Optimal Hedging Monte Carlo Draft

Tengxi Wan

June 2018

1 Introduction

This is the draft of Optimal Hedging Monte Carlo Simulation (OHMC) method in option pricing.

1.1 Background

Under the classic Black-Scholes assumption, dynamic hedging works perfect so that one can replicate an option payoff with linear instrument effectively. However, it is not true in real market considering following problems:

- Transaction cost, continuous "Delta Hedge" in real stock market can be costly given the existence of transaction cost
- Liquidity premium and bid-offer spread, underlying asset can be illiquid, such that the hedge interval can be long in days and months
- hedge slippage - loss may occur in between the discrete hedge points

Therefore, instead of pricing derivatives under the risk-neutral measure, we will create a portfolio including the derivative and hedging position.

2 Methodology

2.1 Time Interval Discrete

Consider an option with maturity T, we will discrete evenly the time interval $[0, T]$ into N sub-interval. With $0 = t_0 < \cdots < t_k < \cdots < t_N = T$, and $t_k = k\frac{T}{N} = k\tau$.

2.2 Portfolio Composition

Consider a portfolio W with one long position in European call option(C) and Φ position in underlying asset(S). (Φ could be positive or negative). Thus at time t_k the portfolio will be

$$W_k = C_k + \Phi_k S_k \tag{1}$$

Define the value changing of our portfolio from time t_k to t_{k+1} to be: (Let r denote the risk-free interest rate)

$$\begin{aligned}\Delta W(k,k+1) &\equiv W_{k+1}e^{-r\tau} - W_k \\ &= [C_{k+1}e^{-r\tau} - C_k] + \Phi_k[S_{k+1}e^{-r\tau} - S_k] \\ &\equiv \Delta C_k + \Phi_k \Delta S_k\end{aligned} \tag{2}$$

Note when we value an option by Monte Carlo simulation, we always do it backward from maturity, just like what is in binomial tree. Thus, when deriving back at time t_{k+1}, the unknowns are C_k and Φ_k (follow some stochastic processes). The goal is to estimate these quantities under following conditions:

$$\begin{aligned}&\textbf{minimize:} \quad E[\Delta W(k,k+1)^2] \\ &\textbf{constraint:} \quad E[\Delta W(k,k+1)] = 0\end{aligned} \tag{3}$$

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

2.3 Basis Function

The original optimization is very difficult to solve due to high dimension in Monte Carlo simulation (often greater than 50,000). Thus we assume an basis function form of unknowns C_k and Φ_k(backward at time t_{k+1}).

$$\begin{aligned}C_k &= \sum_{i=0}^{M_C} a_i A_i(s) \\ \Phi &= \sum_{j=0}^{M_\Phi} b_j B_j(s)\end{aligned} \tag{4}$$

Where $A_i(s)$ and $B_j(s)$ are series of basis function of the price of underlying asset. And a_i and b_j are the unknown coefficients. One can choose polynomial series as basis functions,more advanced orthogonal series should also be in consideration such as Laguerre or Hermite series.

2.4 Matrix Form

Let us continue the scenario in Monte Carlo simulation at time t_{k+1}. Assume we decide to simulate with P paths in total. Thus at each time point, we will have a column of stock and option price:

$$\begin{aligned}S_k &= (S_k^1, S_k^2, \cdots, S_k^P)^T \\ C_k &= (C_k^1, C_k^2, \cdots, C_k^P)^T\end{aligned} \tag{5}$$

And we define the difference of discounted stock price,

$$\Delta S_k \equiv [e^{-r\tau} S_{k+1} - S_k] \tag{6}$$

In matrix form, we define the coefficient a and b (all at time k, so we omit subscription)

$$\begin{aligned} a &\equiv [a_0, a_1, \cdots a_{M_C}]^T \\ b &\equiv [b_0, b_1, \cdots b_{M_\Phi}]^T \end{aligned} \tag{7}$$

Here we combine the coefficient a and b to be a new column denoted c

$$c = [a, b]^T = [a_0, a_1, \cdots a_{M_C}, b_0, b_1, \cdots b_{M_\Phi}]^T \tag{8}$$

We will rewrite the deterministic part into matrix form (abbreviate the subscription of time k):

$$\mathcal{Q} = \begin{bmatrix} -A_0(S^1) & \cdots & -A_{M_C}(S^1) & B_0(S^1)[\Delta S^1] & \cdots & B_{M_\Phi}(S^1)[\Delta S^1] \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -A_0(S^P) & \cdots & -A_{M_C}(S^P) & B_0(S^P)[\Delta S^P] & \cdots & B_{M_\Phi}(S^P)[\Delta S^P] \end{bmatrix}_{p \times (2+M_C+M_\Phi)} \tag{9}$$

For any path p, we have the portfolio change

$$\begin{aligned} \Delta W^p(k, k+1) &= [e^{-r\tau} C^p_{k+1} - C^p_k] + \Phi^p_k \Delta S^p_k \\ &= [e^{-r\tau} C^p_{k+1} - \sum_{i=0}^{M_C} a_i A_i(S^p_k)] + \sum_{j=0}^{M_\Phi} b_j B_j(S^p_k) \Delta S^p_k \end{aligned} \tag{10}$$

Define the column of portfolio change for every path

$$\Delta W(k, k+1) = [\Delta W^1(k, k+1), \Delta W^2(k, k+1), \cdots, \Delta W^P(k, k+1)]^T \tag{11}$$

In matrix form, we will have

$$\Delta W(k, k+1) = e^{-r\tau} C_{k+1} + \mathcal{Q}c \tag{12}$$

2.5 Optimization Problem

Thus the constraint condition in equation 3 will be

$$E[\Delta W(k, k+1)] \approx \frac{1}{P} \sum_{l=1}^{P} \Delta W^l(k, k+1) = \frac{1}{P} \mathbf{1}^T [\Delta W(k, k+1)] \tag{13}$$

That is the equation of constraint:

$$\mathbf{1}^T [e^{-r\tau} C_{k+1} + \mathcal{Q}c] = 0 \tag{14}$$

The optimization problem is to minimize following term

$$\begin{aligned} E[\Delta W(k,k+1)^2] &\approx \frac{1}{P}\sum_{l=1}^{P}[\Delta W^l(k,k+1)]^2 \\ &= \frac{1}{P}[\Delta W(k,k+1)]^T[\Delta W(k,k+1)] \\ &= \frac{1}{P}[e^{-r\tau}C_{k+1}+\mathcal{Q}c]^T[e^{-r\tau}C_{k+1}+\mathcal{Q}c] \\ &= \frac{1}{P}\left[c^T\mathcal{Q}^T\mathcal{Q}c+2e^{-r\tau}C_{k+1}^T\mathcal{Q}c+e^{-2r\tau}(C_{k+1})^TC_{k+1}\right] \end{aligned} \tag{15}$$

Note term $e^{-2r\tau}(C_{k+1})^TC_{k+1}$ is just a constant, and $2e^{-r\tau}C_{k+1}^T\mathcal{Q}$ is known. This will result a quadratic programming with equality constraints. The general form of linear equality constrained quadratic programming is:

$$\begin{aligned} &\textbf{minimize:} \quad \frac{1}{2}\mathbf{x}^T\Omega\mathbf{x}+\mathbf{r}^T\mathbf{x} \\ &\textbf{object to:} \quad \mathcal{A}\mathbf{x}=\mathbf{b} \end{aligned} \tag{16}$$

Corresponding to our case, $\Omega = \mathcal{Q}^T\mathcal{Q}$ is a positive definite symmetric matrix, $\mathbf{r} = [2e^{-r\tau}C_{k+1}^T\mathcal{Q}]^T$. And $\mathcal{A} = \mathbf{1}^T\mathcal{Q}$ is controling the equality condition, and target value is $\mathbf{b} = -\mathbf{1}^Te^{-r\tau}C_{k+1}$.

2.6 Lagrange Multiplier Technique

For equality constrained quadratic programming problem, we can apply the Lagrange Multiplier technique to solve the optimization problem in equation 3. Firstly, we define the objective function

$$L_k(\Delta W,\lambda) = E[\Delta W(k,k+1)^2]+2\lambda E[\Delta W(k,k+1)] \tag{17}$$

And we let

$$\begin{aligned} &\frac{\partial L_k}{\partial a_i}=0, \quad 0\le i\le M_C \\ &\frac{\partial L_k}{\partial b_j}=0, \quad 0\le j\le M_\Phi \\ &\frac{\partial L_k}{\partial \lambda}=0 \end{aligned} \tag{18}$$

2.7 Some problems

When implementing the algorithm above, we find some problems.

1. Negative option value
   some time we get negative option price C_k from linear system solution.
   Possible solution: Maybe we should add constraint to avoid negative option price.

2. Basis functions
   We choose regular polynomial series, but it may cause singularity problem when the order is high.
   Possible solution: select advanced basis function such as Hermite or Laguerre series.

2.8 Robustness

Given there are some problems found above, we will come up with some robustness methods.

2.8.1 Condition on option price

We will add condition to avoid negative option price value. That is