Introduction

This work is an overview of several papers concerning RL methods in portfolio optimization problem. My main task is to explain intuition, main assumptions and results behind these texts.

Traditionally ML algorithms are of little use in that problem, because of their lack of interpretability. They mainly concentrate on predictions and, as we know, stock prices are very difficult to predict.

As for now, portfolio optimization problem is not solved with any of ML methods (including RL ones), meaning that there is no any method which you would choose and receive the best possible results. It is thus a very promising direction to continue to work on.

Problem statement

We, as portfolio managers, want to choose bunch of assets that, we believe, will perform great in the future. We then allocate weights on them in order to optimize that performance in terms of some objective. Obviously, the main indicator is future returns, but in Markowitz paradigm returns always come with risk associated to them. So the traditional objective should take into account the expected portfolio reward and the risk estimation both.

The portfolio is a set of n assets which have their own weights. For each of the next trading periods we would like relocate previous weights, maximizing the given objective. We are interested in the RL agent, which will make those allocations for us.

Why is it good to use RL in portfolio optimization setup?

Efficient markets assumption. We do not need to use all the information (all info about companies) in order to construct the environmental state. EFA states that all the companies histories are included in today's prices.

We assume that agents actions do not have influence on tomorrow assets prices (and if we trade relatively small amounts, it certainly holds), so we can use prices history many times for training/tests, so sample efficiency is great. Also, there is no correlation between our actions and future observations, which is very different from the standard case of computer games. So we can use sequences of observations(fixed windows) as a single state.

We assume that some third party could exercise our bets right after we place them. We would like to have stocks with good liquidities thus.

We have a natural notion of rewards - stock returns. Those returns are always immediate and depend directly on our weights allocations - actions.

Methods and evaluation

In the papers, state is a fixed window of m previous trading periods (in case of one of the papers, it is window of 52 previous periods, each equals to 30 minutes) and previous

portfolio weights (mainly used to estimate transaction costs). The actions are agent's weights allocations for the next trading period.

Our portfolio could contain a various number of assets. The only constraint that sum of their weights equals to one. We end up with continuous space of actions and we can try to discretize them, but it is unnatural in this setup. So main model in those papers is Deep Deterministic Policy Gradient(DDPG), which gives us continuous action space and nice critic's feature which evaluates actor's actions.

The core of the framework is the Ensemble of Identical Independent Evaluators(EIIE) topology. An EIIE is a NN which estimates potential future growths of assets. This architecture is scalable, because it takes only one asset state representation and computes growth score for it. Then it uses softmax layer calculating weights allocations for all assets from portfolio. In order to include transaction costs in the model, the portfolio weights of each period are recorded in a Portfolio Vector Memory. The EIIE is trained using Online Stochastic Batch Learning scheme (OSBL), which is compatible with both pre-trade training and online training during back-tests or online trading. OSBL helps agent to train in more stable way and to adapt networks weights with new samples all the time.

As I stated before, average reward shouldn't be considered as the final metric. Although we have very natural reward function, it is important to include risk measure also. So, many metrics could be used such as Sharpe ratio, Sharpe considering only negative variation, etc. Also you can minimize the general risk, so you can use Maximum Drawdown, Value at Risk, or simple portfolio variance minimization.

Each of the papers have training, validation and test datasets. Each of them trade in different markets (cryptocurrencies, American and China stocks), that's why trading periods, metrics, benchmarks and model hyperparameters differs. It is worth to state that explicit comparisons of the results are not appropriate.

Results

There are several obvious benchmark to which you should compare your algorithms. The first one is some market approximation (S&P 500 index, for example), the second is portfolio which allocations are equally distributed between all assets (1/n approach, could be also market approximation, let's assume we preselected some stocks before that approach). Then you can take whatever algorithm you want.

In general, all RL algorithms (CNN, RNN, LSTM policy network based) outperformed their closest opponents. In cryptocurrency market, the winner is oracle approach (the cheater that uses knowledge about future prices in order to place its bets), but the second place takes the RL (CNN based) agent. It is important to say, that despite the results of RL agents were better on average, 1/n approach gave approximately the same results on the American stock market.

Conclusions

Portfolio optimization problem come under several assumptions which allow us to use RL methods in that problem. We see that the results are quite promising, but few issues are to be resolved. Although all three implementations of EIIE approach gave good results, they all stated several improvements to make. The next two seems crucial for me:

- Clever asset pre-selection. For now, authors of papers only selected assets with high liquidity (it helps in the assumption that we should immediately exercise our bet). Most of the papers worked with some predefined sets of assets (from S&P 500, 10 cryptocurrencies, etc), but we want ideally to examine all the tradable universe. **It is a place where some kind of interpretation comes.**
- Another features apart from prices. The hypothesis of efficient markets is cool and sometimes it seems even correct, but additional features such as EPS, Book to market ratio could be important in predicting prices. There are few market anomalies whose existence was practically used by Fama and French in their 3-5 factor models.

References

Main link to papers/reports:
https://arxiv.org/abs/1706.10059
https://arxiv.org/pdf/1808.09940.pdf
https://ruohanzhan.github.io/fun_proj/trading/report.pdf
https://dias.library.tuc.gr/view/manf/81141

Additional links:
https://github.com/liangzp/Reinforcement-learning-in-portfolio-management-
https://medium.com/swlh/ai-for-portfolio-management-from-markowitz-to-reinforcement-learning-cffedcbba566
https://github.com/Rachnog/Deep-Portfolio-Management/blob/master/Reinforcement%20Learning.ipynb