Q-learning in Binomial Stock Hedging Problem

Jin Tian & Zheng Zihao

2022 April

Abstract

In this paper, we used Q-learning method to solve the dynamic hedging problem for stocks in binomial models. We used stock price as the states and mean-variance return as the target to optimize with. This method could be used to explain the risk-aversion utility function that is reasonable in most cases. Our results showed that the aversion of risk could lead to larger portion of hedging. Also, when the underlying volatility of stocks increases, optimal policies tend to apply deeper hedging.

1 Introduction

Hedging is an important problem for traders. A lot of methods were applied to this problem, such as delta hedging, covariance hedging, duration hedging etc.. They were used for different assets, aimed at different targets. Hedging methods were used for different investors, represented by different utility fuctions. To most investors, risk-aversion models could be used.

In stock models, binomial model is a simple but direct one. It was firstly introduced in Sharpe (1978)[1] and was formalized in Cox et al.(1979)[2]. This method was mainly used for option pricing in discrete time models.

In modern days, reinforcement learning was widely used in many areas, such as AI, engineering and finance. A comprehensive treatment of reinforcement learning is provided by Sutton and Barto (2018)[3]. In this book, they destate elements in reinforcement learning such as states, actions, policy value actions and so on. One important method was called Q-learing. It was a kind of 0ff-policy TD method, first brought by Watkins (1989)[4]. It could be applied in binomial stock hedging problems with risk-aversion investors as mentioned.

In this paper, we will introduce our model using Q-learning method in stock hedging problems. We first constructed our algorithms and then we used Monte Carlo methods to estimate their performance. The optimal policy also showed certain patterns of dynamic hedging.

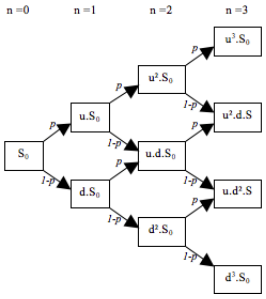
2 Models

In the first place, we introduce binomial model of the stock. The binomial model traces the evolution of stocks in discrete-time. This is done by means of a binomial lattice (Tree), for a number of time steps between the valuation and expiration dates. Each node in the lattice represents a possible price of the underlying at a given point in time. This method was widely used in option pricing.

The algorithm in generating binomial trees depends on (1) interest rate: r (2) volatility: σ. These variables will determine (1)proportion of the stocks' increment: u ,(2) proportion of the stocks' decrement: d and (3) probability when price increases: p. These are key factors determine the shape and values in a tree. This model could be explained as below:

At each step, it is assumed that the underlying instrument will move up or down by a specific factor (u or d) per step of the tree (where, by definition, $u \geqslant 1$ and $0 < d \leqslant 1$). And their probabilities are affected by risk-free interest rate r. So, if S is the current price, then in the next period the price will either be $S_{up} = S \cdot u$ or $S_{down} = S \cdot d$. The up and down factors are calculated using the underlying

volatility, σ , and the time duration of a step, t, measured in years. From the condition that the variance of the log of the price is $\sigma^2 t$, we have the binomial tree like figure 1. So, with binomial tree, we can define the states in our problem to be figure 2:



$$p = \frac{e^{rt/n} - d}{u - d}$$

$$u = e^{\sigma\sqrt{t/n}}$$

$$d = e^{-\sigma\sqrt{t/n}}$$

Figure 1: Binomial Model

t=0	t=1	t=2	t=3	t=4
S(0,0)	S(0,1)	S(0,2)	S(0,3)	S(0,4)
	S(1,1)	S(1,2)	S(1,3)	S(1,4)
		S(2,2)	S(2,3)	S(2,4)
			S(3,3)	S(3,4)
				S(4,4)

Figure 2: State

Here, $S(i, j)$ means the $j + 1$'s largest price at $t = i$ in a binomial tree. It could be stated like:

$$S(i, j) = S_0 \cdot u^{j-i} \cdot d^i \tag{1}$$

Then, we could define our actions to be: $a_t \in \{0, 0.5, 1\}$ where $a_t = 1 - \omega_t$. It means the proportion of hedging. For example, if $a = 0$, we take the whole portfolio to be the underlying stock; if $a = 0.5$, 50% of the portfolio would be stock, others would be cash.

At last, we define return to be:

$$R_t = r_t - \lambda\sigma_t^2 \tag{2}$$

In this formula, r_t, λ and σ_t^2 are defined to be:

$$\mu = a_t[p \cdot ln(u) + (1 - p) \cdot ln(d)] \tag{3}$$

$$\sigma_t^2 = (1 - a_t)^2[p(ln(u) - \mu)^2 + (1 - p)(ln(d) - \mu)^2] \tag{4}$$

These are expected return and expected variance of return in the period where λ is the degree of risk aversion, at each time $t \in [0, T - 1]$, return should be the sum of return from stock and cash:

$$r_t = (1 - a_t)[ln(S_{t+1}) - ln(S_t)] + a_t e^{r/N} \tag{5}$$

Now, we define our algortithm formally:

Algorithm 1 Q-learning for Hedging Problem

Parameters: $\alpha \in (0, 1]$ and small $\epsilon > 0$.
Initialize $Q(s, a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$ arbitrarily, and $Q(terminal - state, \cdot) = 0$
Repeat (for each episode):
  Initialize S;
  Repeat (for each step of episode):
    Choose A from S using policy derived from Q (e.g., ϵ-greedy);
    Take action A, observe R, S';
    $Q(S, A) \leftarrow Q(S, A) + \alpha\left[R + \max_a Q\left(S', a\right) - Q(S, A)\right]$;
    $S \leftarrow S'$;

To this algorithm, we can show that it is a typical Q-learning problem. Here, we set $\gamma = 1$, which means no discounting. By using this algorithm, we could find $\pi^* = argmax_a Q(S, A)$ and $V_*(S_t) = max_a Q(S, A)$ of the problem. The problem is solved then.

3 Empirical Results

In this part, we could calculate our result. As we can see, we used $N = 6$, $\sigma = 0.2$ and $\alpha = 0.2$ in our problem. These pictures are corresponding π^* and $V_*(S_t)$ of the problem when risk free rate R and risk-aversion parameter change.

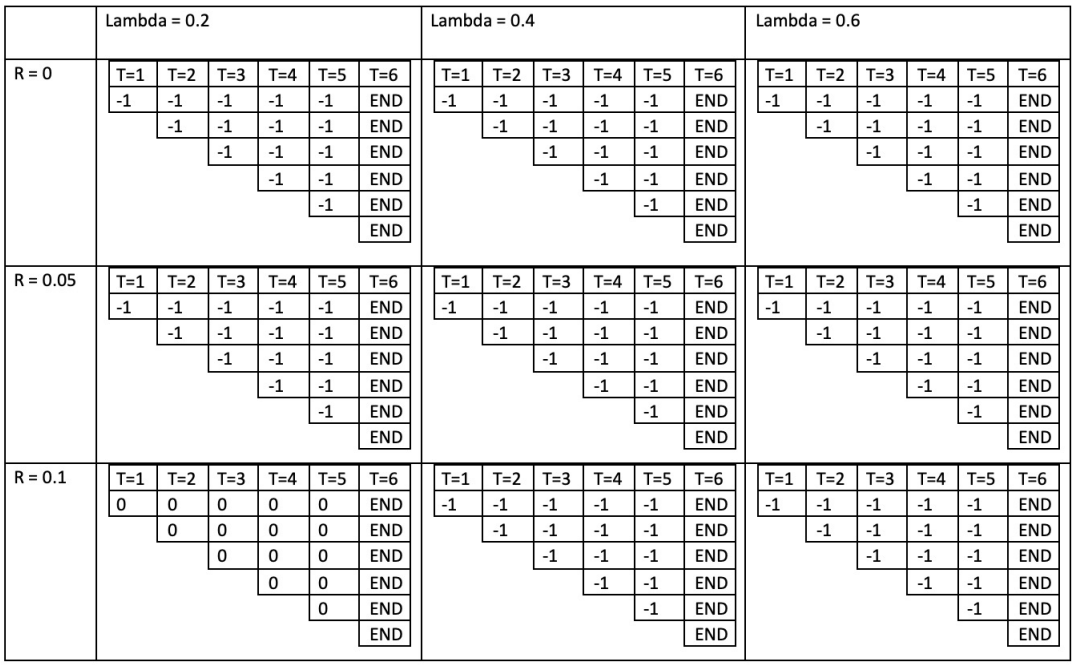


Figure 3: π_*, λ and R

From the results above we may see that when R and λ changes, the optimal policy would also change. When state is made sure, optimal policy will have the same action in every state of an episode. We can see that when R increases, the optimal policy tend to be zero because that will provide higher payoff. Also, when λ increases, optimal policy tends to hedge earlier.

Certainly, to all a, there exsist a λ which will make $\pi_*(S) = a$.

In this part, we proved that using Q-learning, we can find the optimal policy of hedging, it also showed that the hedging policy differs when ones' utility function changes.

4 Conclusions

From this paper, we developed our model to hedge a stock using Q-learning method. We constructed our return to be related to investors utility. With different preference of risk, the optimal policy would change. Here, optimal policy will have the same action in every state of an episode. Generally, when risk-aversion increases, optimal policy tends to have complete hedge; when interest return increases, it is better to hedge less.

5 Reference and Appendix

Please see https://github.com/JinTian0717/hm2