Machine Learning in Finance:
A Q-Learner Based Approach to Pricing Financial Options

Russell Kim
email: *russellkim98@princeton.edu*

September 24, 2019

Abstract

This paper summarizes the discrete-time option pricing model by [1]. More specifically, we focus on the off-policy algorithm well known as the *Q-Learning* algorithm. In order to transform the original problem of pricing a Black Scholes based derivative, we stray from the academic continuous-time limit and use a discretized version of the classic Black Scholes model in order to construct a risk-neutral Markov Decision Process. Here, the option price of the MDP is an optimal Q-function. Pricing and hedging an option when the stock dynamics, such as the volatility, are unknown is now possible through the use of the Q-Learning algorithm. The model itself does not make any guesses or assumptions about the structure of the data it is given, and is guaranteed to asymptotically converge to the optimal solution given enough data and time. As a result, this suggests that the RL environment could be used to provide more optimal pricing of financial derivatives in a practical real world setting.

Introduction

The classical method of pricing BSM models in real world applications is Backwards Dynamic Programming. We use Markowian Portfolio Theory to rehedge our portfolio at each time interval. However, taking away the unrealistic assumption (of the BSM model) of continuous hedging means a finite rebalancing frequency. This finite rebalancing means that there is some mis-hedging risk for which the option buyer/seller should be compensated. Therefore, moving from continuous-time pricing to discrete-time pricing becomes more dependent on investors' risk preferences.

As a result, we construct a sequential decision-making process of minimization of slippage - this is the key objective of options trading and pricing: risk minimization by hedging in a sequential decision making process. Now we walk through this process.

QLBS

Using an MDP, we provide a discrete-time approximation to the hedging and pricing strategy of the BSM. In order to do this, we work in the physical measure $\mathbb{P}$ rather than the probabilistic measure $\mathbb{Q}$. When the transition probabilities and the reward function are given, we can use a recursive backward Value Iteration method to analytically derive a Bellman optimality equation. Optimizing this turns out to be a maximization of a quadratic function, and will be shown later(appendices).

However, in the much more likely scenario that we do not know the true stock price dynamics, we can solve a backwards recursion for the Bellman optimality equation with using only samples of data - otherwise known as **Reinforcement Learning**. This can also be solved semianalytically (due to a quadratic reward function) using the **Q-Learning** method put forth by Watkins, which is guaranteed to converge to the optimal hedge for any time step size given enough time and training data. If the stock price dynamics are lognormal, then the price given by our optimal value function.

State variables

Since stock prices usually include a deterministic drift, we "de-mean" the stocks so we have stationary temporal variables. We define

$$\begin{aligned} X_t &:= -\left(\mu - \tfrac{\sigma^2}{2}\right) t + \log S_t \\ dX_t &= -\left(\mu - \tfrac{\sigma^2}{2}\right) dt + d\log S_t = \sigma dW_t \end{aligned}$$

From this, we can see that X_t is a standard Brownian motion scaled by σ, and therefore a martingale. The state variable X_t is time uniform and the relation above can be used to map non-stationary dynamics of S_T into stationary dynamics of X_t. These relations can still be applied when the stock price dynamics are not lognormal - however, X_t will not be guaranteed to be a martingale in this situation. It is nevertheless very useful for separating non-stationarity of optimization from the non-stationarity of state variables.

Finding the Q-value Function

We then denote actions $a_t = a_t(X_t)$ and $u_t = u_t(S_t)$ to take (hedges in this situation).

$$u_t\left(S_t\right) = a_t\left(X_t\left(S_t\right)\right) = a_t\left(\log S_t - \left(\mu - \frac{\sigma^2}{2}\right)t\right)$$

We can generalize this to find a general *hedging strategy* for any state X_t, rather than worrying about individual hedging decisions $a_t(x_t)$ (which should be interpreted as a hedging decision made given x_t, which is an instance of the random state X_t at time t. Therefore, since $\{X_t\}_{t=0}^{T}$ is a time-homogeneous variable, we define a time-dependent policy

$$\pi : \{0, ..., T-1\} \times \mathcal{X} \to \mathcal{A}$$

This is a deterministic policy that maps the time t and the current state $X_t = x_t$ into the action (hedge) $a_t \in \mathcal{A}$:

$$a_t = \pi(t, x_t)$$

In order to optimize an MDP problem, we need to find the *objective*, or *Q-value* function to maximize. In this case, we are trying to minimize the slippage risk between the true fair price of the option and what the price is calculated as. We now derive this value function by considering an altered version of Markowian's Portfolio Theory.

Classical Markowian Portfolio Theory

A portfolio at time t is represented by

$$\Pi_t = u_t S_t + B_t$$

where

- Π_t = Value of Portfolio
- u_t = The amount of stock held
- S_t = Price of the stock held
- B_t = Value of a risk-free bank deposit

Self Financing Constraint

In order for this model to work we make the following assumption, which implies the following relation that ensures conservation of the portfolio value by a re-hedge at time $t+1$:

$$u_t S_{t+1} + e^{r\Delta t} B_t = u_{t+1} S_{t+1} + B_{t+1}$$

This condition ensures that any future changes in the hedge portfolio should be performed using the liquid assets in the bank.

Monte Carlo Simulation

Usually, values of hedge portfolios are done *pathwise* using Monte Carlo simulations, which takes as input a hedging strategy $\{u_t\}_{t=0}^{T}$ and gives as output the estimated option price. This is done by simulating many paths of the underlying stock price $S_1 \to S_2 ... \to S_N$, then recursively calculates Π_t going backwards using the well known equation

$$\Pi_t = e^{-r\triangle t}[\Pi_{t+1} - u_t \triangle S_t], \triangle S_t = S_{t+1} - e^{r\triangle t} S_t, t = T-1, ..., 0$$

However, this is all done when both a hedging strategy and a terminal condition are specified. In summary, we simulate many forward paths $S_1 \to S_2 ... \to S_N$, then the backwards path is performed using the recursive formula above. The terminal condition is obviously given, but how do we find the optimal hedging strategy?

Optimal Hedging Strategy

Unlike path-wise simulation as with calculating portfolio values, we use *cross-sectional* simulation that simulates every point in time for all Monte Carlo paths. By way of example, if we choose to simulate a million paths, at time t, we know all information about every one of those million paths, but only up to time t, since we cannot know the future when computing a hedge. We let

$$\mathcal{F}_t^{(n)} = \left\{ \left(X_t^{(n)}, a_t^{(n)}, R_t^{(n)}, X_{t+1}^{(n)} \right) \right\}_{t=0}^{T-1}, n = 1, \ldots, N_{MC}$$

which represents the cross-sectional information available up to time t. Since we want to find an optimal hedge for *every* possible situation, it is logical to minimize the variance of the portfolio given every possible situation. This leads to the optimal hedging strategy u_t^* conditioned on the available cross-sectional information $\mathcal{F}_t$ being calculated as such:

$$\begin{aligned} u_t^* (S_t) &= \arg\min_u \operatorname{Var} [\Pi_t | \mathcal{F}_t] \\ &= \arg\min_u Var [\Pi_{t+1} - u_t \Delta S_t | \mathcal{F}_t], \quad t = T-1, \ldots, 0 \end{aligned}$$

This equation implies that all uncertainty in the portfolio value is due to the uncertainty concerning the amount B_t required in the bank to cover any future hedges that may occur. As a result, the optimal hedge should minimize the cost of hedging (capital) for any such position at time t.

We take the derivative of the RHS of the above equation and set it to zero. This gives

$$u_t^\star (S_t) = \frac{\operatorname{Cov} (\Pi_{t+1}, \Delta S_t | \mathcal{F}_t)}{\operatorname{Var} (\Delta S_t | \mathcal{F}_t)}, \quad t = T-1, \ldots, 0$$

Now we have an equation describing the optimal hedge.

Fair option pricing

The theoretical fair price for an option, $\hat{C}_t$, is given by

$$\hat{C}_t = \mathbb{E}_t [\Pi_t | \mathcal{F}_t]$$

However, this is *not* the price that a seller of the option should charge. The actual fair *risk-adjusted* price is given by

$$C_0^{(ask)}(S,u) = \mathbb{E}_0\left[\Pi_0 + \lambda \sum_{t=0}^{T} e^{-rt} \operatorname{Var}\left[\Pi_t | \mathcal{F}_t\right] | S_0 = S, u_0 = u\right]$$

The term $\sum_{t=0}^{T} e^{-rt} \operatorname{Var}\left[\Pi_t | \mathcal{F}_t\right]$ is the sum of time-discounted variances of portfolio values, and λ is a risk parameter chosen by the option seller's risk preferences, as she has to take into account the risk of overdrawing from the bank when rehedging.

Value Function

The overall term $\lambda \sum_{t=0}^{T} e^{-rt} \operatorname{Var}\left[\Pi_t | \mathcal{F}_t\right]$ is referred to as the *risk premium*. Since buyers of the option will want the lowest price available, this becomes a situation of *minimizing* the fair option price. In other words, we can think about it as minimizing the *negative*. Thus we define our value function:

$$V_t\left(S_t\right) = \mathbb{E}_t\left[-\Pi_t - \lambda \sum_{t'=t}^{T} e^{-r\left(t'-t\right)} \operatorname{Var}\left[\Pi_{t'} | \mathcal{F}_{t'}\right] | \mathcal{F}_t\right]$$

We now rewrite this equation in terms of our state variable X_t and our state policy *pi*.

$$\begin{aligned} V_t^{\pi}\left(X_t\right) &= \mathbb{E}_t\left[-\Pi_t\left(X_t\right) - \lambda \sum_{t'=t}^{T} e^{-r\left(t'-t\right)} \operatorname{Var}\left[\Pi_{t'}\left(X_{t'}\right) | \mathcal{F}_{t'} | \mathcal{F}_t\right]\right. \\ &= \mathbb{E}_t\left[-\Pi_t\left(X_t\right) - \lambda \operatorname{Var}\left[\Pi_t\right] - \lambda \sum_{t'=t+1}^{T} e^{-r\left(t'-t\right)} \operatorname{Var}\left[\Pi_{t'}\left(X_{t'}\right) | \mathcal{F}_{t'}\right] | \mathcal{F}_t\right] \end{aligned}$$

where

$$-\lambda \mathbb{E}_{t+1}\left[\sum_{t'=t+1}^{T} e^{-r\left(t'-t\right)} \operatorname{Var}\left[\Pi_{t'} | \mathcal{F}_{t'}\right]\right] = \gamma\left(V_{t+1} + \mathbb{E}_{t+1}\left[\Pi_{t+1}\right]\right), \quad \gamma \equiv e^{-r\Delta t}$$

using the definition of the value function with a temporal shift. Rearranging the value equation, we obtain the value function for our MDP Black Scholes Model

$$V_t^{\pi}\left(X_t\right) = \mathbb{E}_t^{\pi}\left[R\left(X_t, a_t, X_{t+1}\right) + \gamma V_{t+1}^{\pi}\left(X_{t+1}\right)\right]$$

where the one step time-dependent random reward is as follows:

$$\begin{aligned} R_t\left(X_t, a_t, X_{t+1}\right) &= \gamma a_t \Delta S_t\left(X_t, X_{t+1}\right) - \lambda \operatorname{Var}\left[\Pi_t | \mathcal{F}_t\right], t = 0, \ldots, T-1 \\ &= \gamma a_t \Delta S_t\left(X_t, X_{t+1}\right) - \lambda \gamma^2 \mathbb{E}_t\left[\hat{\Pi}_{t+1}^2 - 2 a_t \Delta \hat{S}_t \hat{\Pi}_{t+1} + a_t^2\left(\Delta \hat{S}_t\right)^2\right] \end{aligned}$$

where we use the fact that $\Pi_t = e^{-r\triangle t}[\Pi_{t+1} - u_t \triangle S_t], \triangle S_t = S_{t+1} - e^{r\triangle t} S_t, t = T-1, ..., 0.$

(Although this model varies the reward function, there are other approaches to constructing a risk-sensitive MDP by modifying one-step variance penalties, which may be discussed later.)

We see that the expected reward is quadratic with respect to the action a_t:

$$\mathbb{E}_t\left[R_t\left(X_t, a_t, X_{t+1}\right)\right] = \gamma a_t \mathbb{E}_t\left[\Delta S_t\right] - \lambda\gamma^2 \mathbb{E}_t\left[\hat{\Pi}_{t+1}^2 - 2a_t\Delta\hat{S}_t\hat{\Pi}_{t+1} + a_t^2\left(\Delta\hat{S}_t\right)^2\right]$$

The action value function, or the **Q Function** as it is better known, is similar to the value function, but is conditioned on the current state X_t *and* the initial action or hedge a_t. It then follows a policy π:

$$Q_t^\pi(x,a) = \mathbb{E}_t\left[-\Pi_t\left(X_t\right)|X_t = x, a_t = a\right] - \lambda\mathbb{E}_t^\pi\left[\sum_{t'=t}^{T} e^{-r\left(t'-t\right)}\operatorname{Var}\left[\Pi_{t'}\left(X_{t'}\right)|\mathcal{F}_{t'}\right]|X_t = x, a_t = a\right]$$

The optimal value function satisfies the Bellman optimality equation

$$V_t^\star\left(X_t\right) = \mathbb{E}_t^{\pi^*}\left[R_t\left(X_t, u_t = \pi_t^*\left(X_t\right), X_{t+1}\right) + \gamma V_{t+1}^\star\left(X_{t+1}\right)\right]$$

The Bellman optimality equation for the action-value function reads

$$Q_t^*(x,a) = \mathbb{E}_t\left[R_t\left(X_t, a_t, X_{t+1}\right) + \gamma\max_{a_{t+1}\in\mathcal{A}} Q_{t+1}^*\left(X_{t+1}, a_{t+1}\right)|X_t = x, a_t = a\right], t = 0$$

where the terminal condition at $t = T$ is given by

$$Q_T^\star\left(X_T, a_T = 0\right) = -\Pi_T\left(X_T\right) - \lambda\operatorname{Var}\left[\Pi_T\left(X_T\right)\right]$$

We already know that $\Pi_T = B_T$, since it has been given.

To summarize, we framed the classic BSM model as an MDP such that the Q-value function to be maximized is the negative of the risk-adjusted fair option price, giving explicit formulas for functions like the reward function.

Optimal Policy

Now the optimal policy is the one that maximizes the action-value function $Q_t^\pi(x,a)$. It is given by

$$\pi_t^\star\left(X_t\right) = \operatorname{argmax}_\pi V_t^\pi\left(X_t\right) = \operatorname{argmax}_{a_t\in\mathcal{A}} Q_t^*\left(X_t, a_t\right)$$

Then, we use the expected reward (calculated above) and substitute it into the Bellman optimality equation:

$$\begin{aligned} Q_t^*\left(X_t, a_t\right) = {} & \gamma\mathbb{E}_t\left[Q_{t+1}^*\left(X_{t+1}, a_{t+1}^*\right) + a_t\Delta S_t\right] \\ & - \lambda\gamma^2\mathbb{E}_t\left[\hat{\Pi}_{t+1}^2 - 2a_t\hat{\Pi}_{t+1}\Delta\hat{S}_t + a_t^2\left(\Delta\hat{S}_t\right)^2\right], t = 0, \ldots, T-1 \end{aligned}$$

We follow the standard assumption of BSM that no individual seller/buyer of options produces market impact. Then, we can see that the term $\mathbb{E}_t\left[Q_{t+1}^*\left(X_{t+1}, a_{t+1}^*\right)\right]$ depends only on the current action a_t through decomposing the expectation and looking at the conditional probability $p(X_{T+1}|X_t a_t)$. The next state probability can only depend on the current action when there is an impact of the action on the market, which we have just assumed there is not. Therefore, the next state probability does not depend on the

current action. As follows, the above Q function is a quadratic function of a_t and we can calculate the hedge $a_t^*(S_t)$ that maximizes this Q function analytically:

$$a_t^*\left(X_t\right) = \frac{\mathbb{E}_t\left[\Delta\hat{S}_t\hat{\Pi}_{t+1} + \frac{1}{2\gamma\lambda}\Delta S_t\right]}{\mathbb{E}_t\left[\left(\Delta\hat{S}_t\right)^2\right]}$$

Taking the limit of this as $\triangle t \to 0$ by using Taylor expansions results in

$$\lim_{\Delta t\to 0} a_t^\star = \frac{\partial\hat{C}_t}{\partial S_t} + \frac{\mu - r}{2\lambda\sigma^2}\frac{1}{S_t}$$

If we set $\mu = r$ (or equivalently $\lambda \to \infty$), two things happen. The limit above becomes identical to the BS delta. Secondly, the analytical solution to the optimal hedge includes the finite $\triangle t$ delta which turns into a local-risk minimization delta given by

$$u_t^*\left(S_t\right) = \frac{\mathrm{Cov}\left(\Pi_{t+1}, \Delta S_t|\mathcal{F}_t\right)}{\mathrm{Var}\left(\Delta S_t|\mathcal{F}_t\right)}, \quad t = T-1, \ldots, 0$$

What this means is that the quadratic hedging that approximates option delta looks only at how risky a portfolio when determining the optimal hedge. However, the hedge maximizing the Q-value function above adds a **drift term** $\mathbb{E}[\Pi_t]$ to the objective function in a similar way to Markowitz risk-adjusted portfolio return analysis. This leads to a linear first term in the quadratic expected reward. **Hedges maximizing the Q-value function are therefore different from hedges obtained by only minimizing risk.**

Q-Learning and Iteration

Now how do we approximate this unknown expectation seen in the Q-function from data? We use the Robbins-Monro approximation to write out an explicit update rule given in terms of the already observed datapoint.

After observing the datapoint $\left(X_t^{(n)}, a_t^{(n)}, R_t^{(n)}, X_{t+1}^{(n)}\right)$, we use this recursive equation:

$$Q_t^{\star,k+1}\left(X_t, a_t\right) = \left(1-\alpha^k\right)Q_t^{\star,k}\left(X_t, a_t\right) + \alpha^k\left[R_t\left(X_t, a_t, X_{t+1}\right) + \gamma\max_{a_{t+1}\in\mathcal{A}} Q_{t+1}^{\star,k}\left(X_{t+1}, a_{t+1}\right)\right]$$

which is guaranteed to asymptotically converge to a true optimal action-value function, without making any assumptions about the data-generating distribution that produced the datapoint $\left(X_t^{(n)}, a_t^{(n)}, R_t^{(n)}, X_{t+1}^{(n)}\right)$ (which is a property of the Q-learning algorithm). It takes the datapoint and updates the Q-action-value function for the state-action tuple $\left(X_t^{(n)}, a_t^{(n)}\right)$. Ideally, the algorithm should take in as input

1. Data in the form of historical(or simulated) stock prices and times
2. The risk parameter λ (dependent on how risk averse the option seller/buyer is)
3. The time period
4. Number of MC paths to simulate

5. The type of financial derivative to be priced

and should give as output the optimal Q-function, and therefore the optimal risk-adjusted option price as well.

However, even though the Q-learning algorithm is computationally efficient with respect to other similar RL algorithms, it stil may be too slow in practice to use the pure vanilla model, since we would be taking cross-sectional information from a very large number of simulations at the same time t. This cross-sectional information would not be available for any online learning setting of updating Q-values and optimal hedge rations $a_t^*(X_t)$ on a space grid. The solution to this is to update these values *for all points on the grid simultaneously* by looking at a snapshot of all Monte Carlo paths at time t. This is possible since we are in a **batch-mode** setting of RL, and the data is readily available.

Batch Q-Learning

A much faster way of computing the Q-values and optimal hedge ratios is given by the Fitted Q Iteration algorithm, which uses basis functions to approximate the updated function at each time step. This model is now generalized to a continuous-state space, since the only difference between a continuous-state space and a discrete-state specification is the choice of basis function.

The main idea behind Fitted Q Iteration is that we can model the updated Q-value-action Function as a series of linear regressions, where the regression models themselves are polynomial splines. This is possible since the Q-function $Q_t^\star\left(X_t, a_t\right)$is quadratic with respect to a_t, and therefore can be decomposed as an expansion of basis functions $\{\Phi_n(x)\}$, with time-dependent coefficients which are parameterized by some matrix $\mathbf{W}_t$. Optimizing the choice of basis functions is a topic for further research, but cubic or quartic splines will provide enough complexity to model the data.

$$
\begin{aligned}
Q_t^*\left(X_t, a_t\right) &= \left(\begin{array}{c} 1, a_t, \frac{1}{2}a_t^2 \end{array}\right)\left(\begin{array}{cccc} W_{11}(t) & W_{12}(t) & \cdots & W_{1M}(t) \\ W_{21}(t) & W_{22}(t) & \cdots & W_{2M}(t) \\ W_{31}(t) & W_{32}(t) & \cdots & W_{3M}(t) \end{array}\right)\left(\begin{array}{c} \Phi_1\left(X_t\right) \\ \vdots \\ \Phi_M\left(X_t\right) \end{array}\right) \\
&\equiv \mathbf{A}_t^T \mathbf{W}_t \Phi\left(X_t\right) \equiv \mathbf{A}_t^T \mathbf{U}_W\left(t, X_t\right)
\end{aligned}
$$

which can be rearranged into an product of a parameter vector and a state-action dependent vector:

$$
\begin{aligned}
Q_t^\star\left(X_t, a_t\right) &= \mathbf{A}_t^T \mathbf{W}_t \mathbf{\Phi}(X) = \sum_{i=1}^{3}\sum_{j=1}^{M}\left(\mathbf{W}_t \odot \left(\mathbf{A}_t \otimes \mathbf{\Phi}^T(X)\right)\right)_{ij} \\
&= \overrightarrow{\mathbf{W}}_t \cdot vec\left(\mathbf{A}_t \otimes \mathbf{\Phi}^T(X)\right) \equiv \overrightarrow{\mathbf{W}}_t \vec{\Psi}\left(X_t, a_t\right)
\end{aligned}
$$

Note: $\odot$ stands for an element-wise (Hadamard) product of two matrices.

The vector of time- dependent parameters $\overrightarrow{\mathbf{W}}_t$ is obtained by concatenating columns of matrix $\mathbf{W}_t$, and similarly, $\vec{\Psi}\left(X_t, a_t\right) = vec\left(\mathbf{A}_t \otimes \mathbf{\Phi}^T(X)\right)$ stands for a vector obtained by concatenating columns of the outer product of vectors $\mathbf{A}_t$ and $\mathbf{\Phi}(X)$.

Then, we recursively calculate $\mathbf{W_t}$ backwards in time for $t = T-1, ..., 0$. As a result, we can express the

Bellman Optimality Equation (reproduced below for convencience) as a regression.

$$Q_t^\star(x,a) = \mathbb{E}_t\left[R_t\left(X_t, a_t, X_{t+1}\right) + \gamma \max_{a_{t+1}\in\mathcal{A}} Q_{t+1}^\star\left(X_{t+1}, a_{t+1}\right) | X_t = x, a_t = a\right], t = 0, \ldots, T-1$$

The term inside the expectation is regressed as follows:

$$R_t\left(X_t, a_t, X_{t+1}\right) + \gamma \max_{a_{t+1}\in\mathcal{A}} Q_{t+1}^\star\left(X_{t+1}, a_{t+1}\right) = \overrightarrow{\overline{\mathbf{W}}}_t \vec{\Psi}\left(X_t, a_t\right) + \varepsilon_t$$

with random noise ϵ_t with mean 0 at time t. Therefore,

$$Q_t^\star(x,a) = \mathbb{E}[\overrightarrow{\overline{\mathbf{W}}}_t \vec{\Psi}\left(X_t, a_t\right) + \varepsilon_t]$$

Actually calculating the coefficients $\mathbb{W}_{\approx}$ are found by solving a simple **least squares** optimization problem:

$$\mathcal{L}_t\left(\mathbf{W}_t\right) = \sum_{k=1}^{N_{MC}} \left(R_t\left(X_t, a_t, X_{t+1}\right) + \gamma \max_{a_{t+1}\in\mathcal{A}} Q_{t+1}^\star\left(X_{t+1}, a_{t+1}\right) - \overrightarrow{\overline{\mathbf{W}}}_t \overrightarrow{\overline{\mathbf{\Psi}}}\left(X_t, a_t\right)\right)^2$$

Solving this equation gives

$$\overrightarrow{\overline{\mathbf{W}}}_t^\star = \mathbf{S}_t^{-1}\mathbf{M}_t$$

where

$$S_{nm}^{(t)} = \sum_{k=1}^{N_{MC}} \Psi_n\left(X_t^k, a_t^k\right) \Psi_m\left(X_t^k, a_t^k\right)$$

$$M_n^{(t)} = \sum_{k=1}^{N_{MC}} \Psi_n\left(X_t^k, a_t^k\right) \left(R_t\left(X_t^k, a_t^k, X_{t+1}^k\right) + \gamma \max_{a_{t+1}\in\mathcal{A}} Q_{t+1}^k\left(X_{t+1}^k, a_{t+1}\right)\right)$$

Finding $\max_{a_{t+1}\in\mathcal{A}} Q_{t+1}^\star\left(X_{t+1}^k, a_{t+1}\right)$ is now easy since we know $\mathbb{W}_{t+1}$ and therefore we know

$$\mathbf{U}_W(t+1, X_{t+1}) = \mathbf{W}_{t+1}\mathbf{\Phi}(X_{t_+1})$$

Therefore, we can express the updated Q-function as

$$Q_{t+1}^*(X_{t+1}, a^*t+1) = \mathbf{U}_W^{(0)}(t+1, X_{t+1}) + a_{t+1}^*\mathbf{U}_W^{(1)}(t_1, X_{t+1}) + \frac{(a_{t+1}^*)^2}{2}\mathbf{U}_W^{(2)}(t+1, X_{t+1})$$

Important note: this is a quadratic equation with respect with to a^*t+1, but it is incorrect to use the maximum as a function of a^*t+1 to solve for the optimal value. This would mean that the algorithm uses the same dataset to estimate both the optimal action and the optimal Q-function. This leads to overestimation of

$$\max_{a_{t+1}\in\mathcal{A}} Q_{t+1}^k\left(X_{t+1}^k, a_{t+1}\right)$$

due to Jensen's inequality and the convexity of the $\max(\cdot)$ function. The correct calculation of the optimal action a_{t+1}^* would be to expand the optimal action in terms of basis functions, much like we did with the Q-function.

Estimating the optimal hedge

We express the optimal hedge as

$$a_t^*(X_t) = \sum_n^M \phi_{nt} \mathbf{\Phi_n}(X_t)$$

where we use the same basis functions as before. For convenience, the Q-function is reproduced below:

$$\begin{aligned} Q_t^*\left(X_t, a_t\right) = \gamma \mathbb{E}_t\left[Q_{t+1}^*\left(X_{t+1}, a_{t+1}^*\right) + a_t \Delta S_t\right] \\ - \lambda\gamma^2 \mathbb{E}_t\left[\hat{\Pi}_{t+1}^2 - 2a_t\hat{\Pi}_{t+1}\Delta\hat{S}_t + a_t^2\left(\Delta\hat{S}_t\right)^2\right], t = 0, \ldots, T-1 \end{aligned}$$

We find ϕ_{nt} as follows:

1. Replace the expectation in the equation above by an MC estimate
2. Drop all a_t-independent terms
3. Substitute the expansion $a^*t(X_t) = \sum_n^M \phi_{nt}\mathbf{\Phi_n}(X_t)$ for a_t
4. Change the overall sign to change from maximization to minimization

This equivalently leads to minimization of the following equation:

$$G_t(\phi) = \sum_{k=1}^{N_{MC}} \left(-\sum_n \phi_{nt}\mathbf{\Phi}_n(X_t^k)\triangle S_t^k + \gamma\lambda(\hat{\Pi}_{t+1}^k - \sum_n \phi_{nt}\mathbf{\Phi}_n(X_t^k)\triangle\hat{S}_t^k)^2 \right)$$

Minimizing this with respec to the coefficients ϕ_{nt} leads to the following set of linear equations:

$$\sum_m^M A_{nm}^{(t)}\phi_{mt} = B_n^{(t)}, n = 1, ..., M$$

where

$$\mathbf{A_{nm}^{(t)}} = \sum_{k=1}^{N_{MC}} \mathbf{\Phi}_n(X_t^k)\mathbf{\Phi}_m(X_t^k)(\triangle\hat{S}_t^k)^2$$

$$\mathbf{B_{nm}^{(t)}} = \sum_{k=1}^{N_{MC}} \mathbf{\Phi}_n(X_t^k)\left[\hat{\Pi}_{t+1}^k\triangle\hat{S}_t^k + \frac{1}{2\gamma\lambda}\triangle S_t^k\right]$$

which leads to the solution for the coefficients ϕ_t^* of the optimal action $a_t^*(X_t)$ as

$$\phi_t^* = \mathbf{A_t^{-1}B_t}$$

where $\mathbf{A}$ and $\mathbf{B}$ are given above. Now we are done.

Note: since an analytical optimal action is available, some problems that arise in classical Q-learning are avoided, such as the potential overestimation of the optimal action. Usually, this is addressed by using a variant of the Q-learning method, like Double Q-Learning. However, this issue is avoided in this QLBS model.

Quick Summary

Essentially, this model allows us to price and hedge a stock option by learning **directly from past trading data**. It implements the principle of hedging first and pricing second in a consistent way for a discrete-time version of the Black Scholes model.

If we have an options pricing model where the option price is the negative of a Q-function, and the option hedge is its second argument, the optimal action value Q-function hedges and prices the option **by definition.** The only modelling assumption is that local rewards (or negative losses) are quadratic in terms of hedging actions - this leads to a model/distribution free solution. The only thing that the Q-learning algorithm needs is a dataset in the form of cross-sectional tuples

$$\mathcal{F}_t^{(n)} = \left\{ \left(X_t^{(n)}, a_t^{(n)}, R_t^{(n)}, X_{t+1}^{(n)} \right) \right\}_{t=0}^{T-1}, n = 1, \ldots, N_{MC}$$

across all simulations. It eliminates the need for any volatility surface models or jump-diffusion models. As long as rewards are quadratic with respect to the action taken, the algorithm produces model-independent results that define the optimal price and the optimal hedge.

Since the Q-Learner is an *off policy* algorithm, it will still learn the optimal price/hedge from training data given enough data *even if the actions taken are completely random.* As a result, it is possible to create synthetic trading histories by combining real-market data (stock prices) with a pre-defined trading strategy in stocks.

From a financial point of view, the Black Scholes model washes out any risk out of rehedging a portfolio in lognormal dynamics when the rehedging time step $\triangle t$ is taken to the limit $\triangle t \to 0$

Possible future research directions

Over the course of the year, plans to conduct further research on this topic will start out with a small scope and extending it if time permits. The following will be the order in which results will be given.

Numerical experiments on options

Rigorous and extensive testing with synthetic data will be done to examine the differences between the QLBS price and the Black Scholes price for the following options in this order:

1. European options
2. American options
3. Barrier options
4. Asian options
5. Financial derivatives that have multiple underlying assets
6. Physical options such as oil/gas futures

For each of these options, there will be a different Bellman Optimality equation due to the mechanics of the option itself which will have to be solved analytically in order to guarantee quadratic rewards. The American option, for instance, can choose to either rehedge or exit the position entirely ahead of the expiration date, an action that makes it difficult to price. The interesting aspect about this QLBS model is that it could theoretically rediscover the Black-Scholes formula for the option price by using the QLBS by analyzing data collected with random actions (since it is an off policy method) and using a Neural Network to approximate the value function. However, the question remains as to whether this method is numerically stable in real world applications. As a result, after testing on synthetic data, we propose taking historical stock data and comparing our QLBS calculated option price to the equivalent historical option prices on the market. This can lead to further insights about the financial markets as well from a data science perspective. If our calculated price consistently is lower than the real option price for stocks in a certain industry, then it may be an indication that the market was too bearish at the time and was consistently overpricing stocks due to lack of investor confidence.

Optimizing risk preferences

The parameter λ is part of the *risk premium*, the term that determines how risk-averse the seller of the option is. Usually, this is a small positive number. We propose historical stock analysis from a large number of stocks across multiple industries for the past 5 years in order to determine what historical average λ would be in the market today. This could, in a sense, be called the "fair" or "risk-neutral" (not mathematically speaking) value to be compared against when pricing options using the QLBS model.

References

[1] Igor Halperin. "QLBS: Q-Learner in the Black-Scholes(-Merton) Worlds". In: *arXiv e-prints*, arXiv:1712.04609 (Dec. 2017), arXiv:1712.04609. arXiv: `1712.04609 [q-fin.CP]`.