MAFS5370 Assignment 1

Asset Allocation Based on Reinforcement Learning

SHAO, Ruizhao **WANG, Liangshu**

Problem Statement

Consider the discrete-time asset allocation example in section 8.4 of Rao and Jelvis:

We are given wealth W_0 at time 0. At each of discrete time steps labeled $t = 0, 1, ..., T-1$, we are allowed to allocate the wealth W_t at time t to a portfolio of a risky asset and a riskless asset in an unconstrained manner with no transaction costs.

Suppose the single-time-step return of the risky asset from time t to $t+1$ as $Y_t = a$ with $prob = p$, $and\ b$ with $prob = (1-p)$. Suppose that $T = 10$, sse the TD method to find the Q function, and hence the optimal strategy.

1. Introduction

Wealth allocation is an essential problem in finance and economics, as it involves determining the optimal way to distribute resources among different assets in order to maximize returns while minimizing risk. In this report, we will use reinforcement learning to address the problem of wealth allocation in the context of a portfolio consisting of a risky asset and a riskless asset.

Specifically, we will consider scenario where we are given an initial wealth W_0 and are allowed to allocate this wealth between a risky asset and a riskless asset at each of the discrete time steps labeled $t = 0, 1, ..., T - 1$. The single-time-step return of the risky asset from time t to $t + 1$ is given by $Y_t = a$ with probability p and $Y_t = b$ with probability $1 - p$.

Our goal in this report is to apply the TD (temporal difference) learning method to find the Q function, which estimates the expected cumulative future reward of taking a particular action in a particular state. We will use Q-learning to find the optimal policy for allocating wealth between the risky and riskless assets at each time step.

The report is structured as follows.

- In Section 2, we provide some background on TD method and Q-learning, which will be used in our approach.
- In Section 3, we describe the problem setup and the assumptions we make.
- In Section 4, we present our methodology for applying TD method with Q-learning to the problem.
- In Section 5, we present our results and analyze the performance of our optimal policy.
- In Section 6, we discuss the implications of our approach and suggest possible extensions.

2. Background

We will describe the background of the TD (temporal difference) method and Q-learning, which are the main algorithms used in our approach to solving the asset allocation problem.

TD learning is a form of model-free learning that involves updating estimates of the value function of a policy based on the differences, or "temporal differences," between the predicted and actual rewards received. The basic idea behind TD learning is that the value function estimates are updated iteratively based on the difference between the predicted value of the current state and the actual value received after taking an action and observing the next state. TD methods have two ways to perform, including TD(0), which updates the value function after each step, and TD(λ), which combines the TD(0) and Monte Carlo methods to balance the tradeoff between bias and variance. In this assignment we only apply TD(0).

Q-learning is a reinforcement learning algorithm that is used to find the optimal action-selection policy for any given environment. It is a model-free method that can be applied to problems where the transition probabilities and rewards of the environment are not known. Q-learning works by iteratively updating an estimate of the optimal action-value function using the TD method.

3. MDP Formulation

- **Variables and Parameters related to the problem**

Name	Description	Value
W_0	Wealth at time 0	Constant Integer (set to be 10000)
W_t	Wealth at time t	Integer
T	Terminal time	Constant Integer (set to be 10)
a	Upward return	Constant Float
b	Downward return	Constant Float
p	Upward probability	Constant Float
r	Risk-free return	Constant Float
A_t	Action at time t, the proportion of W_t invest in risky asset	Float in the range of [0, 1]
$U(W)$	Utility function	$U(W) = \frac{1-e^{\alpha W}}{\alpha}$

- **Assumption Related to the work**
  - **Wealth Utility Assumpton:** Our investor is indifferent between two weath with a same integer part (for the scale of the wealth aound 10000). This means a difference smaller than 1/10000 of the total wealth will be igonored by the investor. For example, 10000.1 and 10000.4 will be viewed as the same level of wealth, which will bring same level of utility. Hence, for W_t, we round it to integer.
  - **Allocation Decision Assumption:** The decisions of our investors can be represented by several equally spaced investment ratios. For example, if we divide 0 to 1 into three investment decisions, that is, 0, 0.5 and 1, then 0 represents the unwillingness to invest in risky assets, 0.5 represents the willingness to invest in risky assets, and 1 represents all investment in risky assets. Hence, for A_t, we represent it by finite equally spaced ratios between 0 and 1.
- **MDP Formulation**
  - **State space:** State space in this problem includes all the possible wealth at each time t. Since we round W_t to integer, the state space becomes finite and can be found in advance by doing simulations.
  - **Action space:** Actions space includes all the possible investment ratio to the risky asset, based on the Allocation Decision Assumption, it is finite as well.
  - **Transition probabilities:** The transition probabilities are determined if the state and the action is given, the transition between W_t and W_{t-1}:

$$W_{t+1} = int(W_t + aW_tA_t + rW_t(1 - A_t)) \ \ with \ \ prob = p$$

$$W_{t+1} = int(W_t + bW_tA_t + rW_t(1 - A_t)) \ \ with \ \ prob = 1 - p$$

  - **Rewards:** The rewards in this problem is defined as the utility of owning wealth of W_t, which is $U(W_t)$.
  - **Discount factor:** Since all the wealth will not be consumed until the end of T, the discount factor can be set as 1 in this problem.
  - **Policy:** The policy is to based on the Q funtion value to find the action brings biggest utility in the future.

4. Methodology

The steps we took to implement the TD method based on Q-learning are as follows:

1. Construct the state-action value function to store the Q values for each state-value pair and initiate all of them to zero.
2. Build Q-learning process to estimate each Q vlaue for each state-action pair:
   a. Use epsilon-greedy to perform the ‘exploration and exploitation’, set epsilon to be large at the beginning to explore more and then decay it to relay on the existing Q values.
   b. Update the Q value each trail based on the following transition expression:

$$Q(W_t, A_t) \leftarrow Q(W_t, A_t) + \alpha[R_{t+1} + \gamma * max\ Q(W_{t+1}, a) - Q(W_t, A_t)]$$

c. Stop the learning process when all the Q values converge based on caculating an average Q value for all the actions at each time t.

3. Make the optimal decision based on the Q values, which is our optimal policy.

5. Results

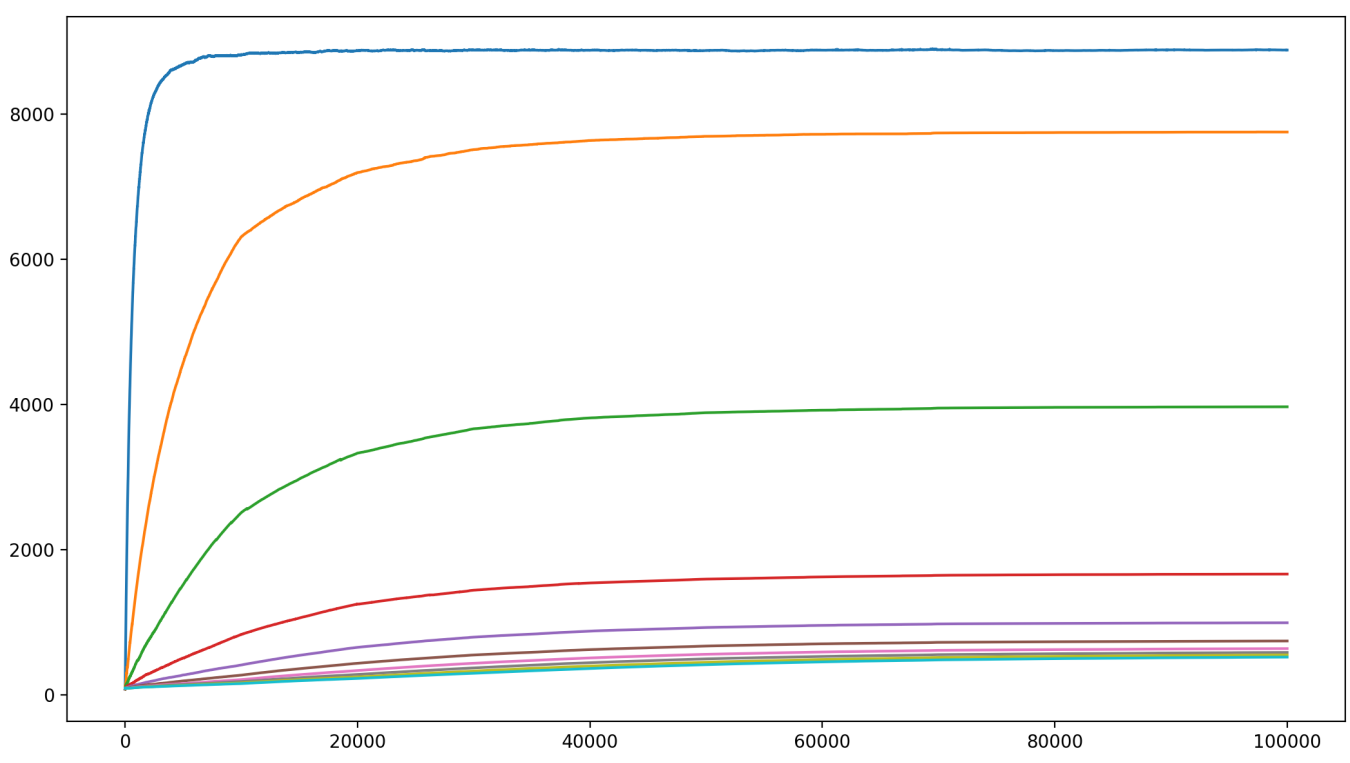
To test our algo, we did a demonstration test, in which we set our action space to be [0, 0.2, 0.5, 0.8, 1.0], which is the invest ratio in the risky asset.

1. **Convergence Test:**

By using the Q learning, the algo converges very fast. We caculate an average Q value for all the actions at each time t to see wether it converges or not.

The xlab is the number of episodes finished, the line with different colors refer to different time point t, which is from 1 to 10.

The ylab is the average Q value for all the valid state-action pairs at each time t, which represents the stable level of Q value at that time t.

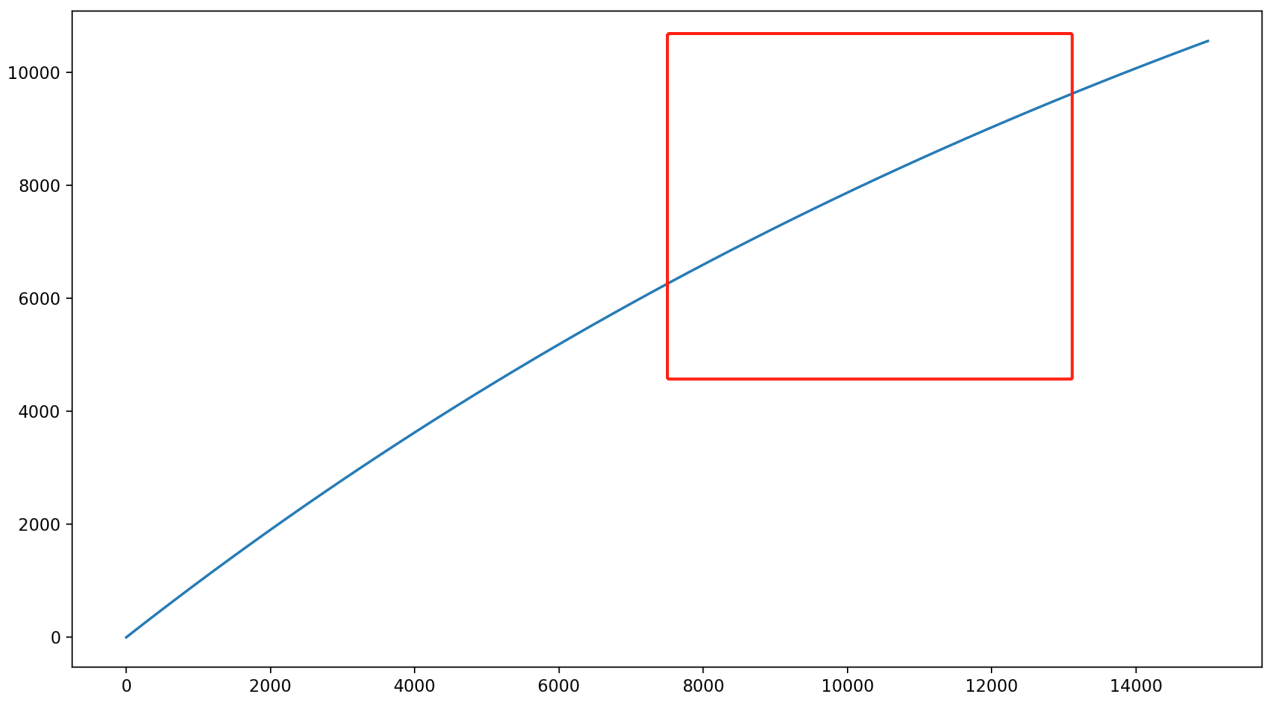


We can find the algo converges after running the algo around 60000 times.

2. **Optimal Policy Test**

To test the algo, we set several conditions for the parameters related to the problem and perform our algo on these conditions:

- **Setting**: utility alpha = 0.00005, which means our investor is risk adverse. The values we mainly pay attention to is in the red square.



- **Condition 1:** $a = 0.05,\ b = -0.05,\ p = 0.7,\ r = 0.02$

  In this condition, the expectation of two asset are the same, which equals to 0.02. However, our investor is risk adverse, which means it will not choose the risky asset.

  - **Test:**

```
new trace
t = 0, wealth = 10000, action = 0.0
t = 1, wealth = 10200, action = 0.0
t = 2, wealth = 10404, action = 0.0
t = 3, wealth = 10612, action = 0.0
t = 4, wealth = 10824, action = 0.0
t = 5, wealth = 11040, action = 0.0
t = 6, wealth = 11260, action = 0.0
t = 7, wealth = 11485, action = 0.0
t = 8, wealth = 11714, action = 0.0
t = 9, wealth = 11948, action = 0.0
```

  We notice that it continued to choose 0 as the investment ratio.

- **Condition 2:** $a = 0.07,\ b = -0.05,\ p = 0.7,\ r = 0.02$

  In this condition, the expectation of risky asset is larger than the risk free asset.

  - **Test:**

```
new trace
t = 0, wealth = 10000, action = 0.8
t = 1, wealth = 10600, action = 0.5
t = 2, wealth = 11077, action = 0.5
t = 3, wealth = 10910, action = 0.8
t = 4, wealth = 11564, action = 0.5
t = 5, wealth = 11390, action = 0.0
t = 6, wealth = 11617, action = 0.0
t = 7, wealth = 11849, action = 1.0
t = 8, wealth = 12678, action = 0.8
t = 9, wealth = 12221, action = 1.0
```

  This time it starts with a relative aggressive strategy, after facing with a loss at time t=5, it changed to a conservative strategy. And back to an aggressive strategy after earning some money.

- **Condition 3:** $a = 0.07,\ b = -0.05,\ p = 0.9,\ r = 0.02$

  In this condition, the prob of profit and the profit itself are both higher than the loss, which means we will choose to invest in the risky asset.

  - **Test:**

```
new trace
t = 0, wealth = 10000, action = 1.0
t = 1, wealth = 10700, action = 1.0
t = 2, wealth = 11449, action = 1.0
t = 3, wealth = 10876, action = 1.0
t = 4, wealth = 11637, action = 1.0
t = 5, wealth = 12451, action = 1.0
t = 6, wealth = 13322, action = 1.0
t = 7, wealth = 12655, action = 0.8
t = 8, wealth = 13414, action = 0.8
t = 9, wealth = 12931, action = 0.5
```

  We notice it continued to invest in the risky asset. After suffering loss at time t=7, it turned to be a little bit conservative but still with large portion in the risky asset.

6. Discussion

- The TD method with Q-Learning can solve this problem by estimating all the Q function values in the discrete state and action spaces easily. However, when it comes to continuous state and action spaces, we need to perform some function approximation to estimate the Q funtions instead of using a tabular to do so. This part could be done using the tools provided by Rao and Jelvis in their book, which is the ***rl*** library. We also completed a demo for that.
- This problem can also be solved by applying dynamic programming since the probability distributions are known. It can also be solved explecitly by hand, same as the deduction in the book of Rao and Jelvis, and simpler.