On Least Square Monte Carlo Method

IEDA 4520 Project, 2023 Fall

Matthew Lau

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

November 29, 2023

Table of Contents

Table of Contents

Traditional Approaches for American Option Pricing

American Options: Options that can be exercised any time before the maturity T (VS Bermudan Options: only allow exercise at set dates).

Two popular methods for pricing American Options:

- Binomial tree: Perform backward induction at discrete time steps for options with payoff function $H(.)$ using the **Bellman Equation**

$$V^*(S_t) = max(H(S_t), e^{-r(t_{n+1}-t_n)} \cdot \mathbb{E}^Q\left[V^*(S_{t+1})|S_t\right])$$

- Finite difference method: Numerical technique to solve the partial differential equation of American Options (see Appendix 1).

Dimension in Option Pricing

Curse Of Dimensionality: Methods working very well in low-dimension become impractical in high-dimensional settings.

Dimension in options: Number of underlying securities involved.

- High dimensional example: Multi-asset American option on underlying $\{S^1, S^2, S^3\}$ with payoff $max(max(S_t^1, S_t^2, S_t^3) - K, 0)$.

Limitations of Traditional Approaches

Problems arise in traditional methods under high-dimensional settings:

- Binomial tree: Number of nodes grows **exponentially** with the number of dimension.
  Consider a n-step binomial *recombining* tree with d underlying assets. At the n^{th} step, each underlying has $(n+1)$ possible prices, and in total there are $(n+1)^d$ possible combinations of prices.
- Finite difference method: Partial differential equations for higher dimension are hard or infeasible to obtain.

Monte Carlo Methods for American Options

It is difficult to apply general Monte Carlo simulation method to price American Options.

- Deriving the **continuation value** (or **holding value**) at any time point t based on a single subsequent path does not make much sense.
- Some may attempt to apply **multiple-tier Monte Carlo simulation** to estimate the continuation value, which is infeasible for large number of discrete time steps.

Monte Carlo Methods for American Options

It is difficult to apply general Monte Carlo simulation method to price American Options.

- Deriving the **continuation value** (or **holding value**) at any time point t based on a single subsequent path does not make much sense.
- Some may attempt to apply **multiple-tier Monte Carlo simulation** to estimate the continuation value, which is infeasible for large number of discrete time steps.

The challenge lies the conditional expectation $\mathbb{E}^Q[V^*(S_{t+1})|S_t]$.
Longstaff and Schwartz came up with an idea to approximate this term.

Table of Contents

LSM ideas and inputs

LSM ideas

The LSM algorithm combines 3 ideas:

- Simulate underlying security movement based on Monte-Carlo
- Regression approximation (i.e., least squares) of continuation value for paths with **in-the-money states**
- Backward induction of early exercise states

LSM ideas and inputs

LSM ideas

The LSM algorithm combines 3 ideas:

- Simulate underlying security movement based on Monte-Carlo
- Regression approximation (i.e., least squares) of continuation value for paths with **in-the-money states**
- Backward induction of early exercise states

Inputs for the algorithm:

- Simulated Monte-Carlo paths indexed $i = 0, 1, ..., m-1$ at time steps indexed $j = n, n-1, ..., 1, 0$
- Payoff function, denoted as $Payoff(S_{i,j})$
- Regressor functions $\phi_0(S_{i,j}), ..., \phi_{r-1}(S_{i,j})$
- Estimated continuation value is given by $f(S_{i,j}) = \alpha + \hat{\beta}^T \cdot \phi(S_{i,j}) = \alpha + \sum_{k=0}^{r-1} \hat{\beta}_k \cdot \phi_k(S_{i,j})$

LSM Example

Example

Vanilla American Put with $S_0 = 100, K = 110, T = 3, r = 0.06, \Delta t = 1$
Naive regressors: $\phi_0(S) = S; \phi_1(S) = S^2$

Step 1: Simulate paths

S_t	t = 0	t = 1	t = 2	t = 3
path 1	100	109	108	134
path 2	100	116	126	154
path 3	100	122	107	103
path 4	100	93	97	92
path 5	100	111	156	152
path 6	100	76	77	90
path 7	100	92	84	101
path 8	100	88	122	134

Step 2: Payoff at $T(t = 3)$

	payoff $(t = 3)$
path 1	0
path 2	0
path 3	7
path 4	18
path 5	0
path 6	20
path 7	9
path 8	0

LSM Example (con't)

Step 3: Discount payoff from $t = 3$, calculate exercise value at $t = 2$, and perform regression

	Exercise value (EV, $t = 2$)	DCF (at $t = 2$)
1	2	0
2	0	0
3	3	$7 \cdot e^{-0.06} = 6.59$
4	13	$18 \cdot e^{-0.06} = 16.95$
5	0	0
6	33	$20 \cdot e^{-0.06} = 18.84$
7	26	$9 \cdot e^{-0.06} = 8.48$
8	0	0

$\Rightarrow$

Y	$\phi_0(S_{\cdot,2})$	$\phi_1(S_{\cdot,2})$
0	108	108^2
6.59	107	107^2
16.95	97	97^2
18.84	77	77^2
8.48	84	84^2

Least squares gives:

$$\hat{\alpha} = -107, \hat{\beta}_0 = 2.983,$$

$$\hat{\beta}_1 = -0.01813$$

LSM Example (con't)

Step 4: Compare the approximation of the continuation value (which is given by the regression result) with the exercise value to make decision

Example for approximating continuation value: for path 1, we have

$$f(S_{1,2}) = -107 + 2.983 \cdot 108 - 0.01813 \cdot 108^2 = 3.69$$

Path	Approx Continuation Value (CV)	EV	Exercise ?
1	3.69	2	No
3	4.61	3	No
4	11.76	13	Yes
6	15.2	33	Yes
7	15.65	26	Yes

LSM Example (con't)

Step 5: Update the cash flow based on the exercise decisions at $t = 2$ and discount the cash flows back to $t = 1$, perform regression afterwards.

Remark: We use the actual cash flow instead of the approximated value given by the previous regression result.

Path	EV ($t = 1$)	DCF (at $t = 1$)	CF ($t = 2$)	CF ($t = 3$)
1	1	0	0	0
2	0	0	0	0
3	0	$7 \cdot e^{-0.06 \cdot 2} = 6.208$	0	7
4	17	$13 \cdot e^{-0.06} = 12.24$	13	0
5	0	0	0	0
6	34	$33 \cdot e^{-0.06} = 31.08$	33	0
7	18	$26 \cdot e^{-0.06} = 24.49$	26	0
8	22	0	0	0

LSM Example (con't)

After regression on the **in-the-money** paths, we obtain

$$\hat{\alpha} = 203.8, \hat{\beta}_0 = -3.335, \hat{\beta}_1 = 0.01356$$

Step 6: Again, make exercise decision at $t = 1$

Path	Approx Continuation Value (CV)	EV	Exercise ?
1	1.39	1	No
4	10.92	17	Yes
6	28.66	34	Yes
7	11.75	18	Yes
8	15.33	22	Yes

LSM Example (con't)

Step 7: Finally, we have made exercise decisions on all the intermediate time steps. We Update the cash flow based on the exercise decisions at $t = 1$, and sum up the discount cash flows as the option value.

Path	$t = 1$	$t = 2$	$t = 3$
1	0	0	0
2	0	0	0
3	0	0	7
4	17	0	0
5	0	0	0
6	34	0	0
7	18	0	0
8	22	0	0

Option value at $t = 0$ is given by

$$\frac{1}{8}(7 \cdot e^{-0.06 \cdot 3} + 17 \cdot e^{-0.06 \cdot 1} + 34 \cdot e^{-0.06 \cdot 1} + 18 \cdot e^{-0.06 \cdot 1} + 22 \cdot e^{-0.06 \cdot 1}) = 11.44$$

Advantages of LSM

Computational efficient: The computational complexity does not grow exponentially as dimension increases.

Flexibility: The LSM method can be readily applied together with **different stochastic models** to price **exotic** products.

- Can incorporate **Merton's jump diffusion** in path simulation.
- Can incorporate **interest rate models** (e.g., Ho Lee Model, CIR Model...) to price exotic fixed-income derivatives.

  Example: Deferred American Swaption, which gives buyers the right to enter into a forward starting swap at a pre-specified swap rate after a lock-in period.

Advantages of LSM (con't)

Computational speed-up: LSM algorithm can be combined with parallel / distributed computing architecture to speed-up running time.

- Simulated paths can be generated in parallel with **multiple CPUs**.
- Can split the simulated paths into several subsets. Run LSM on each subset through distribution and return the average of subsets' result. Remark: Computational complexity is reduced **more than proportional** when matrix size is reduced.

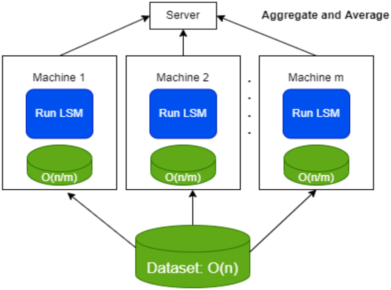


Table of Contents

Convergence of LSM

Accuracy of the approximation on the American Option value can be increased by:

- Increasing the number of simulated paths m
- Increasing the number of regressor functions r
- Increasing the number of time steps n (so the model is **closer to an American Option** instead of a Bermudan Options)

Longstaff and Schwartz and several other researchers after concluded:

Convergence of LSM

The LSM Algorithm converges to the actual value of the corresponding Bermudan Option with n exercise dates if $r \to +\infty$ and $m \to +\infty$.

Bias of LSM

There are 2 main sources of bias in the LSM method, which interestingly run in **opposite direction** (i.e., LSM estimator is *interleaving*).

Low-side bias

Also known as the *sub-optimal* bias, refers to the phenomenon that finite regressor functions cannot fully represent the continuation value.

- Suppose we have an optimal exercise policy P^*
- LSM will return an alternative exercise policy P' such that the realized option value from P' can **never be greater than** that from P^*

High-side bias

Also known as the *look-ahead* or *foresight* bias, arising from the following:

- Original LSM uses one simulation for both the exercise decision and the payoff valuation ("reporting *MSE* from the training set")
- Using future information (S_{t+1}) to make exercise decision at t

Fixing low-side bias (*sub-optimal* bias)

Assuming the absence of high-side bias, *sub-optimal* bias in fact provide us guidance in determining the number of regressor functions required.

- By **gradually increasing the number of regressor functions** until the value implied by the LSM algorithm does not increase, we can obtain the threshold on the number of regressors function to satisfactorily capture the dynamics of the continuation value.

Fixing low-side bias (*sub-optimal* bias)

Assuming the absence of high-side bias, *sub-optimal* bias in fact provide us guidance in determining the number of regressor functions required.

- By **gradually increasing the number of regressor functions** until the value implied by the LSM algorithm does not increase, we can obtain the threshold on the number of regressors function to satisfactorily capture the dynamics of the continuation value.

Longstaff and Schwartz suggested that the high-side bias is **negligible**, but many researchcers after suggested **the opposite**. Therefore, we next look for fixes on the high-side bias.

Fixing high-side bias (*look-ahead* bias)

Method 1: Independent set of Monte Carlo paths with two passes

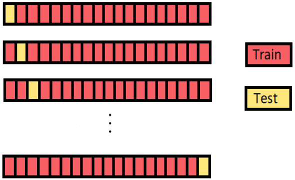
- Similar idea as "train-test split" in ML, removing the correlation between the exercise decision and the simulated payoff.
- First pass: Use the first set of Monte Carlo paths to determine the coefficients of the regressor functions in each time step through backward induction.
- Second pass: Use the second set of Monte Carlo paths to make exercise decision through forward pass, by comparing approximated continuation value (from regression functions) and exercise value.

Drawback: doubling the computational cost, which may be quite heavy if number of regressor functions and paths is large.

Fixing high-side bias (*look-ahead* bias) (con't)

Method 2: Leave One Out (LOO) decision making

- Similar idea as Leave One Out Cross Validation in ML.
- Omitting each simulation path from the regression and make exercise decision on the path based on the self-excluded regression.
- The method is computational efficient, since updating the regression result (by excluding / adding an element) only takes a simple linear algebra operation.



LOO decision making illustration

Example

3 simulated paths at time t, with information of S_t and DCF_t, where DCF_t is discounted continuation value. Strike $K = 0$.
Naive regressor: $\phi_0(S) = S$

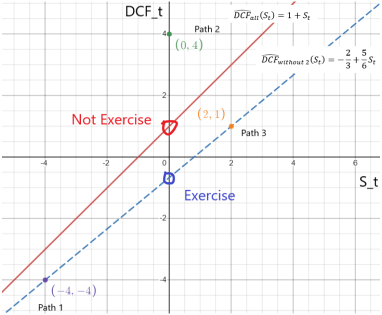
Path	S_t	Exercise Value	DCF_t
1	-4	-4	-4
2	0	0	4
3	2	2	1

Regression result from all 3 paths:
$f_{all}(S_t) = 1 + S_t$

Regression result excluding path 2:
$f_{without2}(S_t) = -\frac{2}{3} + \frac{5}{6}S_t$

Based on f_{all}, we will **not** exercise at path 2 since $f_{all}(0) = 1 > EV_2 = 0$

Based on $f_{without2}$, we will exercise at path 2 since
$f_{without2}(0) = -\frac{2}{3} < EV_2 = 0$



Under the full regression, a "better" decision in path 2 is made due to outlying DCF_t in path 2, illustrating the *look-ahead* bias.

With the help of LOO, the influence of bias is reduced.

Choice of regressor functions

There seems to be no general rule to determine what are the best regressor functions and how many functions to be used in LSM.

Longstaff and Schwartz suggested that the form of regressor functions did not play significant role in the performance of LSM, but the claim was subject to debate.

Choice of regressor functions

There seems to be no general rule to determine what are the best regressor functions and how many functions to be used in LSM.

Longstaff and Schwartz suggested that the form of regressor functions did not play significant role in the performance of LSM, but the claim was subject to debate.

Some suggest the usage of Laguerre polynomials, which have the form

$$L_n(X) = e^{-\frac{X}{2}} \frac{e^X}{n!} \frac{d^n}{dX^n}(X^n e^{-X})$$

- L_i and L_j are orthogonal, which means they are not correlated with each other, preventing multicolinearity and nearly-singular matrix.
- Laguerre polynomials can capture non-linear relationships more effectively, resulting in faster convergence with fewer terms.

Choice of regressor functions (con't)

Some special regressor functions are suggested for pricing different options:

- Asian Option: Option involving $\bar{S}_t$ in payoff.
  $\bar{S}_t$ and $S_t \times \bar{S}_t$ are suggested to be included.
- Lookback Option: Option involving S_{max} / S_{min} in payoff.
  $S_{max(min)}$ and $S_t \times S_{max(min)}$ are suggested to be included.
- Barrier Option: Option being knocked in/out after crossing barrier B.
  Sojourning time above / below the barrier is suggested to be included.

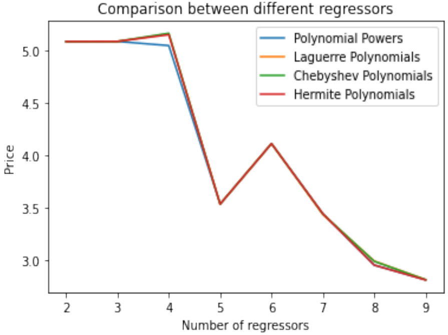
Table of Contents

Comparing different regressor functions

I coded up the LSM method (link) and attempted pricing a vanilla American Put with parameters $S_0 = 44, \sigma = 0.4, T = 2, r = 0.06, K = 40$. The benchmark price from the finite difference method is \$5.647.

I attempted 4 regressor functions:

- Polynomial Power
- Laguerre Polynomials
- Chebyshev Polynomials
- Hermite Polynomials



Unfortunately, the result does not seem to be very promising when the number of regressor functions increases.

Comparing different regressor functions (con't)

A few observations:

- The algorithm returned the closest price from the benchmark when number of regressors is 4.
- Laguerre, Chebyshev, and Hermite polynomials performed generally better than polynomial powers, likely because they are orthogonal.
- Performance of the algorithm worsened drastically when the number of regressors increase, likely due to problem of multi-collinearity.

Table of Contents

Application: Pricing Real Options

Real Options are important in corporate finance, which gives firms the right to undertake certain business opportunities or investments.

Usually, decisions in firms are not enforced to be made in a particular date, so the options can be thought of as American-style option.

Application: Pricing Real Options

Real Options are important in corporate finance, which gives firms the right to undertake certain business opportunities or investments.

Usually, decisions in firms are not enforced to be made in a particular date, so the options can be thought of as American-style option.

Example

Consider an option F_1 for a company to enter into a new industry with value V within T_1 years. With a cost outlay of I_1, the firm can get $e_1 \in (0, 1)$ of the whole industry and the subsequent option F_2.
The option F_2 allows the firm to get the remaining portion of the industry $e_2 = 1 - e_1$ with an cost outlay of I_2, within T_2 years.

The above example illustrates a compound American Option.
Exercising F_1 creates opportunity F_2, and value of F_1 depends on F_2.

Application: Pricing Real Options (con't)

The following illustrates the idea of applying LSM on compound option:

Formulation for compound option

Suppose there are H compounded real options, and the h-th option gives right to exercise the $(h+1)$-th option.
Without loss of generality, suppose maturities for H options are given by $T_1 \leq T_2 \leq ... \leq T_H$.
The Payoff of option h in isolation is denoted as $\Pi_h(t, X_t)$.

Then, current value of the h-th option is given by

$$F_h(t, X_t) = \max_{\tau \in (t, T_h)} \mathbb{E}^Q[e^{-r(\tau - t)}(\Pi_h(\tau, X_\tau) + F_{h+1}(\tau, X_\tau))]$$

We can then apply the LSM algorithm in order for option H (which is a Vanilla American Option), $H-1, H-2, ...,$ all the way back to option 1.

Table of Contents

Conclusion

Advantages of LSM:

- Complexity does not grow exponentially with dimension.
- Flexible to be applied with different stochastic models.
- Support parallel / distributed computing architecture.

Disadvantages of LSM:

- Require determination / calibration on the type of regressor functions and the number of regressor functions to be used.
- *Interleaving* biased property.

Some other techniques for pricing American Option:

- Quasi-Monte Carlo Method combined with LSM
- Reinforcement Learning: Least Squares Policy Iteration (LSPI) and Fitted Q-Iteration (FQI)

Thank You !

References

- Valuing American Options by Simulation - A Simple Least-Squares Approach
- The Valuation of Real Options with the Least Squares Monte Carlo Simulation Method
- Real Options Valuation - A Monte Carlo Approach
- Leave-One-Out Least Square Monte Carlo Algorithm
- Distributed Least-Squares Monte Carlo for American Option Pricing
- Number of paths versus number of basis functions in American option pricing