

# Buffon's Needle 问题以及蒙特卡洛

## Buffon's Needle

平面上有一系列等间距的平行线, 随意抛一支长度比木纹间距小的针, 求针和其中一条木纹相交的概率

解法:来自 [wiki 布丰投针问题](#)

假设针长度为 $l$ , 平行线间距为 $t$ ,  $x$  为针的中心和最近的平行线之间的距离,  $\theta$  为针和线之间的角度, 为锐角. 有

$x \in [0, t/2]$  均匀分布, 概率密度函数为 $2/t$

$\theta \in [0, \pi/2]$  均匀分布, 概率密度函数为 $2/\pi$

两个变量相互独立, 因此两个联合概率密度为两者之积  $\frac{4}{t\pi}$

当  $x \leq \frac{l}{2} \sin \theta$  时, 针和线相交, 然后对  $x, \theta$  积分计算所求的概率.

积分需要分两种情况  $l \leq t$  或者  $l > t$

对于前者(几何概型):

$$\begin{aligned} P &= \int_0^{\pi/2} \int_0^{(l/2) \sin \theta} \frac{4}{t\pi} dx d\theta \\ &= \int_0^{\pi/2} \frac{4}{t\pi} \cdot \left( \frac{l}{2} \cdot \sin \theta \right) d\theta \\ &= \frac{2l}{t\pi} \end{aligned}$$

## 估算 PI

另 $l = 1, t = 2$  则概率为 $1/\pi$

采用蒙特卡洛模拟, 思路是, 可以得到 $[0, 1]$ 均匀分布从其中采样, 但是由于不知道  $\pi$ , 因此不能够有效的到 $[0, \pi/2]$  均匀分布的采样结果.

替代方案是从 $[0, 1]$ 均匀分布里面直接采样 $\sin \theta, \theta \in [0, 2\pi]$

## 采样算法

[Sine and cosine of random angle in 2D](#) 37.4.3

1. 给定两个相互独立 $0,1$  均匀分布的采样  $u_1, u_2$

1.  $v_1 = 2u_1 - 1 \in (-1, 1)$

3.  $v_2 = u_2 \in (0, 1)$

4. 计算  $r^2 = v_1^2 + v_2^2$
5. 如果  $r^2 > 1$ , 重复步骤 1~4
6. 采样出的  $\sin = 2v_1v_2/r^2$ ,  $\cos = (v_1^2 - v_2^2)/r^2$

我找到了证明, [在这里](#) 用的也是几何概型+二倍角公式

$$S = \frac{2uv}{r^2} = \frac{2(r \cos \theta)(r \sin \theta)}{r^2} = 2 \cos \theta \sin \theta = \sin(2\theta)$$

$$C = \frac{u^2 - v^2}{r^2} = \frac{r^2 \cos^2 \theta - r^2 \sin^2 \theta}{r^2} = \cos^2 \theta - \sin^2 \theta = \cos(2\theta)$$

## Python 程序

```

1  import random
2
3
4  def buffon_needle(l, t, N=100000):
5      def _sample_sine():
6          r2 = 2
7          while r2 > 1:
8              u_1 = random.uniform(0, 1)
9              u_2 = random.uniform(0, 1)
10             v_1 = 2 * u_1 - 1
11             r2 = v_1 ** 2 + u_2 ** 2
12             return abs(2 * v_1 * u_2) / r2
13
14     cnt = 0
15     for _ in range(N):
16         center = random.uniform(0, t / 2)
17         sine = _sample_sine()
18         if center < l * sine / 2:
19             cnt += 1
20     p = cnt / N
21     print("P: {:.5f}, Pi:{:.10f}".format(p, 2 * l / (p * t)))
22
23
24 if __name__ == '__main__':
25     for i in range(10):
26         buffon_needle(1, 2)

```

结果如下

1	P: 0.31971, Pi:3.1278346001
2	P: 0.32093, Pi:3.1159442869
3	P: 0.32107, Pi:3.1145856044
4	P: 0.31959, Pi:3.1290090428
5	P: 0.31818, Pi:3.1428751021
6	P: 0.31977, Pi:3.1272477093
7	P: 0.32064, Pi:3.1187624750
8	P: 0.31699, Pi:3.1546736490
9	P: 0.31907, Pi:3.1341085028
10	P: 0.32022, Pi:3.1228530385