

Introduction

My Presentation Using Beamer

Berah, Espel, Kulak, Qiu, Zhang

Imperial College London

February 23, 2018

Generate Random Number I

There are two methods we used to generate random numbers. The first one is to generate a uniform distribution and then transform it into Standard Normal Distribution. The second method we tried is the system build-in function **Randon**, which can directly generate a variable follows Standard Normal Distribution.

To generate a uniform distribution we tried two ways: one can use the system build-in methods **rand**. The function will return a number between 1 and $2^{15} - 1$ randomly. The range is around 30,000, way below 100,000, the sample size. In other words, when we use rand to simulate 100,000 entries, there will be numbers appear more than once, which will create bias. We prefer the second way to generate a uniform distribution: linear congruential generator.

Generate Random Number II

$$n_i = (an_{i-1}) \bmod m$$

for $i = 1, 2, \dots, 10,000$, and we let $a = 7^5$, and $m = 2^{31} - 1$, which give about 2 billion points. We run 100,000 times use 0.005% of all point. Theoretically, no pattern should appear.

To transform the uniform distribution we got into Standard Normal Distribution, we have tried three methods. By Central Limit theory,

$$Z_n = \frac{\sum_{i=1}^n X_i - n\mu}{\sqrt{n}\sigma}$$

where X_i is from the uniform distribution we generated previously. Z_n converges in distribution to SND, however, it requires n , the number of uniform distribution, to be sufficiently large. Therefore this method requires significant large simulations and speed is slow consequently.

Box-Muller methods

We also tried Box-Muller methods

$$Z_1 = \sqrt{-2\ln X_1} \sin(2\pi X_2), \quad Z_2 = \sqrt{-2\ln X_1} \cos(2\pi X_2)$$

Since its simulation involves computation of sine and cosine, the speed is slow. We prefer the last method, Marsaglia Polar method.

$$\text{let } V_1 = 2U_1 - 1, \quad V_2 = 2U_2 - 1.$$

where U_1 and U_2 are two independent uniform distribution. Let $W = V_1^2 + V_2^2$. If $W > 1$, return to the beginning. Otherwise,

$$N_1 = \sqrt{\frac{(-2\log W)}{W}} V_1, \quad N_2 = \sqrt{\frac{(-2\log W)}{W}} V_2$$

As the computation doesn't involve sine and cosine, it is generally faster than Box-Muller.

My table

The following is the simulation result and time used by different methods.

Method	Mean	Variance	Time
<i>rand</i> generated U + CLT	cell5	cell6	..
<i>rand</i> generated U + Box-Muller	cell8	cell9	..
<i>rand</i> generated U + Marsaglia Polar	cell8	cell9	..
linear congruential generator + CLT	cell8	cell9	..
linear congruential generator + Box-Muller	cell8	cell9	..
linear congruential generator + Marsaglia	cell8	cell9	..
<i>Randon</i>	cell8	cell9	..