Advanced Monte Carlo Methods

Emmanuel Odujebe

May 2024

Abstract

In this paper, I will discuss the methodology and usage of advanced Monte Carlo methods in finance. In the first section of this paper, I will briefly discuss the mathematics behind Monte Carlo simulation, using binomial probability estimation as an example. In the second section of this paper, I will introduce two Monte Carlo variations (antithetic and importance sampling) and explain their effectiveness in error reduction. In the third section of this paper, I will introduce Monte Carlo simulation in a financial context. By estimating the call price of a European option using all three methods, I will evaluate the efficiency of each estimator.

1 Introduction

Monte Carlo simulation is a mathematical technique developed during World War Two by John von Neumann and Stanislaw Ulam. Named after Monaco's gambling region with the same name, the method was created whilst von Neumann and Ulam were working on nuclear weapons at the Los Alamos National Library[1].

Monte Carlo simulation is especially effective when dealing with problems that don't have practical analytical solutions or a large number of random variables. When applied in a financial context, it can be used to forecast and analyse the variability of a portfolio or specific asset's performance over time, which can help to influence better decision making.

2 Basic Monte Carlo Simulation

A basic Monte Carlo simulation plan follows two stages. The first step is to generate independent identically distributed random variables $X_1, X_2, \ldots, X_n$ with the same distribution as X. Let X be a random variable such that $\mu = E[X]$. Then our Monte Carlo estimate is given by

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i. \tag{1}$$

This comes from the Law of Large Numbers, which states that for a sequence of independent, identically distributed random variables $X_i, i = 1, 2, ...,$ then

$$\frac{X_1 + X_2 + ... + X_n}{n} \xrightarrow[n\to\infty]{} \mu_X. \tag{2}$$

Monte Carlo Simulation was first used to aid in calculating neutron diffusion paths in nuclear bombs [2]. However, it can be used for other purposes, such as estimating probabilities within continuous distributions. For example, using a standard normal distribution $Z \sim N(0, 1)$, we can estimate the value of $P(-1.96 \leq Z \leq 1.96)$ using Monte Carlo Simulation, and R.

```r
#basic normal estimation
set.seed(234)
runs <- 1000000
simulation <- rnorm(runs, mean=0, sd=1)
mc_estimate <- sum(simulation >= -1.95 & simulation <= 1.95)/runs

mc_estimate
```

```
## [1] 0.948835
```

```r
#we are really close to the actual probability, 0.95
```

Figure 1: Using R and 1,000,000 simulations, an estimate for the standard normal distribution can be obtained.

The standard error of $\hat{\mu}$ generated by Monte Carlo Simulation is given by

$$SE = \sqrt{\frac{1}{n(n-1)}\left(\sum_{i=1}^{n} X_i^2 - n\hat{\mu}^2\right)}. \tag{3}$$

3 Advanced Monte Carlo Methods

Generally, the smaller an estimate's variance, the more efficient it is as an estimator. $\hat{\mu}$, the Monte Carlo estimate, is a consistent estimator of μ:

$$\lim_{n\to\infty} Var[\hat{\mu}] = \lim_{n\to\infty}\left(\frac{1}{n(n-1)}\left(\sum_{i=1}^{n} X_i^2 - n\hat{\mu}^2\right)\right) = 0. \tag{4}$$

However, there are variations of the Monte Carlo simulation method that reduce variance. One of such variations involves antithetic sampling.

Whilst in simple Monte Carlo simulation, samples are all independent and identically distributed, antithetic samples incorporate samples with negative correlations. In an antithetic sample, there are n pairs of samples. Pairs (X_i, Y_i) are independent and identically distributed and Y_i has the same distribution as X_i, but X_i and Y_i are dependent. The estimate is now

$$\hat{v} = \frac{1}{n}\sum_{i=1}^{n}\frac{X_i + Y_i}{2}. \tag{5}$$

The variance of an antithetic sampling estimate $\hat{v}$ with n pairs of 2 samples is given by [3]:

$$
\begin{aligned}
Var[\hat{v}] &= \frac{1}{4n^2}\sum_{i=1}^{n} Var\left[X_i + Y_i\right] \\
&= \frac{1}{4n^2}\sum_{i=1}^{n} \left(Var[X_i] + Var[Y_i] + 2Cov(X_i, Y_i)\right) \\
&= \frac{1}{4n^2}\sum_{i=1}^{n} \sigma^2\left(2+\beta\right) \\
&= \frac{\sigma^2(1+\beta)}{2n},
\end{aligned}
\tag{6}
$$

where β is the correlation coefficient between X_i and Y_i.

Since the variance of a simple Monte Carlo estimate for $2n$ samples is given by

$$
Var[\hat{\mu}] = \frac{\sigma^2}{2n}, \tag{7}
$$

it can be seen that when $\beta < 0$, antithetic sampling produces less variance compared to a simple Monte Carlo simulation. The greater the magnitude of β, the greater the variance reduction. For a given set of samples $X = X_1, ..., X_n$, we can generate an antithetic sample.

Firstly, we can write $X = F^{-1}(U)$, where $U \sim Unif(0,1)$ and F^{-1} is the inverse of X's cumulative distribution function. Then, we can define an antithetic sample of X as $Y = F^{-1}(1-U)$. X and Y are negatively correlated since X is an increasing function of U and Y is a decreasing function of U. X and Y are identically distributed since we also have $(1-U) \sim Unif(0,1)$.

If F^{-1} isn't easily found, then an antithetic sample of $X = h(U)$ can be defined as $Y = h(1-U)$. X and Y are negatively correlated when the function h is monotone.

Another commonly used technique in Monte Carlo simulation is importance sampling. For a density function $g(x)$, we can write

$$\begin{aligned}\mu = \int_{\mathbf{R}} h(x)f(x)dx &= \int_{\mathbf{R}} h(x)\frac{f(x)}{g(x)}g(x) \\ &= E\left[h(Y)\frac{f(Y)}{g(Y)}\right],\end{aligned} \tag{8}$$

where Y is a random variable with a probability density function of g. Since generated samples are now coming from a different probability distribution compared to where they would come from under simple Monte Carlo simulation, each outcome $h(Y_i)$ is weighed by a likelihood ratio $\frac{f(Y_i)}{g(Y_i)}$ to prevent bias. Then, we can take independent and identically distributed samples from g to get a Monte Carlo estimate of

$$\hat{\mu} = h(Y_i)\frac{f(Y_i)}{g(Y_i)}. \tag{9}$$

An important step in importance sampling is choosing an alternative distribution to sample from. A distribution should be chosen as to minimise the variance of the generated estimate. The variance of the importance sampling estimate is given by [3]:

$$Var\left[\frac{1}{n}\sum_{i=1}^{n} h(Y_i)\frac{f(Y_i)}{g(Y_i)}\right] = \frac{1}{n}Var\left[h(Y)\frac{f(Y)}{g(Y)}\right]. \tag{10}$$

4 Applications to Finance

One of many applications of Monte Carlo simulation within finance is in option pricing. Consider an European option with a strike price of \$100, a risk-free interest rate of 2%, volatility of 20% and a current price of \$102, that matures in half a year. Then we can calculate the theoretical value of a call option C using the Black-Scholes formula:

$$C = S_0 \cdot N(d_1) - PV(K) \cdot N(d_2), \tag{11}$$

where

$$d_1 = \frac{1}{\sigma\sqrt{T}}\left[\log\left(\frac{S}{K}\right) + T\left(r + \frac{\sigma^2}{2}\right)\right], \tag{12}$$

$$d_2 = d_1 - \sigma\sqrt{T}, \tag{13}$$

$$PV(K) = Ke^{-rT}, \tag{14}$$

and $N(d) \sim N(0,1)$. In this case, we have

$$d_1 = \frac{1}{0.2 \cdot \sqrt{0.5}}\left[\log\left(\frac{102}{100}\right) + 0.5\left(0.02 + \frac{0.04}{2}\right)\right] = 0.281; \tag{15}$$

$$d_2 = 0.281 - 0.2\sqrt{0.5} = 0.140; \tag{16}$$

$$PV(K) = 100e^{-0.01} = 99.0; \tag{17}$$

giving

$$C = 100 \cdot N(0.281) - 99.0 \cdot N(0.140) = 7.29. \tag{18}$$

We can now estimate this call price by firstly using a simple Monte Carlo simulation. At a given time T, the asset price is given by [4]

$$S_T = S_0 e^{(r-\frac{\sigma^2}{2})T+\sigma W_T}, \tag{19}$$

where $W_T \sim N(0,T)$. The payoff of this call option is equal to $(S_T - K)_+$. Now, we can run a simple Monte Carlo simulation to estimate the call price and standard error.

```r
#setting seed
set.seed(129)

#setting variables
K = 100
r = 0.02
sigma = 0.2
T = 0.5
S0 = 102

#call price using Black-Scholes formula
d1 <- (log(S0/K) + (r + sigma^2/2) * T)/(sigma * sqrt(T))
d2 <- d1 - sigma * sqrt(T)
call_price <- S0 * pnorm(d1) - K * exp(-r * T) * pnorm(d2)
call_price
```

```
## [1] 7.288151
```

```r
#Simple Monte Carlo Simulation

call_mc <- function(sims, tau, r, sigma, s0, K){

  #generating samples from the standard normal distribution
  Z <- rnorm(sims, mean=0, sd=1)
  #calculating W_T and S_T
  W_T <- Z*sqrt(tau)
  S_T <- S0*exp((r-(sigma^2)/2)*tau+sigma*W_T)
  #calculating call price for the sample
  simmed_calls <- exp(-r*tau)*pmax(S_T-K,0)
  #calculating simple monte carlo estimate and std. error
  mc_price <- mean(simmed_calls)
  mc_se <- sd(simmed_calls)/sqrt(sims)
  #returns the estimated call price and its standard error
  return(list(call_price = mc_price, standard_error = mc_se))
}

#the estimate
estimate <- call_mc(sims = 1000000, tau = 0.5, r=0.02, sigma = 0.2, s0 = 102, K = 100)
estimate
```

```
## $call_price
## [1] 7.289501
##
## $standard_error
## [1] 0.01012741
```

Figure 2: Estimating European call prices using a simple Monte Carlo simulation.

In this case, the Monte Carlo estimated call price is very close to the theoretical value calculated using Black-Scholes, with a low standard error. This suggests that even using a simple Monte Carlo simulation can be very effective in estimating call prices. Now, we can compare this to estimated values generated through antithetic sampling and importance sampling.

For our antithetic sampling method, we can use two samples with negative correlations. In this case, we can use $S_{T_1} = S_0 e^{(r-\frac{\sigma^2}{2})T+\sigma W_T}$ and $S_{T_2} = S_0 e^{(r-\frac{\sigma^2}{2})T-\sigma W_T}$. We can find simulated values for both samples and average them to find our estimated call price.

```
#Antithetic Sampling
call_as <- function(sims, tau, r, sigma, S0, K){

  #generating samples from the standard normal distribution
  Z <- rnorm(sims, mean=0, sd=1)
  #calculating W_T
  W_T <- Z*sqrt(tau)
  #generating antithetic samples S_T1 and S_T2
  S_T1 <- S0*exp((r-(sigma^2)/2)*tau+sigma*W_T)
  S_T2 <- S0*exp((r-(sigma^2)/2)*tau+sigma*(-W_T))
  #calculating call prices for both samples
  simmed_calls1 <- exp(-r*tau)*pmax(S_T1-K,0)
  simmed_calls2 <- exp(-r*tau)*pmax(S_T2-K,0)
  #obtaining the average of both samples
  avg_calls <- (simmed_calls1 + simmed_calls2)/2
  #calculating the estimated call price and std. error
  as_price <- mean(avg_calls)
  as_se <- sd(avg_calls)/sqrt(sims)
  #returns the estimated call price and its standard error
  return(list(call_price = as_price, standard_error = as_se))
}
set.seed(129)
estimate_as <- call_as(sims = 1000000, tau = 0.5, r=0.02, sigma = 0.2, S0 = 102, K = 100)
estimate_as
```

```
## $call_price
## [1] 7.287807
##
## $standard_error
## [1] 0.004993132
```

Figure 3: Estimating call prices using antithetic sampling.

Using antithetic sampling gives us a call price that is now slightly under the theoretical Black-Scholes price. However, the standard error produced is less than half of the error produced by the simple Monte Carlo simulation, suggesting that our antithetic sampling estimate is more consistent.

```r
#antithetic sampling method with half of the simulations
set.seed(129)
estimate_as2 <- call_as(sims = 1000000/2, tau = 0.5, r=0.02, sigma = 0.2, S0 = 102, K = 100)
estimate_as2
```

```
## $call_price
## [1] 7.276491
##
## $standard_error
## [1] 0.007056377
```

Figure 4: Antithetic sampling with half the number of simulations.

When using antithetic sampling with 1,000,000 simulations, it creates 2,000,000 values, one for each negatively correlated sample. This naturally leads to a smaller standard error. However, even when simulating 500,000 times to create an equal amount of values as our simple Monte Carlo simulation, the standard error is still smaller.

For our importance sampling method, we can generate an alternative sampling distribution by using an indicator function [4] with:

$$f(x) = \begin{cases} 1 & S_T > K \\ 0 & S_T \leq K \end{cases} \tag{20}$$

```r
call_is <- function(sims, tau, r, sigma, S0, K){

  #generating samples from the standard normal distribution
  Z <- rnorm(sims, mean=0, sd=1)
  #calculating W_T and S_T
  W_T <- Z*sqrt(tau)
  S_T <- S0*exp((r-(sigma^2)/2)*tau + sigma*W_T)
  #calculating call price
  simmed_calls <- (exp(-r*tau)*pmax(S_T-K,0))[S_T>K]
  #calculating the estimated call price and std. error
  is_price <- mean(simmed_calls*mean(S_T>K))
  is_se <- sd(simmed_calls*mean(S_T>K))/sqrt(sims)
  #returns the estimated call price and its standard error
  return(list(call_price = is_price, standard_error = is_se))

}

set.seed(129)
estimate_is <- call_is(sims = 1000000, tau = 0.5, r = 0.02, sigma = 0.2, S0 = 102, K = 100)
estimate_is
```

```
## $call_price
## [1] 7.289501
##
## $standard_error
## [1] 0.005781881
```

Figure 5: Estimating call prices using importance sampling.

Using this alternative sampling distribution, we obtain the same call price as with our simple Monte Carlo simulation, but with a much smaller standard error. The equal call price could be down to our filter of $S_T > K$ not filtering out any of our simulations.

To compare all of our estimators, we can look at their relative efficiencies. The relative efficiency of two estimators E_1 and E_2 is given by

$$RE\left(\frac{E_1}{E_2}\right) = \frac{V(E_2)}{V(E_1)}. \tag{21}$$

In the case of our advanced Monte Carlo estimators, from antithetic sampling (P_{AS}) and importance sampling (P_{IS}), we can see that both are more efficient than our simple Monte Carlo estimator (P_{MC}):

$$
\begin{aligned}
RE\left(\frac{P_{AS}}{P_{MC}}\right) &= \frac{0.01012741}{0.004993132} = 2.028; \\
RE\left(\frac{P_{IS}}{P_{MC}}\right) &= \frac{0.01012741}{0.005781881} = 1.752.
\end{aligned}
\tag{22}
$$

In this scenario, the antithetic sampling estimator is twice as efficient as the simple Monte Carlo estimator, whilst the importance sampling estimator is 75% more efficient. Using an indicator function to generate the alternative sampling distribution led to a higher variance for the importance sampling estimator. Choosing a better distribution would have led to a much more efficient estimator.

5 Conclusion

Simple Monte Carlo simulation is very effective in estimation. In our European call pricing example, despite the variance of the estimator being greater than those found through advanced models, it is still accurate: we can construct a 95% confidence interval for the simple Monte Carlo estimator using the equation

$$CI_{95} = P_{MC} \pm z_{0.025} \cdot SE_{MC} = (7.270, 7.309). \tag{23}$$

Our theoretical value, 7.288, falls within this confidence interval, so we cannot say that this estimator is statistically different to the true value. However, when trying to forecast option prices, minimising variance is very important. Hence, using advanced Monte Carlo methods that reduce variance is helpful. Whilst antithetic and importance sampling are effective by themselves, they can potentially be used in tandem, along with other methods, to produce even better results.

References

[1] ibm.com: *What is Monte Carlo Simulation?*, https://www.ibm.com/topics/monte-carlo-simulation.

[2] O. Summerscales (2023) *Hitting the Jackpot: The Birth of the Monte Carlo Method*, discover.lanl.gov.

[3] H. Wang (2012) *Monte Carlo Simulation with Applications to Finance*, CRC Press.

[4] G. Pipis (2020) *Pricing of European Options with Monte Carlo*, r-bloggers.com.