Monte Carlo Simulation

Rupesh Poudel

07 January 2022

Outline

A little about me!

- Third Semester MEMS Student
- Accounting and Finance Major
- B.Sc. International Business and Economics
- Interest in Private Equity and Quantitative Finance

Monte Carlo Simulation

"Monte Carlo Simulation uses random samples of parameters or inputs to explore the behavior of a complex system or process." - Solver.com [1]

The essence of Monte Carlo Simulations are characterised by following properties:

1. The problem is too complex to be solved analytically.
2. The problem requires large number of experiments.
3. The underlying input for the estimate, forecast or decision comes with significant uncertainty.
4. The higher dimensional nature of problems make numerical evaluation "prohibitively" slow.

[1] https://www.solver.com/monte-carlo-simulation-overview

What was it named after?

Figure: The famous casino of Monte Carlo, Monaco[2]

[2] https://www.lonelyplanet.com/monaco/attractions/casino-de-monte-carlo/a/poi-sig/1229750/359266

Who invented it?

Figure: Ulam

Figure: Neumann

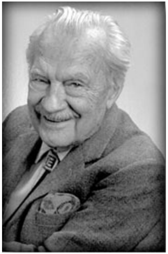

Figure: Metropolis

Stanislaw Ulam [3], John von Neumann[4], and Nicholas Metropolis[5]

See Metropolis (1987) for full story.

[3] https://en.wikipedia.org/wiki/Stanislaw_Ulam

[4] https://en.wikipedia.org/wiki/John_von_Neumann

[5] https://en.wikipedia.org/wiki/Nicholas_Metropolis

The mechanics behind Monte Carlo simulation

Metropolis (1987) outlines the methodology as follows:

1. Generate uniformly distributed pseudo random numbers.
2. These numbers must be converted to a non-uniform distribution as per the requirement of the problem that should be solved.
3. The inverse of the non uniform distribution function $f = g^{-1}$ is our desired conversion function.
4. Use this transformed random numbers to carry out instances of deterministic computations.
5. Aggregate the results of the experiments.

An example will follow to make the procedure clear.

Example: The coin toss Experiment

Setup: Let's toss a fair coin 100 times. For each heads, I will pay you 1 €, and for each tails, you will pay me 1€. **Alternatively,** the losing party may transfer only the difference to the winning party.

The experiment: This is a setup of an ordinary random walk with two nodes. A binomial process like this is represented with the following equation: *Please refer to Franke, Härdle and Hafner (2019) for details*

$$X_t = X_0 + \sum_{k=1}^{t} Z_t, \quad t = 1, 2, 3, ...$$

with $X_0 = 0$, Z_t $i.i.d$, $P(Z_k = 1) = p$ and $P(Z_k = -1) = 1 - p$

The Coin Toss Experiment

The source of uncertainty: We do not know whether the value of Z_t will be 1 or -1 after the end of each coin toss.

Where Monte Carlo comes in? Monte Carlo will make use of random numbers it generates for each trial to suggest the value for Z_t

The random numbers: We give the program a range of numbers from which it can draw random numbers. Say, i) [0, 1) ii) [0, 100).

Experimental Design: i) heads for random number greater than or equal to 0.5, tails for less ii) Round the random numbers to two digits and even is heads.

Aggregation: Use the outcomes resulting from our definition of random numbers to get the binary results for that experiment, i.e. Heads or Tails. Then, find the value of X_{100} to calculate final transfer.

Outline

Relation to Existing Literature of Methods

Statistical Sampling:

1. Statistical sampling is drawing a set of observations randomly from a population distribution. - Riffenburgh (2006).
2. Monte Carlo Methods are basically large scale statistical sampling that uses massive computational prowess.
3. "Monte Carlo method solves deterministic computation using probabilistic metaheuristic"- Wikipedia

Monte Carlo method is like current day smartphone. It has so many use case that its predecessors range from traditional integration to sensitivity analysis to scenario analysis.

Bernouli's law of Large Numbers

Let $\mathrm{X}_1, X_2, \ldots be$ i.i.d. The μ and σ^2 exists and are finite. Let $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ be the sample mean. What happens when n gets larger?

The SLLN, Strong Law of Large Numbers, states that $\bar{X}_n \to 0$ as $n \to \infty$

The sample mean converges to true(theoretical) mean, as the sample size converges to infinity.

The WLLN, Weak Law of Large Numbers,also states that sample mean will be very close to theoretical mean, albeit differently. The WLLN implies convergence in probability.

Mathematically, $P(|\bar{X}_n - \mu| > c) \to 0$ as $n \to \infty$.

Please refer to the Lecture 29 of Statistics 110 by Joe Blitzstein, Harvard University, for full explanation. [6]

[6] https://www.youtube.com/watch?v=OprNqnHsVIA

LLN- Interpretation

Let's take a fair coin with two sides. The probability of heads is p.

The probability distribution closest to the coin toss experiment is Bernouli Distribution with $X_i \sim Bernouli(p)$.

The law of large numbers implies the following:

$$\frac{X_1 + X_2 + ... + X_n}{n} \to p$$

with probability 1

Applications for finance and statistics

Monte Carlo Methods in Statistics:

1. Provide efficient random estimates for Hessian Matrix
2. Bayesian Statistics: Computing posterior quantities using sample from a posterior distribution
3. Experimental Design: Factorial Design vs Monte Carlo methods.
4. Statistical Interference: Compute quantiles of test statistic under the null hypothesis

Monte Carlo Methods in Finance:

1. Analysis of default risks
2. Evaluating investment portfolios
3. Pricing financial derivatives
4. Cash flow analysis of an uncertain project

Pros and Cons

Pros:

1. Deterministically complex problems can be solved numerically
2. For non- invertible functions, Monte Carlo can use reverse search to plot the graph of the inverse when possible.
3. Monte Carlo method is a powerful tool for solving any problem with uncertainty involved.

Cons:

1. Considerable computational time and resources are consumed.
2. GIGO: if our input or parameters are garbage, our output will be garbage too. This includes the domain of input and the underlying probability distribution.
3. Monte Carlo method does not factor in irrationality and behavior aspects of say, finance and economics very well.
4. Monte Carlo methods do not accurately predict tail events.

Outline

Random Numbers

An assumption is made: the sample drawn at random will have same properties as the population.

1. Without random sampling, any statistical estimator will be biased.
2. With (sufficiently) random sampling and (sufficiently)large number of simulations, central moments of the sample will approach central moments of the population.

However, a computer generated random number is technically pseudorandom. For our purpose, we will be using these two terms interchangeably.

Random Number Generator

Some random number generators are given below:

- Linear Congurential Generator : Intel's RANDU
- middle square digits
- Kronecker- Weyl theorem
- Mersenne- Twister algorithm
- Intel's RDRAND
- Yarrow Algorithm
- fortuna
- arc4random
- Keep it Simple Stupid

Middle Square digits

1. Pick a N digit number.Say $N = 4$ and number $= 1111$
2. Square the number. $1111^2 = 1234321$
3. Add a preceeding 0 when necessary. 01234321.
4. The middle 4 digits, 2343, is the new random number.
5. Use the middle digits to start the process anew.
6. The sequence of random number generated will repeat.

Higher value of N will prevent the sequence from repeating soon.

Mersenne Twister

Mersenne Prime: Mersenne Primes are a special set of numbers that take the form $M_n = 2^n - 1$ and are also a prime.

1. The period length of Mersenne Twister (MT) is chosen to be Mersenne Prime.
2. This offers a 623- dimensional equidistribution and up to 32 bits accuracy.
3. Variants of these methods have been invented for specific use cases. e.g. GPU optimised MTGP, cryptographically secure CryptMT, or relatively quick SFMT.
4. Mersenne Twister is a default random number generator in, among other programs, R, Python (including Numpy), and MATLAB.

Outline

Case: Modeling a stock

It all starts with a model. A simulation is only as good as model.

1. We extract n adjusted closing prices from a database.
2. We calculate $n-1$ percentage changes from today to tomorrow.
3. We calculate the standard deviation of those percentage changes.
4. We start with the latest known stock price. Say, today or yesterday.
5. We generate a random variable, **normally distributed**, with $N(0, \sigma^2)$, with σ being the result from step 3.
6. This indicates the up or down movement for the stock on that day. Take the values from the previous day, repeat step 5 and 6 for desired period of, say, 252 trading days.
7. Repeat Step 4, 5, and 6 for desired number of trials. The distribution of final day is our estimate for stock behavior in 1 year.

Case: Data Description

- We take the stock price of Apple Inc from yahoo finance.
- The start date is January 1, 2020
- The end date is November 30, 2021

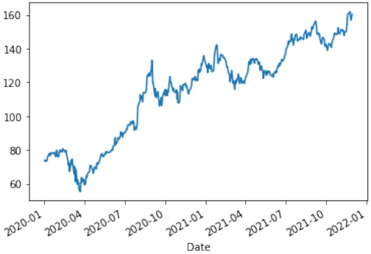

Figure: Apple Stock Price vs Time

Case : Simulation Setting

- Number of trials: 10,000
- Number of days ahead to forecast: 252 trading days.
- Random Number Generator: Numpy's np.random.normal()
- Model explicitly uses normal instead of lognormal returns

Case: Results

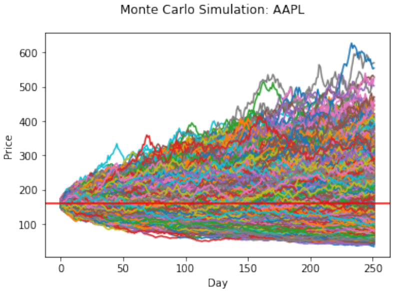

Figure: Paths

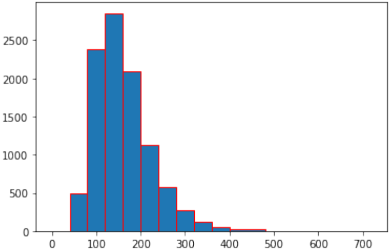

Figure: Histogram

Case: Conclusion

- max $= 690.48$
- min $= 35.02$
- $P(\text{end price} < \text{starting price}) = 0.5795$
- $P(\text{end price} < \text{starting price}(1 + r_f)) = 0.5921$ with $r_f = 0.02$
- Computation time $= 32$ seconds

References

1. https://www.solver.com/monte-carlo-simulation-overview
2. Metropolis, N. (1987). The beginning of the Monte Carlo method. *Los Alamos Science*, 15(584), 125-130.
3. Eckhardt, R., Ulam, S., & Von Neumann, J. (1987). the Monte Carlo method. *Los Alamos Science*, 15, 131.
4. Riffenburgh, R. H. (2006). Methods You Might Meet, But Not Every Day. Statistics in Medicine, 521-529.
5. Baumgärtner, A., Binder, K., Hansen, J. P., Kalos, M. H., Kehr, K., Landau, D. P., ... & Weis, J. J. (2013). Applications of the Monte Carlo method in statistical physics (Vol. 36). Springer Science Business Media.
6. Chen, M. H., Shao, Q. M., & Ibrahim, J. G. (2012). Monte Carlo methods in Bayesian computation. Springer Science Business Media.

References

7. Franke, J., Härdle, W.K., & Hafner, C.M. (2019). Statistics of Financial Markets: An Introduction. (5^{th} edition). Springer.
8. Gentle, J. E. (2009). Monte carlo methods for statistical inference. In Computational Statistics (pp. 417-433). Springer, New York, NY.
9. Van Groenendaal, W. J., & Kleijnen, J. P. (1997). On the assessment of economic risk: factorial design versus Monte Carlo methods. Reliability Engineering System Safety, 57(1), 91-102.
10. Matsumoto, M., & Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation (TOMACS), 8(1), 3-30.