Naive implementation of Reinforcement learning in Portfolio Management and its interpretation

Laurens Weijs
366515lw@eur.nl
https://sites.google.com/view/laurensweijs

July 31, 2017

Research proposal Master Quantitative Finance

Professor: Rutger-Jan Lange

Abstract

All machine learning methods commonly in use today are viewed as black boxes, the goal of this paper is to make one of these methods transparent in the context of Portfolio management. I will interpret the strategies implied by Reinforcement learning problem solving and relate them to academic portfolio advice and professional portfolio advice.
In reinforcement learning, target outputs are not provided. Rather, the system takes actions (makes trades), receives feedback on its performance (the received utility or wealth) and then adjusts its internal parameters to increase its future rewards. With this approach, an ultimate measure of trading performance such as profit, utility or risk-adjusted return is optimized directly. Therefore, this method is able to solve Dynamic Programming problems in comparison to Supervised and Unsupervised learning methods and fit to the goal of solving the intertemporal portfolio problem.

Erasmus
School of
Economics

1 Problem Description

1.1 Introduction

On long-term strategic asset allocation, relatively little literature exists on the out-of-sample performance for dynamic strategies. Besides the extensive research of Diris et al. [2015], this paper will extend the current literature with another algorithm to solve the dynamic asset allocation problem. The algorithm has its roots in Reinforcement learning(RL) and is called Q-learning. The advantage over classical long-term portfolio strategies is that no econometric model is needed for the returns with the corresponding estimation error of the estimated model. A pain in the eye for classical portfolio management is that the model on the returns is generally estimated by a Vector Autoregressive model. In an environment with long periods and many assets, the estimation of the model suffers from the curse of dimensionality and estimation error. In the literature of the computer science discipline, a few papers have been written about RL in the domain of finance. However, these papers lack in terms of economic interpretation, the naive implementation of out-of-sample prediction and the appropriate objective function. However, the paper Moody et al. [1998] is a good starting point. Especially because transaction costs can be incorporated in an easy matter, this is not the case in the value iteration solution with classical portfolio management like used in Diris et al. [2015] due to the backward inductive nature of the solution.

Classical Portfolio choice knows three main steps to solve,

- Formulate an interesting, practical problem
- Obtain a model for the objects of interest
- Solve the optimization problem

The problem this research considers is the intertemporal portfolio problem, the weights of the portfolios needs to be estimated optimally over time and you are only able to invest in three type of stocks. Secondly the problem states that you need to be able to construct an estimation of the asset returns by an econometric model. At last, you can calculate the optimal portfolio weights with the model of the objects of interest accordingly. However, the special method of Q-learning of RL doesn't need to estimate an econometric model in advance. Q-learning learns the dynamics of the objects of interest in an iterative manner and is, therefore, able to find approximate solutions to stochastic dynamic programming problems.

This research also adds value in terms of interpretation of the RL algorithm. In classical portfolio management, one tries to bring the academic and professional advice together in one model which can be achieved by state of the art econometric models and economic theory. These advice are shown in Table 1.

Academic portfolio advice	**Professional portfolio advice**
Combine the market(or tangency portfolio) with the risk-free rate.	Long-term investors should invest more in stocks than short-term investors.
The relative proportion of risky assets is the same.	Conservative investors should hold more bonds, aggressive investors should hold more stocks.
No specific advice on market timing.	Investors should actively time the market, invest more when the economy is in good shape and less when otherwise.

Table 1 – Academic and professional portfolio advice related to each other.

The goal is, therefore, to relate the outcomes of the RL algorithm to the second column of Table 1.

1.2 Literature

Diris et al. [2015] perform an out-of-sample prediction for long-term strategic asset allocation and concludes that with a value iteration method would give same results as a repeated myopic strategy. This is mainly caused by the estimation error in the predictors, the hedge term needs a good predictability in step 2 to function well. Bayesian estimation of step 2, however, increases the performance of the long-term dynamic strategy. Adding transaction costs or limiting the step sizes of the weights can be seen as a shrinkage in the Q-learning solving method.
The long term strategy of Diris et al. [2015] is extended by transaction costs in Gârleanu and Pedersen [2013], here the transaction costs are introduced into the Bellman equation and a closed-form solution is presented. In this paper, I only consider constant transaction costs while Gârleanu and Pedersen [2013] also considers varying transaction costs. This paper states that the newly formed portfolio is the weighted value of the Markowitz portfolio and the aim portfolio, which is the optimal portfolio in the absence of trading costs with the trade-off between risk and return.
Moody et al. [1998] propose a framework for three Recurrent Reinforcement Learning algorithms: backpropagation through time (BPTT), dynamic backpropagation (DBP), and real-time recurrent learning (RTRT). The first one is an off-line/batch algorithm, while the latest two are on-line algorithms. Moody et al. [1998] implement different objective functions like terminal wealth, economic utility, Sharpe ratio and the differential Sharpe ratio. They show that the performance is dependent on the objective function and does outperform portfolio management optimization based on minimization of MSE. This research will start off with the basis case of RTRT, Q-learning without recurrence, and add economic interpretation to the outcomes.
Hens and Woehrmann [2007] implement the RTRT algorithm of Moody et al. [1998] on real world data with bonds and stocks of the United States, Great Britain, and Germany. This paper concludes that the investor actively times the market and outperforms other strategies. This paper, however, considers in-sample prediction, which is not interesting enough and only actually only confirms the equivalence of value iteration and the RTRT algorithm.
For the function approximation of the Q learning methods a Neural Network is constructed, for an introduction to Neural Networks and Deep Neural Networks DeMiguel et al. [1997] is a good start. The importance of the equally weighted portfolio strategy is an important benchmark due to its simplicity and performance, Svozil et al. [2009] show this. This benchmark together with a simple Constant Expectation Return model will be used as a comparison.

1.3 Research Questions

How can reinforcement learning improve on classical portfolio management or the naive portfolio diversification out-of-sample?

Diris et al. [2015] Show that the true data generating process is not known and is prone to misspecification and therefore is prone to parameter estimation errors. This leads to the fact that dynamic portfolio is the same as the repeated myopic portfolio, only looking one step at each time and is not able to beat the naive portfolio diversification of equal weights.

How can reinforcement learning methods be improved by imposing restrictions on the portfolio weights?

Because the RL method does not count on a prediction model of the returns one could not add parameter uncertainty in the prediction model in terms of a Bayesian posterior distribution. However, there can be bounds on the weights in the system, otherwise, the change of getting a high turnover is present. In the classical portfolio management improved by Bayesian statistics, this restriction in combination with economic restrictions does improve the predictive performance significantly.

2 Data

This research will be based on the monthly stock and bond market of the United States. Because this research considers three assets classes to choose from as an investor, I need to gather these three classes from various data sources. Please see the Table 2 for a description of the data and its source.

Stock class	**US Asset**	**source**
Equity	Weighted average of NYSE, NASDAQ, and AMEX	CRSP
Long-term nominal bonds	Nominal 5-year T-Note	FRED
Short-term nominally risk-free T-bills	Nominal 3-month T-Bill	FRED

Table 2 – Data used in this research combined with the data source.

The summary statistics stated in Table 3 shows typical market behavior, the safest asset(with the lowest volatility but also a low return is the Ex-post T-Bill rate r. The stocks, on the other hand, are more volatile but on average have a higher return.

	r	x_s	x_b
Avg.	0.0008	0.0057	0.0013
Std. dev.	0.0033	0.0432	0.0146
Min	-0.0108	-0.2305	-0.0687
Max	0.0193	0.1594	0.0951
AR(1)	0.4456	0.0907	0.1193

Table 3 – Summary statistics of the three assets taken into consideration, the ex-post real T-bill, Value-weighted stock returns, Excess-bond returns. The data set starts in february 1954 and ends in december 2016 and are notated in monthly returns.

In this research, three models are considered to simulate the asset returns from the Constant Expected Return(CER) [1] model, Vector Autoregressive(VAR) model, and a Bayesian Vector Autoregressive(BVAR) model. The models are shown respectively in Equations 1,2, and 3. Where y denotes the vector of asset returns $(r, x_s, x_b)'$.

$$y_t = \mu + \varepsilon_t, \text{ with } \varepsilon_t \sim N(0, \sigma^2) \tag{1}$$

$$y_t = \hat{A} + \hat{B} y_{t-1} + \varepsilon_t, \text{ with } \varepsilon_t \sim N(0, \sigma^2) \tag{2}$$

$$y_t = A + B y_{t-1} + \varepsilon_t, \tag{3}$$

$$\text{with } \varepsilon_t \sim N(0, \sigma_i^2),\ A, B \sim N(\hat{A} \text{ or } \hat{B}, \frac{\sigma_i^2}{T}), \text{ and } \sigma_i^2 \sim iGamma2(SSE, T-1)$$

Where for each simulation i the uncertainty parameter of Equation 3 has a different draw of the inverse Gamma distribution with as parameters the sum squared residual of the shrinkage model stated in Equation 2 and $T - 1$.

Distribution returns	Constant parameters	Time-varying parameters
Known parameters	1	2
Unknown parameters	3	4

Table 4 – Different assumptions involved in the asset allocation problem.

Based on these assets, simulations are made for four different scenarios based on the knowledge of the distribution of returns. See Table 4 for the different scenarios. In scenario 1 and 3 the CER and VAR model are simulated and in scenario 1 one can solve the optimal weights allocation analytically for the CER model and approximate analytically for the VAR model. In scenario 2 and 4 however,

[1] The CER model is introduced because it has a clear analytical solution, the myopic solution.

I simulate from the BVAR model in order to get time varying parameters. Scenario 2 can again be solved approximate analytically but this time for the known BVAR model. Classical portfolio management assumes that by estimating a model of the returns they have found the true Data Generating Process of the returns and thus operate in the domain of scenario 1 and 2. This research, however, doesn't assume it knows the model of the underlying assets and operates in scenario 3 and 4 therefore.

3 Methodology

3.1 Problem Statement

This research focuses on a world where an investor can choose between three assets: equity, long-term nominal T-Notes, and short-term nominal T-Bills. The problem stated for a long-term investor in portfolio management is a dynamic intertemporal weight optimization problem with uncertainty about the future states. The objective function of such an investor with intertemporal utility $U(\cdot)$ is given by:

$$\max_{w_t,\dots,w_{t+K-1}} \mathbb{E}_t[U(W_{t+K})] \quad \text{s.t.} \tag{4}$$

$$W_{s+1} = W_s(w_s' r_{s+1} + R_s^f) \quad ,s = t,\dots,t+K-1 \tag{5}$$

With the second equation being the budget restriction, including the restriction that the weights should count to one. Your wealth one period ahead can only change by the change in the assets times the weights in the assets. The other parameters in these equations are described as follows: K is the number of periods to optimize over in the future, W_s is the current wealth at time s, not to be confused with w_s which is the weight per asset class at time s, r_{s+1} is a vector of excess returns on the asset classes one period in the future, R_s^f is the current Risk-free asset at time s, and $U(\cdot)$ is the utility function, which in this research is the power utility function defined by $\frac{W_{t+K}^{1-\gamma}}{1-\gamma}$, with γ being the risk aversion of the investor.

This dynamic problem can be transformed into the following Bellman equation to show the recursive nature of this dynamic optimal strategy:

$$\begin{aligned} V_t(W_t,\theta) &= \max_{w_t,\dots,w_{t+K-1}} \mathbb{E}_t[U(W_{t+K})] && (6)\\ &= \max_{w_t} \mathbb{E}_t\left[\max_{w_t+1,\dots,w_{t+K-1}} \mathbb{E}_{t+1}[U(W_{t+K})]\right] && (7)\\ &= \max_{w_t} \mathbb{E}_t\left[V_{t+1}(W_t(w_t' r_{t+1} + R_t^f,\theta))\right] && (8)\\ \text{with terminal condition:} \quad & V_{t+K}(W_{t+K}) = U(W_{t+K}) && (9) \end{aligned}$$

With θ representing the parameters of the model from the assets classes stated in the different scenarios of Table 4. When the parameters are unknown it is therefore not possible to estimate the expectation of the right side immediately.

When you now substitute the utility function for the power utility and substitute W_{t+1} by the budget constraint you will get the following formula:

$$V_t(W_t,\theta) = \max_{w_t} \mathbb{E}_t\left[\frac{\left(W_t(w_t' r_{t+1} + R_t^f)\right)^{1-\gamma}}{1-\gamma} \max_{w_{t+1}\cdots w_{t+K-1}} \mathbb{E}_{t+1}\left[\left(\prod_{s=t+1}^{t+K-1}(w_s' r_{s+1} + R_s^f)\right)^{1-\gamma}\right]\right] \tag{10}$$

From this point, you can solve the Bellman equation, which is a necessary condition for optimality, in four different ways with differing assumptions about the distribution of returns see Table 4. I state here, that the classical way in Portfolio Management is the solution method performed by Diris et al. [2015] with the assumption that the distribution is known and covers Scenario 1 and 2 of this table. And the proposed way, which is raised by the Reinforcement Learning community, has no assumptions about the distribution of returns and cover scenario 3 and 4. All four cases will be simulated in this research and as an extension, the methods will be tested against the real data.

3.2 Classical portfolio management

In the classical way, I assume that one knows a model of the returns and are therefore is able to calculate the estimation of the wealth in the next period ($\mathbb{E}_{t+1}[U(W_{t+K})]$). This model is first estimated from the true sample of the historical returns of asset prices and the mathematical estimation is then retrieved from the simulation of this estimated model of returns. The industry workhorse is a VAR model, this can be shown mathematically as:

$$y_t = A + By_{t-1} + \varepsilon_t \tag{11}$$

A is a vector of intercepts, B is an $(n \times n)$ of the slope of the equation, and ε is a vector of idiosyncratic errors.

With this VAR model stated before, the probability distribution of the returns is known. According to Bayesian statistics, you can approximate the expectation of any distribution by the simulated sample averages.

$$\mathbb{E}[f(\theta)] \approx \frac{1}{N}\sum_{i=1}^{N} f(\theta_i) \tag{12}$$

This is exactly what I will do with the help of the MCMC algorithm, an extensive reversible Markov Chain (MC) is sampled from the estimated VAR model and is assumed to be a true representation of the real underlying dynamics of the asset returns. This Markov Process (MP) can be seen graphically in Figure 1, where each transition is chosen by random by the stated transition probability. Please note that in classical portfolio theory the focus is not on the Markov Chains, but in this research, the focus is on MC's due to the clear link with Reinforcement learning methods.

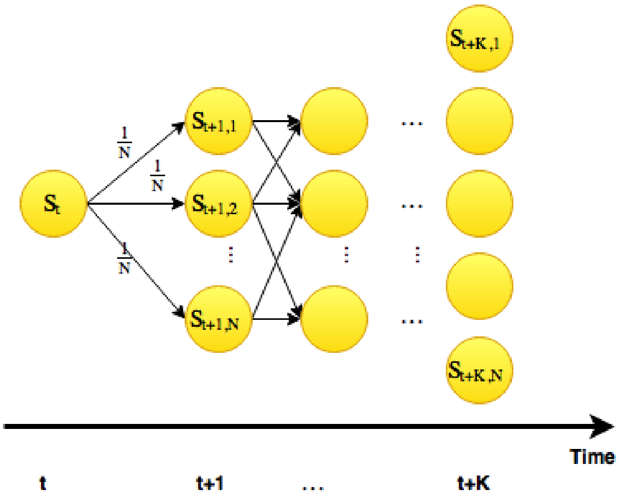

Figure 1 – Graphical representation of the Markov Process of the asset prices, transitions are given in transition probabilities and determined by nature.

Given that you can construct the Markov Process[2] of the asset returns, one can also construct a Markov Decision Process (MDP) in which you make the whole problem statement of the optimal portfolio weight a Markov Process. See Figure 2 for the graphical representation of this decision process. In each node, the value of the objective function is shown and an action should be executed at each time period, this action is the weights in the corresponding asset prices. After that the weights have been chosen, nature decides based on the Markov process of the prices in Figure 1 what the price in the next period will be and therefore the new value of the objective function.

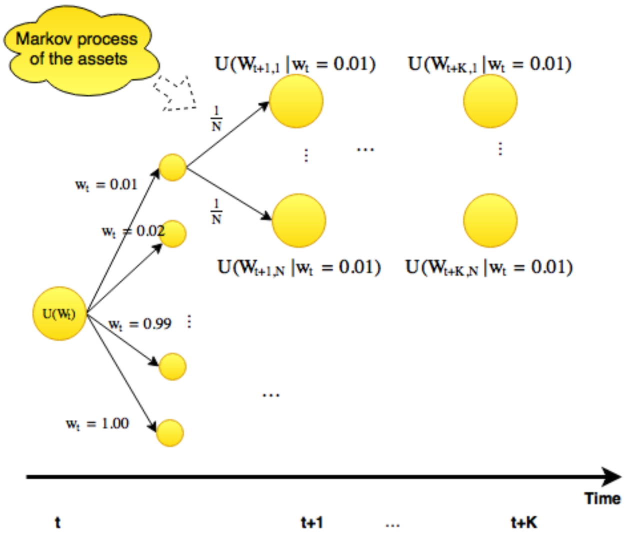
Markov process of the assets
$U(W_{t+1,1}\,|w_t = 0.01)$
$U(W_{t+K,1}\,|w_t = 0.01)$
$\frac{1}{N}$
$\frac{1}{N}$
$w_t = 0.01$
$w_t = 0.02$
$U(W_t)$
$w_t = 0.99$
$w_t = 1.00$
$U(W_{t+1,N}\,|w_t = 0.01)$
$U(W_{t+K,N}\,|w_t = 0.01)$
Time
t
t+1
...
t+K

Figure 2 – Graphical representation of a Markov Decision Progress representing the Portfolio problem statement of Equation 5. Small nodes are actions nodes, where the weights should be determined and large nodes represent the state of the objective function.

Now that the objective function is transformed into a fully deterministic Markov Decision Process we can now solve the problem by Backward induction, this is actually the same as the method of Value iteration in Reinforcement learning. Working from the back you calculate the Expectation of utility for each weight at time $t + K - 1$ by taking the average over the utilities acquired by each w_{t+K-1}. By maximizing each period over the expected utility and starting at the end of the period we maximize the Bellman Equation stated in Equation 8.

3.3 Extensions

Predictability of returns

Diris et al. [2015] not only implements the base case stated in the section before, it also takes the Bayesian estimation of the probability distribution of the assets into account and predictability in the returns. The latter one now implies that instead of $\mathbb{E}[f(\theta)], \mathbb{E}[f(\theta|z_t)]$ needs to be estimated. You can approximate the conditional expectation by the fitted values of the across path regression, that is the fitted values of the regression of the simulated utilities on the state variables.

Bayesian inference

When considering a Bayesian inference, top right cell of the scenario table in Table 4, on the asset prices, one can still make use of the MCMC algorithm to transform the assets into a known Markov

[2] For the ease of consistency in literature I write here Markov Processes instead of Markov Chains, while they are equivalent in discrete state spaces.

Process. Now each path that is sampled by the MCMC has different parameters of the probability distribution. This will in most cases result in a higher variance in the values of the Markov Process.

Transaction costs

By adding transaction costs according to Gârleanu and Pedersen [2013] the Bellman equation in Equation 10 are changes to:

$$\max_{w_t} \mathbb{E}_t \Bigg[\frac{\Big(W_t(w_t' r_{t+1} + R_t^f - \Delta w_t TC)\Big)^{1-\gamma}}{1-\gamma} \max_{w_{t+1} \cdots w_{t+K-1}} \mathbb{E}_{t+1} \Bigg[\Bigg(\prod_{s=t+1}^{t+K-1} (w_s' r_{s+1} + R_s^f - \Delta w_s TC) \Bigg)^{1-\gamma} \Bigg] \Bigg] \tag{13}$$

The term added to the equation is the transaction costs $\Delta w_t TC$, Δw_t is the difference between the weights of time t and $t-1$ and and TC are the constant transaction costs per unit of weight. It is harder to simulate with the numerical solution proposed in Diris et al. [2015], therefore the closed-form solution from Gârleanu and Pedersen [2013] is taken. The optimal weight is the weighted average of the current weight and the aim portfolio (the weighted average of the current and future expected Markowitz portfolios),

$$w_t = \left(1 - \frac{a}{TC}\right) w_{t-1} + \frac{a}{TC} aim_t \tag{14}$$

With a being the optimal weighting scheme futher specified in Gârleanu and Pedersen [2013] For the further specification, I would like to refer to to that paper. The aim of this paper is not about this the dynamic strategy with transaction costs. It is more important for the RL method to include transaction costs because of the fluctuate nature of the machine learning methods.

3.4 Reinforcement learning in portfolio management

In reality, the underlying Markov Process of the asset returns is unknown and also volatile in terms of its model parameters. Therefore, I make use of a model free RL method. Please note, that we need to take the natural log of the objective function and the budget constraint in order to be able to solve the problem by reinforcement learning. This is needed so you can rewrite the objective function in the summation of intermediate functions/rewards, in order for our RL agent to learn from each step in the future. The main difference between the classical method and the reinforcement method is that the latter solves the problem from the beginning propagating forward to the MDP in comparison to the value iteration by backward induction which calculates the optimal path backward. The main problem one faces with RL is that one is unable to get a completely correct estimation of $\mathbb{E}_{t+1}[U(W_{t+K})]$.

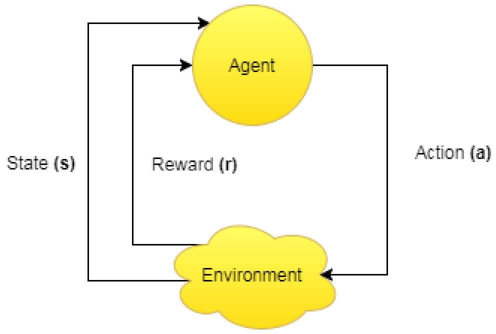

Figure 3 – High-level description of an reinforcement learning agent and how it interacts with the environment, in this case the assets.

A reinforcement learning agent can be represented like in Figure 3. Each point in time the agent takes an action based on its current knowledge of the problem stated and based on its action the environment will give a reward to the agent and the new state of the environment. In this research the state (s) is represented as the current level of the assets, the action (a) is the weights of the assets, and the reward (r) is the log utility gained from moving from state s_t to s_{t+1} with action a(not to be confused with the returns).

Instead of trying to estimate the expectation of utility next period, RL makes use of the Bellman equation and the only known parameter R_{t+1}, the immediate reward from the transition of time period t to $t+1$. Because it is more interesting which actions to take in order to receive the maximum value function, the value function of Equation 8 needs to be decomposed into the action-value function. This is simply the value function given a certain action, these functions are defined in Equation 15 and 16. The value function is actually the same as the intermediate Bellman equation of Equation 6.

$$V_t(s) = \mathbb{E}_t[U(W_{t+K})|S_t = s] \tag{15}$$

$$Q_t(s,a) = \mathbb{E}_t[U(W_{t+K})|S_t = s, A_t = a] \tag{16}$$

This value function can be rewritten to an immediate reward, directly resulting from a specific action performed in a given state, plus the value function of its successor state. This Bellman Equation is shown in Equation 17 and also the corresponding Bellman Equation of the action-value function in Equation 18. With also the power utility, written in the equation, so that you have an analytical term for the immediate return R_{t+1}.

$$\begin{aligned} V_t(s) &= \mathbb{E}_t[R_{t+1} + V_{t+1}(S_{t+1})|S_t = s] \\ &= \mathbb{E}_t\left[\log\left(\frac{\left(W_t(w'_{t,optimal}r_{t+1,S_t} + R^f_{t,S_t})\right)^{1-\gamma}}{1-\gamma}\right) + \log(V_{t+1}(S_{t+1}))\middle| S_t = s\right] \end{aligned} \tag{17}$$

$$\begin{aligned} Q_t(s,a) &= \mathbb{E}_t[R_{t+1} + Q_{t+1}(S_{t+1}, A_{t+1})|S_t = s, A_t = a] \\ &= \mathbb{E}_t\left[\log\left(\frac{\left(W_t(w'_{t,A_t}r_{t+1,S_t} + R^f_{t,S_t})\right)^{1-\gamma}}{1-\gamma}\right) + \log(Q_{t+1}(S_{t+1}, A_{t+1}))\middle| S_t = s, A_t = a\right] \end{aligned} \tag{18}$$

In summary, in RL you assume that the expectations of assets and therefore its respective MP is unknown. By construction, the value function only exists out of the intermediate value functions and is characterized by Equation 17 and 18. If one is in the last period and knows the reward given a certain action, one can back propagate the last action-value function into the rest of the action-value functions, by value iteration. This could also be done for the value function, but we are more interested in which action to take given the state. This method would be an approximate dynamic programming solution and can solve scenario 1 and 3 from Table 4.

Unknown Markov Decision Process / Q-learning

Now I assume that the MDP made with the help of the econometric model is not known, this results that the right side of Equation 18 can not be estimated at all times. Now you need to estimate $Q(s,a)$ at specific states in order to be able to solve for the optimal action-value function at the end of the period. You would, therefore, learn the optimal Q values, in the discrete case a Matrix representing the knowledge of the MDP describing the trading system. First, the Q matrix initialized at zero and updated iteratively with the reward function and the next Q value in the matrix. Because the reward function is known at each state and action pair you can iterate w.r.t. the adjusted greedy policy (ε-greedy heuristic) to update the values of the Q matrix. This heuristic chooses the next action based on the maximal Q function and chooses a random action by the parameter ε. This updating rule follows directly from Equation 18 and is given in Equation 19.

$$q(s_t, a_t) = q(s_t, a_t) + \alpha \left[r + \max_a q(s_{t+1}, a_{t+1}) - q(s_t, a_t) \right], \tag{19}$$

with α being the learning rate, the rate to which extent the value is updated with the estimated value of the Q function. And the action a, here written as the action leading to the highest action-value function in the next period, chosen greedily but also some random deviation included, otherwise only one path will be updated continuously without learning rest of the Q matrix. By iteratively updating $q(s, a)$ it converges to the optimal action-value function $q_*(s, a)$. And from the optimal Q table, one can calculate by value iteration or backward induction the optimal policy/weights for the assets.

3.5 Extensions Reinforcement Learning

Transactions costs

Adding transaction costs to the RL method would mean that the immediate Reward should be reduced in Equation 18 relative to a number of assets traded which has a direct relation of the action to take by the method. The new reward function would, therefore, become equal to Equation 20.

$$R_{t+1} = \log \left(\frac{\left(W_t(w'_{t,A_t} r_{t+1,S_t} + R^f_{t,S_t}) \right)^{1-\gamma}}{1-\gamma} - \Delta w_t TC \right) \tag{20}$$

Function approximation

Because the states are actually not representable in discrete values because the assets have a continuous state space and in this research, even three continuous variables are taken into consideration the Q table would become large. This makes it not viable in terms of memory but also not in terms of learning time. One solution for this would be to estimate the value function by function approximation: like a linear combination of the assets, an econometric model, or a neural network.

$$\hat{Q}(s, a, \theta) \approx Q_t(s, a) \tag{21}$$

This is more efficient because this function could generalize from states already encountered to new states and therefore reduce the memory usage. Although it never converges to the true action-value function in practice it tends to oscillate around and approximate the action-value function closely. In this research, it would not be viable to construct a table of all the states and the infinite actions when considering continuous weights.

To optimize the parameters of θ from the function approximation I make use of the stochastic gradient descent method for Neural Networks[3]. For the ease of generality, Neural Networks are used so that you do not have to optimize over the functional form of the function approximator because you can form many different kinds of functions with Neural Networks. Keep in mind that Neural Networks can represent any abstract relationship between two variables of interest, from linear to nonlinear to high dimensional relationships. The loss function that would be used to propagate back through the Neural Network is defined in Equation 22. Only the loss function is stated because that is the sole instrument needed for the Neural Network.

$$Loss = \sum_a \left(R + \max_{a_{t+1}}(q(s_{t+1}, a_{t+1})) - q(s_t, a_t) \right) \tag{22}$$

[3] The package this paper uses for Neural Networks is Tensorflow for Python.

References

V. DeMiguel, L. Garlappi, and R. Uppal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, 39:43–62, 1997. doi: 10.1016/S0169-7439(97)00061-0.

B. Diris, F. Palm, and P. Schotman. Long-term strategic asset allocation: An out-of-sample evaluation. *Management Science*, 61(9):2185–2202, 2015. doi: http://dx.doi.org/10.1287/mnsc.2014.1924.

N. Gârleanu and L.H. Pedersen. Dynamic trading with predictable returns and transaction costs. *The Journal of Finance*, 67, 2013. doi: 10.1111/jofi.12080.

T. Hens and P. Woehrmann. Strategic asset allocation and market timing: A reinforcement learning approach. *Computational Economics*, 29(3):369–381, 2007. doi: 10.1007/s10614-006-9064-0.

J. Moody, L. Wu, Y. Liao, and M. Saffell. Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting*, 17:441–470, 1998. doi: 10.1.1.87.8437.

D. Svozil, V. Kvasnicka, and J. Pospíchal. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *The Review of Financial Studies*, 22, 2009. doi: 10.1093/rfs/hhm075.