doi: 10.36689/uhk/hed/2023-01-039

Robo-advisor: A Deep Reinforcement Learning Algorithm with the Risk Tolerance Regularization for Portfolio Management

Cong Ma

Northwest University, Xi'an, China; mollyxjtu@163.com

Abstract: In recent years, deep reinforcement learning (DRL) is wildly applied to finance field, especially in portfolio management. DRL algorithm combines the perceptual ability of deep learning with the decision-making ability of the reinforcement learning, showing a better performance. However, the existing studies did not take the risk tolerance of the investor into account when managing the portfolio, leading a wild fluctuating return and a large drawdown to investors. Various emergencies have made the economic environment more uncertain, and posed more severe challenges to portfolio management. Considering this, a novel algorithm is proposed, called β-DRL algorithm. It proposes a novel risk-tolerance-related regularization and introduces it into the objective function. In this way, the proposed algorithm takes into account both the dynamic risk of the portfolio and the investor's risk tolerance. Extensive experimental results on Chinese Stock Market fully illustrate the excellence and robustness of the proposed algorithm.

Keywords: deep reinforcement learning; portfolio management; dynamic risk; risk tolerance

JEL Classification: C53; C58

1. Introduction

Portfolio management (PM) is a key part of the quantitative investment field, whose core issue is to balance the relationship between returns and risks, that is, investors can maximize returns within an acceptable risk. Since the financial market is affected by various complicated external factors such as society, politics, economy, and culture, its risks are time-varying. Therefore, how to build a robust investment portfolio strategy is a key issue of financial technology. Due to the characteristics of high dimensionality, high noise, and nonlinearity of financial data, traditional econometric methods have very limited ability to extract information from financial data, and it is difficult to grasp the non-stationary dynamics and complex interactions of financial markets.

Since the AlphaGo (Silver et al., 2016) debate the top professional player Lee Sedol, Deep Reinforcement Learning (DRL) algorithm have received extensive attention from various fields. DRL algorithm is also used in finance, such as, stock trading (Wu et al., 2020), PM (Betancourt et al., 2021), option pricing (Du et al., 2020), etc. There are many researchers proposed a lot of PM strategies using the DRL algorithm. The first PM system (Moody et al., 1998) using recurrent reinforcement learning (RRL) is proposed and used in S&P 500. A QSR system (Gao et al., 2000) was built based on Q-learning, which generates substantial profits in Forex market. A financial-model-free RL framework (Jiang et al., 2017) was built for cryptocurrency, which combines the Ensemble of Identical Independent Evaluators (EIIE) topology, Portfolio-vector Memory (PVM), a reward function together, and achieves a higher return. An adaptive RRL-PSO portfolio trading system (Almahdi et al., 2019) was proposed by combining particle swarm and RRL, showing a steady profit on S&P100 index stocks. Despite this, fewer researchers consider the dynamic risk tolerance of the investor in the financial market. Besides, different investors have different risk tolerance. Therefore, how to design a robust PM strategy to maximize the return within the risk tolerance of the investor?

This paper aims to build a robust portfolio strategy under the time-varying, dynamic risk. A novel algorithm is proposed for PM, called β-DRL algorithm. Firstly, a risk-related regularization is built by considering both the risk of the portfolio and the risk tolerance of the investor. Then, the regularization is introduced into the objective function. During the trading process, the agent with the improved objective function can find the optimal allocation. Extensive experiments on the Chinese stock market show the β-DRL robo-advisor can yield higher profits with lower risk than the existing state-of-the-art algorithms.

The remainder of this paper is organized as follows. Section 2 describes the Markov process and the proposed DRL algorithm for PM. Section 3 implements many numerical experiments and compares the experimental results of the β-DRL robo-advisor with several existing algorithms in the Chinese Stock Market. Section 4 discusses the limitation of the β-DRL algorithm. Section 5 concludes this paper.

2. Methodology: DRL Algorithm for PM

PM is a process where the investor allocates the fund among several different financial products. Suppose a portfolio contains m risky assets and one risk-free asset, m stocks are used as the risky assets and the remaining cash is used as the risk-free asset. Thus, the aim of PM is to continuously reallocate fund into the $m+1$ assets in each period in order to maximize the profit.

This part will illustrate how to implement PM using the DRL algorithm, as shown in Figure 1. PM can be regarded as a Markov Process (MDP), expressed as $M = (S, A, P, \gamma, r)$. Here, S is a set of states, A is a set of actions, $P: S \times A \times S \to [0,1]$ is the transition probability distribution, $\gamma \in (0,1]$ is the discount factor and $r: S \times A \to \mathbb{R}$ is the immediate reward after taking an action in a certain state. The agent will interact with the environment using the policy π_θ to get the next action, that is, $a_t = \pi_\theta(s_t)$. During the training, the agent will continuously interact with the financial market, and get a sequence of the trajectory including states, actions and rewards, expressed as $\tau = \{s_1, a_1, s_2, a_2, \dots, s_T, a_T\}$. For any trajectory, the transition probability distribution satisfies the Markov property, that is $p(s_{t+1}|s_1, a_1, \dots, s_t, a_t) = p(s_{t+1}|s_t, a_t)$. Then, the optimal policy $\pi_\theta: S \to A$ can be learned by maximizing the expected cumulative discounted return

$$J(\pi_\theta) = E_{\pi_\theta}[r(s,a)] = E_{\pi_\theta}[r(s, \pi_\theta(s_t))] \tag{1}$$

The following parts will illustrate the action space, state space, and reward function.

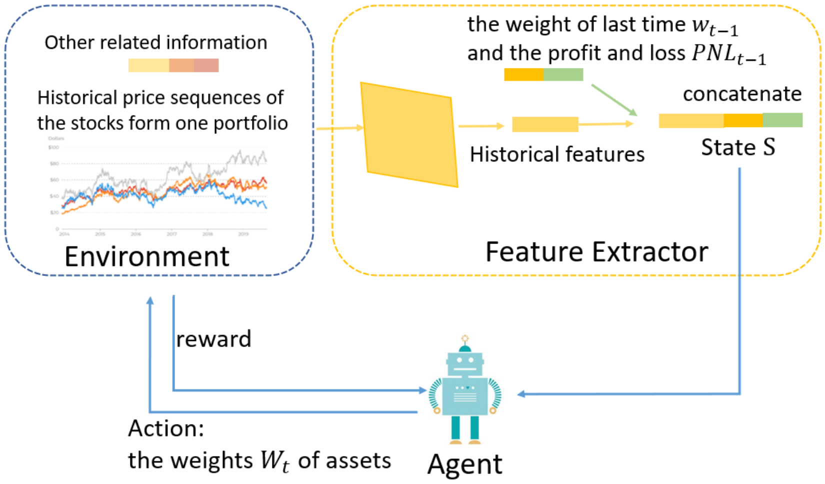
Other related information
Historical price sequences of the stocks form one portfolio
Environment
the weight of last time w_{t-1}
and the profit and loss PNL_{t-1}
concatenate
Historical features
State S
Feature Extractor
reward
Action:
the weights W_t of assets
Agent

Figure 1. The framework of DRL for PM

2.1. State Space

The state space contains the market information of m stocks, denoted as

$$S_t = \{S_{1,t}, S_{2,t}, \dots, S_{m,t}\} = \{S_{i,t}\}_{i=1}^{m}, \tag{2}$$

where $S_{i,t} = [h_{i,t}, o_{i,t}, l_{i,t}, c_{i,t}, V_{i,t}, MA_{i,t}^{5}, MA_{i,t}^{10}, OBV_{i,t}, MACD_{i,t}, w_{i,t-1}, PNL_{i,t-1}]$ is the state of each stock, including high, open, low, close prices, volume, moving average value of 5 days, and moving average value of 10 days. $w_{i,t-1}$ is the weight of last time, which is used to reduce the trading frequency. $PNL_{i,t-1}$ is the profit and loss at time $t-1$.

2.2. Action Space

The action is the proportion of funds allocated to each asset, represented as

$$W_t = \pi_\theta(S_t) = [w_{1,t}, w_{2,t}, w_{3,t}, \dots, w_{m,t}, w_{m+1,t}], \tag{3}$$

where $w_{i,t}(1 \le i \le m)$ is the weights of m stocks at time t, and $w_{m+1,t}$ is the weight of the remaining cash. And $\sum_{i=1}^{m+1} w_{i,t} = 1$, $0 \le w_{i,t} \le 1$, Besides, the initial weight is $W_0 =$ [0,0,. . ,0,1], which means the agent only holds cash at the beginning of the investment.

2.3. Reward with the Risk Tolerance Regularizer

Referring to the previous literature (Jiang et al., 2017), the price relative vector is defined as the price change, denoted as

$$y_t = \left(\frac{c_{1,t}}{c_{1,t-1}}, \frac{c_{2,t}}{c_{2,t-1}}, \frac{c_{3,t}}{c_{3,t-1}}, \dots, \frac{c_{m,t}}{c_{m,t-1}}, 1\right) \tag{4}$$

The portfolio value can be represented as

$$P_t = \mu_t P_{t-1} y_t \cdot W_{t-1}, \tag{5}$$

where μ_t is the remainder factor, which means the remainder of the total asset value after deducting commission. $P_0 = 1$. The logarithmic rate of return at time t can be denoted as

$$r_t = \text{In}\frac{P_t}{P_{t-1}} = \text{In}(\mu_t y_t \cdot W_{t-1}) \tag{6}$$

The portfolio value reward at the end of the trading period T is

$$P_T = P_{T-1}(\mu_T y_T \cdot W_{T-1}) \tag{7}$$

$$= P_{T-2}(\mu_{T-1}y_{T-1} \cdot W_{T-2})(\mu_T y_T \cdot W_{T-1})$$

$$= \cdots$$

$$= P_0 \prod_{t=1}^{T} \mu_t y_t \cdot W_{t-1} \tag{8}$$

The mean of logarithmic cumulative return R for one episode is

$$R(s_1, a_1, s_2, a_2, \dots, s_T, a_T) = \frac{1}{T} \ln \frac{P_T}{P_0}$$

$$= \frac{1}{T} \sum_{t=1}^{T} \ln(\mu_t y_t \cdot W_{t-1}) \tag{9}$$

Here, considering the time-varying risk, the objective function is modified by introducing the risk-related restriction into the return function

$$\max_{\theta} R(s_1, a_1, s_2, a_2, \dots, s_T, a_T) = \max_{\theta} \frac{1}{T} \sum_{t=1}^{T} \ln(\mu_t y_t \cdot W_{t-1}) + \mu(\beta - \sigma_{t-1}) \tag{10}$$

where β is the risk tolerance of the investor, $\sigma_{t-1} = \sigma_{t-1}(r_{t-30}, r_{t-29}, \dots, r_{t-1})$ is the standard deviation of the immediate returns of the past 30 trading days, and $0 < \mu < 1$ is an adjustment parameter. The larger the β value, the stronger of the investor's risk tolerance.

Therefore, the agent with the novel regularization is called β-DRL algorithm to find a robust PM trading strategy.

3. Experimental Results

In this part, the effectiveness and robustness of the proposed algorithm will be illustrated by implementing several experiments on Chinese Stock Market.

3.1. Dataset and Experimental Setup

The daily data of these stocks is public and can be collected from the online quantitative platform (JoinQuant, 2022), which provides Shanghai and Shenzhen Stock Exchanges from 2005 to the present. Table 1 shows three different portfolios. These stocks of each portfolio belong to the same industry. The training period is set from January, 2005 to December, 2018, and the test period is from January, 2019 to December, 2019. The Deep Deterministic Policy Gradient (DDPG) algorithm is used to train the model. Grid search is used to choose the optimal parameters.

Table 1. Three different portfolios

Portfolios	Stocks	Time period
1	000768, 600038, 600391, 002013, 600316	2015/01/01-2019/12/31
2	002032, 002035, 600690, 000521, 000651	2015/01/01-2019/12/31
3	600036, 600000, 600016, 600015, 000001	2015/01/01-2019/12/31

The initial cash is 1,000,000 RMB. The transaction fees for buying and selling stocks are set according to the real trading. The buying and selling are charged at 0.1% and 0.2% of the amount of the transaction, respectively.

3.2. Evaluation Metrics

The performance of different algorithms is evaluated by the following three metrics.

- Cumulative Rate of Return (CRR): The real return rate of an investment over time. It can be expressed as

$$CRR = \frac{P_T - P_0}{P_0} \times 100\% \tag{11}$$

- Sharpe ratio (SR): The additional amount of return that an investor receives per unit of increase in risk. It is defined as

$$SR = \frac{R_p - R_f}{\sigma_p} \tag{12}$$

where R_p is the asset return, R_f is the risk-free return, σ_p is the standard deviation of the asset excess return. Higher SR means better return under the same risk.

Maximum drawdown (MDD): The measure of the decline from a historical peak during the investment. It represents the risk of the portfolio, and can be represented as

$$MDD = \max_{0 \le t_1 \le t_2 \le T} \frac{R_{t_1} - R_{t_2}}{R_{t_1}} \tag{13}$$

Here, T is the investment period, R_{t_1}, R_{t_2} are the cumulative returns at time t_1 and t_2, respectively.

3.3. Experiment Results

In this part, extensive numerical experiments are performed on the three portfolios to compare the performance of the proposed $\boldsymbol{\beta}$-DRL algorithm with several baseline algorithms, including Robust Median Reversion (RMR) (Huang et al., 2016), On-Line Portfolio Selection with Moving Average Reversion (OLMAR) (Li et al., 2015), Passive Aggressive Mean Reversion (PAMR) (Li et al., 2012), Ensemble of Identical Independent Evaluators with CNN instant (EIIE-CNN) (Jiang et al., 2017) and the naive DRL algorithms. The experimental results of several different algorithms are shown in Table 2.

Table 2. The performance of the proposed algorithm and several existing algorithms

Portfolios	Metrics	RMR	OLMAR	PAMR	EIIE-CNN	Naïve DRL	$\boldsymbol{\beta}$-DRL
1	CRR	28.72%	22.01%	18.26%	35.62%	32.26%	49.75%
	MDD	-13.65%	-21.46%	-30.27%	-12.78%	-14.27%	-7.89%
	SR	0.89	0.78	-0.16	0.89	0.76	1.02
2	CRR	38.65%	42.78%	40.69%	55.36%	45.47%	61.26%
	MDD	-9.86%	-11.39%	-8.14%	-6.39%	-7.63%	-4.75%
	SR	1.16	1.14	1.22	1.48	1.32	1.86
3	CRR	26.32%	27.68%	28.66%	42.46%	38.69%	47.25%
	MDD	-17.65%	-18.96%	-16.62%	-14.69%	-12.68%	-9.65%
	SR	0.98	1.01	1.14	1.20	1.26	1.68

From these results, it's clear that the CRR value of β-DRL algorithm significantly exceeds the other baseline algorithms in the three portfolios. Focusing on the results of Portfolio 1, the CRR of β-DRL algorithm is 49.75%, and higher than other algorithms. At the same time, the MDD value of the β-DRL is -7.89%, which is also obviously lower than others. Moreover, its corresponding SR value is the highest among these algorithms. Similar results can be seen from the other two portfolios. These results fully illustrate the superiority of the proposed β-DRL algorithm.

Further, the total asset curves of Portfolio 1 are shown in Figure 2 which are implemented by several different algorithms. It is obvious that total asset values of the β-DRL algorithm is higher than other algorithms. Besides, the curve of β-DRL algorithm is relatively steady growth, while the curves other algorithms have relatively large fluctuations. These results further demonstrate the superiority and effectiveness of our proposed β-DRL.

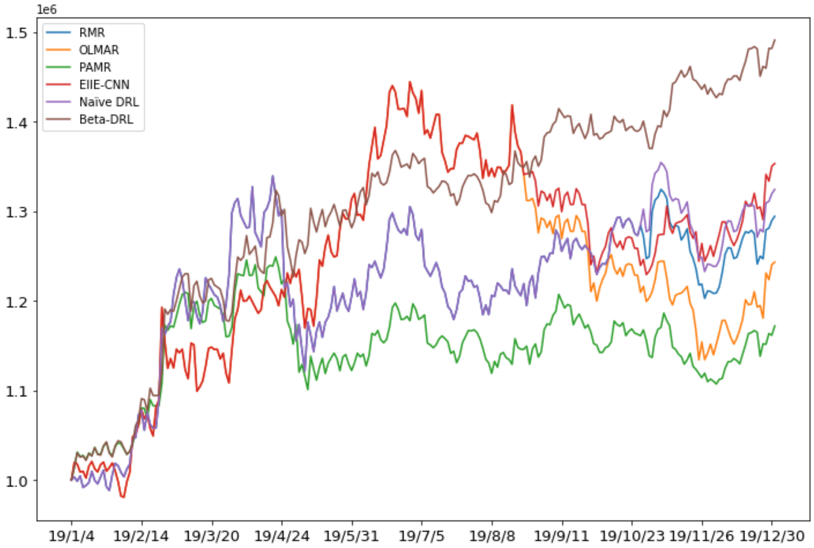

Figure 2. The total asset curve of Portfolio 1. The X-axis represents time, and the Y-axis is the total asset value

4. Discussion

The key of the proposed algorithm is to choose an appropriate value of the risk tolerance β for the investors. A larger value is for the risk appetite; and a smaller value is for risk averse. During the real trading, the risk tolerance of the investors can be assessed through a questionnaire and credit evaluation. Then, the robo-advisor can give a dynamic and profitable portfolio plan. And most of the existing algorithms are tested on American or European Stock Market, but our proposed beta-DRL is more suitable for Chinese Stock Market, and shows a better performance.

It must be emphasized that our proposed algorithm assumes that all orders can be executed at the close price. But in actual transactions, the transaction order may not necessarily be executed at the specified price. It will cause the return of the real trading is worse than the return of the experimental results. But the performance of the proposed algorithm is also better than these several baseline algorithms.

Meanwhile, the proposed algorithm has some limitations. Such as, the risk tolerance of the investors is evaluated by the questionnaire and is set at the beginning the investment. But in real trading, the risk tolerance of the investors is time-varying and will be influenced by the external environment. A dynamic evaluation method for risk tolerance should be studied in the following work. Besides, the proposed β-DRL algorithm can only deal with general unsystematic risks by diversifying the portfolio, and cannot cope with the shocks on the financial market posed by emergencies, such as the COVID-19, extreme weather, war or other extreme events. These emergencies have posed a severe challenge to PM. Benefited from the development and boom of artificial intelligence, some researchers try to predict these emergencies by collecting and extracting the related or useful information from website, newspaper, twitter, Instagram, etc. In the future work, I will try to combine it with the PM to build a more robust strategy.

Besides, in order to further improve the anti-risk ability of the algorithm, I will consider the risk from the perspectives of systematic risk and unsystematic risk. For the unsystematic risk, researchers or investors typically diversify and lower risk by diversifying the investment or building portfolios. Compared with unsystematic risks, systematic financial risks will cause a macroeconomic downturn and a decline in total output. And the financial and economic crisis caused by the outbreak of systematic financial risks will cause significant losses to the social economy. Thus, I should pay more attention to systematic risk. For systematic risk, people tend to focus monitoring risk and early warning in short-term. But systematic risks take a long time to brew and occur very rarely, leading that short-term monitoring and early warning may be difficult to detect. The outbreak of systematic risk is the process of the system from a normal state to an abnormal state, and a structural break occurs. And even if some changes are detected in advance, the risk has quickly spread to other sectors and caused huge damage to all economic and social system. Thus, studying the mechanism of the systematic risk can help us to understand the system evolution logic before the outbreak of the serious risk, so as to grasp the probability of the key systematic risk and reduce their damage.

5. Conclusions

This paper proposes a novel trading strategy for PM based on the DRL algorithm, named β-DRL algorithm. And the proposed algorithm can not only avoid risks, but also maximize the return within the investor's risk tolerance. Many experimental results fully show the effectiveness and superiority of the β-DRL algorithm on the Chinese Stock Market.

Acknowledgments: This work was in part supported by the Ministry of Education of Humanities and Social Science Project of China (No. 22XJCZH004), in part supported by the Natural Science Foundation of Shaanxi Province of China (No. 2023-JC-QN-0799), and in part supported by the Scientific Research Project of Shaanxi Provincial Department of Education (No. 22JK0186).

Conflict of interest: none.

References

Almahdi, S., & Yang, S. Y. (2019). A constrained portfolio trading system using particle swarm algorithm and recurrent reinforcement learning. *Expert Systems with Applications*, *130*, 145-156. https://doi.org/10.1016/j.eswa.2019.04.013

Betancourt, C., & Chen, W. H. (2021). Deep reinforcement learning for portfolio management of markets with a dynamic number of assets. *Expert Systems with Applications*, *164*, 114002. https://doi.org/10.1016/j.eswa.2020.114002

Du, J., Jin, M., Kolm, P. N., Ritter, G., Wang, Y., & Zhang, B. (2020). Deep reinforcement learning for option replication and hedging. *The Journal of Financial Data Science*, *2*(4), 44-57. https://doi.org/10.3905/jfds.2020.1.045

Gao, X., & Chan, L. (2000). An algorithm for trading and portfolio management using q-learning and sharpe ratio maximization. In *Proceedings of the international conference on neural information processing* (pp. 832-837).

Huang, D., Zhou, J., Li, B., Hoi, S. C., & Zhou, S. (2016). Robust median reversion strategy for online portfolio selection. *IEEE Transactions on Knowledge and Data Engineering*, *28*(9), 2480-2493. https://doi.org/10.1109/TKDE.2016.2563433

Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. https://doi.org/10.48550/arXiv.1706.10059

JoinQuant. (2022). *An online quantitative trading platform*. https://www.joinquant.com/

Li, B., Zhao, P., Hoi, S. C., & Gopalkrishnan, V. (2012). PAMR: Passive aggressive mean reversion strategy for portfolio selection. *Machine Learning*, *87*(2), 221-258. https://doi.org/10.1007/s10994-012-5281-z

Li, B., Hoi, S. C., Sahoo, D., & Liu, Z. Y. (2015). Moving average reversion strategy for on-line portfolio selection. *Artificial Intelligence*, *222*, 104-123. https://doi.org/10.1016/j.artint.2015.01.006

Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting*, *17*(5-6), 441-470. https://doi.org/10.1002/(SICI)1099-131X(1998090)17:5/6<441::AID-FOR707>3.0.CO;2-%23

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484-489. https://doi.org/10.1038/nature16961

Wu, X., Chen, H., Wang, J., Troiano, L., Loia, V., & Fujita, H. (2020). Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, *538*, 142-158. https://doi.org/10.1016/j.ins.2020.05.066