件的频率近似于它的概率”，这个统计学规律在数学上被称作“大数定律”，也很符合我们的自然直观。蒙特卡洛方法正是在这个规律的指导下，应用随机手段来逼近一些难以直接求解的数值。

“蒙特卡洛方法”这个名称看起来奇奇怪怪的，其实当中的“蒙特卡洛”指的就是著名赌城蒙特卡洛，传闻是由于该方法的发明者之一乌拉姆的叔叔常在此处输钱而得名——不得不说这个命名确实是很随意哈哈。但是认真来说，赌博与概率/统计的学科发展相依相伴，贯穿始终，既是统计学的发源地，又是概率论的演练场，以赌城之名来命名这样一个完全依赖于随机性的方法，也果然是相得益彰，十分到位。

说到这里不得不提一嘴的是，同为著名赌城，拉斯维加斯也有自己的“冠名算法”，本文就不做详述了，感兴趣的同学可以自行了解。

1. 蒙特卡洛方法的原理

实际上之前我们已经提到过了，蒙特卡洛方法的有效性是建立在大数定律的基础上的，也就是说我们需要通过模拟这样一个不断重复的随机过程，来获得与正常的反复随机试验相同的结果，因此该方法也被称为“蒙特卡洛模拟法”。

随着实验次数（即随机样本）的增加，从统计学意义上来说，得到的结果会越来越精确，与正确结果的误差会越来越小。之所以说是“统计学意义上”，是因为这种方法并不保证 2001 次随机试验的结果一定比 2000 次随机试验的结果更加准确，甚至不能保证比 1 次实验的结果更准确；但总体来看，实验次数越多，得到的结果确实更加可信。

通过上面的分析我们可以看出，蒙特卡洛方法使用的场景是相对比较灵活的，并且更适合对数据的精度要求并不太严格的场合。一般来讲，工业领域的精度要求是完全可以被蒙特卡洛方法满足的。

2. 两个应用实例

> 这两个实例本质上是一样的

2.1 求 π 的值

凡讲到蒙特卡洛方法，这几乎都是一个必被提及的应用实例。

正方形与其内切圆

上图所示，阴影部分是一个半径为 1 的圆形，另有一个边长为 2 的正方形与之相切。

我们从小就知道，圆面积公式为：

$$S_1 = \pi r^2$$

其中，S1 为圆面积，r 为圆半径，π 则是一个常数。

正方形面积公式为：

$$S_2 = l^2$$

其中，S2 为正方形面积，l 为正方形边长。

显然，对于特定的正方形，其内切圆的直径一定与正方形边长相等，也就是说圆半径是正方形边长的一半：$r = \frac{l}{2}$。这样，我们只要再知道 S1 和 S2 的比值，就可以根据上面这两个面积公式求出 π 的具体数值了。

更进一步地，由于圆形和正方形都具有特殊的对称性，因此我们可以只考察一部分图形，同样可以得到相同的结果：

正方形与其内切圆-部分

那么问题的关键就在于，这个比值到底应该怎么求呢？

方法有很多，最容易想到的就是对圆形求积分，得到对应的面积。对计算机来讲，我们可以返璞归真，用积分的思想，将图中这个扇形划分为大量小“矩形”，对小矩形面积求和即可得到扇形面积。

但是还有一种更加直接的方法。我们可以在图示的正方形中直接随机撒下一些点，然后统计落在扇形内部的点的个数，这个个数比上我们撒下的点的总个数，也就近似等于扇形面积与正方形面积之比。

正方形与其内切圆-部分-随机撒点

结合上述分析，我们可以得到 π 值的计算式：

$$\pi = 4\frac{S_{扇}}{S_{方}}$$

好了，铺垫了这么多，接下来让我们直接上代码：

```
>>> import random
>>> REPEAT = 20000 # 实验次数
>>> count = 0 # 用于记录落在扇形内部的随机点数
>>> for i in range(REPEAT):
...     x = random.random() # 生成[0.0, 1.0)区间内的均匀分布随机数
...     y = random.random()
```

```
...     if x*x + y*y < 1.0:
...         count += 1
...
>>> ratio = count / REPEAT
>>> PI = 4 * ratio
>>> PI
3.1388
```

可以看到，最后得到的 π 值与实际值是比较接近的。

2.2 求积分

同样地，在很多情境下，对于一些比较难以求出解析式的积分，或是即使知道解析式计算起来也比较麻烦的积分，我们并不需要一味地求出准确积分，而只需要通过蒙特卡洛方法得到一个粗糙的近似值即可，大大降低了计算的成本。

实际上，第一个例子的扇形面积可以从积分的角度考虑，而这个例子中的积分也同样可以从面积的角度考虑，二者本质上并无区别。

这里我们就以一个简单的积分为例，演示一下用蒙特卡洛方法求解积分的过程。

x 的平方

> 作图工具为 GeoGebra

图中所示函数为$f(x)=x^2$，所以$\int_0^1 f(x)$应该等于 1/3，也就是 0.666... 。

放码过来看看：

```
import random


def solve_integral(repeat = 20000) -> float:
    count = 0
    for i in range(repeat):
        x = random.random()
        y = random.random()
        if y > x*x:
            count += 1

    ratio = count / repeat
    integral = ratio * 1
    return integral

if __name__ == "__main__":
    repeat = int(input("请输入实验次数："))
    print(solve_integral(repeat))
```

```
    print(solve_integral(repeat))


# 请输入实验次数：500000
# 0.666066
```

3. 总结

本文简单介绍了一种简单的随机算法——蒙特卡洛方法。这种方法看起来非常“低级”，没有太多的技术含量，但实际上却正体现出了一种简单之美，用概率的方法战胜了复杂的计算，反而十分优雅。

同时这种方法也十分灵活，可以应用于许多不同的领域，实现起来门槛也不高，读者可以另行探究。

参考资料

大数定律-百度百科

蒙特卡洛方法-维基百科

蒙特卡罗方法入门-阮一峰的网络日志

> 示例代码：Python-100-days

系列文章

第 111 天：Python 垃圾回收机制
从 0 学习 Python 0 - 110 大合集总结

PS: 公号内回复 ：Python，即可进入Python 新手学习交流群，一起**100天计划！**