Reinforcement Learning for Options Pricing under discrete hedging

Vitaly Minkevich[*]

New Economic School

February 4, 2021

Abstract

In this paper, I investigate optimal option pricing in discrete dynamic hedging case using ML approach. For discrete hedging, it is impossible to hedge perfectly, which is equivalent to the situation of incomplete market. Wilmott (1994) derived the correction to the option price in this situation (assuming an options dealer is risk-neutral). Also recently, there have been several papers applying ML methods to solve this problem. One of them is the QLBS model by Halperin (2019) based on reinforcement learning. In this work, I realized learning an optimal option price and its optimal hedging based on the Q-learning Black-Scholes (QLBS) algorithm in comparison with Black-Scholes model with adjusted volatility and test them for equivalence. As the result, I demonstrate that the problem of hedging and pricing in discrete time with Wilmott‘s volatility adjustment is

[*] Master’s Student at NES. Email: vminkevich@nes.ru

equivalent to finding optimal price and hedge received Halperin‘s algorithm (QLBS) with some assumptions.

Contents

1 Introduction

An optimal option price and its optimal hedging are fundamental research topics in Mathematical Finance. The use of sophisticated models based on Artificial intelligence (AI) approach for valuing derivatives is becoming a necessary condition to be effective in the market. Artificial intelligence (AI) is making its impact on many areas of finance, particularly pricing and hedging. A diverse range of artificial intelligence sub-fields such as deep learning, reinforcement learning, and computer vision are currently being utilized to predict asset movements. However, the use of machine learning methods is often criticized due to weak theoretical justification and lack of assumptions based on Theory of Finance.

Reinforcement learning as a branch of machine learning in which an agent learns how to act with an particular environment for maximizing its total reward, which is defined in relationship to the actions it takes. The enthusiast who implemented of RL algorithms into optimal pricing under discrete hedging was Halperin with his work Halperin (2019). It is alternative approach for pricing and hedging discretely in time considered in the paper Wilmott (1994).

2 Literature review

In this section I describe Black-Scholes model with volatility adjustment for discrete pricing and hedging from Wilmott (1994) and the combination of Q-learning algorithm and BSM model as a Dynamic Programming problem with discrete-time stochastic setup and continuous state space described described in this paper Halperin (2019).

Different types of models can be used for describing the price of an option. The original model is Geometric Brownian Motion (GBM). It is the most simple and popular one which models how stock prices move.

In the most popular option pricing model Black-Scholes-Merton (BSM), the $S_t \sim$ GBM assumption holds, where S_t is underlying asset. In the article Wilmott (1994) the problem of option hedging in discrete time is considered. As mentioned earlier, analysis through a pure BSM requires prolonged hedging, which is acceptable in theory but not in practice. The model for discrete hedging is to rehedge at fixed intervals of time δt. A strategy commonly used with ranging from one day to a week. Some of the ideas in this article can be found in P. Boyle (1980). They considered the errors in following a pure Black-Scholes hedging strategy discretely in detail. Moreover, Wilmott did some correction of $\sigma^{*}{=}\sigma\left(1+\frac{\delta \mathrm{t}}{2\sigma^2}(\mu-r)\left(3(\mu-r)+\sigma^2\right)\right)$. He also supposed that continuous-time stochastic differential equation for motion of underlying asset $\mathrm{dS} = \mu \mathrm{Sdt} + \sigma \mathrm{SdX}$ can be replaced with a discrete version $S = e^x$, where $\delta x = \left(\mu - \frac{\sigma^2}{2}\right)\delta t + \sigma\phi\delta t^{1/2}$, $\phi \sim N(0,1)$. The detailed formula derivation we can see in this paper Wilmott (1994).

On the other hand, we have a modern ML approach for the discrete-time option pricing described in Halperin (2019). In this approach, author derived risk-adjusted Markov Decision Process for the discrete version of BSM model. The Q-learning Black-Scholes (QLBS) model is a discrete-time option hedging and pricing model which is based on Dynamic Programming (DP) and Reinforcement Learning (RL). It combines the famous Q-Learning method from RL with the Black-Scholes model's idea of reducing the problem of option pricing and hedging to the problem of optimal rebalancing of a dynamic replicating portfolio for the option. Portfolio is made of a stock and cash. The option price is an optimal Q-function with the optimal hedge as an argument, so that both the price and hedge are parts of the same formula. Using Q-learning method he learnt the optimal price and optimal hedge directly from data, without knowing model of world. I will explain the model in detail further.

3 Theoretical model

In this section, I give the intuition of Reinforcement Learning problem, the link between Reinforcement Learning and Dynamic Programming, describe the QLBS model as a Dynamic Programming problem with discrete-time stochastic setup and continuous state space. I also describe synthetic data generated by Monte-Carlo simulation.

Reinforcement learning is a type of machine learning approach that consider a sequence of decisions have to be made optimally in a stochastic and dynamic environment. At every moment of time t, there are a number of states S_t and a number of possible actions A_t. The agent takes an action, A_0 when the state S_0 is known. This results in a reward, R_1, at time 1 and a new state, S_1. The agent then takes new action, A_1 which results in a reward, R_2 at time 2 and a new state, S_2; and so on. The aim of reinforcement learning is to maximize expected discounted future payoff (reward), G_t. Figure 2 The process of learning is illustrated in Figure 2.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-1} R_T \tag{3.1}$$

where T is a horizon date and γ from $(0; 1]$ is a discount factor. We define R_t as the the cash flow received at time t multiplied by γ (i.e., discounted by one period). It needs for ensuring that (2.10) showed the time value of money.

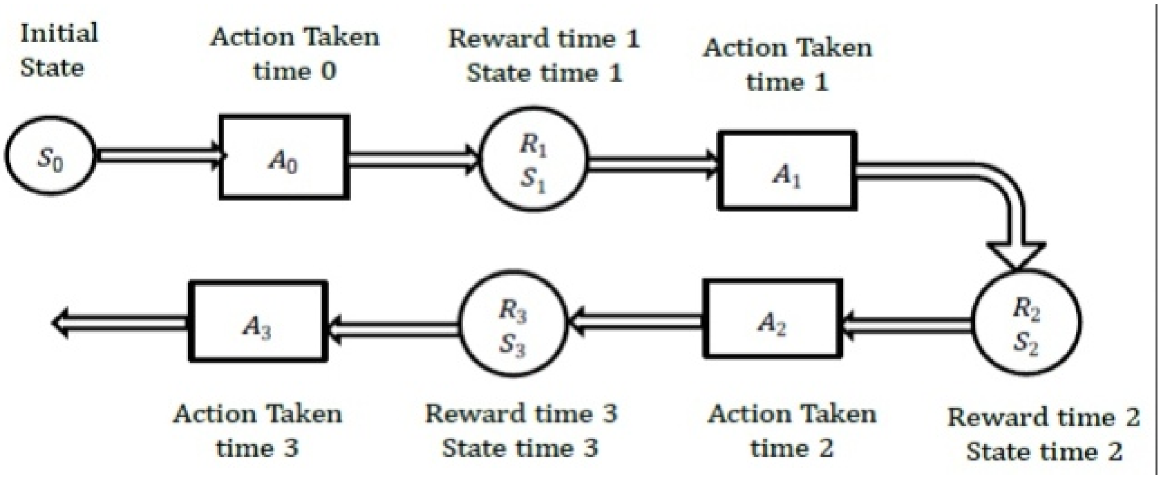

Figure 1: S_t is the state at time t, A_t is the decision made at time t, and R_{t+1} is the resulting payoff at time $t+1$.

For maximizing expected discounted future reward G_t, the agent needs to set of some rules for what decision to make in any given state, for any time moment t. This set of rules is called a policy function $\pi : \mathcal{S} \rightarrow \mathcal{A}$, where $\mathcal{A}$ and $\mathcal{S}$ are sets of all possible actions and states of the world, not necessarily known. Agent use policy π for a given state S_t at time t, then the decision made is $A_t = \pi(S_t)$. The policy function π is updated with a progress of RL algorithm (Q-learning, in our case).

For any policy π and for a given state S_t, we define the value of taking action A_t as a $\mathbb{E}\left(G_t \mid S_t, A_t\right)$ - expected discounted total reward. The value of each pair (S_t, A_t) is represented by a Q-function, $Q : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$.

$$\mathbb{E}\left(G_t \mid S_t, A_t\right) = Q(S_t, A_t) \tag{3.2}$$

The Q-function is learnt from a big numbers of pairs such that (S_t, A_t). They are also referred as episodes and are generated from either historical or simulated data. If the agent who make a decision optimally has a good estimate of the value of each action A_t for any state S_t for any $t \in T$, then the decision maker can compare policies in effective way. With the growth of episodes, the quality of the choice of the optimal policy increases, as more and more the agent learns about the stochastic nature of the environment.

The agent generally starts with a random policy $\hat{\pi}$. It allows the agent to investigate the environment. She observes the payoff received when choosing certain actions in the states and updates the Q-function estimates after each episode of interaction. The decision maker then updates the policy $\hat{\pi} \rightarrow \hat{\pi}_{updated}$, s.t. it is consistent with the new Q-function estimates. The process of updating is stopped when the agent will be enough confident that the optimal policy has been found.

For discrete time we usually need to solve following maximization problem:

$$A_t = \pi^*(S_t) = \arg\max_{a \in \mathcal{A}} Q\left(S_t, a\right) \tag{3.3}$$

Previously, the problem of learning the optimal policy $\pi^* = Q(S, A)$ was considered when action space A is discrete. Watkins (1989) introduced the approach of solving maximization problem for continuous action space A. This method allows the agent to explore the environment through sub-optimal policy.

In this approach, Q-function $Q(A_t, S_t)$ is updated in following rule:

$$Q\left(S_t, A_t\right) \leftarrow Q\left(S_t, A_t\right) + \alpha\left(R_{t+1} + \gamma \max_{a \in \mathcal{A}} Q\left(S_{t+1}, a\right) - Q\left(S_t, A_t\right)\right) \tag{3.4}$$

where $\alpha \in (0;1]$ is a constant parameter and γ is the discount factor.

Intuitively, the agent starts with the state S_t, chooses action A_t based on the currently policy function π and observes ahead the payoff R_{t+1}. The next step is update of the value $Q(S_t, A_t)$. Update is realized according to the expected reward $\mathbb{E}\left(G_t \mid S_t, A_t\right)$ if the current optimal policy π^* is followed as $A_t = \pi^*(S_{t+1})$.

The algorithm of solving an MDP problem is to solve the following Stochastic Bellman equation:

$$Q_t^\star(s,a) = \mathbb{E}_t\left[R_t\left(S_t, A_t, S_{t+1}\right) + \gamma \max_{A_{t+1}\in\mathcal{A}} Q_{t+1}^\star\left(S_{t+1}, A_{t+1}\right) | S_t = s, A_t = a\right], t = 0, ..., T-1 \tag{3.5}$$

where γ is continuous discount factor, $R_t\left(S_t, A_t, S_{t+1}\right)$ is the one-step random payoff with time dependency and $A_t\left(S_t\right)$ is the agent‘s action. In other words, we define hedge as an action.

With this set of basis functions $\left\{\Gamma_n\left(S_t\right)\right\}_{n=1}^N$ for $n = 1, ..., N$; $t = T-1, .., 0$, expand the optimal hedge $A_t^\star\left(S_t\right)$ and optimal Q-function (option price) $Q_t^\star\left(S_t, A_t^\star(S_t)\right)$ in basis functions with time-dependent coefficients.

$$A_t^\star\left(S_t\right) = \sum_n^N \phi_{nt}\Gamma_n\left(S_t\right) \qquad Q_t^\star\left(S_t, A_t^\star(S_t)\right) = \sum_n^N \omega_{nt}\Gamma_n\left(S_t\right)$$

.

To find optimal option price $Q_t^*(S_t, A_t^*(S_t))$ and optimal hedge $A_t^*(S_t)$, we need to find

$\omega_{nt} = \mathbf{C}_t^{-1}\mathbf{D}_t$ and $\phi_{nt} = \mathbf{A}_t^{-1}\mathbf{B}_t$ which are computed recursively backward in time, where $\mathbf{A}_t$, $\mathbf{B}_t$, $\mathbf{C}_t$ and $\mathbf{D}_t$ are matrices and vectors respectively with elements given by:

$$A_{nm}^{(t)} = \sum_{k=1}^{N_{MC}} \Gamma_n\left(S_t^k\right)\Gamma_m\left(S_t^k\right)\left(\Delta\hat{P}_t^k\right)^2 \qquad B_n^{(t)} = \sum_{k=1}^{N_{MC}} \Gamma_n\left(S_t^k\right)\left[\hat{\Pi}_{t+1}^k\Delta\hat{P}_t^k + \frac{1}{2\gamma\lambda}\Delta P_t^k\right] \tag{3.6}$$

And the equations for the matrix C and the vector D, respectively:

$$C_{nm}^{(t)} = \sum_{k=1}^{N_{MC}} \Gamma_n\left(S_t^k\right)\Gamma_m\left(S_t^k\right) \qquad D_n^{(t)} = \sum_{k=1}^{N_{MC}} \Gamma_n\left(S_t^k\right)\left(R_t\left(S_t, A_t^\star(S_t), S_{t+1}\right) + \gamma \max_{A_{t+1}\in\mathcal{A}} Q_{t+1}^\star\left(S_{t+1}, A_{t+1}\right)\right) \tag{3.7}$$

where $\Delta P_t = P_{t+1} - e^{-r\Delta t}P_t \quad t = T-1, ..., 0$; $\Delta\hat{P}_t$ is the sample mean of all values of ΔP_t. The complete solution of this functional equation is given in Halperin (2019).

By Halperin, the following relation is required for ensuring conservation of the portfolio value by a re-hedge at time $t+1$:

$$u_t P_{t+1} + e^{r\Delta t}H_t\left(P_t\right) = u_{t+1}P_{t+1} + H_{t+1}\left(P_{t+1}\right) \tag{3.8}$$

where $H_T\left(P_T\right)$ is condition on equality of portfolio and payoff which should be hold.

Doing some algebra we can derive a recursive relation for Π_t in terms of its values at later times, which can therefore be solved backward in time, starting from $t = T$ with the terminal condition (2), and continued all the way to the current time $t = 0$:

$$\Pi_t = e^{-r\Delta t}\left[\Pi_{t+1} - u_t\Delta P_t\right], \quad \Delta P_t = P_{t+1} - e^{r\Delta t}P_t, \quad t = T-1, \ldots, 0 \tag{3.9}$$

Option pricing occures by following equation:

$$C_0(P, u) = \mathbb{E}_0 \left[\Pi_0 + \lambda \sum_{t=0}^{T} e^{-rt} \operatorname{Var}\left[\Pi_t \mid \mathcal{F}_t\right] \mid S_0 = S, u_0 = u \right] \tag{3.10}$$

where $u_t^{\star}\left(S_t\right) = \arg\min_u \operatorname{Var}\left[\Pi_t \mid \mathcal{F}_t\right] = \frac{\operatorname{Cov}(\Pi_{t+1}, \Delta S_t | \mathcal{F}_t)}{\operatorname{Var}(\Delta S_t | \mathcal{F}_t)}, \quad t = T-1, \ldots, 0.$

Learning optimal option price $C_0(P, u)$ is equivalent to problem of maximization value function $-V_t(P_t)$:

$$-V_t\left(P_t\right) = -\mathbb{E}_t \left[-\Pi_t - \lambda \sum_{t'=t}^{T} e^{-r(t'-t)} \operatorname{Var}\left[\Pi_{t'} \mid \mathcal{F}_{t'}\right] \mid \mathcal{F}_t \right] \tag{3.11}$$

where $\mathcal{F}_t$ is available cross-sectional information

Next step, we need to test the following approach on a synthetic data and compare with the results received by BSM model for discrete-time case. Let‘s start with the BSM formula as a stochastic differential equation:

$$dP_t = rP_t dt + \sigma P_t dZ_t \tag{3.12}$$

Where P_t stock price, r risk-free rate, σ volatility, Z_t Brownian motion.

By Euler Discretization Scheme, we solve the equation. The solution is given by the expression:

The MC simulation of stock prices for BSM model is performed by this formula:

$$P_{t+1} = P_t \exp\left(\left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t} z_t\right) \tag{3.13}$$

After taking $log(P_{t+1})$ we will get:

$$\log P_{t+1} = \log P_t + \left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t} z_t \tag{3.14}$$

Based on log-simulation, we can calculate current state of the environment S_t:

$$S_t = \log P_t - \left(\mu - \frac{1}{2}\sigma^2\right) t \tag{3.15}$$

We also need to compute ΔP_t and $\Delta\bar{P}_t$ as sample mean of all possible ΔP_t.

Let‘s look at the plots of stock price P_t and state variable S_t with 100 different paths with these parameters:

Parameter	
Initial Stock Price:	100
Drift:	0.15
Volatility:	0.45
Risk-free Rate:	0.01
Risk aversion parameter (λ):	0.0001
Strike:	90
Maturity:	0.5

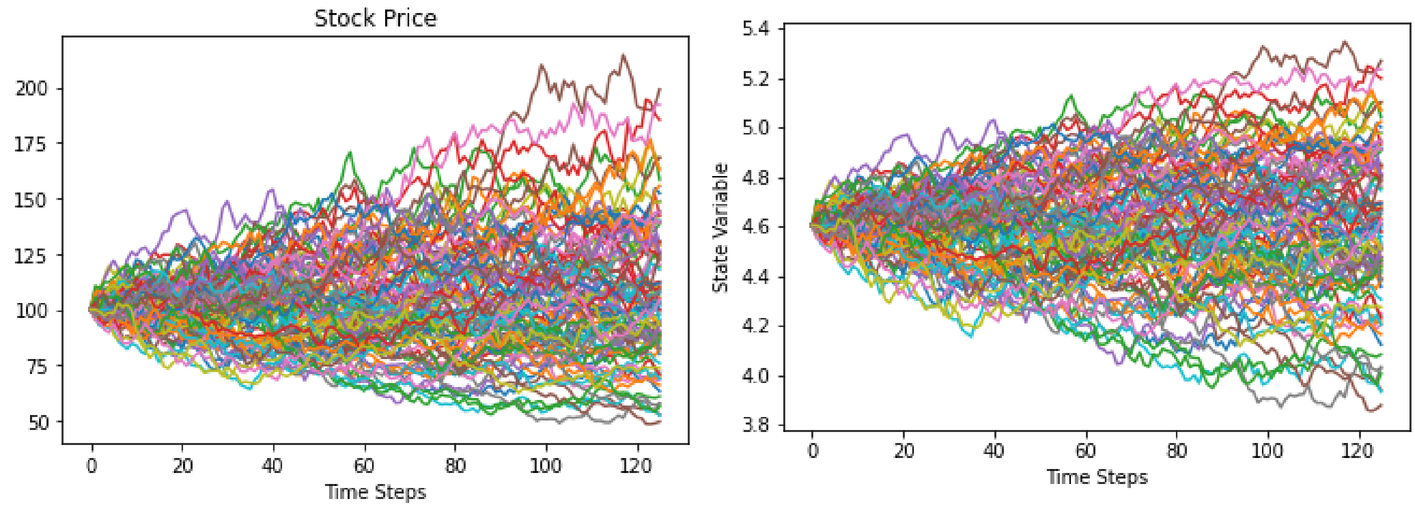

Figure 2: 100 paths for stock price and state with initial stock price = 100

3.1 Results

After defining function for terminal payoff of a European put option

$$C_T\left(P_T\right) = \max\left(K - P_T, 0\right) \tag{3.16}$$

with strike price K and risk-neutral, we calculate our coefficient ϕ_{nt} for $A_t^{\star}\left(S_t\right)$ numerically, with starting condition $A_T^{\star}\left(S_t\right) = 0$. Portfolio value Π is also computed backward by $\Pi_t = \gamma\left(\Pi_{t+1} - A_t^{\star}(S_t) * \Delta P_t\right)$.

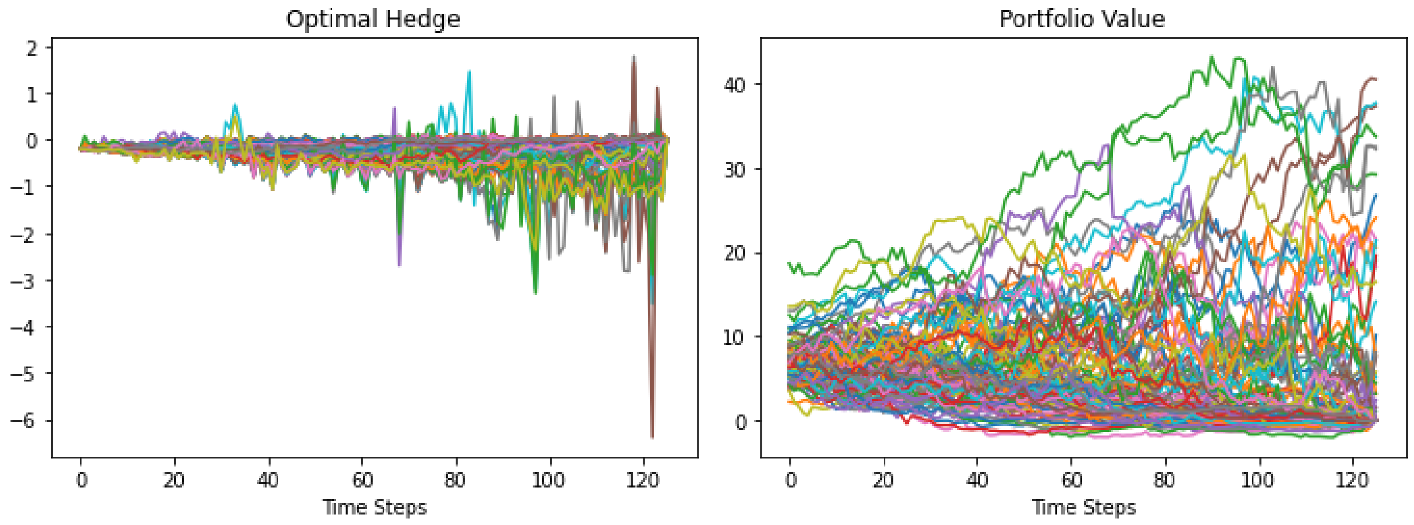

Figure 3: Plots of 100 paths for Optimal Hedge and Portfolio value

Since optimal hedge and portfolio values for different paths are computed, we can calculate payoff function for all states $R(S_t, A_t, S_{t+1}) = \gamma A_t \Delta P_t - \lambda \operatorname{Var}(\Pi_t)$

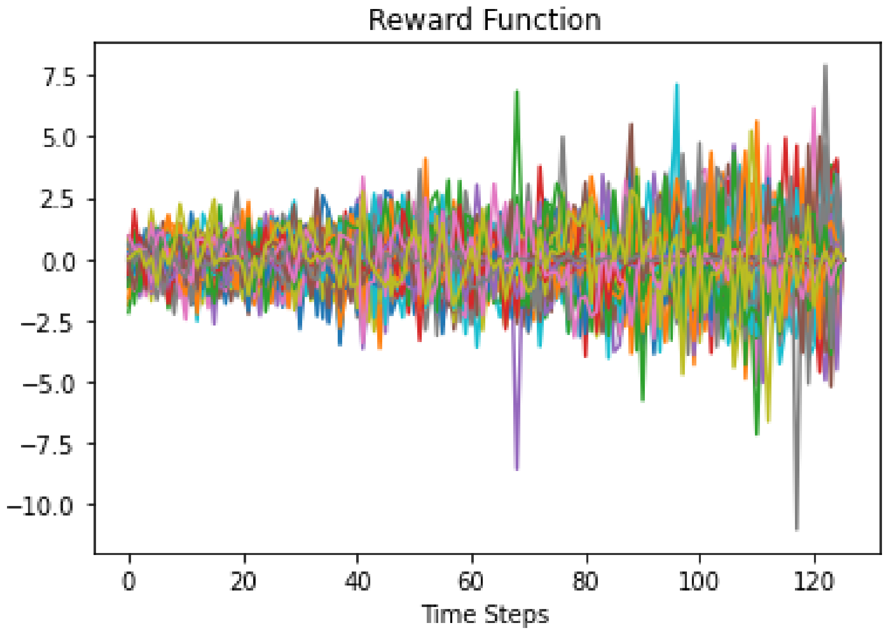

Figure 4: Reward function for all 100 paths

We need to learn optimal Q-function, which will be put option price, and need to compare with option price based on BSM model with Wilmott‘s volatility adjustment. By Halperins‘ assumptions on risk-neutral agent ($\lambda \to 0$); $S_t \sim$ GBM; the following equality must hold:

$$\lim_{\lambda \to 0} C_t^{(QLBS)}\left(P_t, \lambda\right) = C_t^{(BS)} \tag{3.17}$$

We calculate our coefficient ω_t for $Q_t^*\left(S_t, A_t^\star\right)$ with backward recursively with starting condition $Q_T^* = -\Pi_T\left(S_T\right) - \lambda Var\left[\Pi_T\left(S_T\right)\right]$ numerically.

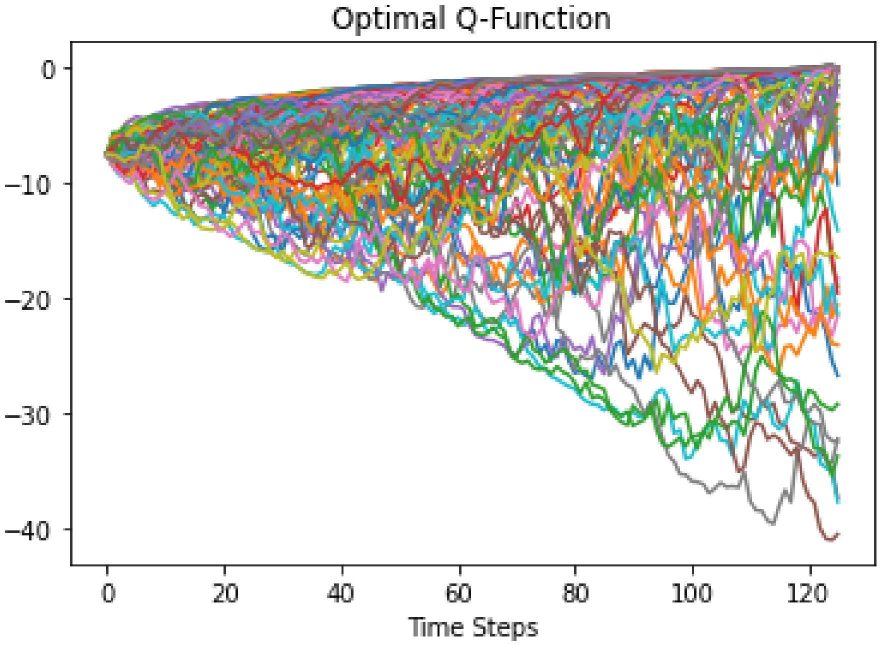

Figure 5: Q-function for all paths

The Q-function has the following form since we only consider European put option prices and hedge process. Let‘s take a look on analytical form of optimal option price:

$$C_t^{(QLBS)} = -Q_t\left(S_t, A_t^\star\right) \tag{3.18}$$

and plot with all steps:

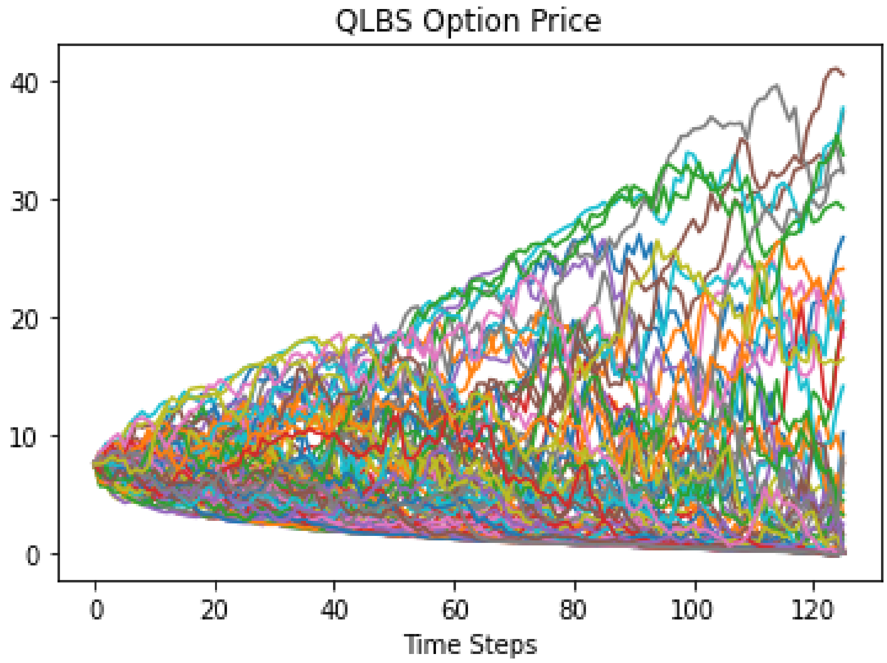

Figure 6: QLBS option price

Finally, we need to compare the difference European put option prices received by BSM model with Wilmott‘s adjustment and QFBS algorithm numerically with fixed parameters:

QLBS Put Price:	7.48 (1.023)
Black-Scholes Put Price:	7.46
Black-Scholes Adjusted Put Price:	7.47

Both methods gave comparatively similar results. QLBS has SE = 1.023 with the following confidence interval (4.878; 8.892) computed at the 95% confidence level.

4 Conclusion

In this study, I employ Black-Scholes model and QLBS algorithm to estimate the optimal European put option price and optimal hedge in discrete hedge for portfolio for fixed parameters on synthetic data. I show that the $\lim_{\lambda \to 0} C_t^{(QLBS)}(P_t, \lambda) = C_t^{(BSadj)}$ with assumptions on ΔS_t, $\lambda \to 0$ and existences of closet-form solution for option price. In Figure 7, we can see that with $\lambda \approx 0$, $C_t^{(QLBS)} \approx C_t^{(BSMadj)} \approx C_t^{(BSM)}$

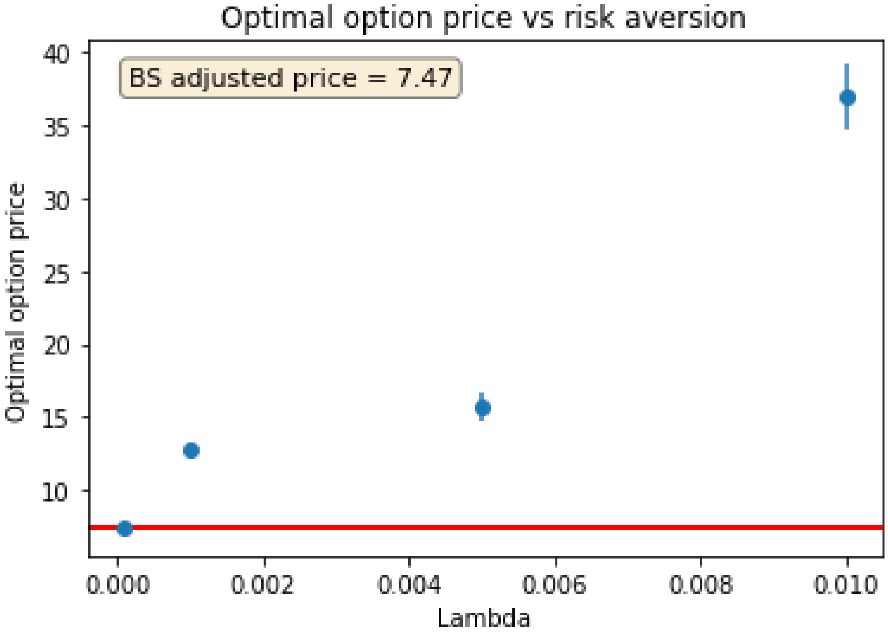

Figure 7: QLBS option price vs attitude to risk

There is another approach to verify the assumption on $C_t^{(QLBS)} \to C_t^{(BSM)}$. We need to consider an agent with $\lambda > 0$, $\Delta S_t \sim GBM$ and hedge in discrete time hedge. With these assumptions we receive $\lim_{\Delta t \to 0} C_t^{(QLBS)}(P_t, \lambda) = C_t^{(BS)}$. With $\lambda = 0.001$ $C_t^{(QLBS)} = 12.8517$. We can start with this price to show convergence to $C_t^{(BS)}$. The both method of verifying by convergence performer well.

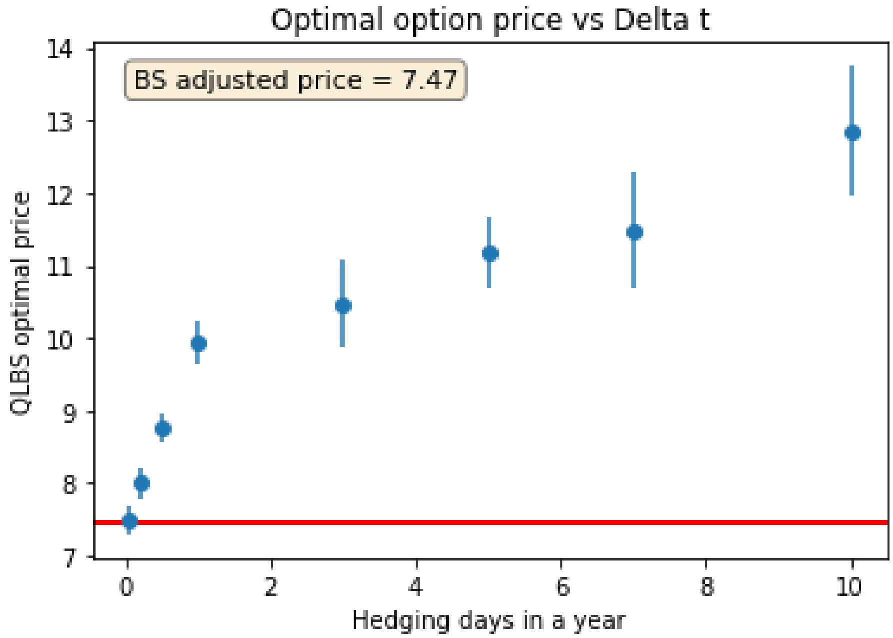

Figure 8: Optimal option price vs Delta hedging.png

5 Extensions

In the future work, SV with jump in underlying on market data can be tested. Also, neural networks such as Recurrent Neural Network into Q-learning algorithm could be implemented.

References

Halperin, I. (2019). QLBS: Q-Learner in the Black-Scholes(-Merton) Worlds. *arXiv:1712.04609v3*.

P. Boyle, D. E. (1980). Discretely adjusted option hedges. *Journal of Financial Economics*, 8(3), 259–282.

Watkins, C. J. C. H. (1989). Learning from Delayed Rewards. *Machine Learning*, 8, 279–292.

Wilmott, P. (1994). Discrete Charms. *Risk Magazine*, 7(3), 48–51.

Appendix

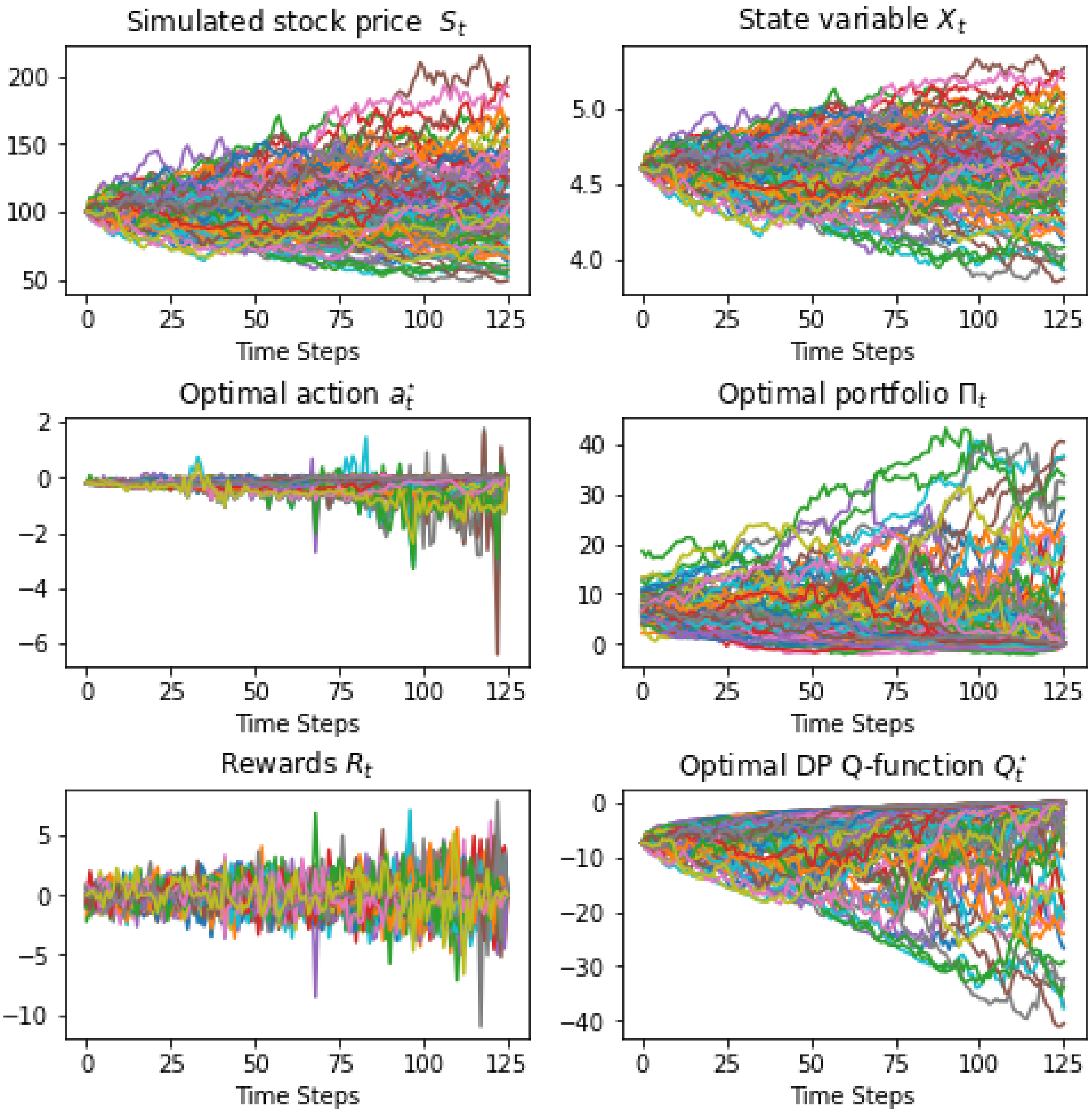

Figure 9: Summary

Parameters	
Initial Stock Price:	100
Drift of Stock:	0.15
Volatility of Stock:	0.45
Risk-free Rate:	0.01
Risk aversion parameter:	0.0001
Strike:	90
Maturity:	0.5

QLBS Put Price:	7.48 (1.023)
Black-Scholes Put Price:	7.46
Black-Scholes Adjusted Put Price:	7.47