Sixteenth International Conference
Financial and Actuarial Mathematics – FAM-2023

Monte Carlo simulation of Switch-Time distribution

Pavel Stoynov
pavel_todorov_stoynov@yahoo.com

Abstract.

The linear combination of Cumulative Distribution Functions (CDF) or Probability Density Function (PDF) with positive weights summing up to one is again a CDF, or PDF respectively. This fact can be used for expressing a probability distribution function in a simpler form and is the basis of the composition method for Monte Carlo simulation of random numbers from some mixed distributions.

In this article, we consider the composition method for Monte Carlo simulation of random numbers from Switch-Time (ST) distribution.

Keywords: switch-time distribution; Monte Carlo simulation; composition method.

Introduction

The Monte Carlo technique (Metropolis, 1953; Hastings, 1970) is based on repeated generation of random numbers (Campolietti, 2014; Voitishek, 2006).

The linear combination of Cumulative Distribution Functions (CDF) or Probability Density Function (PDF) with positive weights summing up to one is again a CDF, or PDF respectively. This fact can be used for expressing a probability distribution function in a simpler form and is the basis of composition method for Monte Carlo simulation of random numbers from some mixed distributions (Campolietti, 2014).

In this article, we consider the composition method for Monte Carlo simulation of random numbers from Switch-Time (ST) distribution (Stoynov 2016, 2019).

To code the algorithms, we use the programming language R (Show, 2017; Kabacoff, 2015).

The main approach

Let ξ be a random variable with PDF $f(x)$ and let $f(x)$ can be presented as a linear combination of m other density functions $f_1(x), f_2(x), \dots, f_m(x)$. So, we have $f(x) = \sum_{j=1}^{m} w_j f_j(x), x \in R,$ where $w_j > 0,$ for every $j = 1,2,\dots,m$ and $\sum_{j=1}^{m} w_j = 1.$

In this case, the PDF $f(x)$ is called mixture PDF. The support of $f(x)$ is a union of the supports of $f_1(x), f_2(x), \dots, f_m(x).$ If the intersection of any two supports of $f_j(x)$ is empty, the mixture is also called stratification.

To simulate the random variable ξ with a mixture PDF, we use the following Composition Sampling Method (Shaw, 2017):

1. As input, we take w_j, $j = 1,2,\dots,m$ and f_j, $j = 1,2,\dots,m.$
2. We generate D from the probabilities $P(D = k) = w_j$, $j = 1,2,\dots,m.$
3. We generate ξ from the PDF $f_D(x).$
4. We return $\xi.$

Switch-time distributions

Switch-Time (ST) distributions are presented in details by Stoynov (2016, 2019). Different aspects of ST distributions were considered in Stoynov (2019, 2022). Here we just remember some basic facts about these distributions.

The random variable ξ with ST distribution has probability density function

$$f_{\xi}(x) = \begin{cases} C(n,\beta)e^{-\beta x}(1+x)^n, \; x \geq 0 \\ 0, x < 0, \end{cases} \tag{1}$$

where $C(n,\beta)$ are normalizing coefficients for which

$$C(n,\beta) = \frac{1}{I(n,\beta)} \tag{2}$$

and

$$I(n,\beta) = \frac{1}{\Gamma(1)}\int_0^{\infty} e^{-\beta t}(1+t)^n \, dt = \\ = \int_0^{\infty} e^{-\beta t}(1+t)^n \, dt \tag{3}$$

Monte Carlo simulation of random variable having ST distribution is based on the following theorem presented in (Metropolis et al., 1959):

Theorem 1. The probability density function $f_{\xi}(x)$ for $x \geq 0$ of the random variable $\xi \in ST(n,\beta)$ can be presented as a mixture of Erlang distributions:

$$f_{\xi}(x) = \sum_{k=1}^{n+1} P(D^n = k) f_{G^k}(x), \tag{4}$$

where D^n are random variables with parameter $n,$ taking values $k = 1,\ldots,(n+1)$ with probabilities $P(D^n = k) = \dfrac{C(n,\beta)n!}{\beta^k (n-k+1)!}, k = 1,\ldots,(n+1),$ and G^k is a random variable with parameter $k,$ having distribution $G^k \in \Gamma\left(k, \dfrac{1}{\beta}\right) \equiv Erlang\left(k, \dfrac{1}{\beta}\right).$

Proof:

We have:

$$C(n,\beta)e^{-\beta x}(1+x)^n = \\ = C(n,\beta)e^{-\beta x}\sum_{k=0}^{n}\binom{n}{k}x^k = \\ = \sum_{k=0}^{n} C(n,\beta)\binom{n}{k}x^k e^{-\beta x} = \\ = \sum_{k=0}^{n} \frac{C(n,\beta)n!}{\beta^{k+1}(n-k)!}\frac{\beta^{k+1}x^k e^{-\beta x}}{k!} =$$

$$
\begin{aligned}
&= \sum_{k=0}^{n} P(D^n = k+1)\, f_{G^{k+1}}(x) = \\
&= \sum_{k=1}^{n+1} \frac{C(n,\beta)n!}{\beta^k (n-k+1)!} \frac{\beta^k x^{k-1} e^{-\beta x}}{(k-1)!} = \\
&= \sum_{k=1}^{n+1} P(D^n = k)\, f_{\xi}(x \mid D^n = k) = \\
&= \sum_{k=1}^{n+1} P(D^n = k)\, f_{G^k}(x).
\end{aligned} \tag{5}
$$

The result of this theorem we use in the next section.

Application of composition sampling method to Switch-time distribution

To simulate ST distribution using the Composition Sampling Method (Campolietti and Makarov, 2014), we use the result of Theorem 1. For $\xi \in ST(n,\beta)$ we apply the presentation in (4).

Applying this approach, we derive the following Monte Carlo algorithm for simulating ST distribution:

1. As input information we have: the parameters of the distribution n and β.

2. We repeat the following iteration:

2.1. We generate a value $k \in \{1,\ldots,(n+1)\}$ of the random variable D^n using probabilities $P(D^n = k) = \dfrac{C(n,\beta)n!}{\beta^k (n-k+1)!}, k = 1,\ldots,(n+1).$

2.2. We generate a value x of the random variable G^k with parameter k, having distribution $G^k \in \Gamma\left(k, \dfrac{1}{\beta}\right) \equiv Erlang\left(k, \dfrac{1}{\beta}\right).$

2.3. We return x as value of the random variable $\xi \in ST(n,\beta).$

3. The number of iterations m can take different values.

4. Using the generated sequence, we may approximate different quantities related to ST distribution, like mean, variation, empirical distribution etc.

Numerical experiments with R

As numerical examples, we present simulations of ST(0,1), ST(1,1) and ST(2,1) distributions.

The R code for simulating ST(0,1) distribution with 10 iterations is presented on Figure 1.

```
N<-10
ST_n<-0
ST_beta<-1
ST_simulated<-1:N
for (i in 1:N)
{
ST_simulated[i]<-rgamma(1,ST_n+1,1/ST_beta)
}
```

Fig. 1. R code for simulation of ST(0,1) distribution with 10 iterations.

The output is presented in Table 1.

Table 1. The simulated values from ST(0,1) distribution with 10 simulations.

Number of iteration	ST(0,1) simulated value
1	0.3259309
2	1.3749967
3	0,8733983
4	2.0158807
5	0.3631403
6	0.3314720
7	0.1879224
8	0.9025136
9	4.3076417
10	2.7448360

The R code for simulating ST(1,1) distribution with 10 iterations is presented on Figure 2. The output is presented in Table 2.

```
ST_n<-1
ST_beta<-1
#In_beta function
In_beta<-function(n,beta, recursive=TRUE)
{if (n==0) return (1/beta) else return ((1/beta)+(n/beta)*In_beta(n-1,beta))}

#Cn_beta function
Cn_beta<-function(n,beta) {1/In_beta(n,beta)}

#FDn_inverse function
#n>0
FDn_inverse<-function(y,n,beta){
Dn_prob<-1:(n+1)
for (i in 1:(n+1))
{
Dn_prob[i]<-(Cn_beta(n,beta)*factorial(n)/(beta^i*factorial(n-i+1)))
}
Prob_acum<-Dn_prob[1]+Dn_prob[2]
for (k in 1:n)
{
if (y< Prob_acum) return (k)
else
Prob_acum<-Prob_acum+Dn_prob[k+2]
}
return(n+1)
}
Dn_prob<-1:(ST_n+1)
for (i in 1:(ST_n+1))
{
Dn_prob[i]<-(Cn_beta(ST_n,ST_beta)*factorial(ST_n)/(ST_beta^i*factorial(ST_n-i+1)))
}
ST_simulated<-1:N
Dn_simulated<-1:N
for (i in 1:N)
```

```
{
St_unif_value<- runif(1, min = 0, max = 1)
Dn_simulated[i]<-FDn_inverse(St_unif_value,ST_n,ST_beta)
ST_simulated[i]<-rgamma(1,Dn_simulated[i],1/ST_beta)
}
```

Fig. 2 R code for simulation of ST(1,1) distribution with 10 iterations.

Table 2. The simulated values from ST(1,1) distribution with 10 simulations.

Number of iteration	ST(1,1) simulated value
1	0.9658339
2	0.2706930
3	0.4928619
4	1.3158975
5	0.2554345
6	0.9130880
7	0.3471026
8	0.1748429
9	1.5303544
10	0.2364177

```
ST_n<-2
ST_beta<-1
#In_beta function
In_beta<-function(n,beta, recursive=TRUE)
{if (n==0) return (1/beta) else return ((1/beta)+(n/beta)*In_beta(n-1,beta))}

#Cn_beta function
Cn_beta<-function(n,beta) {1/In_beta(n,beta)}

#FDn_inverse function
#n>0
FDn_inverse<-function(y,n,beta){
Dn_prob<-1:(n+1)
for (i in 1:(n+1))
{
Dn_prob[i]<-(Cn_beta(n,beta)*factorial(n)/(beta^i*factorial(n-i+1)))
}
Prob_acum<-Dn_prob[1]+Dn_prob[2]
for (k in 1:n)
{
if (y< Prob_acum) return (k)
else
Prob_acum<-Prob_acum+Dn_prob[k+2]
}
return(n+1)
}
Dn_prob<-1:(ST_n+1)
for (i in 1:(ST_n+1))
```

```
{
Dn_prob[i]<-(Cn_beta(ST_n,ST_beta)*factorial(ST_n)/(ST_beta^i*factorial(ST_n-i+1)))
}
ST_simulated<-1:N
Dn_simulated<-1:N
for (i in 1:N)
{
St_unif_value<- runif(1, min = 0, max = 1)
Dn_simulated[i]<-FDn_inverse(St_unif_value,ST_n,ST_beta)
ST_simulated[i]<-rgamma(1,Dn_simulated[i],1/ST_beta)
}
```

Fig. 3 R code for simulation of ST(2,1) distribution with 10 iterations.

The R code for simulating ST(2,1) distribution with 10 iterations is presented on Figure 3.
The output is presented in Table 3.
Here we provide some additional remark related to Theorem 1.

One corollary of Theorem 1 is that $\xi \mid D^n \equiv Erlang\left(D^n, \frac{1}{\beta}\right)$.

Also, we could consider the variables $\tilde{D}^n = D^n - 1$, taking values $k = 0, \dots, n$ with probabilities $P(\tilde{D}^n = k) = \frac{C(n,\beta)n!}{\beta^{k+1}(n-k)!}, k = 0, \dots, n.$

In the special case when of ST distribution of type ST(1,b) which is actually Lindley distribution with parameter b, we have the presentation

$$\begin{aligned} f_\xi(x) &= \frac{\beta^2}{\beta+1} e^{-\beta x}(1+x) = \\ &= \frac{\beta}{\beta+1}\beta e^{-\beta x} + \frac{1}{\beta+1}\beta^2 x e^{-\beta x} = \\ &= p\beta e^{-\beta x} + (1-p)\beta^2 x e^{-\beta x}, \end{aligned} \tag{6}$$

where $p = \frac{\beta}{\beta+1}$.

Table 3. The simulated values from ST(2,1) distribution with 10 simulations

Number of iteration	ST(2,1) simulated value
1	0.1260225
2	3.1695044
3	0.5714216
4	2.0577410
5	1.1900808
6	0.5791173
7	1.5522574
8	1.4666762
9	0.3122641
10	0.7756049

$$f_\xi(x) = \frac{\beta^2}{\beta+1} e^{-\beta x}(1+x) = \\ = \frac{\beta}{\beta+1}\beta e^{-\beta x} + \frac{1}{\beta+1}\beta^2 x e^{-\beta x} = \\ = p\beta e^{-\beta x} + (1-p)\beta^2 x e^{-\beta x}, \tag{6}$$

where $p = \dfrac{\beta}{\beta+1}$.

This presentation is actually discrete mixing of exponential distribution and Erlang distribution of type $Erlang(2,\beta)$.

To remember that the Erlang distribution has probability density function $f_\xi(x;k,\beta) = \dfrac{\beta^k x^{k-1} e^{-\beta x}}{(k-1)!}$.

In the special case when of ST distribution of type ST(2,b) the probability density function is

$$f_\xi(x) = \begin{cases} \dfrac{\beta^3}{\beta^2+2\beta+2} e^{-\beta x}(1+x)^2 = \\ = C(\beta)e^{-\beta x}(1+x)^2,\ x \ge 0 \\ 0,\ x<0. \end{cases} \tag{7}$$

This is actually weighted exponential function with weighting function $(1+x)^2$. This probability density function is close to the probability density function of Lindley distribution, where the weighting function is $w(x) = 1+x$. Also, this probability density function is close to the probability density of gamma distribution where the weighting function is $w(x) = x^{\alpha-1}$.

Conclusion

Applying the composition sampling method for simulation of ST distribution is an effecting approach for obtaining samples of this kind of distribution.

We demonstrated this for several types of ST distribution: with parameters $b=1$ and $n=0,1,2$ correspondingly.

References

Campolietti, G. and R. Makarov (2014) Financial Mathematics. A comprehensive treatment. CRC Press: Taylor & Francis Group. ISBN 978-1-4398-9242-8.

Hastings, W. (1970) Monte Carlo Sampling Methods Using Markov Chains and their Applications," in Biometrica, vol. 57(1), 1970, pp. 97-109, doi: 10.1093 /biomet/57.1.97.

Kabacoff, R. (2015) R in Action. Manning Publications Co. New York.2015. ISDN: 9781617291388.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller (1953) Equations of State Calculations by Fast Computing Machines- Journal of Chemical Physics, vol. 21(6), 1953, pp. 1087-1092.

Shaw, B. (2017) Uncertainty Analysis of Experimental Data with R. CRC Press: Taylor & Francis Group. ISBN 978-1-4987-9732-0.

Stoynov, P. (2016) Switch Time Family of distributions and processes and their applications to reflected surplus models - Annual of the Faculty of Economics and Business Administration, Sofia University "St. Kliment Ohridski", 2016, pp. 255-285.

Stoynov, P. (2019) Switch-time distributions and processes and their applications in finance and insurance - Pliska Studia Mathematica, vol. 30, 2019, pp. 201-214.

Stoynov, P. (2022) Applications of ST Distributions to Neural Networks and Regression Models - Proceedings of Eighth International Conference on New Trends in the Applications of Differential Equations in

Science (NTADES’21), 6-9 September, St. Constantine and Helena, Bulgaria, On-line: AIP Conference Proceedings 2459, 030036 (2022); https://doi.org/10.1063/5.0084006. Published Online: 05 April 2022.

Stoynov, P. (2022) Time Activation Function and Stopit Regression - Some Examples - XXXI International Scientific Conference Electronics (ET), 2022, pp. 1-6, doi: 10.1109 /ET55967.2022.9920205.

Voitishek, V. and G. Mihailov (2006) Numerical statistical modeling: Monte Carlo methods. Akademiya. Moscow. 2006.