FinRL

Team 25
B09901041 Tsai, Yun-Tong
B08505048 Liu Ming-Kai
B09901142 Lu, Jui-Chao
P11942A03 Hsieh, Ting-Sheng

1 Introduction

The realm of stock trading has long been dominated by human decision-making, which may face challenges due to the ever-fluctuating nature of financial markets, and makes it hard for stockholders to maximize returns. Generally, human judgements suffer from following constraints:

Susceptibility to Emotions: Human traders are inherently emotional beings, and emotions can significantly impact decision-making in the stock market. Fear and greed, often triggered by market volatility, can lead to impulsive actions, deviating from a rational investment strategy.

Cognitive Biases: Cognitive biases, stemming from psychological tendencies, can cloud judgment and introduce errors in decision-making. Anchoring, confirmation bias, and overconfidence are just a few examples that can impede an investor's ability to accurately assess market information.

Time Constraints: Monitoring the stock market in real-time requires constant attention, a resource that humans often find limited due to other responsibilities and commitments. This limitation may result in missed opportunities or delayed responses to changing market conditions.

To maximize assets of stockholders with sensible and intelligent strategies, the exploration of alternative approaches becomes indispensable. An wise method is to train an agent to support stockholders to make decisions. However, the agent must have strong adaptability to changes of stock prices, and be trained without using much time. Therefore, in this paper, a method that combines a meta learning algorithm called Model-Agnostic Meta-Learning, and the concept reinforcement learning is introduced in the following sections.

2 Related work

Model-Agnostic Meta-Learning (MAML) has emerged as a powerful technique in machine learning, particularly in scenarios where fast adaptation to new tasks is crucial. [1] introduced MAML as a meta-learning algorithm that facilitates rapid learning across diverse tasks by efficiently parameterizing models. MAML's ability to quickly adapt to new tasks by fine-tuning its initial parameters has found applications in various domains, including computer vision, natural language processing, and robotics. In the context of reinforcement learning (RL), MAML has been extended to adapt to sequential decision-making problems. This extension enables agents to efficiently learn from small amounts of experience, making it highly promising for training agents capable of swift adaptation to new environments or changing dynamics, such as financial markets.

Reinforcement Learning in Finance Reinforcement learning (RL) has gained traction in financial applications due to its ability to learn from interactions with an environment to optimize decision-making processes. In finance, RL models seek to maximize cumulative rewards by making sequential decisions in dynamic and uncertain environments, such as stock markets. RL algorithms, including Q-learning, Deep Q Networks (DQN), and policy gradient methods like Proximal Policy Optimization (PPO) and Actor-Critic models, have been employed to address various financial tasks. These tasks include portfolio allocation [2, 3], algorithmic trading [4, 5, 6, 7, 8, 9, 10], risk management, and market prediction. However, traditional RL methods often face challenges when dealing with non-stationary and volatile financial data. Sudden changes in market conditions or the emergence of new patterns can lead to poor performance or the inability to adapt swiftly. This limitation underscores the need for more adaptive and flexible reinforcement learning approaches in financial domains.

Reinforcement Learning (Meta-RL) represents a paradigm within RL that focuses on learning procedures or algorithms that can adapt to new tasks quickly [1, 11, 12, 13, 14, 15, 16]. Meta-RL algorithms aim to generalize knowledge across tasks, enabling agents to learn efficiently from limited data. Within the domain of financial trading, Meta-RL has garnered attention for its potential to equip trading agents with the ability to adapt to various market conditions [17, 18, 19]. Meta-RL techniques, combined with RL algorithms, aim to enable agents to continuously improve their trading strategies by leveraging meta-learning principles.

3 Problem formulation

Goal We aim to train an agent that can swiftly adapt to unseen stock price trend and make accurate trading choices. We meta-train the agent to acquire the parameters that best adapts to all the sampled meta-training tasks and meta-test the agent on meta-testing tasks. We want to maximize the rate of return with fixed initial asset, trading on a single stock over the trading period.

State **s** consists of daily stock prices, daily trading volume and daily technical indicators. Technical Indicators includes Moving Average Convergence/Divergence (MACD), Bollinger Bands (BOLL), Relative Strength Index (RSI), Commodity Channel Index (CCI), Directional Movement Index (DX), closing price 30/60 Simple Moving Average (SMA).

Action **a** is a float between -1$\sim$1. Action **a** will be multiplied by a factor $hmax$ to represent the amount of shares being bought/sold at that day (e.g **a** = 0.5 means buying $0.5 * hmax$ shares, **a** = -0.3 means selling $0.3 * hmax$ shares)

4 Methods

4.1 pre-categorization

Our dataset covers a comprehensive array of financial market data, consisting of daily stock prices, daily trading volume and daily technical indicators. Empirically, we observe the agent would stagnating from making trading decisions if we don't pre-categorized the dataset. We believe this can attribute to the agent's overfitting upward, downward or bumpy segments. Such a bias could prompt the agent to adopt a conservative approach, refraining from making trading decisions to retain the asset. Figure 1 shows the categorization of tasks.

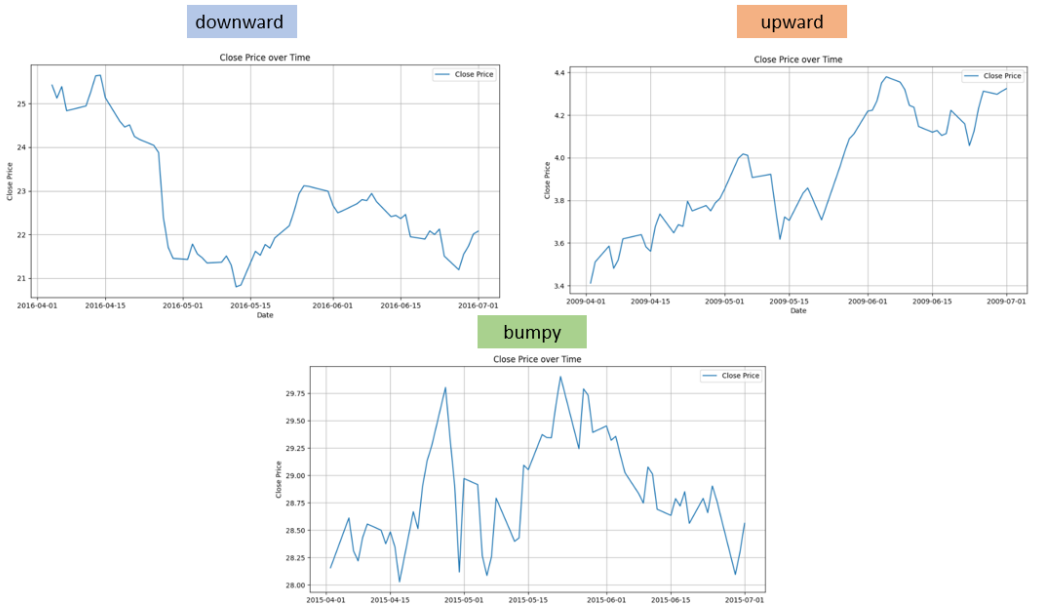

Figure 1: examples from each categories

Therefore, we partitioned the dataset into segments based on their closing price trends. Three categories were identified: Upward, Downward, and Bumpy. Each segment encompassed a 3-month duration.

4.2 meta-training

Meta-Training Algorithm:
L = loss function within A2C algorithm

Algorithm 1 Meta-Training

Require: $p_{upward}(\tau), p_{downward}(\tau)$, and $p_{bumpy}(\tau)$: distribution over segments in each categories
Require: β: step size hyperparameter
random initialize θ
while not done **do**
    Sample batch of tasks $\tau_u \sim p_{upward}(\tau), \tau_d \sim p_{downward}(\tau)$, and $\tau_b \sim p_{bumpy}(\tau)$
    ∇_{meta} = **0**
    for all τ_u, τ_d, and τ_b **do**
        Sample k trajectories D_i using θ
        θ' = use A2C algorithm and D_i to update θ
        Sample k trajectories D_i' using θ'
        ∇_{meta} += $\nabla_{\theta'} L(D_i')$
    end for
    Update $\theta \leftarrow \theta - \beta \nabla_{meta}$
end while

4.3 meta-testing

Meta-Testing algorithm:

Algorithm 2 Meta-Testing

Require: df: stock data
Require: θ: meta-trained parameters
Require: f_{adapt}: frequency of adaptation
adapted = **False**
for $i = 1, 2, \ldots, len(df)$ **do**
    if i >= 90 and i % $f_{adapt} = 0$ **then**
        adapted = **True**
        Sample k trajectories D_i from df[i - 90 : i] using θ
        θ' = use A2C algorithm and D_i to update θ
        Predict action **a** using state **s** = df[i] and θ'
    else
        if adapted **then**
            Predict action **a** using state **s** = df[i] and θ'
        else
            Predict action **a** using state **s** = df[i] and θ
        end if
    end if
end for

5 Experiment

5.1 Dataset

We chose the DOW-30 stocks as our dataset and download it from the Yahoo Finance. The total time span is from 2009/01/01 to 2021/10/29. We split the train/test dataset at 2020/07/01. For the criteria of determining the Upward, Downward, and Bumpy categories, we set the threshold to 0.05. That is, the difference between the three-month time span being greater than five or below minus five percent of the original price would be in the Upward or Downward group, and the others would be in the Bumpy groups.

5.2 Environment

The environment we trained the agents is StockTradningEnv provided by AI4Finance-Foundation. Since we perform trading on single stock so we set the stock dimension to the total indicators of one stock. For each task (three month), we formulate an environment which are used to training the agent in an episode. The action of the agent is a real number in[-1, 1], which range from selling the max number of shares to buying the max number of shares. We also set the max number of trading and the initial amount of funds.

5.3 Baseline

For the baseline we trained a simple A2C agent, which along with its hyperparameters are the same as the agent we used in MAML. We trained the baseline agent in the same environment mentioned above.

5.4 Results

After meta-training the model, we performed meta-testing. We first load the meta-trained parameters into a model, and then adapted the model to the 90 days before a time point, and let the agent trade for the following 15 days. By repeating the above actions, we got the total returns at the end time point. afterwards, we compared our meta-trained agent with the baseline, the results on the first eight stocks of DOW-30 are shown on Table 1 below. We also compared the meta-trained agent adapting to different time span, which is, 0, 10, and 15 days. As the result has shown, agent without adapting actually perform better than either adapting to 10 and 15 days of stock data.

Ticker	**Baseline**	**MAML-RL**
AAPL	68.21	**73.36**
AMGN	16.91	**19.9**
AXP	79.51	**126.48**
BA	124.45	58.11
CAT	59.83	**63**
CRM	82.82	**109.13**
CSCO	39.05	30.19
CVX	61.42	45.21

Table 1: Return Rate(%)

MAML-RL			
Ticker \ f_{adapt}	0	10	15
AAPL	**73.36**	71.11	72.66
AMGN	**19.9**	-1.92	11.35
AXP	**126.5**	67.4	87.35
BA	58.11	**62.72**	48.14
CAT	**63**	57.01	38.01
CRM	**109.1**	80.12	70.89
CSCO	30.19	22.37	**33.09**
CVX	45.21	34.88	**45.89**

Figure 2: Adapting agent to different time span

6 Conclusion

The application of MAML algorithm in training agents has shown promising results in terms of adaptability and performance. For future work, refining hyperparameter settings stands out as a crucial aspect of further improving the efficacy of MAML in this domain. Hyperparameters such as learning rate, timesteps, etc, are the important keys for reinforcement learning. By optimizing hyperparameters, we can unlock more potential of meta-learning techniques and develop stock trading agents that are not only adaptive but also consistently outperform traditional approaches.

Besides, currently our results focus on training and test an agent using the same ticker, but using different timeframes. The possible future work could extend to more general scenarios, that is, in one task, the agent is trained based on portfolio which allocates assets on different tickers at a same time. The scenario With training agent doing portfolio would be more realistic and practical for stockholders.

References

[1] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." International conference on machine learning. PMLR, 2017.

[2] Xinyi Li, Yinchuan Li, Yuancheng Zhan, and Xiao-Yang Liu. Optimistic bull or pessimistic bear: Adaptive deep reinforcement learning for stock portfolio allocation. ICML Workshop on Applications and Infrastructure for Multi-Agent Learning, 2019.

[3] Zhengyao Jiang, Dixing Xu, and J. Liang. A deep reinforcement learning framework for the financial portfolio management problem. ArXiv, abs/1706.10059, 2017.

[4] John Hull et al. Options, futures and other derivatives/John C. Hull. Upper Saddle River, NJ: Prentice Hall, 2009

[5] Jinke Li, Ruonan Rao, and Jun Shi. Learning to trade with deep actor critic methods. 2018 11th International Symposium on Computational Intelligence and Design (ISCID), 02:66–71, 2018.

[6] Lin Chen and Qiang Gao. Application of deep reinforcement learning on automated stock trading. In 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), pages 29–33, 2019.

[7] Quang-Vinh Dang. Reinforcement learning in stock trading. In ICCSAMA, 2019

[8] Zhuoran Xiong, Xiao-Yang Liu, Shan Zhong, Hongyang Yang, and Anwar Walid. Practical deep reinforcement learning approach for stock trading. NeurIPS Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy, 2018.

[9] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. Deep reinforcement learning for automated stock trading: An ensemble strategy. ACM International Conference on AI in Finance (ICAIF), 2020.

[10] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep reinforcement learning for trading. The Journal of Financial Data Science, 2(2):25–40, 2020.

[11] TomZahavy, Zhongwen Xu, Vivek Veeriah, Matteo Hessel, Junhyuk Oh, Hado van Hasselt, David Silver, and Satinder Singh. "A self-tuning actor-critic algorithm". In: arXiv preprint arXiv:2002.12928 (2020).

[12] Zhongwen Xu, Hado P van Hasselt, Matteo Hessel, Junhyuk Oh, Satinder Singh, and David Silver. "Meta-Gradient Reinforcement Learning with an Objective Discovered On line". In: Advances in Neural Information Processing Systems. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 15254–15264. URL: https://proceedings.neurips.cc/paper/2020/file/ae3d525daf92cee0003a7f2d92c34ea3-Paper.pdf.

[13] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. "Learning to reinforce ment learn". In: arXiv preprint arXiv:1611.05763 (2016).

[14] Junhyuk Oh, Matteo Hessel, Wojciech M Czarnecki, Zhongwen Xu, Hado van Hasselt, Satinder Singh, and David Silver. "Discovering reinforcement learning algorithms". In: arXiv preprint arXiv:2007.08794 (2020).

[15] Louis Kirsch, Sjoerd van Steenkiste, and Jürgen Schmidhuber. "Improving generalization in meta reinforcement learning using learned objectives". In: arXiv preprint arXiv:1910.04098 (2019).

[16] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. "RL2: Fast reinforcement learning via slow reinforcement learning". In: arXiv preprint arXiv:1611.02779 (2016).

[17] Wong, Wei Jie. "Meta-reinforcement learning in quantitative trading." (2022).

[18] Sorensen, Erik, et al. "Meta-learning of evolutionary strategy for stock trading." Journal of Data Analysis and Information Processing 8.2 (2020): 86-98.

[19] Harini, S. I., et al. "Neuro-symbolic Meta Reinforcement Learning for Trading." arXiv preprint arXiv:2302.08996 (2023).