Reinforcement Learning for Portfolio Managment

Oleksandr Vovchuk
Ukrainian Catholic University
vovchuk@ucu.edu.ua

Abstract

Portfolio management (PM) is a fundamental financial engineering task that is performed to achieve investment goals, maximal profits and minimal risks. This process requires continuous decision making and optimization using the derivation of valuable information from various data sources. Despite the popularity of Machine Learning and Big data, portfolio management is still highly relies on linear models and the Markowitz framework known as Modern Portfolio Theory (MPT). Accurate prediction of market prices and evaluation of past asset probability distributions are the main concepts of MPT.

Nowadays, Signal Processing and Control Theory have been a lynchpin of Financial Engineering. More recently, discoveries in sequential decision making, mainly through the concept of Reinforcement Learning, developed the instrumental for the multistage stochastic optimization, a concept that can be effectively used in Portfolio Management

1. Introduction

Accurate allocation of portfolio assets can lead to lucrative results, so no wonder why investors are turning towards Machine Learning algorithms. In this paper, I want to look on the Portfolio management from the side of Reinforcement Learning.

Traditionally, portfolio management techniques are viewed as classic model- based algorithms, where you need to predict the future and then make decision. However, its more natural to consider this problem as game, where the players needs to make some decisions based on the state of the game. Using this approach we can consider the Portfolio Management as sequential decision making process with some environment and agents operating inside it. We will develop this idea through the next parts of this paper.

Of course, to make our research self-contained we will evaluate Reinforcement Learning approach along with other classical algorithms and strategies. We will consider Naïve Portfolio Management Strategies, Portfolio optimization methods like: Follow the winner strategy, Maximum Diversification and etc. We will evaluate all this methods on the real stock data and will develop a few portfolio types that will provide us with a valuable findings.

1.1 Problem Statement

The aim of this report is to investigate the applicability of Reinforcement Learning to the Portfolio Management and to compare it with more traditional algorithms. We will train our agent on a real life data and it will be able to obtain stock weights and optimally allocate assets in portfolio given the frame of the last stock data.

2. Related Work

The usage of neural networks for management stock portfolios is not a novel concept, although the application of Reinforcement Learning to this sphere is not much developed With the availability of different market data, it's natural to employ deep learning (DL) model which can exploit the potential laws of market in PM. Prior arts (Heaton, Polson, and Witte 2017; Schumaker 2012; Nguyen, Shirai, and Velcin 2015) in training a neural network (NN) model for market behavior prediction have shown their effectiveness in asset price prediction and asset allocation. However, DL models that do not interact with the market and environment - has a natural disadvantage in decision making problem like PM.

Reinforcement learning algorithms have been proved effective in decision making problems in recent years and deep reinforcement learning (DRL) (Chen 2019). For instance, (Almahdi and Yang 2017) proposed a recurrent reinforcement learning (RRL) method, with a coherent risk adjusted performance objective function named the Calmar ratio, to obtain both buy and sell signals and asset allocation weights. (Jiang, Xu, and Liang 2017) use the model- free Deep Deterministic Policy Gradient (DDPG) (Lillicrap 2015) to dynamically optimize cryptocurrency portfolios.

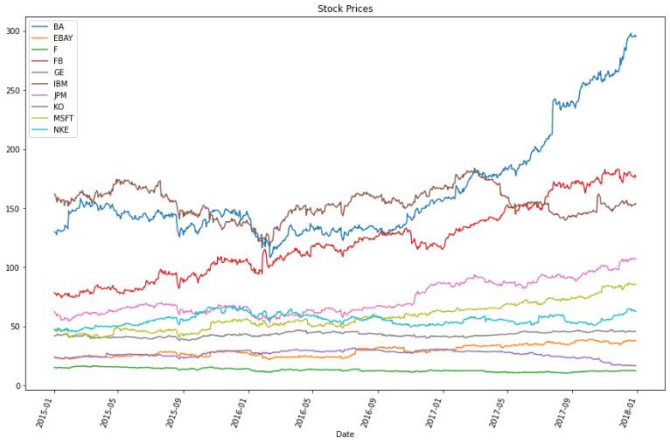
Similarly, (Liang 2018) optimize asset portfolios by using the DDPG as well as the Proximal Policy Optimization (PPO) (Schulman et al. 2017). (Buehler 2019) presents a DRL framework to hedge a portfolio of derivatives under transaction costs, where the framework does not depend on specific market dynamics. However, they mainly tackle the PM problem by directly utilizing the direct observation of historical prices for RL training, which may largely overlook data noise and overestimate the model's learning capability.

3. Dataset and environment

3.1 Data

For our research we are using the historical stock prices gathered from Yahoo Finance API. We have constructed the 2 types of portfolios for methods evaluation.
For the first portfolio (Portfolio 1) we have the stock prices of the following 10 companies: Boeing, eBay, Ford, Facebook, General Electric, IBM, JP Morgan, Coca Cola, Microsoft, and Nike. This portfolio contains companies from different industries. We picked companies in this way, to have a well-balanced, diversified portfolio, close to real one.
The second portfolio (Portfolio 2) contains stocks of the technology companies like: Google Facebook, IBM and etc. We constructed this portfolio to obtain how our methods will manage non -diversified type of portfolio

We have loaded the stock data for a period of 3 years and split it to train/test parts. The period of 2 years (from 2015/01/01 to 2017/01/01) – is used for training and period of 1 year (from 2017/01/01 to 2018/01/01) – for testing.



1.Historical stock prices. Portfolio 1.

3. Historical stock prices. Portfolio 2

Open	High	Low	Close	Adj Close	Volume
78.58000183105469	78.93000030517578	77.69999694824219	78.44999694824219	78.44999694824219	18177500.0
77.98000335693336	79.25	76.86000061035156	77.19000244140625	77.19000244140625	26452200.0
77.23000335693336	77.58999633789062	75.36000061035156	76.15000152587891	76.15000152587891	27399300.0
76.76000213623047	77.36000061035156	75.81999969482422	76.15000152587891	76.15000152587891	22045300.0
76.73999786376953	78.23000335693336	76.08000183105469	78.18000030517578	78.18000030517578	23961000.0

2. Example of the stock data

3.2 Environment

In our work we represent the portfolio management problem, as state-based in order to use the reinforcement learning algorithms. We give our agent a budget of 1 million dollars to construct the portfolio. Also, we are setting the rebalance period, let's say n days. So, on each of this time-stamps, that occur every n days agent should make some action – construct new updated optimal weights for the portfolio to maximize its objective. The action is represented as a vector with a portfolio weights.
Another, additional concept that we added to our environment is the opportunity to short the assets. So the agent could hedge against the downside risk of a long position.

4. Methods

In this paragraph, we will review the main methods, we put on evaluation. For our research, we picked 3 types of methods that can be used for portfolio management: Naïve Portfolio Management Strategies, Portfolio optimization methods, Deep Q-learning (Reinforcement learning).

4.1 Naïve Portfolio Management Strategies

This class of methods is used by portfolio managers for optimal allocation of portfolio assets. Each of this strategies is represented by a basic set of rules followed by the agent.

4.1.1 Follow the Winner

Follow the Winner approach assumes that the outperforming assets will remain profitable. So the behavior of this method is characterized by transferring portfolio weights from the under-performing assets (experts) to the outperforming ones.

4.1.2 Follow the Loser

Follow the Loser approach assumes that the under-performing assets will revert and outperform others in near future. Thus, their common behavior is to move portfolio weights from the outperforming assets to the under-performing assets.

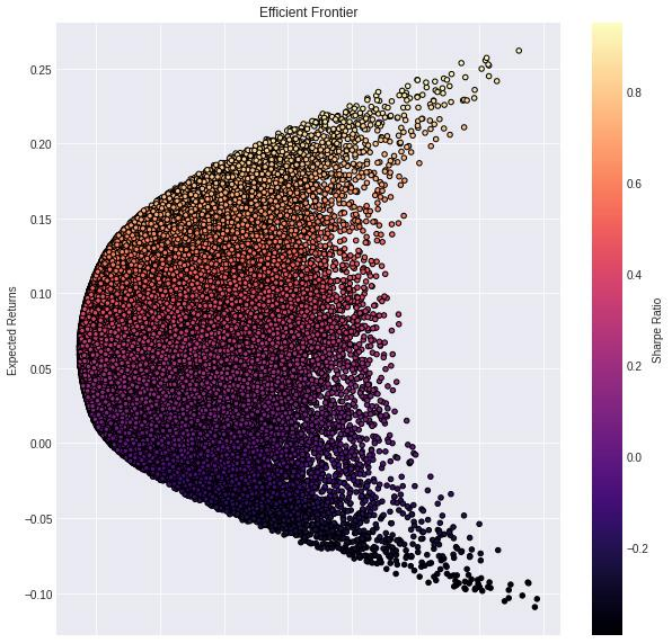
4.1.3 Uniform Constant Rebalanced Portfolios

A constant rebalanced portfolio (CRP) is an investment strategy which keeps the same distribution of wealth among a set of stocks from day to day. The UCRP approach suggests that the wealth be equally distributed between the chosen assets in a portfolio without making any kind of changes

throughout the trading period. This helps avoid the transaction costs incurred by the trading agency.

4.2 Portfolio Optimization Methods

Modern portfolio theory was introduced in 1952 by Harry Markowitz. It relies on the idea that investor wants to maximize the portfolio returns contingent on any given amount of risk. This relationship is represented by a curve known as efficient frontier. All of the portfolios on this curve are well diversified. The portfolio optimization problem is specified as a constrained utility-maximization problem, There are a lot of optimization criteria that can be used for this purpose.



3. Efficient Frontier obtained by Monte Carlo Simulation

4.2.1 Minimum Variance

If all investments have the same expected return independent of risk, investors seeking maximum returns for minimum risk should concentrate exclusively on minimizing risk. This is the explicit objective of the minimum variance portfolio:

$$w^{MV} = arg\ min\, w^T \cdot \Sigma \cdot w$$

where Σ is the covariance matrix.

4.2.2 Maximum Diversification

Consistent with the view that returns are directly proportional to volatility, the Maximum Diversification optimization substitutes asset volatilities for returns in a maximum Sharpe ratio optimization, taking the following form.

$$w^{MD} = arg\ max\ \frac{w \times \sigma}{\sqrt{w^T \cdot \Sigma \cdot w}}$$

where σ and Σ reference a vector of volatilities, and the covariance matrix, respectively.

4.2.3 Maximum Decorrelation

Maximum Decorrelation described by (Christoffersen et al. 2010) is closely related to Minimum Variance and Maximum Diversification, but applies to the case where an investor believes all assets have similar returns and volatility, but heterogeneous correlations. It is a Minimum Variance optimization that is performed on the correlation matrix rather than the covariance matrix. W Interestingly, when the weights derived from the Maximum Decorrelation optimization are divided through by their respective volatilities and re-standardized so they sum to 1, we retrieve the Maximum Diversification weights. Thus, the portfolio weights that maximize decorrelation will also maximize the Diversification Ratio when all assets have equal volatility and maximize the Sharpe ratio when all assets have equal risks and returns. The Maximum Decorrelation portfolio is found by solving for:

$$w^{M\,Dec} = arg\ min\, w^T \cdot \mathrm{A} \cdot w$$

where A is the correlation matrix.

4.2.4 Sharpe Ratio

Sharpe ratio is a measure of the performance of an investment's returns given its risk:

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$$

Where: $R_p = return\ of\ portfolio$, $R_f = risk\ free\ rate$, $\sigma_p = standard\ deviation\ of\ the\ portfolio's\ excess\ return$

4.3 Reinforcement Learning

The main concept of Reinforcement Learning consists of 2 things : Environment and Agent. The Agent interacts with the environment (everything beyond agent). And as a result, the agent changes makes some changes to the environment (changes state) and gets some feedback from it to plan future interaction with it. There are four main subelements of a reinforcement learning system that we can define: a policy, a reward signal, a value function, and, optionally, a model of the environment.

Policy

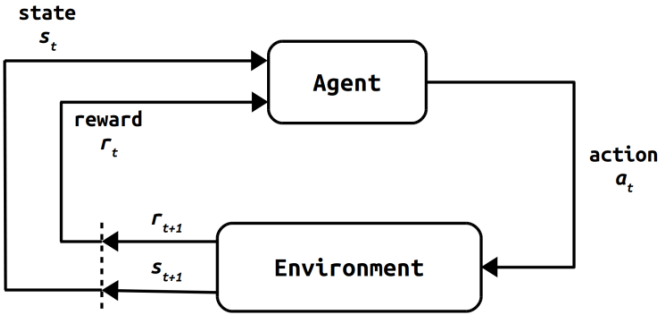
A policy defines the learning agent's way of behaving at a given time. Roughly speaking, a policy is a mapping from perceived states of the environment to actions to be taken when in those states

Reward

A reward signal defines the goal of a reinforcement learning problem. On each time step, the environment sends to the reinforcement learning agent a single number called the reward. The agent's sole objective is to maximize the total reward it receives over the long run.

Value function
Whereas the reward signal indicates what is good in an immediate sense, a value function specifies what is good in the long run. Roughly speaking, the value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state.

Markov decision process (MDP)
All this core concepts are forming a model named Markov decision process or MDP. MDPs are meant to be a straightforward framing of the problem of learning from interaction to achieve a goal. The learner and decision maker is called the agent. The thing it interacts with, comprising everything outside the agent, is called the environment.



Considering our environment a finite MDP, we can claim that Reinforcement Learning is applicable to our problem
Now lets discuss how our algorithm is working.
Agent observes the state (prices of stocks), given this information it can use policy function to find the best action or to take random action. (This is called observation – exploitation problem), we will define special hyperparameter *epsilon* that will decide the ratio of picking this methods. When we exploit the system we just following the best given answer to the state that we have. When we take random actions we are learning new ways of interaction with environment and evaluating them using Deep Learning. Following this process we can explore and evaluate the actions to find the best policy for the given problem

5. Results and discussion

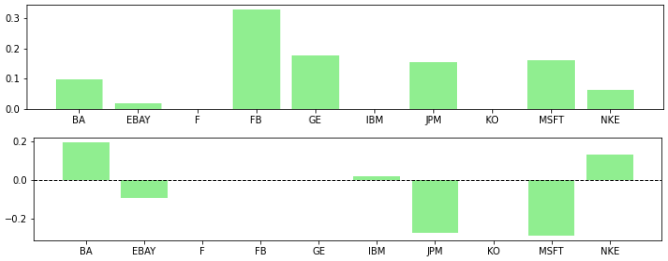
As we stated previously, we trained our Reinforcement Learning on the period of 2 years of stock prices. The other algorithm covered here do not require training. Our main benchmark of model performance was the cumulated returns on the test data. So lets review our results.

5. 1 Portfolio 1 Results

As we can see on the first figure, we got the best results using Follow the winner (Maximized returns) strategy, Maximum Sharpe and Reinforcement Learning. The reason for that is diversified stocks of big companies in this portfolio. Usually, this stocks has a long-term trends, so the Follow the Winner Strategy performs well here. Due to the same reasons Maximum Sharpe strategy also shows good result.

1. Cumulative returns. Portfolio 1. Without shortage

2. Cumulative returns. Portfolio 1. With shortage



3. Example of Portfolio 1 weights generated with Reinforcement Learning in a single time-stamp..

	UCRP	Min Variance	Max Returns	Max Sharpe	Max Decorrelation	Reinforcement Learning
Mean returns	0.0005	0.0004	0.0019	0.001	-0.0001	0.0012
Volatility	0.0049	0.0032	0.0135	0.0043	0.0031	0.0068
Sharperation	1.5757	1.9469	2.1867	3.9018	- 0.3431	2.812
Alpha	0.0	0.0003	0.0018	0.0009	- 0.0002	0.0012
Beta	1.0	0.265	0.0534	0.2144	0.3496	0.3784

4. Cumulative returns. Portfolio 2. Without shortage

5. Cumulative returns. Portfolio 2. With shortage

6. Example of Portfolio 2 weights in a single time-stamp.

	UCRP	Min Variance	Max Returns	Max Sharpe	Max Decorrelation	Reinforcement Learning
Mean returns	0.001	- 0.0001	0.0005	-0.0001	0.0005	0.00012
Volatility	0.0076	0.0046	0.0104	0.0052	0.004	0.0074
Sharperation	2.0404	- 0.4162	0.8024	-0.3443	1.8831	2.341
Alpha	0.0	- 0.0004	- 0.0001	-0.0003	0.0004	0.0003
Beta	1.0	0.294	0.6772	0.1985	0.0435	0.0378

5. 2 Portfolio 2 Results

Reinforcement learning algorithm performed well on the Portfolio 2 data.
The stocks used in this portfolio are all dependent on the technology industry, and therefore are non-diversified. That's why more traditional algorithms failed on this data and Reinforcement Learning agent adopted well.

5.3 Future work

In my opinion our work is a successful proof of concept, that have opened a few great directions for the future research.

In this research, we haven't considered the trading fee – one of the highly important things in real life trading. Adding such fee, will penalize the frequent weight transferring of portfolio and respectively will make agent behavior closer to real-life environment.
Another thing that we want to upgrade is the number of traded stocks in portfolio, it will be interesting to extend number of stocks in the portfolio. In addition, we can add derivatives to our portfolio to make our action space wider and create even more optimized portfolios

Moreover, in this paper we have focused only on the model free methods of Reinforcement Learning. So , the next research topic could be the usage of model based algorithms for portfolio management .

6. Conclusion

In this project we successfully utilized Reinforcement Learning to Manage portfolio of ten stocks. Exploring different portfolio management strategies and evaluating them on different data, provided us with a great knowledge of practical portfolio management. If given more time, we would like to make deeper research of different techniques and increasing complexity of our model to optimize the performance.

GitHub

https://github.com/OleksandrVovchuk/Reinforcement-Learning-for-Portfolio-Managment

References

[1] Modern Invest Theory, Robert A. Haugen, Prentice Hall, 1986

[2] Deep direct rein- forcement learning for financial signal representation and trading, Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai, IEEE transactions on neural networks and learning systems,

[3] Silver, David. Deep Reinforcement Learning. Accessed: 2016-12-14.

[4] Portfolio optimization: Simple versus optimal methods

[5] Reinforcement Learning for Portfolio Management Angelos Filos

[6] J. Moody and M. Saffell. Reinforcement Learning for Trading Systems

[7] A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem, Zhengyao Jiang, Dixing Xu, Jinjun Liang

[8] Deep learning for finance: deep portfolios, J. B. Heaton, N. G. Polson, and Jan Hendrik Witte, Applied Stochastic Models in Business and Industry, 2016, ISSN 1526-4025. doi: 10.1002/ASMB.2209

[9] Forecasting S&P 500 index using artificial neural networks and design of experiments, Seyed Taghi Akhavan Niaki and Saeid Hoseinzade, Journal of Industrial Engineering International,

[10] B. Lau. Using Keras and Deep Deterministic Policy Gradient to play TORCS, Oct 2106. Accessed: 2016-12-14.

[11] A. Fernndez and S. Gmez. Portfolio selection using neural networks. Computers and Operations Research, 34(4):1177 – 1191, 2007.