FinRL Contest 2024 Task I: BTC Trading Via Online Adaptation of a Mixture of Offline Experts

Giovanni Dispoto
Politecnico di Milano
Milan, Italy
giovanni.dispoto@polimi.it

Lorenzo Campana
Politecnico di Milano
Milan, Italy
lorenzo.campana@mail.polimi.it

Antonio Riva
ML cube
Milan, Italy
antonio.riva@mlcube.it

Amarildo Likmeta
Politecnico di Milano
Milan, Italy
amarildo.likmeta@polimi.it

Keywords

Offline Reinforcement Learning, Online Adaptation, BTC Trading

ACM Reference Format:
Giovanni Dispoto, Lorenzo Campana, Antonio Riva, and Amarildo Likmeta. 2024. FinRL Contest 2024 Task I: BTC Trading Via Online Adaptation of a Mixture of Offline Experts. In . ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Automated cryptocurrency trading has emerged as one of the most compelling applications of artificial intelligence techniques in finance, as the economic outcomes of each action, namely profits or losses (P&L), can be measured precisely and leveraged as feedback for training Reinforcement Learning (RL) models.

However, the success of these techniques is highly contingent on the robustness of their validation and selection procedures, which typically rely on back-testing candidate strategies on the most recent historical data [1]. This approach is inherently limited, as it assumes that future trading patterns will resemble those observed in the validation set, whereas, in practice, trading opportunities are often transient and highly sensitive to shifting market conditions. Financial markets, especially cryptocurrency markets, frequently experience regime shifts which violate the stationarity assumptions on which standard RL frameworks are built [2].

In this report, we propose a novel ensemble-based approach designed to address the challenges posed by the intrinsic non-stationarity of financial markets, providing a solution to the limitations of traditional validation techniques. Building on the work of [3], we designed a two-tier architecture, where an adaptive, profit-driven online validation layer is added on top of a core RL component.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Our method consists of two phases: first, multiple RL models are trained offline to generate a diverse set of tradings strategies. Then, an online learning algorithm dynamically adjusts the weights of each strategy based on recent P&L performance. More specifically, we update these weights at the start of each trading episode and use them to sample the strategy to follow throughout the episode.

The report is organized as follows: Section 2 provides a brief introduction of the theoretical background of the implemented solution. Section 3 describes the problem formulation, focusing on the expert training procedures and the OAMP algorithm. Section 4 presents the trading performance of the ensemble method, and Section 5 offers our concluding thoughts.

2 Preliminaries

2.1 Online Learning with expert advice

In online learning with expert advice [4], the agent has to predict the outcome $y_t \in \mathcal{Y}$ based on the past sequence $y_1, y_2, ..., y_{t-1}$ of events that occurred in some outcome space $\mathcal{Y}$. To do that, it is allowed to choose among the suggestions of a set of experts $\mathcal{K}$. At each step in the prediction game, every expert $k \in \mathcal{K}$ predicts an element $\widehat{y}_t^k \in \mathcal{Y}$ and incurs a loss $f(\widehat{y}_t^k, y_t)$. After its choice, the agent suffers a loss corresponding to the selected expert, but it is also allowed to observe all the other losses, differently from the *bandit* [5] and RL settings.

Optimistic Adapt ML Prod. The Optimistic Adapt ML Prod (OAMP, [6]) is an online, parameter-free, expert learning algorithm that aims at finding the optimal policy in non-stationary environments. Analyzing in detail the steps of OAMP described in Algorithm 1, we can notice that, at each time step t, the weights $\mathbf{w}_t$, and consequently the probabilities $\mathbf{p}_t$ associated with each expert are updated based on the instantaneous regret $\mathbf{r}_t$ and an appropriate estimate $\mathbf{m}_t$ of the former defined as follows:

$$r_t^k = \langle \mathbf{p}_t, \mathbf{l}_t \rangle - l_t^k, \qquad m_t^k = \langle \mathbf{p}_t, \mathbf{l}_{t-1} \rangle - l_{t-1}^k, \tag{1}$$

where $\mathbf{l}_t$ is the vector of losses measured by the experts at time t. Specifically, the higher the regret suffered by the agent or the greater the deviation from its previous performance, the lower the probability of selecting that expert.

Algorithm 1: Optimistic Adapt ML Prod (OAMP)

Initialize: K experts
Set $w_0^k = \frac{1}{K}$, $l_0^k = 0$, $\eta_0^k = \frac{1}{4}$, $\forall k \in [K]$
for $t = 1, 2, ..., T$ **do**
  Update $\widetilde{w}_{t-1}^k = w_{t-1}^k e^{\eta_{t-1}^k m_t^k}$
  Update $p_t^k = \frac{\eta_{t-1}^k \widetilde{w}_{t-1}^k}{\langle \eta_{t-1}, \widetilde{w}_{t-1} \rangle}$
  Sample an expert according to $\mathbf{p}_t$, then receive loss vector $\mathbf{l}_t$
  Update $\eta_t^k = \min \left\{ \frac{1}{4}, \sqrt{\frac{\ln K}{1+\sum_{s\in[t]}(r_s^k - m_s^k)^2}} \right\}$
  Update $w_t^k = \left[w_{t-1}^k e^{\eta_{t-1}^k r_t^k - (\eta_{t-1}^k)^2 (r_t^k - m_t^k)^2} \right]^{\frac{\eta_t^k}{\eta_{t-1}^k}}$

3 Problem Formulation

3.1 Expert Training

The effectiveness of our trading architecture improves with the diversity of the set of RL experts. Ideally, if each expert was specialized in a specific market regime, the online learning algorithm could identify the expert most suited to the prevailing market conditions and assign it a higher weight. Conversely, the more similar the experts are, the greater the likelihood that, when a new market regime emerges, none of them will be able to generate profit.

To create a well-diversified set of experts, we split the training data into 10 folds, each representing a calendar day. In each fold, excluding the first two which will be used for validating the OAMP algorithm[1], we train three RL agents (DQN [7], PPO [8], and FQI [9]) validating their hyperparameters on the subsequent day. All agents are trained and evaluated in an episodic setting, with each episode consisting of 480 2-seconds steps. Note that FQI requires an offline dataset of agent-environment interactions as input. For each training day, we construct this dataset by generating 1000 episodes following four baseline policies: i) Random Agent, ii) Short Only Agent, iii) Long Only Agent, iv) Flat Only Agent.

Finally, from this set of 21 agents we select the best agent for each trading day based on the performance on the validation set which are reported in Table 1.

3.2 Ensemble Trading

During the ensemble trading phase, we use the same episodic setting adopted in training. At the start of each episode, the OAMP algorithm updates the probability of selecting each expert based on its most recent performance, defined as the sum of the last N episode returns. After the weights are updated, the expert with the highest probability is selected and followed throughout the episode.

Note that N, known as the *loss function window*, is the only hyperparameter of OAMP that requires tuning. To do so, we test the performance of the ensemble method on the first day with five different values of N, selecting the one with the highest average episode return for the final evaluation of the trading architecture on the second day. The results of these tuning procedures are reported in Table 2.

[1]We choose the first two days for validating the ensemble, as the last two are the only ones characterized by a price downtrend. This approach approach ensures that we have a diverse set of experts, each exposed to different market regimes.

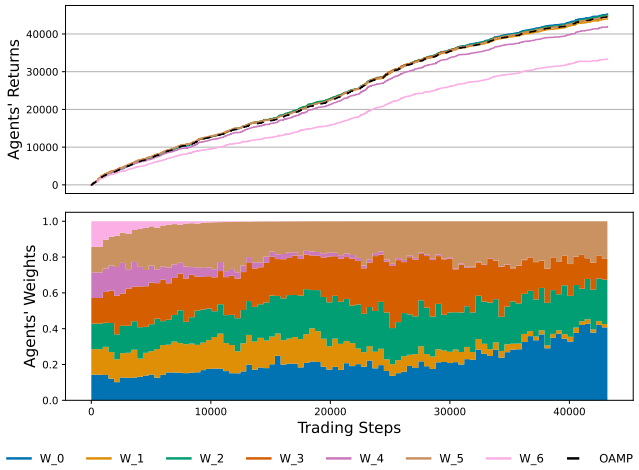

Figure 1: Performance of OAMP on April 8th with $N = 1$. Experts cumulative return and ensemble weights.

4 Experimental Results

Figure 1 shows the final evaluation of our ensemble method performed on April 8th with the 7 best agents and $N = 1$. The top row shows the cumulative return obtained by the ensemble method against those gained by the single agents. Although our trading procedure was not able to produce a well-diversified set of experts, we can observe that the performance of the OAMP algorithm is close to the ones of the best agents even though it chooses which expert to employ at runtime. Moreover, looking at the evolution of the experts' weights, one can see how the ensemble method quickly discards the worst agents to focus on the most promising ones.

April	9th	10th	11th	12th	13th	14th	15th
DQN	617	**524**	**590**	702	**1276**	113	**196**
FQI	647	511	565	**818**	1189	**158**	125
PPO	**687**	496	557	805	1233	111	152

Table 1: Trained agents average episodic returns.

N	1	5	10	20	50
Return	**38750**	38692	38460	37833	37276

Table 2: OAMP cumulative episodic returns on April 7th.

5 Conclusions

In this report, we presented the approach of RL3 for trading BTC via online model selection of Offline trained agents. We trained an ensemble of experts capable of capturing varying market regimes, and proposed a method to select the best expert online while trading. This way we are able both to perform model selection at test time, as well as adapt to the changing market conditions addressing the non-stationarity of the market.

References

[1] Antonio Riva, Lorenzo Bisi, Pierre Liotet, Luca Sabbioni, Edoardo Vittori, Marco Pinciroli, Michele Trapletti, and Marcello Restelli. Learning fx trading strategies with fqi and persistent actions. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–9, 2021.

[2] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[3] Antonio Riva, Lorenzo Bisi, Pierre Liotet, Luca Sabbioni, Edoardo Vittori, Marco Pinciroli, Michele Trapletti, and Marcello Restelli. Addressing non-stationarity in fx trading with online model selection of offline rl experts. In *Proceedings of the Third ACM International Conference on AI in Finance*, ICAIF '22, page 394–402, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393768. doi: 10.1145/3533271.3561780. URL https://doi.org/10.1145/3533271.3561780.

[4] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[5] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[6] Chen-Yu Wei, Yi-Te Hong, and Chi-Jen Lu. Tracking the best expert in non-stationary stochastic environments. *CoRR*, abs/1712.00578, 2017. URL http://arxiv.org/abs/1712.00578.

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 00280836. URL http://dx.doi.org/10.1038/nature14236.

[8] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://dblp.uni-trier.de/db/journals/corr/corr1707.html#SchulmanWDRK17.

[9] Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.*, 6:503–556, dec 2005. ISSN 1532-4435.