On Simulation of Multidimensional Random Points

Jinliang Wang[a,*], and Jiangfeng Li

Science Department of Jiujiang University, Jiujiang 332005, China.

[a]wjlwyk@126.com

*Corresponding author

Keywords: Multidimensional random variable; Generating random numbers; Conditional distribution; Matlab software

Abstract: According to a distribution function of a random variable, generating one dimensional random number is very popular in numerical simulation. But in many cases, we need produce multidimensional random points that are rarely discussed in references. Here in our paper, by the application of conditional distribution for a random vector, we obtain a method that has wide applications in obtaining multidimensional random points according to some given random vector distributions.

1. Introduction

Numerical simulation has wide applications nowadays. On how to generate random numbers according to a random distribution, there are mainly two types of generators that are respectively named as Physical true-random number generators and Mathematical Pseudo-random number generators. Here we focus on the latter one. For relevant studies, we can have references such as (1) and (2). As these generators only produce one dimensional random number, we will give a method which can be applied in many conditions to generate multidimensional random points according to some given probability distributions for random vectors.

2. Review the common traditional method for generating random numbers of one dimension

As we see in almost all kinds of popular calculus software, generating one dimensional random number is the basic content. Normally a simple coding is enough for that work. For example, to generate 18 random numbers according to the uniform distribution u [0,1] by the utility of Matlab software, we only need type the command ‘unifrnd (0,1,2,9)’ to derive 18 numbers in the form of a 2-by-9 matrix

ans =

0.7922 0.6557 0.8491 0.6787 0.7431 0.6555 0.7060 0.2769 0.0971

0.9595 0.0357 0.9340 0.7577 0.3922 0.1712 0.0318 0.0462 0.8235.

Now we let ξ be a random variable uniformly distributed over the interval [0,1]. As is well known, if a random variable has a probability distribution function $\mathrm{F}(x)$ with an inverse function $\mathrm{G}(x)$, then the function $\mathrm{F}(x)$ is the right distribution function of the random variable $\mathrm{G}(\xi)$. For example, as the function $\mathrm{G}(x) = \ln\left(\frac{1}{1-x}\right)$ is the inverse function of $\mathrm{F}(x) = 1 - exp(-x)$ where $\mathrm{x} \geq 0$ (the distribution function of an exponential distribution $\mathrm{Exp}(1)$ with a parameter ‘1’), we see that the random variable $\ln\left(\frac{1}{1-\xi}\right)$ distributes according to $\mathrm{Exp}(1)$. In other words, let ξ stands for the 18 numbers obtained as above mentioned, then the 18 random numbers yielded according to $\ln\left(\frac{1}{1-\xi}\right)$ are the corresponding simulated random numbers according to the distribution of $\mathrm{Exp}(1)$.

3. Discussion on one method for generating multidimensional random numbers

As the discussions of simulation of a multidimensional random vector are rarely found in all the references we could find, here we make some explorations about that.

For a two-dimensional population (X, Y) where X and Y are independent, we can easily generate an observation of a random sample with size ‘n’. As a matter of fact, we firstly obtain ‘2n’ numbers x1, xn, y1, yn where x1, xn are the simulated numbers according to the marginal distribution of X while y1, yn, according to the marginal distribution of Y. Then the number pairs ((x1, y1), (xn, yn)) form a simulated observation of a random sample ((X1, Y1), (Xn, Yn)).

For a two-dimensional population (X, Y) where X and Y are dependent according to a specified distribution, say

F(x,y), as F(x,y)=Fx(x)×FY|X(y|x), namely the joint distribution F(x,y) can be rewritten as the product of a marginal distribution Fx(x) and a conditional distribution FY|X(y|x), we firstly generate ‘n’ numbers x1, xn according to the marginal distribution Fx(x) of X, and then for each given X=xi, we generate one corresponding simulate number yi according to the conditional distribution FY|X(y|xi). the number pairs ((x1, y1), (xn, yn)) form a simulated observation of a random sample ((X1, Y1), (Xn, Yn)) arising from the population (X, Y).

As we can understand, the same procedure can be normally extended to generate any n-dimensional random numbers according to an explicit distribution. That gives an answer to the question on how to generate multidimensional random points. To see that mentioned question, we can refer to the following network:

https://stackoverflow.com/questions/2969593/generate-n-dimensional-random-numbers-in-python
.

4. An application example in generating 2-dimentional random numbers

Let (X, Y) be a random population distributed uniformly over a circle surface $x^2+y^2 \leq 1$, generate a sample with sample size 100 by Matlab software.

First, we generate 100 random numbers according to the uniform distribution u [0,1] by coding

U=unifrnd (0,1,1,100);

As the marginal distribution F(x) of X can be easily figured out as

$$F_X(x)=\begin{cases} 0 & x<-1 \\ \iint\limits_{\substack{s^2+t^2\leq 1\\ -1\leq s\leq x}} 1/\pi\, ds dt & -1\leq x\leq 1 \\ 1 & x>1 \end{cases}$$

$$=\begin{cases} 0 & x<-1 \\ \int_{-1}^{x} 2\sqrt{1-s^2}ds & -1\leq x\leq 1 \\ 1 & x>1 \end{cases}$$

$$=\begin{cases} 0 & x<-1 \\ \frac{1}{2\pi}\left(2x\sqrt{1-x^2}+2\arcsin(x)+\pi\right) & -1\leq x\leq 1 \\ 1 & x>1 \end{cases}.$$

We then simulate 100 numbers of X by calculating the values of the inverse function $G(x)$ of FX(x) by coding

```
rand('seed',77);
n=100;
syms x
rand('seed',77);
```

U=unifrnd (0, 1, 1, n); %to generate ‘n’ numbers distributed uniformly over the interval [0,1] and form a matrix U.

Now noting that Y is uniformly distributed over the interval [-$\sqrt{1-x^2}, -\sqrt{1-x^2}$], for each given X=xi we simulate corresponding yi, i=1, 2, n. Moreover, to see the result intuitively, we draw in the figure 1.1 the unit circle as well as the 100 simulated random points.

for i=1: n

X=solve ((1/2) *(2*x*sqrt(-x^2+1) +2*asin(x)+pi)/pi-U (1, i));

Y=unifrnd(-sqrt(1-X^2), sqrt(1-X^2),1,1);

plot (X, Y,'*'); % we draw each random point (xi, yi), i=1, 2, n

hold on % to see the efficiency, we draw the unit circle to see if the simulated points are uniformly distributed

end

theta=0: pi/100:2*pi;

plot(cos(theta), sin(theta),'r')

axis equal

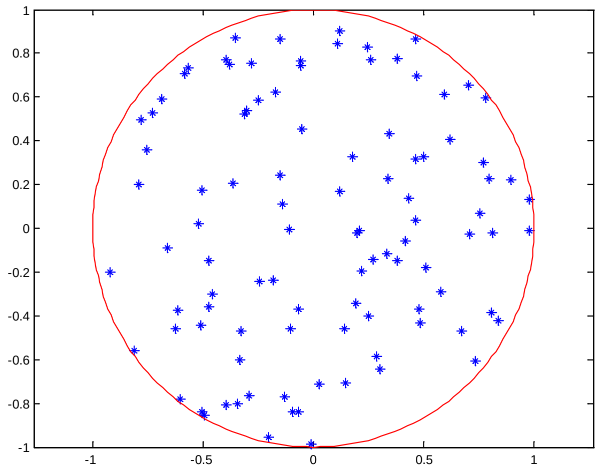

Figure 1. The result of 100 points simulated uniformly distributed over a unit circle surface

Just as is shown in the above figure 1, the simulation effect seems desirable.

Admittedly, there are other ways to generate random points that are uniformly distributed over a circle surface, see reference (3) for an example, but the method presented in reference (3) is based on a transformation that results in two one-dimensional independent random distributions. It is obvious that the generating method presented here in our paper has wider applications.

References

[1] Koeune, François. "Pseudorandom Number Generator. " Stream Ciphers 36.1(2009):57-67.

[2] Marsaglia, George, A. Zaman, and W. Wan Tsang. "Toward a universal random number generator."

[3] Statistics & Probability Letters 9.1(1990):35-39.

[4] Li Xu-dong, ZHAO Xue-jiao. “Theorems of uniformly distributed stochastic points in rectangle and ellipse and their application.” Journal of Chengdu University of Technology (Science and Technology Edition). 39.5 (2012) 555-558.