Chapter 1

Introduction

1.1 Introduction

Portfolio management is a critical area in finance, studied by investors at all levels, from individuals to large institutions. It can be challenging to decide how to allocate capital, but portfolio management provides a framework for making these decisions.

Starting with the traditional Markowitz mean-variance model, researchers have developed many investment theories, including the 60/40 rule as a special case of fixed mix allocations. These traditional investment strategies are effective for single-period optimization problems, but they neglect key factors such as transaction costs, taxes, and underlying market regimes. Multi-period portfolio optimization addresses these issues by allowing information to flow between different time periods. However, finding an analytical solution for a multi-period portfolio problem is challenging and complex.

The work in this report mainly focuses on testing different algorithms for finding an effective trading strategy in a market with multiple regimes that are not directly observable by the investor and where linear transaction costs incur.

1.2 Related Work

The Markowitz mean-variance portfolio, introduced by Markowitz (1952), offers a systematic method for balancing the trade-off between returns and risk. It works effectively for single-period portfolio problems but is not readily applicable to multiperiod scenarios. Merton (1969) pioneering work in mathematical finance, addresses continuous-time portfolio optimization without transaction costs, assuming asset prices follow geometric Brownian motion. An important extension to Merton's model is the inclusion of linear transaction costs, as demonstrated by Davis and Norman (1990). They illustrate that, under reasonable assumptions, the optimal trading strategy in the presence of linear transaction costs involves establishing a no-transaction zone, where trading occurs only when the current position deviates from this zone. In the context of solving multistage asset-allocation problems with linear transaction costs, Mulvey et al. (2001) noted that the optimal policy typically involves creating no-trade zones around asset targets or proportions. These studies lay the foundation for our approach, which focuses on identifying the boundaries of these no-trade zones rather than solving complex optimization problems. The dynamic programming algorithm and the neural network model to find no trade zone proposed by Li and Mulvey (2021) are taken as references for the project.

1.3 Problem Setup

The following equation recognises a general multi-period portfolio optimization problem that addresses the challenges. let there are n $\geq$ 1 risky assets and one risk-free asset in the market, and the prices of risky assets follow trends specific to the regime. Investors cannot directly observe the market regime but can make informed estimates about their likelihood based on asset returns. They choose to adjust their portfolio periodically to optimize their long-term utility. However, these adjustments come with transaction costs that increase proportional to the investment.

$$\begin{aligned} \underset{\pi_0,\pi_1,\ldots,\pi_{T-1}\in\mathbb{R}^n}{\text{Maximize}} \; & Utility[Z_1, Z_2, ...] \\ \text{subject to } & \mathbf{1}^T\pi_t = 1 & \forall t = 0, ..., T-1 \\ & W_t{}^{\rightarrow} = W_t(\pi_t{}^T(1+r_t)) & \forall t = 0, ..., T-1 \\ & \pi_t{}^{\rightarrow} = \frac{\pi_t \odot (1+r_t)}{\pi_t{}^T(1+r_t)} & \forall t = 0, ..., T-1 \\ & W_{t+1} = W_t{}^{\rightarrow} - C(W_t{}^{\rightarrow}; \pi_t{}^{\rightarrow}, \pi_{t+1}) & \forall t = 0, ..., T-1 \end{aligned} \tag{1.1}$$

where $\pi_0, ..., \pi_{T-1}$ are decision variables which represnt the allocation at the beginning of each period, W_T is the wealth at the beginning of period t, $W_t{}^{\rightarrow}$ is the wealth at the end of period t, $\mathbf{1} \in R^n$ is vector with all ones,$r_t \in \mathbb{R}^n$ is the returns vector in period t, $\pi_t{}^{\rightarrow}$, $\odot$ is the element-wise multiplication operator, and $C(W_t{}^{\rightarrow}; \pi_t^{\rightarrow}, \pi_{t+1})$ is the transaction value (can be in Rupee) when the allocation is rebalanced from π_{t+1} to $\pi_t{}^{\rightarrow}$. Initial wealth W_0 is assumed to be given. The arguments of utility function can vary from $\mathbb{E}[W_T]$ to standard deviation and many more complex functions depending on the use case.

1.4 The Market

A market is taken into consideration that features different regimes. Within each of these regimes, the returns on risky assets follow a multivariate normal distribution. The specific parameters of these distributions are tied to the respective regime. To keep things manageable, an assumption is made that the regime switches at the end of each period. Furthermore, assume that this regime-switching behavior satisfies the Markov property. In other words, the probability of the next period's regime, given the current period's regime, is independent of all prior periods. This probability transition is expressed through a transition matrix. Mathematically, the vector of risky assets in period i,$r_t = [r_{t,1}, ...r_{t,n}]'$, follow the dynamic

$$r_t \sim N(\mu_{S_t}, \Omega_{S_t})S_t \in \{1, 2, ..., N\} \tag{1.2}$$

where N is the number of possible regimes, and μ_k and Ω_k for $k \in \{1, 2, ..., N\}$ are the parameters of the normal distribution under regime k. Furthermore, within this market, a risk-free asset is present, and its price experiences growth at a rate denoted as r_f. It is worth noting that the risk-free rate might exhibit dependence on the underlying regimes within the market. It is also considered that the market operates under the assumption of linear transaction costs.

For an investor possessing total wealth represented as W and holding a position denoted as π_1, any adjustment made to this position, labeled as π_2, results in the incurrence of transaction costs. These transaction costs are quantified as $cW||\pi_1 - \pi 2||_1$, where the constant c assumes values between 0 and 1.

1.5 The Investor

Investors participate in the market to earn profits and achieve their investment objectives. It is assumed that they are restricted to trade only at the end of specific time intervals, after knowing the returns for that period. Even though they lack direct insight into the current market situation, they use historical data to estimate the likelihood of future market conditions. Consequently, their decisions to adjust their investments are influenced by the present market performance and their expectations of the future.

The primary objective is to maximize the anticipated benefits in final wealth, denoted as $\mathbb{E}[U(W_T)]$. To achieve this, a utility function is used known as the Constant Relative Risk-Aversion (CRRA) function, chosen for its property of consistently increasing with terminal wealth growth.

$$U(W) = \begin{cases} \frac{W^\gamma}{\gamma}, & \gamma \neq 0 \\ log(W), & \gamma = 0 \end{cases} \tag{1.3}$$

When $\gamma \leq 0$, the utility function is strictly concave and the investor is risk-averse.

Chapter 2

Background

2.1 Portfolio Optimization

Portfolio management aims to generate higher returns with lower risk. The key component is to decide how much capital to invest in each asset, which is often done through optimization. Traditional portfolio optimization involves two steps: estimating parameters of interest (e.g., expected returns and covariance matrix) and solving an optimization problem with selected objective functions (e.g., mean-variance, risk-based, or utility-based).

Parameter estimation is crucial, as it directly affects portfolio performance. An intuitive way to estimate expected returns and covariance is through historical estimators. The hidden Markov Model (HMM) is one of the well-known statistical models to estimate the parameters.

2.1.1 Markowitz Mean-Variance Portfolio Theory

Investors prefer high returns and low risk, but it is unrealistic to pursue both simultaneously. In equilibrium, higher expected returns tend to come with higher risk. The Markowitz mean-variance portfolio (Markowitz (1952)) provides a systematic approach to finding a balance between return and risk. In this model, the risk is defined as variance of the portfolio. Mathematically, an investor solves the optimization problem

$$\underset{w\in\mathbb{R}^n}{\text{Maximize}} \quad \mu^T w - \lambda w^T \Sigma w \tag{2.1}$$

$$\text{subject to} \sum_{i\in 1,\ldots,n} w_i = 1 \tag{2.2}$$

$$\mathbf{w} \geq 0 \tag{2.3}$$

where $\lambda \geq 0$ is the risk aversion is the risk aversion coefficient that describes the risk-reward preference of the investors. Constraint 1.2 is the budget constraint, and Constraint 1.3 is the non-negativity constraint that corresponds to a long-only portfolio. If shorting or leveraging is allowed, Constraint 1.3 can be relaxed accordingly.

The problem can have variations by setting a target portfolio return and minimize variance, or set a target variance tolerance and maximize the expected return. In the standard mean-variance formulation, a larger λ corresponds to more risk-averse investing behavior. In particular, when $\lambda = 0$, the investor cares solely about the expected portfolio return $\mu^T w$. On the other hand, when λ approaches infinity, the portfolio variance gains more weight, leading to a portfolio mimicking the allocation of a minimum-variance portfolio. As λ varies from 0 to infinity, the associated optimal portfolio has decreasing expected return and volatility. Plotting the expected portfolio return versus its volatility on a graph produces the mean-variance efficient frontier.

2.1.2 Risk based Portfolio

Minimum Variance

This is a variant of Markowitz Mean-Variance portfolio problem which mimics the nature when λ approaches infinity. The objective of a minimum-variance portfolio is to minimize the portfolio variance. The optimization problem for a minimum-variance portfolio is

$$\underset{w\in\mathbb{R}^n}{\text{Minimize}} \quad w^T \Sigma w \tag{2.4}$$

$$\text{subject to} \sum_{i\in 1,\ldots,n} w_i = 1 \tag{2.5}$$

$$\mathbf{w} \geq 0 \tag{2.6}$$

As the expected returns are no longer considered here, the optimization problem is no longer sensitive to returns estimation.

2.2 Hidden Markov Model

A hidden Markov model consists of a pair of series (X_t, Y_t) indexed by t, where X_t is a state series whose elements X_t follow a Markov process, and Y_t is an X_t-measurable series whose distributions decided by the corresponding state X_t . The states X_t 's are not directly observable, as suggested by the name hidden Markov model. On the other hand, Y_t is observable, with which one may infer the underlying state X_t. The most widely applied parameter estimation method for hidden Markov models is the forward-backward algorithm Baum (1972), a special case of expectation-maximization algorithm.

For the purpose of this project, HMM is used to estimate the expected returns and covariance matrices of the assets using historical data when the market follows a regime-switching Markov process.

2.3 Machine Learning Applications in Finance

Machine learning algorithms have become increasingly popular in recent years, as the amount of available data has grown exponentially. These algorithms can improve model performance and allow us to learn patterns from high-dimensional data that traditional methods cannot process.

Machine learning methods have been used in the financial sector since the 1990s. They have established their role as an alternative to traditional methods and an efficient solution to the growth of state space. Traditional methods quickly reach their limits in multi-period portfolio models with no analytical solution, as the planning horizon extends or the number of possible regimes increases. We can leverage machine learning methods to find promising solutions to these problems.

2.3.1 Neural Networks

Artificial neural networks (ANNs) are machine learning models inspired by biological neural networks. They are widely used for predictive modeling and adaptive control. ANNs consist of interconnected nodes (neurons) with weighted connections. Figure 2.1 exhibits an example of a simple neural network with one hidden layer. During training, the ANN is fed with training samples and it adjusts the weights using gradient descent to minimize the loss function. The loss function measures how well the ANN's predictions match the actual outputs of the training data. The process of adjusting the weights is called backpropagation. Selecting a suitable learning rate is important for training an ANN. A too-small learning rate can lead to slow training, while a too-large learning rate can lead to instability.

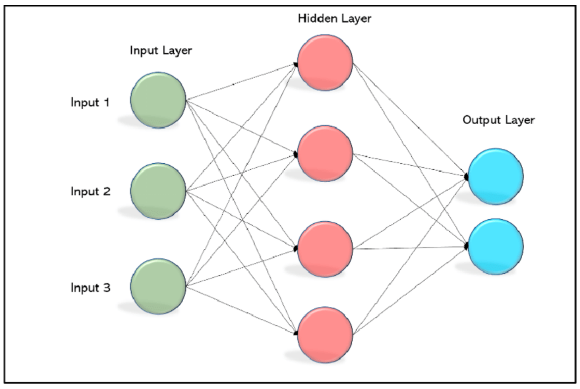


FIGURE 2.1: An artificial neural network with one hidden layer

2.3.2 Recurrent Neural Networks

A recurrent neural network (RNN) is an artificial neural network designed for problems with temporal dependencies. Unlike feedforward networks, RNNs establish directed connections between nodes. In a basic RNN, the model processes input at each time step by combining the input at the current time step with the hidden state from the previous step. This is illustrated in the computational graph in Figure 2.2 which unfolds the RNN over time. Formally, given a sequence $x_0, ..., x_T$, an RNN calculates the hidden states h_t and outputs o_t as follows

$$h_t = \Theta(W_{hh}h_{t-1}, W_{xh}x_t), o_t = \Phi(W_{ho}h_t) \tag{2.7}$$

In these equations, W_{hh}, W_{xh}, and W_{ho} represent the connection weights between nodes, and Θ and Φ are activation functions tailored to the specific problem. RNNs have extensions such as long short-term memory (LSTM) and bi-directional RNNs.

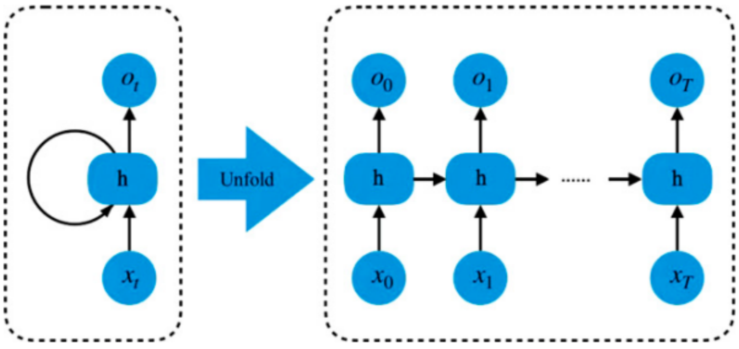


FIGURE 2.2: RNN Architecture

2.3.3 Reinforcement Learning

Reinforcement learning is a machine learning method that teaches an agent how to behave in an environment to maximize its reward. The problem involves an environment and an agent, and defines the set of feasible action as well as the transition probabilities of state switching. Typically, the environment is described as a Markov decision process (MDP), where state space $\mathbb{S}$, action space $\mathbb{A}$, transition probabilities $\mathbb{P}(s^{'}|s, a)$ for $s, s^{'} \in \mathbb{S}, a \in \mathbb{A}$ and reward functions $R(s, s^{'}, a)$ are carefully stated. The agent learns by exploring the environment and taking actions that lead to higher rewards.

Chapter 3

Methodology

Li and Mulvey (2021) proposed a two-stage approach to address the multiperiod portfolio allocation problem. In the initial stage, numeric dynamic programming is employed to determine the optimal allocation strategy when transaction costs are zero. In the subsequent stage, the obtained solution strategy is utilized as an initial input for a neural network, which is employed to devise an investment plan for the complete problem detailed in 1.3.

3.1 Dynamic Programming approach

Dynamic programming leverages the inherent backward recursive nature to address the optimal portfolio problem within the context of regime switching and in the absence of transaction costs.

3.1.1 Construction of state space

In a multi-period portfolio optimization problem, the state space encompasses all the information used by an agent to make trading decisions. For this particular problem, assuming no transaction costs, the minimal necessary state consists of the probability distribution of the current regime and the time remaining until the end.

Mathematically, this state is represented as a pair (p, τ), where $p = [p_1, ..., p_N]$ is a vector denoting the probabilities of the current period belonging to each regime, and τ represents the current time period.

To facilitate dynamic programming, the state space is discretized to make it finite. The probability vector $p = [p_1, ..., p_N]$ is constrained to take values within a discretized set $P = \{[p_1, ..., p_N] | p_i \in 0, \delta,\ 2\delta,...,\mathrm{M}\delta = 1\}\ \forall i \in \{1, ..., N\}, \sum_{i=1}^{N} p_i = 1$ where δ is a small unit change, Now let S be the state space containing all valid (p,τ) as described.

3.1.2 Action Space

The action space, denoted as A, encompasses all feasible actions adhering to the nonanticipatory criteria. In the context of the multiperiod portfolio optimization problem, the agent's actions revolve around determining the allocation of wealth into the risky asset. Therefore, the action space is represented as. $\{\pi = [\pi^1, ..., \pi^n] | \pi^j \in [\pi_l, \pi_u] \forall j \in \{1, ..., n\}\}$. Here, π_l and π_u denote the lower and upper bounds on position, respectively.

3.1.3 Value Function

A value function, denoted as $V : S \to \mathbb{R}$, serves as a mapping that associates states with real numbers. It quantifies the anticipated outcomes for an agent starting from a particular state $s \in S$. Specifically, for this problem, the value function is defined as the expected terminal utility, considering the optimal trading strategy from the current state to the horizon, assuming the initial wealth is Rs.1:

$$V(s_i) = \underset{\pi}{\text{Maximize}} E[U(W_T) | s_i, W_i = 1] \forall s_i \in S \tag{3.1}$$

This value function must adhere to the Bellman equation:

$$V(s_i) = \underset{\pi_i}{\text{Maximize}} \sum_{s_{i+1}, W_{i+1}} p(s_{i+1}, W_{i+1} | s_i, \pi_i) \mathbf{E}[{W_{i+1}}^{\gamma} V(s_{i+1})] \tag{3.2}$$

Here, $p(s_{i+1}, W_{i+1}|s_i, \pi_i)$ represents the probability of transitioning to state s_{i+1} with wealth W_{i+1} at the next stage, given that the agent takes action π_i under the current state s_i.

3.1.4 Algorithm

The following pseudo-code outlines the dynamic programming algorithm employed to seek the optimal portfolio under regime switching conditions and in the absence of transaction costs:

```
Require: Optimal allocation of risky assets $\pi^*(p,t)$
  t = T-1
  for each p = $[p_1,...,p_N] \in$ P do
      $V(p,T) = \frac{1^\gamma}{\gamma}$
  end for
  function UPDATEBELIEF(r,p)
      $p_k{}^{new} = \sum_{i=1}^{N} pdf(r; \mu_i, \sigma_i) * p_i * tpm[i][k]$
      $p^{new} = [p_1{}^{new}, ..., p_N{}^{new}]$
      Normalize $p^{new}$ so that the sum of probabilities is 1
      return $p^{new}$
  end function
  function NEWWEALTH(r,$\pi$)
      $W^{new} = \sum_{i=0}^{n} \pi^i * (1 + r^i)$
      return $W^{new}$
  end function
  return $p^{new}$
  while t≥0 do
      for each p = $[p_1,...,p_N] \in$ P do
          Simulate M return scenarios $r_1, ..., r_M \in \mathbb{R}^{n+1}$ in period t
          based on probability of regimes p
          $\pi^*(p,t) = argmax_\pi(\frac{sum_{s=1}^{M} \text{NEWWEALTH}(r_s,\pi)^\gamma V(\text{UPDATEBELEIF}(r_s,p)),t+1)}{M})$
          $\pi^*(p,t) = max(\pi_l, min(\pi_u, \pi^*(p,t)))$
          $V(p,t) = (\frac{sum_{s=1}^{M} \text{NEWWEALTH}(r_s,\pi^*)^\gamma V(\text{UPDATEBELEIF}(r_s,p)),t+1)}{M})$
      end fort = t-1
  end while
```

Here, tpm is the transition probability matrix between regimes, and tpm[i][j] represents the probability that the next regime is j given current regime is i. The time complexity of a dynamic programming approach is closely tied to the sizes of the state and action spaces. When either the state or action space expands, it necessitates an increase in the number of simulations to explore all potential actions across all states. This phenomenon is commonly referred to as the "curse of dimensionality."

Conversely, if transaction costs are considered, the state space must also encompass the current position π to provide comprehensive information to the agent which increases the state space exponentially. Therefore, neural networks are used in the second stage of the algorithm to devise a strategy that addresses transaction costs.

3.2 No Trade Zone

Assuming zero transaction costs, a dynamic programming solution provides a solid starting point for further exploration to determine optimal trading strategies when proportional transaction costs are taken into account. However, in the presence of non-negligible transaction costs, it becomes inappropriate to directly apply the no-transaction-cost strategy.

Investors face a trade-off in this scenario. On one hand, adhering closely to the optimal position from the no-transaction-cost case can lead to higher expected terminal utility. On the other hand, frequent transactions can be costly, and investors aim to minimize total transaction costs.

In cases involving multivariate normal distribution returns of assets and proportional transaction costs, it is well-known that the optimal trading strategy involves establishing a "no-trade zone," as introduced by Davis and Norman (1990). When the current position falls within this no-trade zone, no action is required. However, if the current position is outside this zone, the agent should rebalance the portfolio to the nearest point, either the lower or upper boundary, within the no-trade zone.

Applying this strategy to a regime-switching market, let's consider a no-trade zone with a shape as depicted in Figure 3.1. Assume a current regime probability of 0.2 and three different current positions:

• In Case (a), the current position of the risky asset is at point a. Since this position exceeds the upper bound of the no-trade zone, selling some of the risky assets and rebalancing to position a' on the upper boundary is necessary.

• In Case (b), the current position of the risky asset is at point b. As point b falls within the no-trade zone, no action is required.

• In Case (c), the current position of the risky asset is at point c. Since this position is below the lower bound of the no-trade zone, purchasing some risky assets and rebalancing to position c' on the lower boundary is necessary.

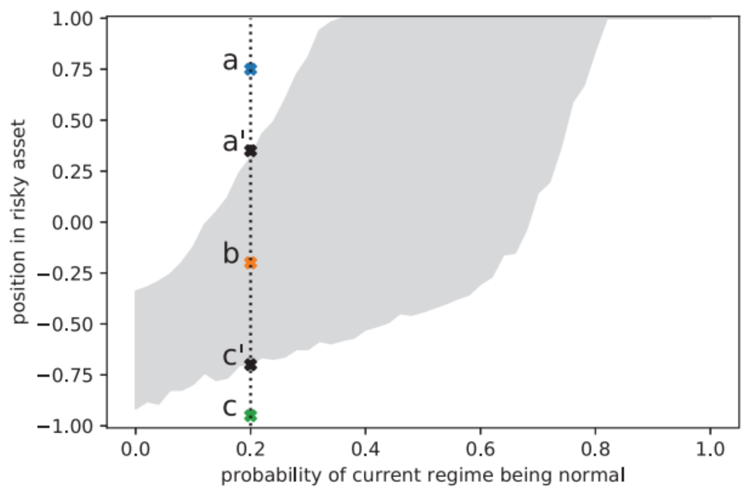


FIGURE 3.1: Example of No Trade Zone

3.3 Neural Network Method

Searching for the Optimal No-Trade Zone

In recent years, neural networks have found extensive utility in a variety of domains, ranging from applications in partial differential equations to their role in mathematical finance. Neural networks are particularly effective tools, especially when they can leverage an advanced starting point. The algorithm's approach involves employing a neural network in conjunction with dynamic programming as an initial reference point. The primary objective is to delineate a "no-trade zone" around this allocation. Within this strategy, the upper and lower boundaries of the no-trade zone are parameterized as

$$u(p,\tau) = \pi(p,\tau) + f_u(p) \tag{3.3}$$

$$l(p, \tau) = \pi(p, \tau) - f_l(p) \tag{3.4}$$

where $\pi(p, \tau)$ is the optimal position under no transaction cost, and $u(p, \tau)$and $l(p, \tau)$are lower and upper boundaries of the no-trade zone, respectively. The neural networks approximate the non-negative functions of interest, f_u and f_l, that is, the "width" of the no-trade zone around $\pi(p, \tau)$.

The network based architecture is based on the structure proposed by Mulvey et al. (2020)

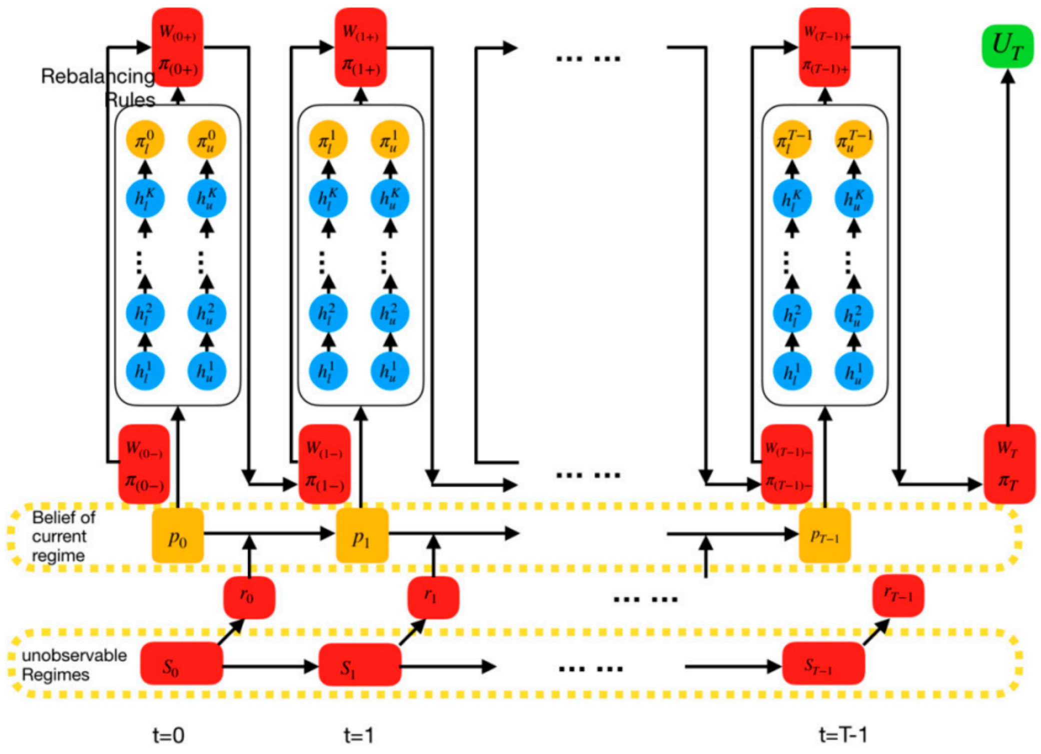


FIGURE 3.2: Neural Network Computational graph

The computational graph for the neural network appears in Figure 3.2. The relationships inside the computational graph are as follows:

• S_t for t=0, ...,T represents the underlying regime of the market in period t. The regime evolves according to a Markov process and is not directly observable to the investors.

• $r_t \in \mathbb{R}^n$ for t=0, ...,T represents the realized return of the risky assets in period t. It is normally distributed with mean μ_{S_t} and covariance Ω_{S_t}.

• $p_t \in \mathbb{R}^N$ for t=0, ...,T is the investor's belief of the current regime. The investor updates her belief of the underlying regime based on her previous belief and the new realized return using the Bayesian rule.
• ${h_l}^k$ and ${h_u}^k$ represent the hidden layers. To increase the stability of the results, same set of weights are shared across all periods. For t=0, ...,T-1, ${\pi_l}^t$ and ${\pi_u}^t$ are the output lower and upper boundaries, respectively, of the no-trade zone learned by the neural network.
• $W_{(t-)}$ and $\pi_{(t-)}$ for t=0, ...,T-1 depict the wealth and asset allocation at time t before rebalancing, respectively. At time t, the investors can rebalance their portfolio. The new allocation $\pi_{(t+)}$ is chosen based on Section 3.2 according to the old allocation $\pi_{(t-)}$ and the no-trade zone calibrated by the neural network. The wealth after rebalancing is calculated as $W_{(t+)} = W_{(t-)}$ - $cW_{(t-)}||\pi_{(t+)} - \pi_{(t-)}||_1$

Monte Carlo simulations are used to generate regimes and prices of the risky assets based on estimated parameters in order to train the neural network. At the end of period T, the loss function equals the negative of expected terminal utility,

$$loss = -E[U(W_T)], \tag{3.5}$$

and is minimized using gradient backpropagation

Chapter 4

Experimental Results

4.1 Dataset

The NIFTY50 is a benchmark Indian stock market index that represents the weighted average of 50 of the largest Indian companies (by market capitalization) listed on the National Stock Exchange. For testing the algorithms, stock prices of Axis Bank which is one of the NIFTY50 assets are chosen over the past 5 years i.e., from 29-10-2018 to 23-10-2023. The dataset is collected from a trusted source (Yahoo Finance). It consists of Open, High, Low, Close, Adjusted Close, and Volume for each trading day.

Feature Name	Description
Open	Price at which stock opens in the market when trading begins
High	The highest trading price of a stock
Low	The lowest trading price of a stock
Close	Price at which stock closes in the market when trading ends
Volume	Number of shares traded in a day
Adjusted Close	Price of the stock after paying off the dividends

To calculate weekly returns data for further proceedings, the closing price is chosen as a representation of the price of the stock on each trading day. The obtained plot is as follows

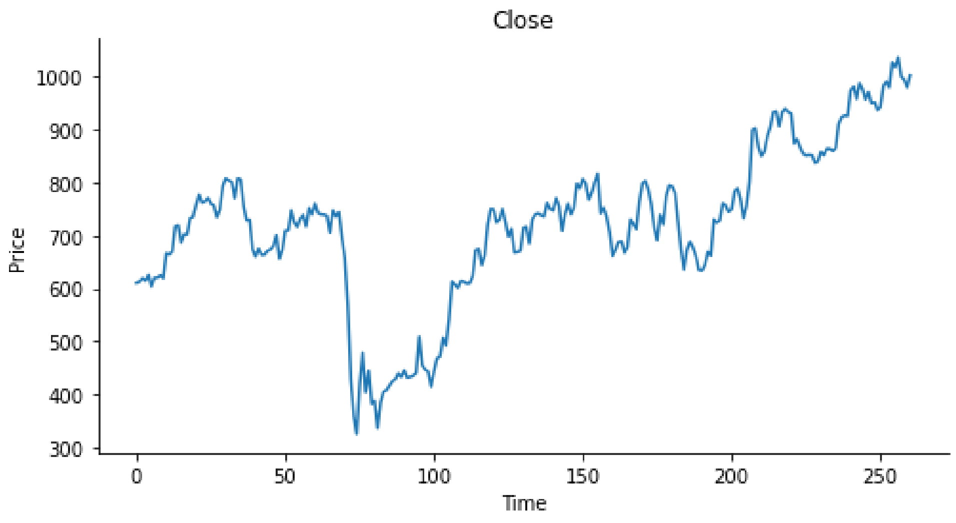


FIGURE 4.1: Closing price of Axis Bank from 29-10-2018 to 23-10-2023

Now, weekly returns are calculated from the closing price values over the entire time period. The plot of weekly returns is obtained as follows

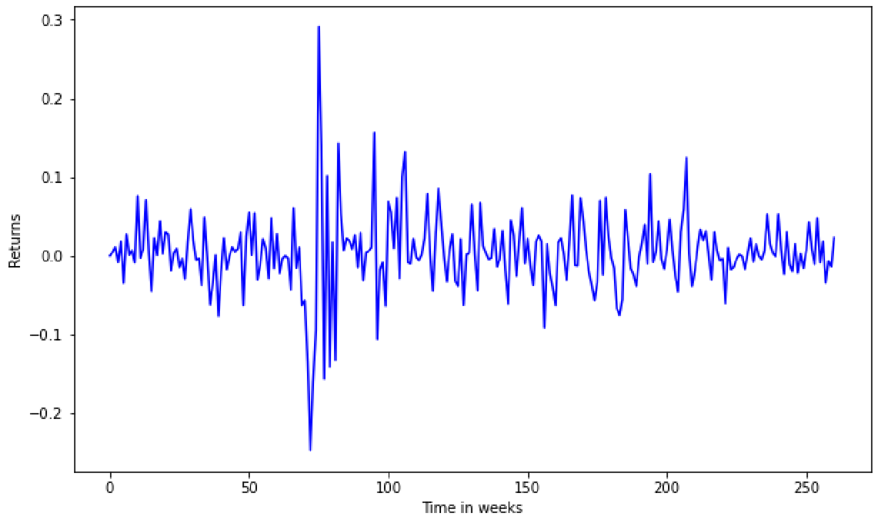


FIGURE 4.2: Returns of Axis Bank from 29-10-2018 to 23-10-2023

Statistics of returns of the asset are shown in the following table

Feature	Description
count	261.000000
mean	0.003172
std	0.050405
min	-0.247275
25%	-0.017021
50%	0.002421
75%	0.022502
max	0.290982

4.2 Parameter estimation

Now, assume that the market has two underlying regimes that are unobservable by the investor. There is one risky asset which is Axis Bank in this case and one risk-free asset in the market. The annualized return on the risk-free asset, r_f is 5% (estimated from Fixed Deposits Interest Rates of various banks) irrespective of the regime. To estimate the estimated returns, covariance matrix, and transition probability matrix of the risky asset, a Gaussian HMM is used which is available in the library hmmlearn in Python.

The HMM is learned for 10000 iterations and the estimated parameters of the risky asset are obtained as:
Regime 1: Mean: 0.00368291, Covariance: 0.00114034

Regime 2: Mean: -0.00135955, Covariance: 0.01559686

Transition Probability matrix of the underlying regimes: $\begin{pmatrix} 0.98581142 & 0.01418858 \\ 0.12509587 & 0.87490413 \end{pmatrix}$
Also, the allocation in the risky asset is limited to the range [100%, 100%] to avoid excess computation. The transaction cost is chosen to be 0.5% and CRRA coefficient γ = -1

Here, regime 1 is considered to be a normal regime since it has a positive mean and a lesser covariance compared to the other regime. Regime 2 is considered to be a crash regime since the returns have a negative mean value associated with it.

A random return path of the risky asset generated using the given parameters using Monte-Carlo simulations is as follows

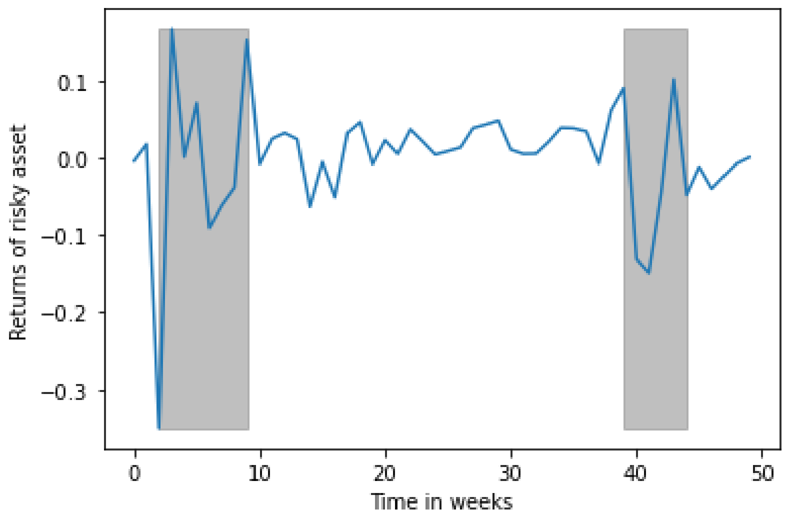


FIGURE 4.3: Random returns path of risky asset

The area shaded in grey in Figure 4.3 refers to the market being in a crash regime (Regime 2). It can be observed that the market is relatively more stable in the normal regime and also has greater returns.

The following random paths show the belief update of the regimes varying to the observed returns from the market.

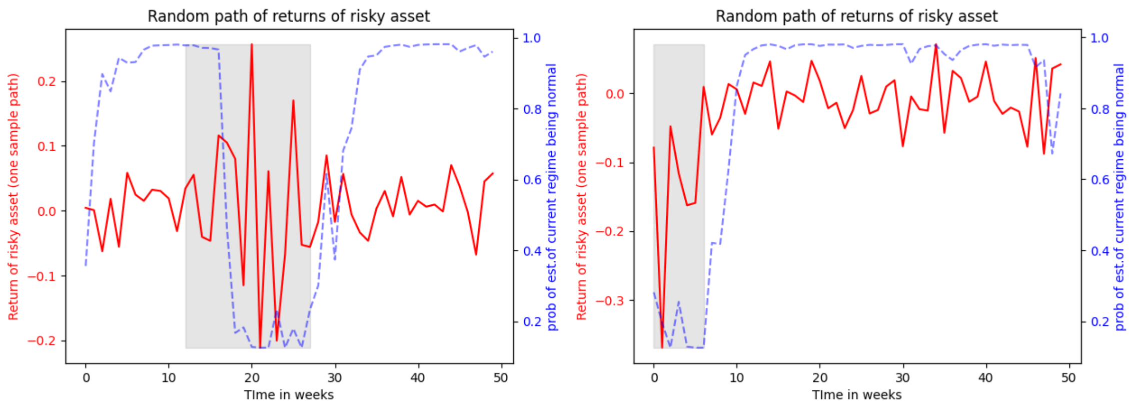


FIGURE 4.4: Returns vs.belief about regimes

4.3 Dynamic Programming

Because of the lack of computation resources on my laptop, the horizon is set to T = 5 weeks from here on. Also, the probability space over regimes and allocation vector in the assets are discretized with = 0.02 to declare state space for dynamic programming.

The number of return scenarios (M) considered at every time period in dynamic programming algorithm as described in 3.1 is taken to be 10000. Upon executing the algorithm, the plot of optimal allocation in risky asset v/s belief about the current regime being normal at t = 2 weeks is obtained as follows:

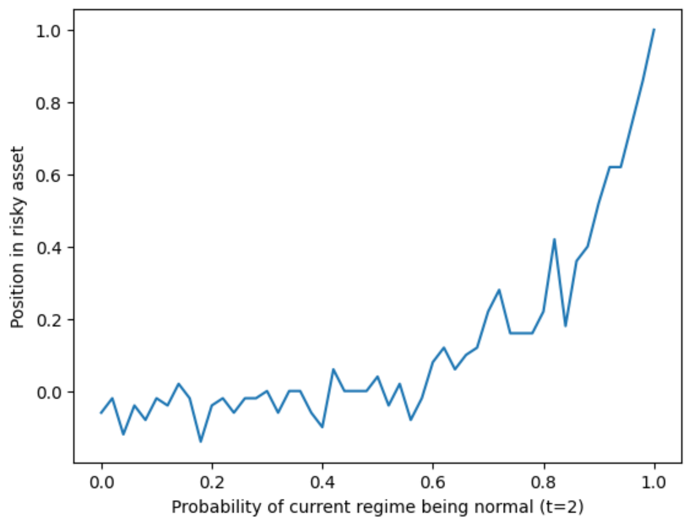


FIGURE 4.5: optimal allocation v/s belief that current regime is normal

Investing 60% percent in risky assets and 40% in risk-free assets is one of the most widely known investment strategies. This is an example of fixed mix allocation. Now, random paths of the market returns are generated to compare the dynamic approach performance with the fixed mix allocation strategy and also the strategy where the entire amount is invested in the risky asset itself. The plots obtained are as follows

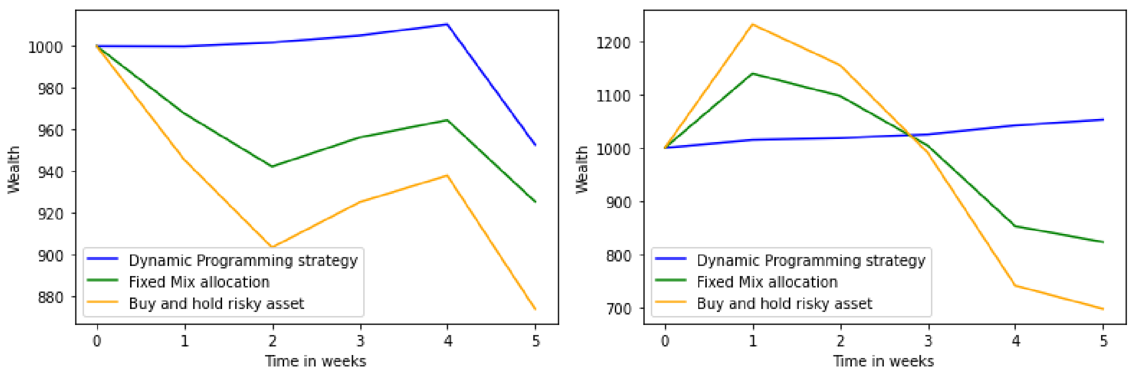


FIGURE 4.6: Random wealth path comparison

Now, the transaction costs are considered with a transaction rate of 0.5%. Random wealth paths are given below to show how the transaction costs affect the total wealth at the end of the horizon.

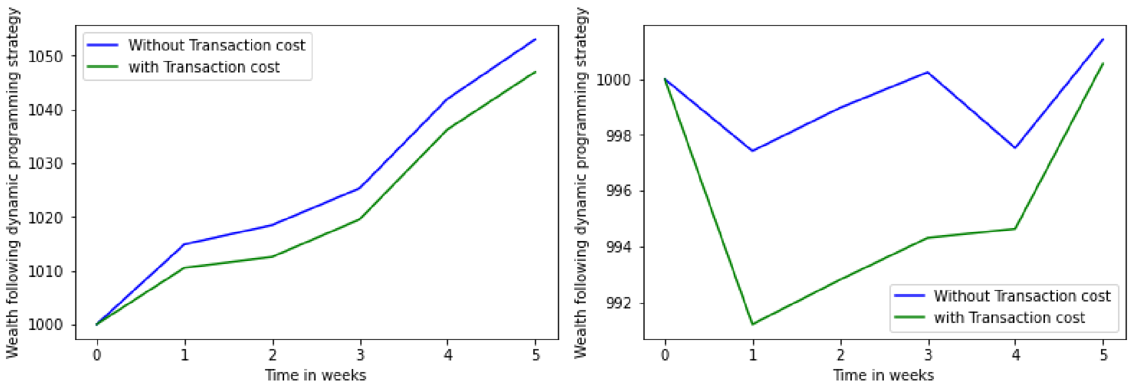
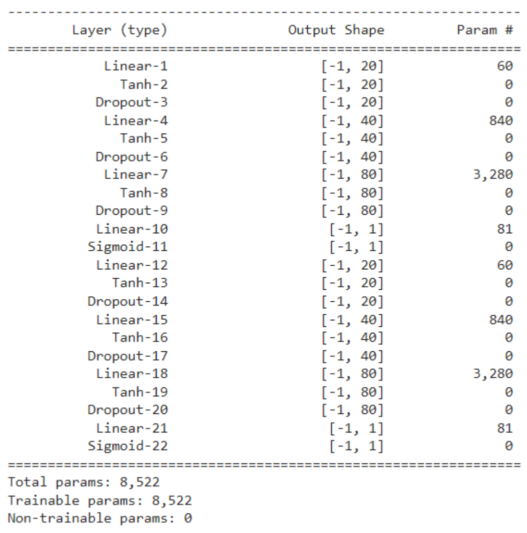


FIGURE 4.7: Wealth path difference in presence of transaction costs

4.4 Combined Method

Using the results obtained from dynamic programming as an effective start to find optimal strategy when there are transaction costs, a neural network is considered which has three hidden layers with neurons 20,40 and 80 respectively, and with Tanh as an activation function. Dropout layers are also added to reduce overfitting with a dropout probability of 0.2. The final output vector of n dimension is taken from a sigmoid layer to ensure non-negativity.



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                [-1, 20]              60
              Tanh-2                [-1, 20]               0
           Dropout-3                [-1, 20]               0
            Linear-4                [-1, 40]             840
              Tanh-5                [-1, 40]               0
           Dropout-6                [-1, 40]               0
            Linear-7                [-1, 80]           3,280
              Tanh-8                [-1, 80]               0
           Dropout-9                [-1, 80]               0
           Linear-10                 [-1, 1]              81
          Sigmoid-11                 [-1, 1]               0
           Linear-12                [-1, 20]              60
             Tanh-13                [-1, 20]               0
          Dropout-14                [-1, 20]               0
           Linear-15                [-1, 40]             840
             Tanh-16                [-1, 40]               0
          Dropout-17                [-1, 40]               0
           Linear-18                [-1, 80]           3,280
             Tanh-19                [-1, 80]               0
          Dropout-20                [-1, 80]               0
           Linear-21                 [-1, 1]              81
          Sigmoid-22                 [-1, 1]               0
================================================================
Total params: 8,522
Trainable params: 8,522
Non-trainable params: 0
```

FIGURE 4.8: Neural network summary

After training the neural network for 15 epochs with a dataset consisting of 5000 Monte Carlo simulations using Adam optimizer with a learning rate of $1e^{-5}$, the loss curve and example of no trade zone are obtained as follows

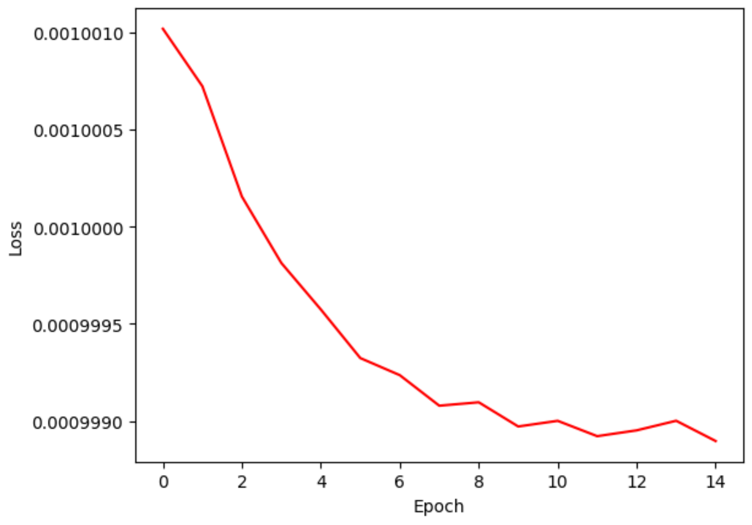


FIGURE 4.9: Loss Curve

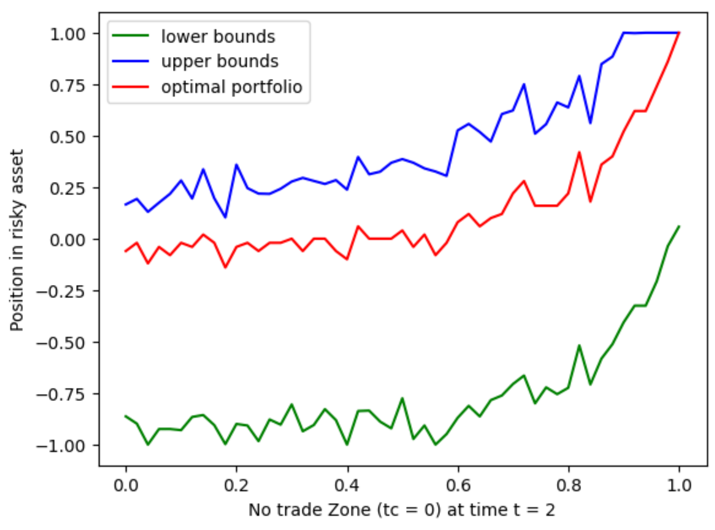


FIGURE 4.10: No Trade Zone at t = 2 weeks

Now, random paths of the market returns are generated to compare the combined method performance with the fixed mix allocation strategy and Buy and Hold strategy.

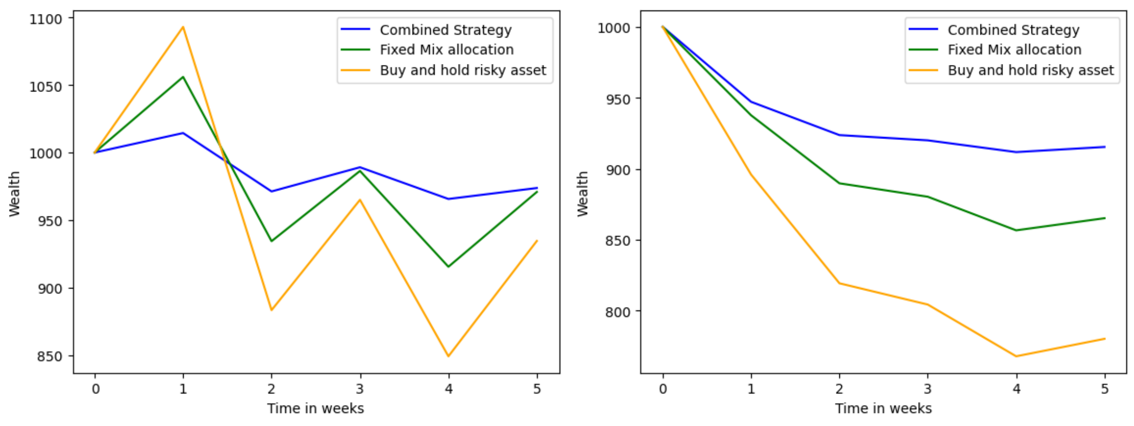


FIGURE 4.11: Combined method Performance comparison

It is observed from Figures 4.6 and 4.11 that dynamic programming and combined method algorithms perform really well when the market is in a crash regime or about to crash. This conveys that the allocation is being shifted from risky assets to risk-free asset to avoid market losses. In most cases, regime-switching behavior is superior to the other two naive strategies discussed.

Chapter 5

Conclusion and Future Work

The neural network demonstrates its efficiency in discovering a trading strategy within a market characterized by unobservable regime-switching, accompanied by linear transaction costs. This algorithm utilizes a dynamic program to identify the optimal allocation in the absence of transaction costs and then employs this solution as a starting point for a neural network to determine the no-trade zone. Notably, it effectively mitigates the challenges associated with high dimensionality as the number of risky assets in the market increases. It provides a solution that closely approximates optimality, particularly in scenarios where there are only two possible regimes and one risky asset.

Due to computational constraints, our algorithm testing has been restricted to a single risky asset within the market. However, we aspire to expand our approach to encompass multiple assets, better reflecting real-world market conditions. Additionally, we plan to adapt the algorithm to dynamic markets, where parameters evolve over time, rather than remaining static. Incorporating price increment probabilities of assets offers a promising avenue for training diverse models, which we intend to explore. Moreover, given the recent success of transformer-based architectures, particularly in processing temporal data, such as speech, we aim to investigate their potential for enhancing the algorithms.

Bibliography

Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation of probabilistic functions of a markov process.

Davis, M. H. A. and Norman, A. R. (1990). Portfolio selection with transaction costs. *Math. Oper. Res.*, 15:676–713.

Li, X. and Mulvey, J. M. (2021). Portfolio optimization under regime switching and transaction costs: Combining neural networks and dynamic programs. *INFORMS Journal on Optimization*, 3(4):398–417.

Markowitz, H. (1952). Portfolio selection*. *The Journal of Finance*, 7(1):77–91.

Merton, R. C. (1969). Lifetime portfolio selection under uncertainty: The continuous-time case. *The Review of Economics and Statistics*, 51(3):247–257.

Mulvey, J., Lu, N., and Sweemer, J. (2001). Rebalancing strategies for multi-period asset allocation. *The Journal of Wealth Management*, 4:51–58.

Mulvey, J., Sun, Y., Wang, M., and Ye, J. (2020). Optimizing a portfolio of mean-reverting assets with transaction costs via a feedforward neural network. *Quantitative Finance*, 20:1–23.