Assignment 2: Q-Hedging for Delta One Stock

XU Guosong, 20830011, gxuae@connect.ust.hk,
CHEN Zuozhi, 20797609, zchenfu@connect.ust.hk
(Github: https://github.com/LegendreXu/ReinforcementLearning/tree/main/Q-Learning)

Abstract. Reinforcement Learning has many applications in financial areas, primarily to derive optimal hedging strategies for derivatives. However, we would like to apply Q-Learning to hedge a simple stock under the binomial price assumption as a naive sample. Moreover, this report also compares the risk of the output policies and other policies for optimality.

1.Introduction

Hedging is a vital part of derivative trading and ”dynamic hedging” is fundamental to derivative pricing. We assume the asset price follows a binomial-tree motion without transaction cost and would like to apply Q-Learning to hedge a stock.

2.Problem Formulation

Suppose we have one unit of stock that follows the binomial model, while we only have three possible actions: short 0, 0.5, 1 unit of the underlying stock at each state. Then we want to hedge this portfolio. The ideal policy should be the one that can maximize the total return at terminal time T and minimize the risk.

2.1.Intuitive Formulation

Intuitively, the expectation of the return during period T is 0 because of the features of the binomial model. Since we have one stock, shorting an equivalent value of the stock position(=1) will guarantee a 0 return and minimize our risk. Therefore, we want to use Q-learning to train an optimal hedging policy to test our assumption.

2.2.Theoretical Formulation

For this Q-learning model, we define the state with two variables stock price P_t and time t as $S_t = (P_t, t)$. Each time, the environment will give us a new stochastic stock price and lead us to the next state. We obtain an action based on the $\varepsilon-$greedy algorithm for each state. Besides, we can have three possible actions: have 0, 0.5, and 1 short position on this stock. To be specific, we are going to decide the amount of a short position during the period from S_t to S_{t+1}. Then we set the reward for each period as

$$R_{t+1} = LongPosition * (S_{t+1} - S_t) + ShortPosition \times (S_t - S_{t+1})$$

The Q-learning algorithm we use is based on the following formula:

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

2.3.Risk Evaluation

There are different risk measures for the reward distribution, and we chose three of them to quantify the risk:

Volatility

Sample standard deviation is a measurement of investment volatility and is often simply referred to as “volatility”.

$$\hat{\sigma} = \sqrt{\frac{1}{N_1 - 1} \sum_{i=1}^{N_1} (X_{1i} - \bar{X}_1)^2}$$

Value at Risk

Value-at-Risk is a widely used risk measure, which is defined as the loss level that will not be exceeded with a certain confidence level during a certain period of time.

$$VaR_p = \inf\{x | Loss(x) \geq p\}$$

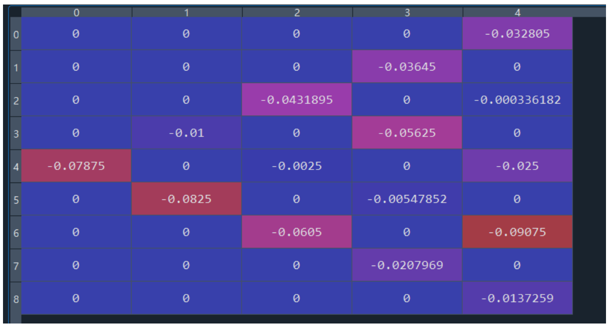
Expected Shortfall

Expected Shortfall is the expectation of the Loss where the Loss is greater than Value-at-Risk. It’s commonly recognized as a better measure since it is sub-additive, which means it is a coherent risk measure.

$$ES_p = [Loss(x) | Loss(x) \geq VaR_p]$$

3.Methodology

In this experiment, we define the probability of going upside $p = 0.5$ and the up ratio $u = 1.1$ and down ratio $d = 0.9$ for the binomial model. For the learning part, we set the $\gamma = 1$ to have no discounting effect and $\alpha = 0.5$. As for the exploration rate, we set initially $\varepsilon = 0.1$. Furthermore, in order to save the computation cost, we will set decrease the ε every 1000 episodes by setting $\varepsilon = \varepsilon \times 0.8$ We use a three-dimensional array for the coding realization to save the Q-value table.



	0	1	2	3	4
0	0	0	0	0	-0.032805
1	0	0	0	-0.03645	0
2	0	0	-0.0431895	0	-0.000336182
3	0	-0.01	0	-0.05625	0
4	-0.07875	0	-0.0025	0	-0.025
5	0	-0.0825	0	-0.00547852	0
6	0	0	-0.0605	0	-0.09075
7	0	0	0	-0.0207969	0
8	0	0	0	0	-0.0137259

Figure 1. A trivial example.

The row index stands for the state of stock price and column index stands for the passing time. For example, suppose the initial state is set as (i_0, j_0), then a state (i, j) represent the state with time $t = j$ and price $S_t = u^h \times d^{j-h}$ where

$$h = (i - i_0 + (j - (i - i_0))/2) \times I_{i>=i_0} + (j + (i - i_0)/2) \times I_{i<i_0}$$

Since in a T-period binomial model, we will have $2 \times (T) + 1 = 2 \times T + 1$ possible prices. Since we set the terminal value $Q(T, :, :) = 0$ and $Q(:, T, :) = 0$, we will have a Q table with sizes $(2 \times T - 1) \times (T) \times (3)$ and we set initial state at $S_0 = (T - 1, 0)$

4.Results

We perform the algorithm for the $T = 5$ scenario as a naive example, with the optimal policy and the comparison with other policies.

4.1.Optimal Policy

After 100,000 training iterations, our model provides the optimal policy to always short one unit of stock, which coincides with our financial intuition: the stock is a “Delta One” product.

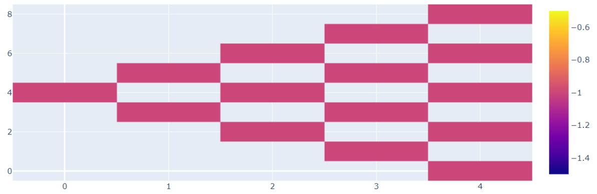


Figure 2. Optimal Policy for Different States.

The x-axis represents the time t from starting from t_0, while the y-axis represents the state of the stock’s price. There are four possible values in each box of the heat-map, which present the optimal action:

- Null: There is no such possible states in our experiment
- 0: The best action is to short 0 position of stock
- -0.5: The best action is to short 0.5 position of stock
- -1: The best action is to short 1 position of stock

4.2.Risk Comparison

The corresponding rewards series of the optimal policy is always zero, which means it has literally no risk. However, the rewards results of the other three policies have different distributions. The left part of the following graph shows the time-series average value of rewards, while the right part describes their distributions by the Box-graph and Histogram.

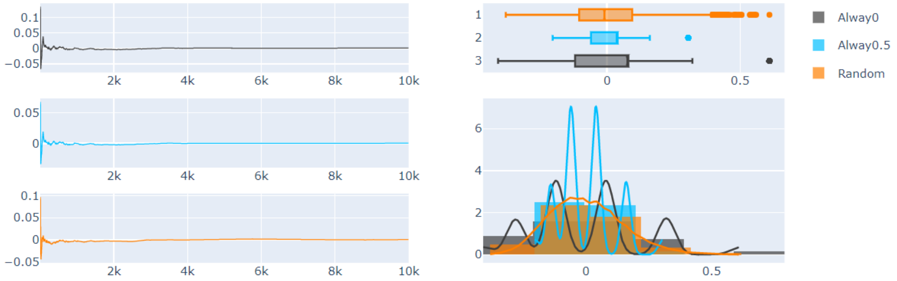


Figure 3. Reward for Different Policies.

We can figure out that the 'Random' policy and the 'Always Zero' policy share approximately the same variation, while the 'Always 0.5' policy has a smaller variation. Moreover, the rewards of 'Always Zero' and 'Always 0.5' concentrate more on several discrete values, while the 'Random' policy rewards have a smooth distribution.

In addition, we calculated the risk measures for these rewards for the quantified comparison of risk.

Table 1: Table of risk measures for different policies

indicator	Non-Optimal			Optimal
	Alway0	Always0.5	Random	Always1
Volatility	0.225844	0.112922	0.145006	0
Value at Risk(95)	0.2810	0.1405	0.22	0
Expected Shortfall(95)	0.409510	0.204755	0.259823	0

5.Conclusion

As the left part of Figure 3 shows, all policies have expected rewards of zero. So we pay more attention to the risk of different policies: the optimal one(short 1 unit) always has zero rewards, which means zero risk. In comparison, the other three policies have different levels of risk, as Table 1 illustrates. Hence, we apply the Q-Learning algorithm in hedging and get the actual Delta successfully. Furthermore, we are willing to try to hedge other financial instruments by Reinforcement Learning if we have further research opportunities.

Acknowledgments

This group assignment is completed by XU Guosong and CHEN Zuozhi with equal contribution. We had many times of video and face-to-face discussions to design this algorithm and experiment. CHEN focus more on the implementation of the experiment, and XU focus more on the data visualization. And this report is wrote by us together as well, and XU is responsible for the LaTeXcode. Also, we would like to express our sincere appreciation to Professor Chak WONG and TAs because of their patient help.