Stock Trading Strategy Developing Based on Reinforcement Learning

Zeyu Xia[1], Mingde Shi[2], and Changle Lin[3,4](✉)

[1] Department of Economics, Shenzhen University, Shenzhen, China
[2] Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
smd18@tsinghua.org.cn
[3] China Aerospace Academy of Systems Science and Engineering, Beijing, China
changlelin@tsinghua.edu.cn
[4] Institute for Interdisciplinary Information Core Technology, Xi'an, China

Abstract. Reinforcement learning has achieved superhuman performance on many sequential decision-making problems, but only very few works are done on applying reinforcement learning to market trading. In this study, we take the stock trading problem as an Markov decision process, and applied PPO algorithm to solve the problem on the Dow Jones 30 stocks for the past 10 years. Our reinforcement learning agent is able to achieve significantly higher returns and higher Sharpe ratio than the broader market index on the test dataset of about a year. By adjusting the reward function of the PPO agent, we found that agents with appropriate risk aversion properties can achieve even higher Sharpe ratio than the risk-neutral agents.

Keywords: Reinforcement Learning · PPO Algorithm · Stock Trading · Introduction

1 Introduction

Reinforcement learning is one of the branches of artificial intelligence that is developing rapidly in recent years. It can model common sequence decision problems in daily applications such as Markov Decision Process (MDP). Reinforcement learning can learn action policies by sample learning, so that good strategies can be learned without relying on the environment model.

Reinforcement learning have successfully solved many complex robot control tasks [1], like solving the complex game playing tasks go-chess [2], solving complex video games like StarCraft [3] and etc. These achievements showed the power of reinforcement learning in solving sequential game decision making problems.

Although we all know that, comparing to the former sequential decision problems, the signal-to-noise ratio in financial market data is relatively low. So parameter presetting maybe very important to ensure basic economic rationality in decision making, or it might be too hard for the model to converge to a robust strategy solution. There are also many literature tried to apply reinforcement learning to financial investment

J. Yen et al. (Eds.): ICBIS 2023, AHCS 14, pp. 156–164, 2023.
https://doi.org/10.2991/978-94-6463-198-2_18

fields in recent years, such as stock trading [4], portfolio management [5], derivative pricing and hedging [6–8], etc. At the same time, a number of tools that can easily apply reinforcement learning in finance and market trading are developed. Here we are going to use an open source project "FinRL", which can be found in open source communities and have showed great potential of deep reinforcement learning in finance.

In this study, we focus on the problem of main stock trading strategies generating through reinforcement learning algorithms. The data we used is Dow Jones 30 Industrial Index in daily frequency. The main algorithm we apply is PPO algorithm [8].

In addition to the conventional reward function setting (set the stock trading daily return as reward), we extended the function to incorporate more penalty of loss, which could encourage the agent to engage more risk-averse behavior in stock trading. On the test dataset (July 2020 to October 2021), the strategies trained by the two reward functions can achieve significantly better performance than the market index Dow Jones 30 Index. Incorporating appropriate loss penalty to the reward function with lead to better performance, but if we set too big penalty parameter, the strategy will cause the strategy to stop taking risks and stop trading.

2 Preliminary Knowledge

2.1 Reinforcement Learning on Markov Decision Process

Markov decision process describes a decision making problem on a Markov process. It can be denoted as a six-tuple $< S, A, R, P, \gamma, H >$, in which S is the state space, A is the action space, R is the reward function, P is the state transition function of the Markov process, $\gamma$is a decay factor and H is the decision round length.

The market process is usually considered as a low signal-to-noise ratio process. It is appropriate to assume that the process can be marked as a Markov stochastic process with single-step memory, because setting longer steps of memory will further increase the parameter noise of the model and make the model not stable as [9] did.

The goal of reinforcement learning on Markov decision process is to learn a policy $\pi(a|s)$ as (1) shows, which specifies the corresponding actions given each state, to maximize the goal return G of the round which we can denote as $G^{\pi} = E_{\pi}[\sum_{t=0}^{H-1} \gamma^t R(s_t, a_t)]$, through interaction with the environment.

$$\pi^* = \text{argmax}_{\pi \in \Pi} E_{\tau \sim p_{\pi}(\tau)} \left[\sum_{(s_t, a_t) \in \tau} \gamma^t r(s_t, a_t, s_{t+1}) \right] \tag{1}$$

2.2 PPO Algorithm

PPO algorithm (Proximal Policy Optimization Algorithm 109) 8 is an important policy optimization method where the policy is simply modeled as a function of probability distribution of actions conditional on states, and the agent will learn the best policy by pushing up the probabilities of actions that lead to higher cumulative rewards. This way, it's simple to add rational decision space, while at the same time, easy to understand and easy to implement.

The fundamental solving method is to apply gradient descent method to gradually approach the best policy. While in stochastic process, the gradients are usually very noisy, and PPO algorithm is designed to take safe and stable steps, while maximizing the amount of reward gain. PPO algorithm uses a clipped objective that can discourage the new policy from stepping far away from the old policy, and takes the biggest improvement step using the current data, to have more reliable performance.

Project FinRL provides multiple fine-tuned standard deep reinforcement learning algorithms, such as DQN, DDPG, PPO, SAC, A2C and TD3 as Fig. 1 showed.

3 Algorithm Design

3.1 Data Preprocessing

We use the Dow Jones 30 Industrial Index, from January 1, 2009 to October 31, 2021, as the experiment dataset. Of which, take data from January 1, 2009 to July 1, 2020 as the training dataset, and the data from July 2, 2020 to October 31, 2021 as the test dataset.

All data are obtained from open source yahoo finance. By deleting one stock with too many missing values, the final dataset contains only 29 stocks' returns. We replace the few missing values of the left returns data with 0s. To help learning, we also calculated important technical factors such as MACD, CCI, RSI, Bollinger Band, DX, SMA, etc.

3.2 Environment Setting

The state space of this problem contains 291 dimensions, including:

1. Current idle funds, which can buy more stocks,
2. Current holding shares of the stocks,
3. Current prices of 29 stocks,
4. Current indicators, including the Bollinger upper and lower bounds, MACD indicators, RSI indicators, SMA30 and SMA60 indicators, DX indicators, CCI indicators, RSI indicators.
5. Current Action Space, which is a 29-dimensional continuous action space, each one represents the share of each stock to be purchased or to be sold. The transaction interval is set to be daily, at the beginning of each trading day, the agent decides to buy or sell actions (with 0.1% as transaction fee) based on the state of the previous day.
6. The reward function (of the normal agent) is designed as the difference between the total asset value at the end of the day and the total asset value at the end of the previous day.
7. Parameter gamma (the decay factor) is set to be 0.99.

3.3 The Risk Averse Agent Setting

The original normal agent only considers the total return as reward. We can adjust the reward function to incorporate penalty on loss and encourage the agent to balance return and risk.

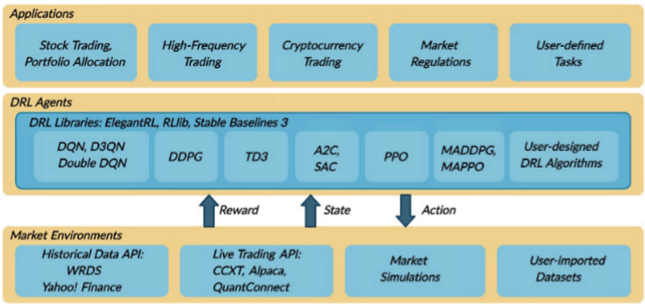

Fig. 1. Layers of FinRL package

A straightforward idea is to add the maximum drawdown over a period of time (or other risk measurement indicator over a period of time) as a penalty to the reward function, but risk indicators are usually calculated based on a period of time data and cannot be assigned to each time step, which makes it difficult for the agent to optimize its strategy due to reward distribution, resulting in poor actual performance for this approach.

So, we take another approach, directly encouraging the agent to avoid risk by increasing the penalty for return loss. Specifically, on the basis of the original reward function, if the income of the day is negative, the reward is multiplied by 1.05 (or other similar penalty parameter) compared to the original loss; if the income is greater than or equal to 0, it remains the same. Such a reward function will make the trained agent more sensitive to stock return losses, thus the agent could learn a risk-averse trading strategy.

In the experiment, we have tried a variety of coefficients of the punishment on losses, and found that if the coefficient is too large, the agent would be too conservative to make any transactions, in order to avoid suffering return loss caused by trading fees. Finally, we found that 1.05 is a reasonable punishment parameter.

4 Experiment Results

We have experimented both the ordinary agent and the risk-averse agents (with many different risk-averse parameters) under the PPO algorithm.

4.1 The Ordinary Agent

For ordinary PPO agent, we only consider the effect of payoff in the reward function. On the test dataset from Jul, 2020 to Oct, 2021, the annualized return of the Dow Jones index as a comparison benchmark was 27.0%, and the Sharpe ratio was 1.79. While the annualized return obtained by the reinforcement learning agent was 40.8%, and the Sharpe ratio is 2.01. The annualized excess return is as high as 13.8%, results can be found in Fig. 2.

As for the risks of the strategy, we calculated the rolling volatility and the drawdown of the ordinary PPO agent. As can be found in Figs. 3 and 4, PPO agent got a higher volatility than the benchmark, and its max drawdown is about 8%.

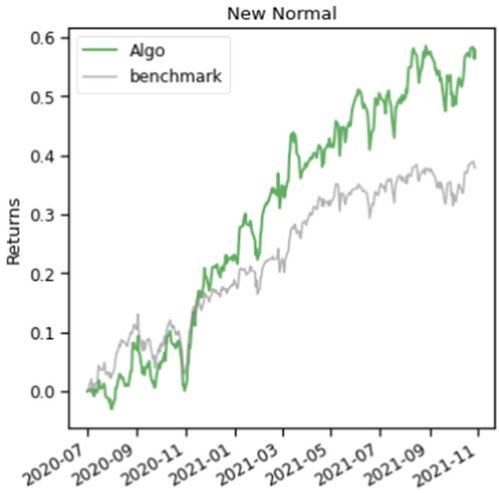

Fig. 2. Normal Agent Cumulated Return

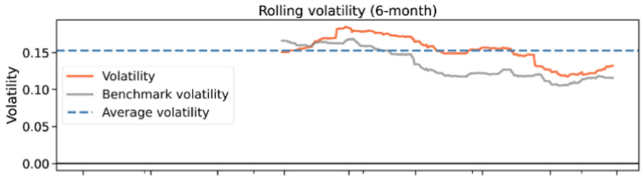

Fig. 3. Rolling volatility of Ordinary Agent and Benchmark

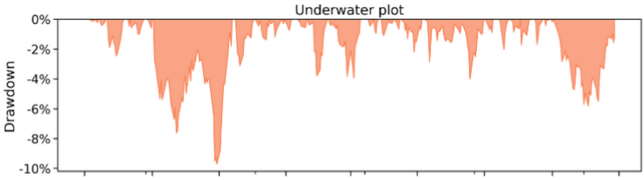

Fig. 4. Drawdown of the NM Agent

It's easy to understand that the ordinary agent actually obtained excess returns by taking more risks. But from the point view of Sharpe ratio, the PPO agent get a higher Sharpe ratio, so it gets more average risk compensation of the risk than the Dow Jones 30 Index.

Considering the test period, the Dow Jones 30 index, which is the benchmark for comparison, showed an overall growth trend, we should draw a conclusion that the agent strategy might essentially be an index enhancement strategy. By optimizing the proportion of individual stock positions and capital allocation in different time periods, higher risk compensation is achieved, and the optimized portfolio obtained a higher Sharpe ratio than the original index. By learning the performance of on the test dataset, we

verified that the PPO strategy should be an index enhancement strategy. Because it is greatly affected by macroeconomic and industry factors, and is still strongly correlated with index returns. On the other hand, when the macroeconomic is undertaking a downturn period, the index enhancement strategy may not be able to achieve satisfactory performance.

4.2 The Risk-Averse Agent

For the risk-averse PPO agent, we encourage the agent to avoid risks by increasing the penalty of losses in the reward function.

Setting the penalty parameter on losses to be 1.05, in the test period, the risk-averse PPO agent achieved an annualized rate of return of 41.5% and a Sharpe ratio of 2.28, both the annualized return and the Sharpe ratio are even higher than the ordinary agent. The portfolio return data for the Risk-averse agent can be found in Fig. 5.

As for the risk measure, the variance of the Risk-averse strategy and the benchmark rate of return is shown in Fig. 6, the risk-averse agent still bears more volatility risk than the benchmark. And the strategy drawdown is shown in Fig. 7, with a maximum drawdown being about 9.5%, also greater than the normal agent.

The return of risk-averse agent over time still has a strong synergistic correlation with the Dow Jones Industrial Index, indicating that the above analysis for ordinary agent is also applicable to the risk-averse agent, that the risk-averse agent is still an index enhancement strategy.

The surprising result is that the risk-averse PPO agent, not only improved the Sharpe ratio, but also increased the annualized rate of return, compared to the normal PPO agent. This may be because the penalty on losses, reduced the agent's overfitting to the training data, thereby improved the portfolio performance. Specifically, the risk averse PPO agent not only increased the rate of return but also decreased the volatility a little bit, which indicate that being a risk-averse investors will have extra advantage over the risk-indifference investors, in the domain of Dow Jones 30 industry index stocks.

In the economic literature, risk aversion is usually represented by a smooth monotonic concave utility function, while in our research we presented a linear punishment on daily losses, this forms a two-line function which is also monotonic and concave. But researchers can also try other continuous smooth function to represent the risk-aversion characteristics.

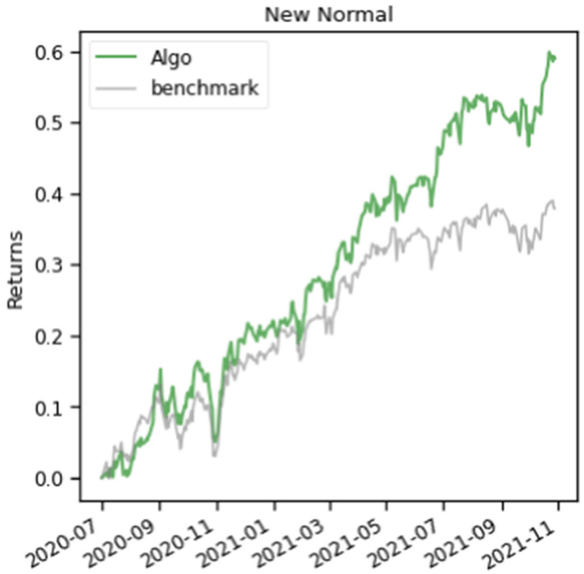

Fig. 5. Risk-Averse Agent Cumulated Return

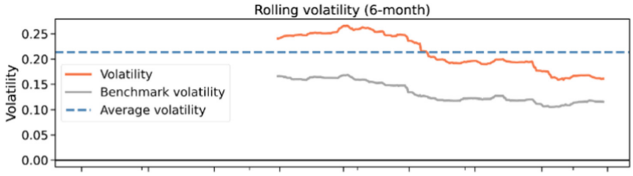

Fig. 6. Rolling volatility of RA Agent and Benchmark

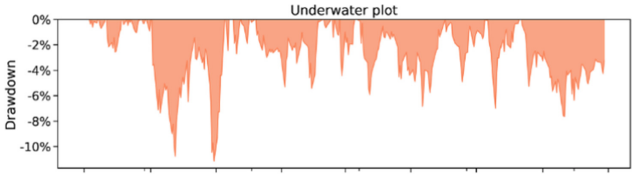

Fig. 7. Drawdown of the RA Agent

5 Conclusion

In this paper, we have built up a reinforcement learning environment based on more than 10 years daily data of Dow Jones Industrial Index stocks, and carried out the learning process to get better trading strategies. By altering multiple learning algorithms and multiple parameters, we have the following conclusions:

1) Stock trading problem can be viewed as a Markov decision process.

2) PPO algorithm and many other reinforcement learning methods can be used to solve MDP problems, while PPO algorithm provided stable and good out-of-sample performance. Strategies learnt by PPO algorithm get much higher expected return and higher Sharpe ratio than simply holding the benchmark index in the test dataset.
3) But the strategies learnt by PPO algorithm, should be considered as index enhancement strategies, at least based on the simple return and loss as reward function.
4) By adding the risk averse feature, PPO agent can achieve even better out-of-sample performance, if the parameter on loss as penalty was appropriately set.

References

1. Nguyen, H., & La, H. (2019, February). Review of deep reinforcement learning for robot manipulation. In *2019 Third IEEE International Conference on Robotic Computing (IRC)* (pp. 590–595). IEEE.
2. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676), 354-359.
3. Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., ... & Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782), 350-354.
4. Xiong, Zhuoran, et al. "Practical deep reinforcement learning approach for stock trading." arXiv preprint arXiv:1811.07522 (2018): 1–7.
5. Jiang, Zhengyao, Dixing Xu, and Jinjun Liang. "A deep reinforcement learning framework for the financial portfolio management problem." arXiv preprint arXiv: 1706.10059 (2017):1–31.
6. Kolm, Petter N., and Gordon Ritter. "Dynamic replication and hedging: A reinforcement learning approach." The Journal of Financial Data Science 1, no. 1 (2019): 159-171.
7. Buehler, Hans, Lukas Gonon, Josef Teichmann, Ben Wood, Baranidharan Mohan, and Jonathan Kochems. "Deep hedging: hedging derivatives under generic market frictions using reinforcement learning." Swiss Finance Institute Research Paper 19–80 (2019).
8. Liu, Xiao-Yang, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang. "FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance." arXiv preprint arXiv:2011.09607 (2020):1–12.
9. Chang, Ying-Hua, and Ming-Sheng Lee. "Incorporating Markov decision process on genetic algorithms to formulate trading strategies for stock markets." Applied Soft Computing 52 (2017): 1143-1153.
10. Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347 (2017): 1–12.