Pricing American Option with Neural Network

Ludovic Goudenège and Thomas Sainrat

Premia 22

1 Problem formulation

We consider the following Black-Scholes model for the stock price:

$$dS_t = rS_t dt + \sigma S_t dW_t$$

where W is Gaussian processes, where r is the interest rate, σ is the volatility.

Using a martingale approach for an european or an american option (call or put), we can prove that the price is given by the solution of the following partial differential equation

$$\frac{\partial U}{\partial t} = \frac{1}{2} s^2 \sigma^2 \frac{\partial^2 U}{\partial s^2} + rs \frac{\partial U}{\partial s} - rU,$$

with the following boundary conditions for the call option

$$\begin{aligned} U(s,t) &= 0 && \text{whenever } s = 0, \\ \frac{\partial U}{\partial s}(s,t) &= \exp(-q\ t) && \text{whenever } s = S_{\max}, \end{aligned}$$

and the following boundary conditions for the put option

$$\begin{aligned} U(s,t) &= K \exp(-r\ t) && \text{whenever } s = 0, \\ \frac{\partial u}{\partial s}(s,t) &= 0 && \text{whenever } s = S_{\max}. \end{aligned}$$

By default, the initial values are $S = S_0 = 100$ and $\sigma = 0.01$, the maturity T is one year and the strike value is 100, such that $U(s,0) = \phi(s) = (b(s-K))^+$ where $b = 1$ for the call and $b = -1$ for the put, where ϕ is the payoff of the option.

In the case of the american options, we add the possibility to exercice the option before the maturity, such that the partial differential equation is actually a PDE with free boundary, which is a difficult problem to solve.

2 Longstaff-Schwarz algorithm

The Longstaff-Schwarz algorithm is based on a dynamic programing to solve the free boundary PDE for the american option. We discretize the time interval between 0 and the maturity T in N sub-intervals. The principle is to compute the optimal stopping times (when the solution touches the free boundary) rather than the solution of the free boundary PDE. Let $(\tau_j^*)_{0\leq j\leq N}$ be a time sequence defined backwardly

$$\left\{ \begin{array}{rcl} \tau_N^* & = & t_N \\ \tau_j^* & = & t_j \chi_{\{\phi(X_j)\geq u(t_j,X_j\}} + \tau_{j+1}^* \chi_{\{\phi(X_j)\geq u(t_j,X_j\}} \end{array} \right.$$

where χ is the indicator function with value 0 or 1, and where X_j are the value at time t_j of a trajectory under the Black-Scholes dynamics.

Note that

$$\begin{aligned} \{\phi(X_j) < u(t_j, X_j)\} &= \{\phi(X_j) < P_j u(t_{j+1}, X_j)\} \\ &= \left\{\phi(X_j) < \mathbb{E}\left[B(t_j, \tau_{j+1}^*)\phi(X_{\tau_{j+1}^*})|X_{t_j}\right]\right\} \end{aligned} \quad (1)$$

where P_j is the transition kernel of the Markov chain $(X_{t_j}, j = 0, \ldots, N)$ between time t_j ad time t_{j+1}, and $B(t_j, t)$ is the actualization coefficient from t_j to time t.

Because of (1), the optimal stopping time could be computed if we can approximate the conditional expectation

$$\mathbb{E}\left[B(t_j, \tau_{j+1}^*)\phi(X_{\tau_{j+1}^*})|X_{t_j}\right].$$

The Longstaff-Schwarz algorithm consists in approximating this value by a least-square regression on linear combination of basis functions e_k, such that

$$\mathbb{E}\left[\left(B(t_j, \tau_{j+1}^*)\phi(X_{\tau_{j+1}^*}) - \sum_{k\geq 1}\alpha_k e_k(X_j)\right)^2\right].$$

was minimal among all the choices for the coefficients α_k of linear combination.

With a Monte-Carlo procedure, and M paths of Black-Scholes sampled, we can estimate the previous expectation, which leads to the optimization problem in the coefficients α_k which minimizes

$$\frac{1}{M}\sum_{m=1}^{M}\left(\alpha\cdot e(X_{t_j}^{(m)}) - B(t_j, \tau_{j+1}^{(m)})\phi(X_{\tau_{j+1}^*}^{(m)})(X_{t_j}^{(m)}\right)$$

with optimal stopping times

$$\tau_j^{(m)} = t_j\chi_{\{\phi(X_j)\geq\alpha\cdot e(X_j)\}} + \tau_{j+1}^{(m)}\chi_{\{\phi(X_j)\geq\alpha\cdot e(X_j\}}.$$

When everything has been computed, we can have an estimator of the price by

$$\frac{1}{M}\sum_{m=1}^{M} B(0, \tau^{(m)})\phi\left(X_{\tau^{(m)}}^{(m)}\right)$$

3 Computing price by Neural Network

In order to price american options, we have te following algorithm

OFFLINE computation

- We fix the value of r and σ to default values.
- We use the Longstaff-Schwarz algorithm with a Monte-Carlo sampling composed of a large number of paths. (Time consuming)

  Backward loop on j

  - At the regression step j, we use the classical regression to compute the conditional expectation. (Time consuming)
  - We train a neural network at each time t_j, backwardly in time, to mimic the approximation of the conditional expectation. (Time consuming)
- We obtain N neural networks, one for each step $0 \leq j < N$, corresponding to time t_j.

ONLINE computation

- We fix the value of r and σ to customer values.
- We use the Longstaff-Schwarz algorithm with a Monte-Carlo sampling composed of a small number of paths. (Quick step)

  Backward loop on j

  - At the regression step j, we use the neural network of time t_j to compute conditional expectation. (Quick step)
- We obtain the price of american option for customer values r and σ.