Concepts : Probability,
Method : Generation algorithm, Sampling techniques

Random web generators

For the following websites, explain how random numbers are obtained :

- A Million Random Digits with 100,000 Normal Deviates Rand corporation
- RandomNumber.org
- Hotbits http ://www.fourmilab.ch/hotbits/
- Marsaglia's generator http ://www.stat.fsu.edu/pub/diehard/

Find other sites that produce random numbers

Random in programming languages

For your favorite programming language,
- What are the libraries for generating pseudo-random numbers ?
- How the seeds of the generators are chosen ?

For dungeon and dragons role players

You forgot your 6 favorite dices (4, 6, 8,10, 12, 20 faces). You have just some coins in your pocket.

- Could you, by flipping several times the coins, simulate a 4 faces dice ?
- Same question with the 8 faces dice.
- What is different with the 6 faces dice ? Propose a method to generate uniformly on $\{1, 2, 3\}$ from a coin.
- Deduce a method to simulate the 6, 10, 12, 20 faces dices.

Propose a generator on $\{1, \cdots, N\}$ from a random generator of bits.

Popular pages

On web servers it has been shown experimentally that hits on pages follow a Zipf's law. This law appears also in documents popularity in P2P systems, words occurrences in texts,...

Consider a web server with N pages. Pages are ranked by their popularity and let p_i be the probability of requesting page i. We have

$$p_1 \geqslant p_2 \geqslant \cdots \geqslant p_N$$

For the Zipf's law we have $p_i = \frac{1}{H_N}\frac{1}{i}$. This means that the second web page occurs approximately 1/2 as often as the first, and the third page 1/3 as often as the first, and so on. H_N is the N^{th} harmonic number :

$$H_N = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{N}$$

which could be approximated by $\log N + \gamma + o(\frac{1}{N})$ with $\gamma = 0.5772156649$ the Euler constant.

- If N is small, classical techniques could be used. But what happens when N is large (10.000 or 100.000) ?

Probability and simulation

- Propose an algorithm that generates, approximatively, with the Zipf's law from a generator of real numbers on $[0, 1[$.
- Generalize this algorithm for "heavy-tail" laws (Benford's laws, Pareto's laws) with probability

$$p_i = \frac{1}{H_{N,\alpha}} \frac{1}{i^\alpha},$$

with α some "sharpness" coefficient and the normalization coefficient $H_{N,\alpha} = \left(\sum_1^N \frac{1}{i^\alpha}\right)^{-1}$.

General Law

Design an algorithm that simulate a pseudo-random variable according the empirical law :

i	1	2	3	4	5	6	7	8
$\mathbb{P}(X = i)$	0.10	0.20	0.05	0.05	0.05	0.15	0.35	0.05

Compute the cost of the generation algorithm.

Binomial law

Propose several algorithms that simulate a variable following the binomial distribution $\mathcal{B}in(n, k)$

$$\mathbb{P}(X = k) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k}$$

Use the fact that the binomial law could be expressed form a sequence of n flips of a coin.

Geometric distribution

Propose several algorithms that simulate a variable following the geometric distribution $\mathcal{G}(p)$

$$\mathbb{P}(X = k) = (1-p)p^{k-1}$$

Hint : compute the law of the first Head in successive coin flips.

Configuration of philosophers

In the dining philosopher's problem, a configuration is said to be admissible if two neighbors are not simultaneously eating. Suppose that the N philosophers are one one line.

Compute the recurrence equation verified by the F_N total number of philosopher's configuration.

Deduce an algorithm that generates uniformly a configuration on N philosophers on the line.

What is modified if philosophers are at a round table ?

Poisson process

Poisson processes are usually used to model arrivals on a system. Such a process is characterized by its inter-arrival time that follows an exponential distribution.

Write a program that simulates the occurrence time of a Poisson process.

Libraries

What are the libraries in C, C++, Java,.. that generates pseudo-random variables according classical distributions ?