Deep Reinforcement Learning for Portfolio Optimisation

Dimosthenis Karafylias
Supervisor: Kourosh Marjani Rasmussen

July 28, 2022

Table of Contents

Introduction

Research Objectives

Objective:

- 🏁 Minimum Spanning Tree to reduce a universe of assets to an uncorrelated subset
- 🏁 Applicability of Deep Reinforcement Learning in portfolio optimisation

📝 Portfolio Optimisation

Strategic allocation of wealth into a fixed number of financial instruments in order to achieve an objective, such as maximising returns or minimising risk

Portfolio Optimisation

Financial Time Series I

A portfolio of m assets $w_t = [w_{1,t}, \dots w_{m,t}]$, at time t, should satisfy:

$$\sum_{i=1}^{m} w_{i,t} = 1.$$

Gross or simple returns of a portfolio p that contains m assets at time t:

$$R_{p,t} = \sum_{i=1}^{m} w_i R_{i,t} = w_1 R_{1,t} + w_2 R_{2,t} + \dots$$

For the log returns it holds:

$$r_{p,t} = \log\left[1 + \sum_{i=1}^{m} w_{i,t} R_{i,t}\right] = \log\left[1 + w_{1,t} R_{1,t} + w_{2,t} R_{2,t} + \dots\right]$$
$$= \log\left[\sum_{i=1}^{m} \frac{P_{i,t}}{P_{i,t-1}}\right] = \log\left[\sum_{i=1}^{m} P_{i,t}\right] - \log\left[\sum_{i=1}^{m} P_{i,t-1}\right].$$

Financial Time Series II

Examples of returns' distributions:

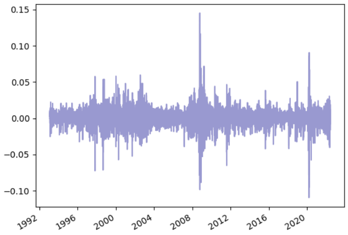


Figure: SPY ETF Returns

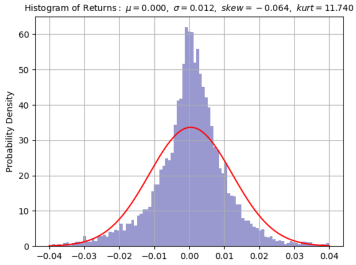


Figure: Distribution of SPY ETF Returns

Financial Time Series III

Covariance and correlation of assets' returns:

Figure: Covariance matrix of returns

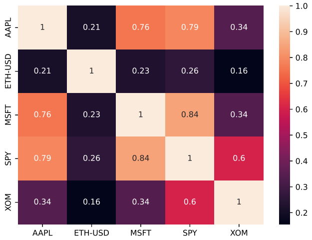


Figure: Correlation matrix of returns

Portfolio Objectives

	PnL	$\mu_{returns}$	$\sigma_{returns}$	$Skew$	$Kurt$	SR
[MSFT, SPY]	1.001	0.0001	0.014	−0.406	0.646	0.118
[MSFT, SPY, XOM]	1.199	0.0008	0.012	−0.603	1.503	1.073

Table: Statistics for two different portfolio constructions

Investment Advice

- **IA1:** Large standard deviation ⇒ returns fluctuate a lot [Rice, 2006]
- **IA2:** Risk-averse investors ⇒ positively skewed returns [Harvey et al., 2010]
- **IA3:** Excess kurtosis ⇒ fat-tail risk [Naqvi et al., 2017, Khan et al., 2020]
- **IA4:** Prefer uncorrelated securities to assets that belong in the same sector[1] [Rice, 2006]
- **IA5:** Choose the alternative with the higher Sharpe Ratio [Dowd, 2000]

[1] assuming this would imply a strong correlation

Modern Portfolio Theory & CAPM

[Markowitz, 1952]

$$\underset{w}{\text{Minimize}} \quad \frac{1}{2} w^T \sum_{i,j=1}^{m} w_i w_j \sigma_{i,j}$$

$$\text{subject to} \quad \sum_{i}^{m} w_i \bar{r}_i = \bar{r}_{target}$$

$$\text{and} \quad \sum_{i=1}^{m} w_i = 1$$

Capital Market Line

CML: $\bar{r} = r_f + \frac{\bar{r}_M - r_f}{\sigma_M} \sigma$

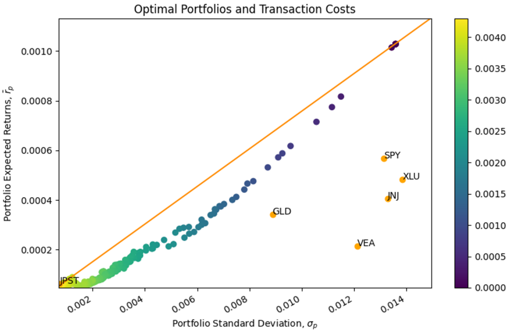


Figure: Efficient Frontier without short-selling

Asset Decorrelation

Minimum Spanning Tree I

Edge Distance

Clusters of assets with strong positive correlations

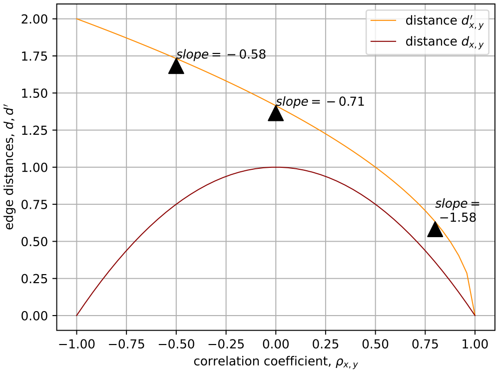


Figure: $d'(x, y) = \sqrt{2(1 - \rho_{x,y})}$ shown in orange; and $d(x, y) = 1 - \rho^2_{x,y}$ shown in dark red.

Minimum Spanning Tree II

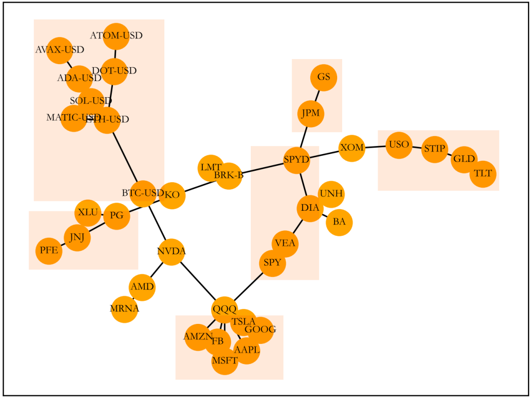


Figure: Universe of assets in a MST

Minimum Spanning Tree III

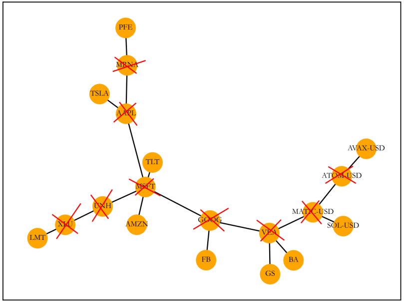


Figure: Node reduction

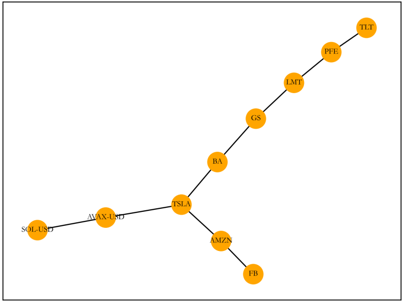


Figure: Final subset

Reinforcement Learning

Why Reinforcement Learning ?

- ✏️ Learns from scratch, by trial and error, without prior financial knowledge or a market model
- ✏️ Does not have any inherent assumptions about the market
- ✏️ Discounted rewards that give importance to future returns
- ✏️ Design of objectives/reward functions (e.g. profits or risk-adjusted)

Agent - Environment interaction in MDPs

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots .$$

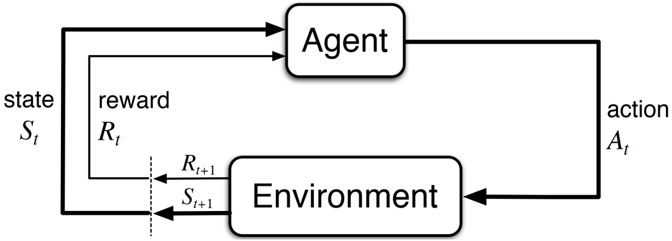


Figure: Agent - Environment setting in a MDP, source: [Sutton and Barto, 2018]

Key Elements

- **policy π:** Mapping from states to actions, stimulus-response rule, determines the agent's behaviour
- **state-value function V_π:** Value of a state (reward expected to be accumulated in the future), how good is to be in a given state

$$u_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S}$$

- **action-value function Q_π:** Value of an action (expected return of action α), how good is to perform a particular action in a given state

$$q_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right], \text{ for all } s \in \mathcal{S}$$

Actor - Critic

Architecture

Actor $\Rightarrow$ policy π_θ
Critic $\Rightarrow$ action-value function Q_π

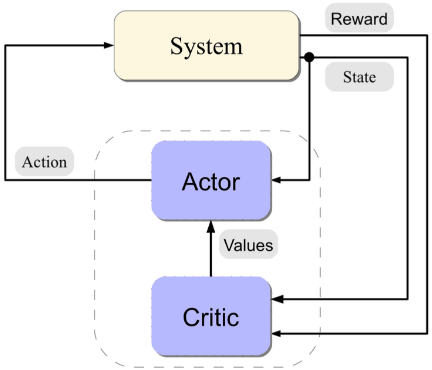


Figure: The Actor-Critic framework[2], source: [Szepesvari, 2010]

[2] System or Environment

Twin Delayed Deep Deterministic policy gradient algorithm (TD3)

1 Clipped Double Q-Learning

$$y_1 = r + \gamma Q_{\theta_2'}\left(s', \pi_{\phi_1}(s')\right)$$

$$y_2 = r + \gamma Q_{\theta_1'}\left(s', \pi_{\phi_2}(s')\right)$$

$$\Downarrow$$

$$y = r + \gamma \min_{i=1,2} Q_{\theta_i'}\left(s', \pi_{\phi_1}(s')\right)$$

📌 Taking the minimum between two target networks to limit overestimation bias

2 Delayed policy updates

A deterministic target policy is learned by:

$$\max_{\phi} \mathbb{E}_{s\sim\mathcal{B}}\left[Q_{\theta_1}\left(s, \mu_{\phi_{target}}(s)\right)\right]$$

📌 Policy updates are delayed for higher quality policies

3 Target policy smoothing

To avoid overfitting, the target is updated as:

$$y = r + \gamma Q_{\theta'}\left(s', \pi_{\phi'}(s') + \epsilon\right)$$

$$\epsilon \sim \text{ clip } \left(\mathcal{N}(0,\sigma), -c, c\right)$$

📌 Policy smoothing averts overfitting and leads to actions more robust to noise

Experiments

Trading Agents I

Algorithm 1: General setup of TD3 as a trading agent

input : Normalised prices of assets,
Objective function $\mathcal{J}(\phi) = \mathbb{E}_{\pi_\phi}\left[\sum_{t=0}^{T} \gamma^t r(s,a)\right]$

output : Parameters for the actor network ϕ and for the critic networks θ_1 and θ_2

for $t = 1, \ldots, T$ time steps in each episode **do**

Observe normalised prices of assets

Select portfolio weights / actions $\alpha \sim \pi_\phi(s)$, and observe reward r and new state s'

Store transition tuple (s, a, r, s') in replay buffer $\mathcal{B}$

Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$

Training:

Select action: $\tilde{\alpha} \leftarrow \pi_{\phi'}(s') + \epsilon, \quad \epsilon \sim \text{ clip } (\mathcal{N}(0, \tilde{\sigma}), -c, c)$

Target network: $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tilde{\alpha})$

Update critics $\theta_i \leftarrow \arg\min_{\theta_i} N^{-1} \sum \left(y - Q_{\theta_i}(s,a)\right)^2$

if $t \mod d$ **then**

Update ϕ by the deterministic policy gradient:
`% [Silver et al., 2014]`

$\nabla_\phi J(\phi) = N^{-1} \sum \nabla_\alpha Q_{\theta_1}(s,a) \,|_{\alpha=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$

Update target networks: θ_i' and ϕ' `% Adaptive gradient optimiser ADAM [Kingma and Ba, 2014]`

end

end

Trading Agents II

Normalised price time series - Training input:

$$P_t = \frac{p_t - p_{min}}{p_{max} - p_{min}}, \quad P_t \in [0, 1].$$

The final portfolio wealth is derived as:

$$\mathbb{W} = \prod_{t=1}^{T} \Big[(1 - c_t) r_t \cdot w_{t-1} + 1 \Big].$$

Reward functions:

1 Simple returns

$$\rho_t = \sum_{i=1}^{m} (1 - c_t)\, r_{i,t} \cdot w_{i,t-1},$$

$$i = 1, \cdots m \text{ assets}$$

2 Log returns

$$R_t = \log\left(\sum_{i=1}^{m} w_i p_{i,t}\right) - \log\left(\sum_{i=1}^{m} w_i p_{i,t-1}\right)$$

3 Differential Sharpe Ratio

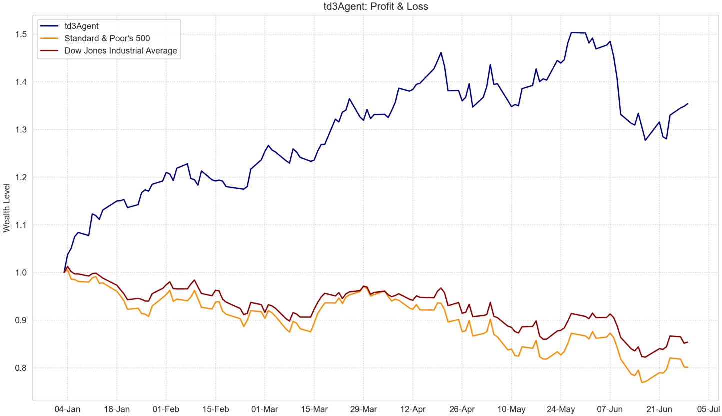
[Moody and Saffell, 1998]

$$\frac{\partial D_\eta(t)}{\partial R_t} = \frac{B_{t-1} - A_{t-1} R_t}{\left(B_{t-1} - A_{t-1}^2\right)^{\frac{3}{2}}}$$

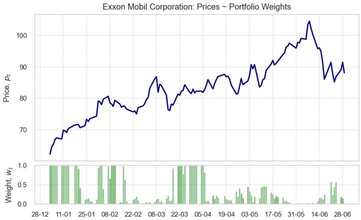
Results I

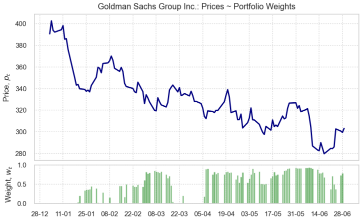
	Start date	End date
Training	02/01/15	30/12/20
Back-test	03/01/22	28/06/22

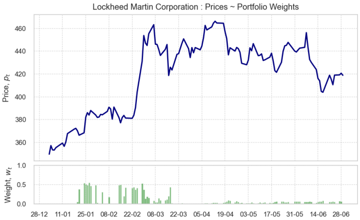
	TD3	S&P500	DJIA
PnL	1.35	0.80	0.85
Std. deviation	0.02	0.02	0.01
Skewness	-0.21	-0.21	-0.21
Kurtosis	0.43	-0.34	0.03
Sharpe ratio	1.69	-1.17	-1.07
MDD	0.33	0.24	0.19
CVaR −99%	-0.04	-0.04	-0.03



Results II


Exxon Mobil Corporation: Prices ~ Portfolio Weights
Price, p_t
100
90
80
70
Weight, w_t
1.0
0.5
0.0
28-12 11-01 25-01 08-02 22-02 08-03 22-03 05-04 19-04 03-05 17-05 31-05 14-06 28-06


Goldman Sachs Group Inc.: Prices ~ Portfolio Weights
Price, p_t
400
380
360
340
320
300
280
Weight, w_t
1.0
0.5
0.0
28-12 11-01 25-01 08-02 22-02 08-03 22-03 05-04 19-04 03-05 17-05 31-05 14-06 28-06

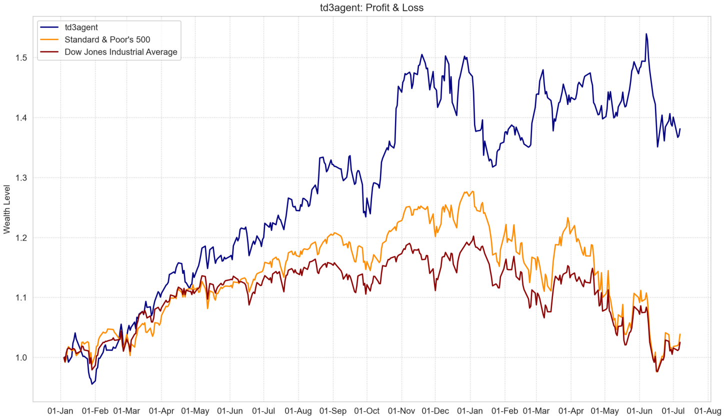

Lockheed Martin Corporation : Prices ~ Portfolio Weights
Price, p_t
460
440
420
400
380
360
Weight, w_t
1.0
0.5
0.0
28-12 11-01 25-01 08-02 22-02 08-03 22-03 05-04 19-04 03-05 17-05 31-05 14-06 28-06

Experiments

Results III

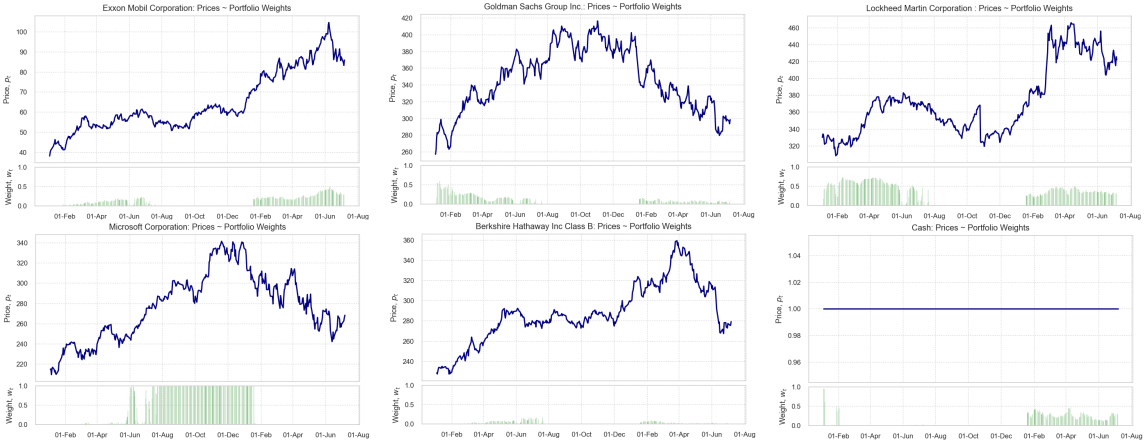
	Start date	End date
Training	02/01/15	30/12/20
Back-test	04/01/21	07/07/22

	TD3	S&P500	DJIA
PnL	1.38	1.04	1.03
Std. deviation	0.01	0.01	0.01
Skewness	-0.27	-0.49	-0.42
Kurtosis	1.23	1.16	1.02
Sharpe ratio	1.53	0.28	0.23
MDD	0.38	0.24	0.19
CVaR −99%	-0.04	-0.04	-0.03



Experiments

Results IV



Experiments

Reward functions comparison

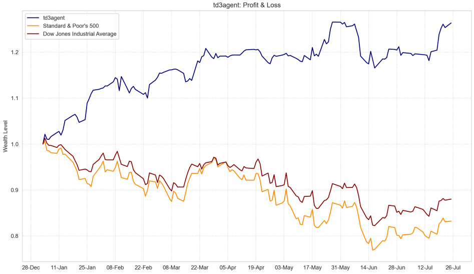


Figure: Differential Sharpe Ratio

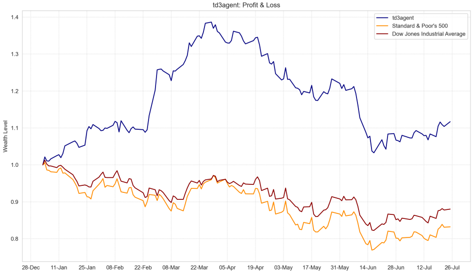


Figure: Log returns

Conclusions

Summary of Results

1. Minimum Spanning Tree used as an aid to topologically reveal clusters of assets with similar returns' patterns
2. Node reduction to obtain a smaller less correlated subset
3. Deep Reinforcement Learning finds patterns in the returns' series
4. Differential Sharpe Ratio, as a reward function, leads to more stable portfolio performances than log returns

Addressing Limitations

- MST selects for the least correlated asset's; albeit not taking their performance into account
- Interpretability; neural networks often described as "black-box" architectures
- Reproducibility issues

Future Work

- Optimise for simplicity in deep architectures to avoid overfitting and improve generalisation ability
- Reward signals that integrate financial knowledge; inclusion of technical indicators, sentiment analysis and fundamentals [Sato, 2019, Hambly et al., 2022, Filos, 2019]
- Learning through imitation; observing experts
- Offline reinforcement learning where data collection is expensive; the agent does not interact with the environment; safer for testing purposes [Fujimoto and Gu, 2021, Levine et al., 2020]

Questions ?

📧 e-mail: s202632@student.dtu.dk
dimostheniskarafylias@gmail.com

Bibliography

References I

Dowd, K. (2000).

Adjusting for risk: An improved sharpe ratio.

International review of economics & finance, 9(3):209–222.

Filos, A. (2019).

Reinforcement learning for portfolio management.

Fujimoto, S. and Gu, S. S. (2021).

A minimalist approach to offline reinforcement learning.

CoRR, abs/2106.06860.

Hambly, B. M., Xu, R., and Yang, H. (2022).

Recent advances in reinforcement learning in finance.

References II

Harvey, C. R., Liechty, J. C., Liechty, M. W., and Müller, P. (2010).
Portfolio selection with higher moments.
Quantitative Finance, 10(5):469–485.

Khan, K. I., Naqvi, S. M. W. A., Ghafoor, M. M., and Akash, R. S. I. (2020).
Sustainable portfolio optimization with higher-order moments of risk.
Sustainability, 12(5).

Kingma, D. P. and Ba, J. (2014).
Adam: A method for stochastic optimization.

Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020).
Offline reinforcement learning: Tutorial, review, and perspectives on open problems.
CoRR, abs/2005.01643.

References III

Markowitz, H. M. (1952).

Portfolio selection.

The Journal of Finance, 7(1):77–91.

Moody, J. and Saffell, M. (1998).

Reinforcement learning for trading.

In Kearns, M., Solla, S., and Cohn, D., editors, Advances in Neural Information Processing Systems, volume 11. MIT Press.

Naqvi, B., Mirza, N., Naqvi, W. A., and Rizvi, S. K. A. (2017).

Portfolio optimisation with higher moments of risk at the pakistan stock exchange.

Economic Research-Ekonomska Istraživanja, 30(1):1594–1610.

References IV

Rice, J. A. (2006).

Mathematical Statistics and Data Analysis.

Belmont, CA: Duxbury Press., third edition.

Sato, Y. (2019).

Model-free reinforcement learning for financial portfolios: A brief survey.

ArXiv, abs/1904.04973.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014).

Deterministic policy gradient algorithms.

In Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14, page I–387–I–395. JMLR.org.

References V

Sutton, R. S. and Barto, A. G. (2018).
Reinforcement Learning: An Introduction.
Cambridge, MA : The MIT Press, second edition.

Szepesvari, C. (2010).
Algorithms for Reinforcement Learning.
Morgan and Claypool Publishers.