Read Me

Author: Jerry Xia

Date: 2018/06/19

Note: The advanced Marckdown features such as math expression may not be compatible in GitHub, please see README.pdf instead if you want more details

Implementation

Please feel free to see the Monte Carlo engine: MonteCarlo.py

Classification

- regular Monte Carlo simulation
- optimal hedged Monte Carlo simulation
- delta-based Monte Carlo simulation
- Monte Carlo with antithetic variates
- Least square method of Longstaff and Schiwatz (LSM)
- Hedeged Least Square method (HLSM)

Underlying Process

- geometric Brownian motion
- CIR model
- Heston model

Boundary Scheme (CIR model)

- absorption
- reflection
- Higham and Mao
- partial truncation
- full truncation

Optimal Hedged Monte Carlo

Model Inventors: Marc Potters, Jean-Philippe Bouchaud, Dragan Sestovic

- ## 1 Introduction

This is a Python Notebook about variance reduction Monte Carlo simulations. In this script, I implemented the following variance reduction methods as well as their antithetic variates' version:

- regular Monte Carlo
- Monte Carlo with delta-based control variates
- optimal hedged Monte Carlo

Due to the significance and robustness, I mainly focus on the optimal hedged Monte Carlo (OHMC) in option pricing. We invoke this method to price European options and make comparison with other methods.

1.1 Facts

- The option price is not simply the average value of the discounted future pay-off over the objective (or historical) probability distribution
- The requirement of absence of arbitrage opportunities is equivalent to the existence of "risk-neutral measure", such that the price is indeed its average discounted future pay-off.
- Risk in option trading cannot be eliminated

1.2 Objective

- It would be satisfactory to have an option theory where the objective stochastic process of the underlying is used to calculate the option price, the hedge strategy and the *residual risk*.

1.3 Advantages

- It is a versatile methods to price complicated path-dependent options.
- Considerable variance reduction scheme for Monte Carlo
- It provide not only a numerical estimate of the option price, but also of the optimal hedge strategy and of the residual risk.
- This method does not rely on the notion of risk-neutral measure, and can be used to any model of the true dynamics of the underlying

2 Underlying dynamics

Black-Scholes Model

$$dS = \mu S dt + \sigma S dW_t$$

$$logS_{t+1} = logS_t + (\mu - \frac{\sigma^2}{2})\Delta t + \sigma\sqrt{\Delta t}\epsilon$$

where

$$\epsilon \sim N(0, 1)$$

In risk neutral measure, $\mu = r - q$.

Heston Model

The basic Heston model assumes that S_t, the price of the asset, is determined by a stochastic process:

$$dS_t = \mu S_t dt + \sqrt{v_t} S_t dW_t^S$$
$$dv_t = \kappa(\theta - v_t)dt + \xi\sqrt{v_t} dW_t^v$$

where

$$E[dW_t^S, dW_t^v] = \rho dt$$

In risk neutral measure, $\mu = r - q$.

3 Methodology

3.1 Simbol Definition

Option price always requires to work backward. That is because the option price is known exactly at the maturity. As with other schemes, we determine the option price step by step from the maturity $t = K\tau = T$ to the present time $t = 0$. The unit of time being τ, for example, one day. We simulate N trajectories. In trajectory i, the price of the underlying asset at time $k\tau$ is denoted as $S_k^{(i)}$. The price of the derivative at time $k\tau$ is denoted as C_k, and the hedge function is H_k. We define an optimal hedged portfolio as

$$W_k^{(i)} = C_k(S_k^{(i)}) + H_k(S_k^{(i)})S_k^{(i)}$$

The one-step change of our portfolio is

$$\Delta W_k^{(i)} = df(k,k+1)C_{k+1}(S_{k+1}^{(i)}) - C_k(S_k^{(i)}) + H_k(S_k^{(i)})(df(k,k+1)S_{k+1}^{(i)} - S_k^{(i)})$$

Where $df(k,k+1)$ is the discounted factor from time $k\tau$ to $(k+1)\tau$, $df2(k,k+1)$ is the discounted factor considering dividend $e^{-(r-q)(t_{k+1}-t_k)}$

3.2 Objective

The optimal hedged algorithm can be interpreted as the following optimal problem

$$\begin{array}{ll} \text{minimize} & Var[\Delta W_k] \\ \text{subject to} & E[\Delta W_k] = 0 \end{array}$$

It means we should try to minimize the realized volatility of hedged portfolio while maintaining the expected value of portfolio unchanged.

3.3 Basis Functions

The original optimization is very difficult to solve. Thus we assume a set of basis function and solved it in such subspace. We use N_C and N_H to denote the number of basis functions for price and hedge.

$$C_k(\cdot) = \sum_{i=0}^{N_C} a_{k,i}A_i(\cdot)$$

$$H_k(\cdot) = \sum_{i=0}^{N_H} b_{k,i}B_i(\cdot)$$

The basis functions A_i and B_i are priori determined and need not to be identical. The coefficients a_i and b_i can be calibrated by solving the optimal problem.

3.4 Numerical Solution

$$\begin{array}{ll} \text{minimize} & \frac{1}{N}\sum_{i=1}^{N} \Delta W_k^{(i)2} \\ \text{subject to} & \frac{1}{N}\sum_{i=1}^{N} \Delta W_k^{(i)} = 0 \end{array}$$

Denote the discounted forward underlying price change at time $k\tau$ as

$$\Delta S_k = df2(k,k+1)S_{k+1} - S_k$$

Define

$$Q_k = \begin{bmatrix} -A_{k,1}(S_k^{(1)}) & \cdots & -A_{k,N_C}(S_k^{(1)}) & B_{k,1}(S_k^{(1)})\Delta S_k^{(1)} & \cdots & B_{k,N_H}(S_k^{(1)})\Delta S_k^{(1)} \\ -A_{k,1}(S_k^{(2)}) & \cdots & -A_{k,N_C}(S_k^{(2)}) & B_{k,1}(S_k^{(2)})\Delta S_k^{(2)} & \cdots & B_{k,N_H}(S_k^{(1)})\Delta S_k^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -A_{k,1}(S_k^{(N)}) & \cdots & -A_{k,N_C}(S_k^{(N)}) & B_{k,1}(S_k^{(N)})\Delta S_k^{(N)} & \cdots & B_{k,N_H}(S_k^{(N)})\Delta S_k^{(N)} \end{bmatrix}$$

$$c_k = (a_{k,1}, \cdots a_{k,N_C}, b_{k,1}, \cdots, b_{k,N_H})^T$$

$$v_k = df(k, k+1) C_{k+1}(S_{k+1})$$

As for v_k, note that we know the exact value at maturity, which means there is no need to approximate price in terms of basis functions, that is

$$v_k = \begin{cases} df(N-1, N)\ payoff(S_N), & k = N-1 \\ df(k, k+1)\ \sum_{i=1}^{N_C} a_{k+1,i} A_i(S_{k+1}), & 0 < k < N-1 \\ df(0,1)\ C_1(S_1), & k = 0 \end{cases}$$

Then, the optimization problem can be expressed as

$$\begin{aligned} \underset{c_k}{\arg\min} \quad & (v_k + Q_k c_k)^T (v_k + Q_k c_k) \\ \text{subject to} \quad & 1^T_{[N\times 1]}(v_k + Q_k c_k) = 0 \end{aligned}$$

In step k, since we already know the information (v_k) in step k+1. By canceling the constant term, the optimal problem can be simplified as the following

$$\begin{aligned} \underset{c_k}{\arg\min} \quad & 2v_k^T Q_k c_k + c_k^T Q_k^T Q_k c_k \\ \text{subject to} \quad & 1^T_{[N\times 1]} v_k + 1^T_{[N\times 1]} Q_k c_k = 0 \end{aligned}$$

3.5 Convex Optimization Problem

Let us first review the standard form of linear constrained quadratic programming problem:

$$ \min_{x} \quad \frac{1}{2} x^T P x + q^T x\

```
\mbox{subject to} \quad G x \preceq h\\

A x = b
```

$*Notethat*x^T*meansthetransposeofvectorx, and*G x \preceq h *denotestheinequalityistakenelement − wiseoverthevectors*G x*and*h *. Theobjectivefunctionisconvexifandonlyifthematrix*P$ is positive-semidefinite(Hermitian matrix all of whose eigenvalues are nonnegative), which is the realm we concern with.

Recall that the constrained optimization problem:

$$ \arg\min{c*k}\quad \quad v{k}^T Q*k c*k + \frac{1}{2}c*k^T Q*k^T Q*k c_k\

\mbox{subject to}\quad \quad 1{[N\times1]}^T v{k} + 1*{[N\times1]}^T Q*k c_k=0 $$ Correspondingly, we make the connection by letting

$$ x = c_k\

```
P = Q_k^T Q_k\\

q = Q_k^T v_k\\

A = 1_{[N\times1]}^T Q_k\\

b = -1{[N\times1]}^T v{k}
```

$$ The hard work is almost over right now. As you would always find, formulating the problem is usually the hard step. Invoking a solver is straightforward.

Note that when $k = 0$, the degree of freedom of the quadratic problem decreases to 2. Because here the only concerns are price and hedge at time zero (we don't need to project them into a high dimension space). Let $x = [C_0, H_0]^T$

$$ Q_0 = \begin{bmatrix}

```
-1 & \Delta S_0^{(1)}\\

\vdots & \vdots\\

-1 & \Delta S_0^{(N)}

\end{bmatrix}\\

P = Q_0^T Q_0\\

q = Q_0^T v_0\\

A = 1_{[N \times 1]}^T Q_0\\

b = -1_{[N \times 1]}^T v_0
```

$$

4 Variance reduction and other methods

The rate of convergence of the Monte Carlo simulation is $O\left(\max\left(\Delta t, \frac{1}{N_x}\right)\right)$. The variance reduction techniques are used to reduce the constant factor corresponding to the Monte Carlo approximation $O\left(\frac{1}{N_x}\right)$. Some of the most used variance reduction techniques are:

- Control Variates
- Antithetic Variates
- Moment Matching

In this part we selected antithetic variates and delta-based control variates methods as a supplement to optimal hedged monte carlo simulation.

4.1 Antithetic variates

The main idea of this technique is to look at the asset equation that you aretrying to simulate:

$$dS_t^{(1)} = rS_t^{(1)}dt + \sigma S_t^{(1)}dW_t$$

and recognize that sinceztis a standard Brownian motion so will be−ztandthey will have the same exact distribution. This means that the equation:

$$dS_t^{(2)} = rS_t^{(2)}dt - \sigma S_t^{(2)}dW_t$$

will also generate paths of the same asset. The variance depends on the sign of the covariance of $payoff(S_t^{(1)})$ and $payoff(S_t^{(2)})$. It can increase the eventual variance or decrease it, both case do arise. One sufficient condition to insure variance reduction is the monotony of the payoff function. Then, when using both in the calculation of the final Monte Carlo value the variance of the estimate will be reduced.

4.2 Delta-based control variates

Delta hedging can be summarized succinctly in the following way: Suppose that at time $t = 0$, we receive C_0 the price of an option that pays C_T at time T. The price of this option at any time t is a function $C(t, S)$. Then, if we hold at any moment in time $\frac{\partial C}{\partial S}(t, S) = \frac{\partial C_t}{\partial S}$ units of stock, then we will be able to replicate the payout of this option C_T at time T. This is in theory since of course we cannot trade continuously. So in practice we perform a partial hedge where we only rebalance at some discrete moments in time say $t_1, t_2, \cdots, t_N$. The replicating strategy can be expressed as follow:

$$W(t_i, S_i) = C(t_0, S_0)e^{r(t_i - t_0)} + \sum_{j=0}^{i} \Delta(t_j, S_j)(S_{j+1}e^{-r(t_{j+1} - t_j)} - S_j)e^{r(t_i - t_j)} = C(t_i, S_i)$$

which is similar to the strategy in the optimal hedged Monte Carlo simulation where the only difference is that in OHMC, we use option and delta hedging to replicate the cash flow and here we do the opposite operation. But when implementing the delta-based control variates, we should move the hedging term to the right hand side which make it identical to the OHMC strategy. Note that here we are assumed to know the delta hedging function. It explains a lot why OHMC can reduce the variance.

4.3 Optimal hedged Monte Carlo simulation

In conclusion, OHMC is just a control variates method with an optimization on top and it is more practical because we do not have an analytical formula for the hedge sensitivity (i.e. delta, gamma, etc.)

5 Add Hedging portfolio with the Least Square Monte Carlo (LSM)

In order to price Amrican type options, we need to consider the problem of optimal exercise. LSM is a well-defined method to tackle this problem. In contrast, here we only utilize the information of exercise points along each simulation path using cross-sectional regression. Different from the original LSM, here we equipe basis functions to approximate price and hedge at each step similar to OHMC. And discuss independently at the inception.

This combination create a magic reaction. Now we can not only price the American options but also hedge it! Moreover, it's model independent, model parameters or construction, dimension doesn't matter at all! We use Black-Scholes and Heston model as examples. What only matters is the underlying price trials. With it, we can calculate the following stuffs.

- American options price
- American options Greeks
- American options optimal exercise boundary

Here, Bouchard and Warin concluded two main dynamic strategy in American options pricing, A1 and A2. Besides, I equiped them with a hedging strategy:

5.1 A1 strategy with optimal exercise time estimate

- Initialization: $\tau(t_J) = T$
- Backward induction:

- $\tau(t_j) = t_j\mathbf{1}_{\{g(t_j)\geq C(t_j)\}} + \tau(t_{j+1})\mathbf{1}_{\{Z(t_j)<C(t_j)\}}$
- Price estimator at 0:$P_0 = E[g(\tau(t_0), X_{\tau(t_0)})]$

5.2 A2 strategy with American values estimate

- Initialization: $P_T = g(T, X_T)$
- Backward induction: $P_{t_j} = max\{g(t_j, X_{t_j}), E[P_{t_{j+1}}]\}$
- Price estimator at 0: P_0

5.3 A2b strategy with optimal exercise time estimate and American values estimate

- Initialization: $\tau(t_J) = T$
- Backward induction:
    - $\tau(t_j) = t_j\mathbf{1}_{\{g(t_j)\geq C(t_j)\}} + \tau(t_{j+1})\mathbf{1}_{\{Z(t_j)<C(t_j)\}}$
    - Price estimator at j:$P_j = E[g(\tau(t_j), X_{\tau(t_j)})]$ for $j = J, J-1, \cdots, 1$
- Price estimator at 0 (one-step hedged MC): $arg\, min_{P_0,H_0} E[(\Delta W_0)^2]$

5.5 Performance Test:

Black-Scholes model: HLSM-BlackScholes-American.ipynb Heston model: HLSM-Heston-American.ipynb

In this document, we take Black-Scholes model as an example

Parameters:

```
risk_free_rate = 0.06
dividend = 0.0
time_to_maturity = 1
volatility = 0.3
strike = 1.1
stock_price = 1
n_trials = 4000
n_steps = 20
func_list = [lambda x: x**0, lambda x: x] # basis for OHMC part
option_type = 'p'
```

Results:

American Options

Algorithm	Price	Delta
A1	0.1499	N/A
A2	0.1590	0.585
A2b	0.1500	0.491

European Options

- BS Formula: 0.1401
- BS Binomial Tree: 0.1410
- Regular MC: 0.1453
- OHMC: 0.1426