

Report2

PB2002480

王润泽

1 Question

用 16807 产生器测试随机数序列中满足关系 $X_{n-1} > X_{n+1} > X_n$ 的比重。讨论 Fibonacci 延迟产生器中出现这种关系的比重。

2 Algorithm

2.1 16807随机数生成器

Schrage 方法主要解决的是线性同余法产生随机数时可能出现的越界问题。其递推关系为:

$$I_{n+1} = \begin{cases} a(I_n \bmod q) - r[I_n/q], & \text{if } I_{n+1} > 0 \\ a(I_n \bmod q) - r[I_n/q] + m, & \text{otherwise} \end{cases}$$

实际计算中取 $q=127773$, $r=2836$, $a=16807$, $m=2147483647$ 。当提供了第一个小于 m 的初始值 I_0 之后, 就可以根据递推公式得到后续的伪随机数。

代码实现在 `Schrage_16807.py` 中封装

2.2 Fibonacci延迟产生器

Fibonacci 延迟产生器的思想是利用序列中的两个整数进行操作后得到后续的整数。即 $I_n = (I_{n-p} \otimes I_{n-q}) \bmod m$ 。其中操作符可以为: $+$, $-$, $*$ 等。本实验用的是带载减法产生器。其递推关系为

$$\begin{aligned} t_n &= I_{n-22} - I_{n-43} - C \\ C &= 0, I_n = t_n, & \text{if } t_n \geq 0 \\ C &= 1, I_n = t_n + (2^{23} - 5), & \text{if } t_n < 0 \end{aligned}$$

在具体生成随机数的时候, 首先指定一个初始值, 然后用 16807 产生器产生前 43 个随机数, 然后用 Fibonacci 产生器产生剩余的随机数序列。

代码实现在 `LaggedFibonacci` 类中。

2. Experiment

实验结果如下, 每组有 $1e5$ 个数据点

```
Seed = 1106799772, Schrage 16807: 0.169300, Fibonacci: 0.167000
Seed = 1248144892, Schrage 16807: 0.166600, Fibonacci: 0.163900
Seed = 1424826292, Schrage 16807: 0.170100, Fibonacci: 0.166400
Seed = 1566171412, Schrage 16807: 0.166600, Fibonacci: 0.163400
Seed = 1707516532, Schrage 16807: 0.168600, Fibonacci: 0.172200
Seed = 1848861652, Schrage 16807: 0.166800, Fibonacci: 0.160900
Seed = 1990206772, Schrage 16807: 0.166500, Fibonacci: 0.171800
Seed = 11974012, Schrage 16807: 0.165300, Fibonacci: 0.164000
Seed = 153319132, Schrage 16807: 0.162800, Fibonacci: 0.168300
Seed = 294664252, Schrage 16807: 0.165900, Fibonacci: 0.163300
Average of Schrage 16807 : 0.166850, Average of Fibonacci: 0.166120
```

图1：随机数序列中满足关系 $X_{n-1} > X_{n+1} > X_n$ 的比重

分析：

在完全随机的理想情况下，假设a,b,c是相互独立且都是均匀分布在[0,1]的随机数，那么

$$P(a > b \& b > c) = \int_0^1 da \int_0^a db \int_0^b dc = \frac{1}{6}$$

理想比重应当是 $\frac{1}{6}$ ，而从整体实验结果来看，两种生成器的计算结果都比较接近理想值，但都存在一定的偏差。而且可以看到，实验结果受到初始值影响较大。

3.Summary

两种产生器所生成的随机数组里面满足 $X_{n-1} > X_{n+1} > X_n$ 比重都接近于 1/6。两种生成器的计算结果都比较接近理想值，结果与初始条件选择有关。

本实验掌握了另一种随机数产生器的原理，并学习了一种新的衡量伪随机数产生器优劣的方式。