VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
MODELLING AND DATA ANALYSIS
MASTER'S STUDY PROGRAMME

Algorithmic Portfolio Management: Markov Chains Case

Master's thesis

Author: Anton Sauchanka
VU email address: anton.sauchanka@mif.stud.vu.lt
Supervisor: Assist. Prof. Saulius Jokubaitis

Vilnius

2022

Contents

Abstract

This paper develops a systematic algorithmic portfolio management framework, which consists of 4 core parts: information extraction, asset preselection, portfolio optimization and online regulation. As a case study for the proposed framework, we use Markov Chains model to extract information from the assets return series and forecast the future wealth distribution attainable by each asset. We utilize these forecasts to reduce dimensionality of the portfolio optimization problem via asset preselection using two algorithms: Data Envelopment Analysis and simple ranking algorithm of our own design. The portfolio weights are optimized by maximizing 3 semi-parametric functions of the forecasted portfolio wealth distribution, prevalent in the related literature, and a proposed extension of Omega Ratio to the Markovian, which demonstrates good performance in the empirical tests. Optimization is performed using two heuristic algorithms – Genetic Algorithm and a problem-specific simplex exploration algorithm, the latter of which consistently demonstrated better performance. Moreover, we propose and validate a simple regularization method in order to control the portfolio weight redistribution in time performed by the considered model. Within the developed framework, we were able to construct multiple end-to-end portfolio management algorithms that outperformed the benchmark S&P500 Index both under favorable market conditions and during the 2022 recession.

Keywords: Portfolio management, Markov Chains, optimization, forecasting, framework, time-series, finance.

Notation

In this work we generally stick to the following notation

- Index $t = 1, \dots, T$ is consistently reserved for identifying time step, with $h > 0$ identifying the forecasting horizon. We assume that in static perspective the decision on portfolio allocation is made at time $t = T$. We also introduce the notation of $^{(t)}$ to denote some vector or matrix at time t in order to avoid expanding dimensionality to more than 2d where possible.
- N is the number of assets available for trade.
- S is the number of states of the Markov Process.
- W_{T+h} is the discrete distribution of future portfolio wealth at time T+h, estimated by evaluating all the possible realizations of the Markov wealth process of length h. It is described by two vectors: $w^{(T+h)}$ and $p^{(T+h)}$ denoting respectively all the possible values of wealth that a given Markov process can achieve and the probabilities of achieving them.
- $x \in \mathbb{R}^N$ denotes the vector of portfolio weights allocated among assets.
- $Z \in \mathbb{R}^{N,K}$ denotes the matrix of historical returns.
- $\overline{b}, \underline{b}$ denote upper (lower) market stochastic bounds.
- τ - Kendall correlation
- ρ - Pearson correlation

Introduction

Algorithmic portfolio management can be generally defined as a field of applied research aimed at developing end-to-end algorithms for financial portfolio weights allocation among given assets. It originates from the Modern Portfolio Theory proposed by [34], who suggested that portfolio optimization is a multi-objective problem, since investors strive to maximize the expected returns, while simultaneously minimizing the associated risk. Since that time a variety of complex approaches for algorithmic portfolio management have been formulated, see e.g. [1, 18, 5]. However, despite the numerous attempts to systematize the state of the field (for instance, [27, 16, 41]), these works are limited by the nature of literature review concept to purely describing and classifying the approaches considered. At the same time, they can hardly answer the question most essential to practitioners: according to certain specific criteria what approach is the best under what conditions? Answering such questions requires a much more technical analytical framework for model comparison, which would allow directly benchmarking the approaches against each other in a similar manner as it is being done in other fields of machine learning application, for example image classification.

These considerations lead to the main inspiration behind current work, which is the creation of a framework, both conceptual and programmatic, which would incorporate the variety of currently available approaches in algorithmic portfolio management. As a theoretical foundation for the proposed framework, we suggest viewing the algorithmic portfolio management model as inherently consisting of four core subtasks, which in practical terms are typically represented as separate algorithms combined together to construct the end-to-end model. These subtasks include information extraction, asset preselection, portfolio optimization and online regulation. Information Extraction refers to any processing of the available raw data with the purpose of extraction of meaningful information about the underlying assets, which can be used for asset selection or portfolio optimization. Asset Preselection refers to explicit selection of a pool of assets to be considered for portfolio out of more wide overall market known and available to the investor or researcher. Portfolio optimization is a problem of allocating funds (or portfolio weights in more generalized form) among the selected set of assets according to some (often multiple) optimality criteria under given constraints. Finally, online regulation refers to the strategy on how the whole portfolio management algorithm is being adjusted in time given the current state of the portfolio and changing market conditions.

In order to support the argument for the development of the discussed framework, we consider the case of a Markovian portfolio management approach consistently developed by [18, 40, 4, 3, 39]. The core idea behind the approach is modeling portfolio returns as a Markov chain process, thus extracting meaningful information about their structure, which allows forecasting the future wealth distribution as described in Section 2.1. Applying this same model to individual assets allows estimating multiple desirable and undesirable statistics, which characterize asset performance and with the application of additional algorithm, such as Data Envelopment Analysis [52, 18, 15], is used to select the most perspective assets as discussed in Section 2.2. Moreover, we optimize portfolio weights by maximizing one of 4 objective functions of the Markov Chains model: MSG Sharpe Ratio, MSG Stable Ratio, MSG Pearson ration proposed in [18] and our own extension of Omega Ratio to Markovian context - MSG Omega Ratio, which was found to be the best objective during favorable market conditions. We use a problem-specific optimization heuristic presented in Section 2.3, which we demonstrate to outperform widely used Genetic Algorithm for the given problems. Finally, we develop certain top-level extensions, which are showed to regulate the overall algorithm performance in time, smoothing the sharp changes in the portfolio weights produced by the direct application of the algorithm, which might be undesirable in real-life context due to the presence of transaction costs as discussed in Section 2.4. Thus, this approach represents a good illustrative example for the proposed framework, as it incorporates all the aforementioned parts of the theoretical algorithmic portfolio management framework we propose.

Moreover, this work additionally contributes to the practical research by developing an open source Python implementation of the proposed framework (see Appendix A). With each of the subtasks essentially corresponding to a separate algorithm in the end-to-end pipeline, a module based code structure allows developing and testing a large variety of models by essentially constructing end-to-end algorithms out of several building blocks. A more advanced version of such framework might facilitate the research in the field immensely, as it would provide an environment for further experiments where researchers or just enthusiasts would not have to implement, test and analyze the whole end-to-end algorithm from scratch, but could rely on the existing infrastructure and focus their effort on making specific improvements. While there are still improvements to make as outlined in Section 4, our implementation is highly optimized, which facilitates the use of this otherwise extremely computationally expensive approach even not so powerful machines.

This paper is structured in 4 sections. In the first section, we develop the theoretical argument for the proposed framework, discuss its structure and support it by discussing the variety of methods found in literature, which typically can be attributed to one of the mentioned subtasks. In the second section, we describe the Markovian approach to portfolio management using it is as a case study to demonstrate all the constituting parts of the framework on specific algorithms. In the third section, we report the results of testing the considered approach on 2 datasets corresponding to different market conditions and discuss the empirically identified nuances. In the fourth section we summarize the paper, provide conclusions and directions for future research.

1 Algorithmic Portfolio Management Framework

In this Section we first propose a novel theoretical framework for systematic understanding of the Algorithmic Portfolio Management models as consisting of 4 primary subtasks: : Asset Preselection, Information Extraction, Portfolio Optimization and Online Regulation. After the presentation of the general idea, we provide an overview of the previous research with primary accent on the specific parts of the proposed approaches corresponding to each of the subtasks in the proposed framework.

1.1 Motivation and core idea

Portfolio management is a very broad term and many researchers have provided their definitions, which follow the same general idea, yet highlight different aspects of this problem. One definition within the field was provided by [41], who very broadly defined portfolio management as "the process of choosing various assets within the portfolio for a predetermined period" and suggested that it "covers the following closely related areas: Portfolio Optimization, Portfolio Selection, Portfolio Allocation".

The authors highlight one of the conceptual problems in the field, which is that these terms are often being used interchangeably, thus complicating the theoretical discussion. Therefore in this work we strive to clearly elaborate on the distinction between the constituting parts of the generalized portfolio management algorithm outlined in the definition. Another definition can be found in [1], who defines portfolio management as "the process of making decisions for the allocation of resources among a set of assets while satisfying financial constraints (such as cardinality, round-lot, asset class, etc.) in order to maximize the return". This definition highlights the fact that portfolio management in mathematical terms is a problem with many constraints, but simplifies the target of the problem as pure return maximization, which is typically not the only objective of the investor. This latter point is generalized further by [20], who defines portfolio management as "a process of selecting and supervising a group of financial products that meet an individual's long-term financial objectives.".

In this thesis, we propose a more structured view on algorithmic portfolio management as a dynamic process of information extraction, asset preselection, portfolio optimization and online regulation with the ultimate goal of producing a financial portfolio best tailored to the investor's preferences under given constraints at each moment in time based on available historic information. In order to justify this definition and show it's relevance, we shall consider both the practical problem that investor faces and expects to be solved by data modeling and the current state of research, to demonstrate how previously proposed methods can be unified and structured by our framework.

First, let us consider the generic, real-life problem that investor faces and then relate it to our definition and existent literature. Consider a novel investor who has some savings and wants to put them to work by forming an investment portfolio. One would typically open an account at some broker and find himself in the following environment. Modern days broker offers a possibility to purchase tens of thousands various instruments including stocks from the entire world's markets, corporate and governmental bonds, commodities, cryptocurrencies, not to mention more sophisticated financial instruments such as options, futures and swaps. There is a vast amount of diverse information available about each of these instruments, including, besides the standard price and volume, an enormous variety of technical indicators developed by generations of traders, financial statements of the companies, breaking news, analysts' reports and an occasional tweet from Elon Musk. Moreover, even if one manages to select the instruments of preference, there is only a limited amount of money to invest, which has to be split somehow among the chosen assets. Clearly, rational investor wants to get as much return as possible, but economics teaches us that there is no such thing as a free lunch in the markets. Bitcoin has been generating enormous returns for a while now, so one might get an impression that buying it is the way to get rich. On the other hand, if it will go bust, all the savings might be lost at once, so, perhaps, it might be wiser to spread the investment and buy something else as well. Now one might think that maybe it would be wise to buy as many instruments as possible, after all, chances of all of them simultaneously going bust are extremely small. But the broker also has it's stake in the game, so he is likely to charge a commission on each transaction made and put the lower limits on the quantity of each asset you may purchase, thus making full diversification costly and technically constrained. Let's say one managed to make all these choices and acquired a fair portfolio. After a month, the situation in the markets might change drastically and, suddenly, the investor finds out there is an unwinding financial crisis and half of his portfolio is under risk of huge losses according to experts.

Well, probably what is left of it has to be reallocated. And here they come again – the transaction costs.

While this motivational example is rather informal, we believe it provides a good illustration of the complex problem this paper addresses, while embedding all the significant aspects of it. Now we turn to the more formal consideration.

The most widely researched part of portfolio management within data modelling research domain is Portfolio Optimization, which is at times being called Portfolio Management, thus it is important to highlight the distinction between the two. Portfolio optimization is a problem of allocating funds (or weights in more generalized form) among a predefined set of assets according to some (often multiple) optimality criteria under given constraints. As such, it is purely a mathematical optimization problem, which was first formulated by Markowitz [34], who suggested a solution in the form of Mean-Variance Optimization model. Markowitz suggested that the optimal portfolio is the one that maximizes the returns, while simultaneously minimizes the associated risks. Since it is a multi-objective optimization problem, there exist multiple optimal solutions. The set of these optimal solutions Markowitz labelled Efficient Frontier and suggested that it is up to investor's preferences to select the particular risk-return trade-off. The problem of portfolio optimization is still challenging on both theoretical and practical levels. On theoretical level, the following areas remain open: how optimal portfolio should be defined, how risks and returns should be measured and estimated, and what real-life constraints should be incorporated in the models. [15, 45] On the practical level, portfolio optimization with multiple-objectives and various constraints is an inherently difficult problem in the first place, so various algorithms and methods are being proposed that try to approximate the optimal solution, which can be found analytically only for some non-realistic formulations of the problem. [16] A more comprehensive discussion on the nuances and approaches to portfolio optimization is presented in Section 1.2.1.

At this stage, however, it is more important to highlight that portfolio optimization is purely a problem of allocation of weights (funds). Thus, before portfolio optimization problem can and shall be solved, two other components of Portfolio Management framework have to be already in place. That is the discrete set of assets must be defined and all the inputs for optimization shall be estimated. It appears that most of the current research typically pays certain attention to the second part, even though sometimes it is being incorporated into the optimization model without the distinction, mostly neglecting the first. A very typical, though unfortunate from the practical point of view, approach is when researchers test their optimization algorithms or models on an arbitrarily selected small pool of assets, often belonging to the same class. There is a whole range of potential problems associated with this approach, including generalizability, scalability, flexibility and, ultimately, practicality. Generalizability refers to the concern that models developed for a small, manually preselected pool of assets and showing good results on it, even if the testing procedures are done with due diligence, might not work well when faced with a different pool of assets. Scalability refers to the concern that many models and methods might not scale well for certain reasons, for instance showing instability of performance when presented with a significantly larger pool of assets than initially considered or resulting in prohibitive computation time. [21] Flexibility refers to a very prominent problem that many methods can hardly incorporate the dynamic changes in the pool of assets. As a result, models become impractical for the real world where investor faces a dynamically changing pool of tens of thousands of assets belonging to different classes. In a sense, this issue can be thought of as researchers solving highly constrained optimization problems for a particular small part of the market, without acknowledging the fact.

It is important to highlight that asset selection and portfolio optimization can be viewed as the same problem from mathematical perspective. If the pool of assets for optimization includes all the available assets, then indeed any optimization algorithm will perform an implicit asset selection by assigning weights of zero to the not-selected assets. Thus, we can distinguish two types of asset selection: implicit, done during the optimization, and explicit, done prior to optimization, which is further referred to as asset preselection. [18] Another important note is that it is practically impossible to define the market in general as assets appear and disappear all the time and only some portion of them is known and available for trading to any given investor. Therefore, the notion of the overall market is conceptual, rather than deterministic. Thus, in the proposed framework and definition of portfolio management, Asset Preselection refers to explicit selection of a subset of assets to be considered for portfolio out of bigger initial set of assets known and available to the investor or researcher.

Information Extraction is another part of portfolio management that is being performed almost all the time but is rarely explicitly acknowledged by researchers. Information Extraction refers to any processing of the available raw data with the purpose of extraction of meaningful information about the assets, which can be used for asset selection or portfolio optimization. The most straightforward example is prediction of future prices of the assets in order to use them as input for portfolio optimization. Various raw inputs can be used to make such a prediction including historical prices, technical indicators, fundamental indicators or textual information, which are being somehow processed in order to estimate the future prices (new information).

The context in which these tasks are performed, according to the proposed definition, is dynamic, subject to investor preferences and constraints. It is dynamic because market conditions continuously change and, therefore, portfolio should be adjusted accordingly in time. This in-time adjustment corresponds to the fourth and last subtask in the general algorithmic portfolio management framework that we label online regulation. The primary distinguishing feature of the algorithms falling into this class is that they take into account the current state of the portfolio at the time of making a decision. Thus, online regulation refers to the process of portfolio weights adjustment in time, which takes into account the meta context of the end-to-end algorithm, such as, for example, current state and or previous history of the portfolio performance, potential portfolio allocations produced by different optimization algorithms, current transaction costs level. [27] It should be mentioned that implementation of a portfolio management algorithm that would incorporate all the new information as it appears and make corresponding decisions in real time is extremely challenging. Thus, the standard approach in the field is discretization of time domain, which corresponds the assumption that portfolio management decisions are done with a certain frequency (daily, for example). [18]

Portfolio management is subject to investor's preferences because there might be different interpretations of the optimality of the portfolio to pursue, as highlighted by Markowitz's Efficient Frontier. [34] For instance, some investors might value security over high returns and require much higher diversification then the others. Thus, the possibility to tune the optimization target is a desirable property for portfolio management system. Moreover, any portfolio management system that strives to be practical shall be able to incorporate the real world constraints, such, as transaction costs (often variable), slippage (the fact that it is not always possible to execute the transaction for a given price), limits on purchase of certain assets (either upper or lower).

Finally, it is necessary to acknowledge the directly related projects. While there were just a few attempts to formulate and implement a generalized programmatic framework for end-to-end portfolio management, it is necessary to discuss them in order to highlight the differences with the currently proposed approach and, thus, justify the value of the current thesis. The two most developed frameworks for portfolio management are presented by OLPS [26] and Qlib [57]. The earlier OLPS framework was implemented in Matlab and Octave and was focused solely on the online portfolio management strategies outlined in [27] and discussed in more details in the corresponding section devoted to portfolio optimization approaches. Judging by the contents of the project's GitHub repository, the framework did not get much attention from the practitioners and was not developed further since its initial implementation. Among clear disadvantages of the OLPS framework, also highlighted by [57], is the choice of the implementation languages, since Matlab and Octave are clearly much less widespread and, arguably, are less flexible than the mainstream Python language. Moreover, the conceptual scope of the framework is rather narrow, especially in comparison to the currently suggested one.

The most recent framework – Qlib – initially developed by the team of Microsoft researchers [57] and turned into an open-source project later on requires closer consideration. This framework is implemented in Python and is being actively maintained, developed and used by practitioners. Thus, it can be considered a state-of-the-art implementation of the end-to-end portfolio management framework. It has to be admitted that in terms of technical implementation, the current work will hardly be able to compete with the Qlib due to severe disparity in available resources. Therefore, this thesis does not pursue a goal to compete with Qlib in terms of creating a more advanced technical framework, but rather takes a somewhat different and more general approach to formulating the framework on the theoretical level. In particular, on the theoretical level, Qlib follows a purely quantitative paradigm and the standard strategy development workflow suggested by the framework is presented by a price-based return forecasting model or ensemble thereof, the predictions of which are then used to solve the optimization task. While this is the most conventional and widespread approach in automated portfolio management, the overall field of portfolio management strategies is more diverse than that. Thus, the current work strives to capture and utilize this diversity as much as possible, instead of focusing on a more efficient implementation of a particular group of approaches.

1.2 Literature review

In this section we provide an overview of the existent literature, with the primary focus on highlighting the relevant details of methodologies corresponding to specific parts of the proposed framework: asset preselection, information extraction, portfolio optimization and online regulation. Thus, this literature review pursues two primary goals. First of all, we illustrate how the existent research can be viewed through the prism of the proposed framework, thus putting it into the context. Secondly, we present and discuss the specific approaches and algorithms corresponding to the particular parts of the framework.

1.2.1 Portfolio optimization

Since portfolio optimization is the core part of the general portfolio management framework, it is necessary to consider it in more details and highlight the main approaches used in the field. Classification of such approaches is a challenging task by itself due to their number and diversity. Thus, we review them in the following manner. First, we discuss the main conceptual paradigms considering what the optimal portfolio is theoretically. Then we briefly review of the most pronounced general classes of approaches.

On the theoretical level, three main paradigms in portfolio optimization can be discerned, namely, the Markowitz's Mean-Variance, the Kelly Criterion and the Risk Parity. [45] Markowitz's paradigm originating from [34] suggests that optimal portfolio is the one providing the best possible trade-off between risk and return. Kelly Criterion suggests that the optimal portfolio is the one maximizing the wealth in the long-run and is originally based on [24]. Risk parity paradigm suggests that the optimal portfolio is the one which equalizes each assets contribution to the overall portfolio riskiness. [12] Thus, Risk Parity focuses purely on the risk minimization, Kelly's Criterion focuses primarily on return maximization and Mean-Variance takes a sort of middle ground suggesting that a balance should be achieved. [45] showed that under certain conditions, both Risk Parity and Kelly's Criterion optimal portfolios lie on the Efficient Frontier of Mean-Variance optimal portfolios, thus suggesting that the three paradigms are not mutually exclusive, but rather just focus on the different sides of the problem.

However, there is another, more soft dimension in which the aforementioned paradigms differ. [27] suggest that Mean-Variance is a static paradigm, focusing purely on a single time portfolio allocation, while Kelly's Criterion, incorporating the long-run notion of wealth maximization, is dynamic. Thus, it is important to highlight, that within our proposed framework we interpret portfolio optimization task as static, while the dynamic aspect of portfolio adjustment is time is considered as a separate, higher level task of online regulation.

We further propose the following classification of the portfolio optimization approaches based on two dimensions. First of all, we distinguish direct optimization and proxy optimization. By direct optimization we understand the mathematical models that directly estimate the optimal weight allocation among given assets from raw data. On the other hand, proxy optimization consists of two parts: the objective function of data to be optimized and the optimization algorithm itself, which is used to find the maximum (minimum) value of this objective function. Moreover, the proxy optimization methods can be further classified into analytical and heuristic. Analytical approaches require the objective function to be formulated in a particular form, for example the standard Mean-Variance optimization formulates the objective function in quadratic form, which then can be solved efficiently either by quadratic programming or using critical line algorithm [35]. On the other hand, heuristic algorithms do not impose any constraints on the form of the objective function.

First, let us consider the direct optimization algorithms. Originally, the idea of directly estimating portfolio weights as a function of asset characteristics was proposed by [9], who named their approach Parametric Portfolio Policies (PPPs). PPPs estimate the portfolio weights as a function of fundamental characteristics of the firms, such as market equity and book-to-market ratio. However, the most modern approach in direct portfolio optimization that has been rapidly developing in the recent years is deep reinforcement learning (RL). Reinforcement learning is not a strict model itself, but rather an approach to training a machine-learning model, which "aims to teach software agents how to optimally perform a series of decisions by interacting with their environment so as to maximize some notion of cumulative return" [1]. Clearly, this definition aligns well with the portfolio optimization problem, which is precisely the problem of making a series of decisions about the distribution of weights among given assets, so that the overall wealth is maximized. Thus, reinforcement learning is dynamic in nature and primarily follows the Kelly's criteria of portfolio optimality. There are 3 primary types of reinforcement learning: value-based, policy-based and hybrid. [1], [45]

The value-based approach, also known as "critic", strives to find the optimal state-action value function $Q^*(s,a)$, where s is the current state and a is the taken action, which is essentially the expected cumulative return for any action in the current state. The state-value function is typically modeled by a neural network, which is known as a Q-learning approach. [45] The actual actions taken by the trading-agents are then aim to maximize this cumulative reward functions, i.e. for each current state the agent takes an action that is expected to maximize the cumulative reward. A recent example of such approach can be found in [30], who proposed a deep Q-learning portfolio management framework, tested it in cryptocurrency market and suggested that results were promising. However, there was some critique to value-based approaches application to portfolio management. For instance, [51] suggested that Q-learning turns out to be unstable when presented with noisy data and the performance highly depends on the choice of value function, while [1] noted that maximization operation performed at each decision-making step might be computationally infeasible.

Policy-based RL, also known as "actor" models, strives to learn the policy for action selection directly. Thus, instead of estimating the cumulative-reward function $Q^*(s,a)$, which is then maximized, the policy based approach strives to estimate the policy, which can be viewed as a function $a = a(s)$, generally approximated by neural networks. In a sense, policy-based approach is more flexible since it allows the practitioner to choose the utility function used to train the model, which can be any financially meaningful measure of risk, return or both [1]. In fact, policies trained within the policy-based RL framework can be seen as a generalization of the parametric portfolio policies approach proposed by [9]. Recent example of policy-based RL application are provided by [21], who developed a policy-based RL framework for cryptocurrencies portfolio management with a policy topology labeled the Ensemble of Identical Independent Evaluators (EIIE).

Finally, the hybrid RL approaches, also known as "actor-critic", essentially combine the two paradigms presented above. In hybrid RL, both the cumulative reward function and the policy are modelled by 2 neural networks (named critic and actor, respectively) and are trained simultaneously in such a manner that the cumulative reward function learned by critic serves as a utility function for actor. In plain terms, on each iteration of the model training, the actor takes the current state as input and suggests the action based on its current policy, the critic takes the state and actor's proposed action as inputs and outputs an expected cumulative reward of the proposed action, which is then used by actor to adjust its policy. A recent example of such approach is provided by [1], who attempted combining the two classical paradigms in portfolio optimization by designing the actor to follow Kelly's criteria and critic to follow Markowitz's criteria. Moreover, the authors tested eight different policy-based learning algorithms and for the first time incorporated the cross-sectional analysis widely used in traditional finance into the RL policy. However, authors warn that while RL approaches seem to be promising, in the current state they tend to be unstable.

Despite the fact that RL-based models for portfolio optimization are being actively researched, the width of this field is enormous and is unlikely to be exhausted any time soon. It feels necessary to highlight the number of layers of flexibility that is available within RL framework, because the 3 high-level paradigms presented above are just the tip of the iceberg. Since Reinforcement Learning is a philosophy of training, the opportunities for specific RL-based portfolio optimization implementation designs are infinite, due to flexibility in choice of policy model designs, utility functions, training algorithms, representations of the market states among the most apparently components, but not limited to them.

Next, we consider the examples of proxy optimization approaches, which we further classify into two groups: analytical and heuristic. The most straightforward example of analytical optimization approach is given by the Mean-Variance Optimization itself [34]. In its classical form it requires solving the following optimization problem

$$\begin{aligned} &\min_x (x^{'}\Sigma x - \alpha\mu^{'}x) \\ \text{s. t.} &\sum_{n=1}^{N} x_n = 1 \\ &0 \le x_n \le 1 \forall n \in 1, \ldots, N \\ &\alpha \ge 0 \end{aligned} \tag{1}$$

where N is the number of assets, x is the vector of portfolio weights, Σ is the covariance matrix of asset returns, μ is the vector of expected asset returns and α is a user-specified constant typically interpreted as a risk-tolerance level. [25] Notice that this minimization problem can be solved by either quadratic programming [25] or critical line algorithm originally developed specifically for this purpose [35]. We also would like to highlight the distinction between information extraction and portfolio optimization parts of the end-to-end algorithm. The procedure of solving the optimization problem in Statement 1 is an example of portfolio optimization method, which takes covariance matrix of returns, vector of expected returns and hyperparameter α as inputs. However, there are plenty of ways that the covariance matrix and expected returns can be estimated. For example, [31] estimate the expected stock returns based on fundamental characteristics of the firms and then use these estimated returns as inputs to the classical MVO model. Such approach could be described as a 2 step process of information extraction (expected return estimation) and portfolio optimization (MVO) from the proposed framework perspective.

Another example of analytical portfolio optimization is given by [22], who reformulated the classical Omega Ratio measure of portfolio performance [23] as a linear function, which then can be optimized by the means of linear programming.

As opposed to the analytical optimization approaches, which require the objective function to have a specific form suitable for optimization, heuristic approaches are based on the empirical exploration of the function space and, thus, can be viewed as more flexible. Most often they are used to directly optimize the non-convex measures of portfolio performance, the most well known being Sharpe Ratio [48] and Omega Ratio in its initial form [23]. However, even much more sophisticated functions might be optimized, such as for example MSG Sharpe Ratio, MSG Pearson Ratio and MSG Stable Ratio, which are non-parametric, making the analytical solution very difficult to formulate. [18] These functions are described in more details in Section 2.3 of this work.

Within the subclass of heuristic optimization algorithms, we can further distinguish between problem-specific heuristics and meta-heuristics. Problem-specific algorithms are specifically tailored to the given problem context, thus they typically perform better on it. One example of such heuristic for portfolio optimization is given by [4] and is described in greater detail in Section 2.3 of this work. On the other hand, meta-heuristic algorithms are universal (i.e. not problem-specific) methods that are used to approximate the solution of a complex problem efficiently via intelligent, often nature-inspired strategy for the solution space exploration and exploitation. Thus, meta-heuristic algorithms do not necessarily find the globally optimal solution for a given problem formulation, but only approximately optimal (satisfactory) solution. The core advantage of such methods is that typically they are very flexible in terms of the formulation of the problem, thus allowing the incorporation of almost any objective function and constraints. Moreover, many meta-heuristic algorithms are able to consider several objectives simultaneously and potentially finding multiple efficient solutions, thus explicitly outputting the efficient frontier.

One of the most popular classes of meta-heuristics actively applied in portfolio optimization is Evolutionary Algorithms (EA), inspired by the process of natural selection of genes during evolution. An extensive review of the application of EAs for portfolio optimization can be found in [37], who analyzed 91 paper considering application of EA to portfolio optimization. [16] also reviewed EAs, but also highlighted another class of MHAs, which gained popularity more recently – quantum-inspired MHAs, which are based on the ideas from quantum physics. While these two classes of evolutionary and quantum meta-heuristics seem to be the most popular, the diversity of such algorithms is enormous, not to mention that they can be combined with each other and with other modelling techniques. For example, [47] tested five MHAs: particle swarm optimization (PSO), grey wolf optimization (GWO), whale optimization algorithm (WOA), Jaya algorithm (JA) and spotted hyena optimization (SHO) – and concluded that the most successful portfolios are being constructed by the hybrid algorithm integrating Jaya and spotted hyena ones labelled J-SHO. Novel algorithms are still being developed, for example, [28] proposed a novel MHA inspired by the virus spreading behavior and validated it on the portfolio optimization problem.

1.2.2 Information extraction

In the proposed portfolio management framework, information extraction refers to the task of processing the available raw data with the purpose of extraction of meaningful information about the assets, which can be used for asset selection or portfolio optimization. In essence, almost any kind of portfolio optimization requires some kind of information extraction. In the simplest form, the most traditional Markowitz' Mean-Variance optimization requires estimation of the means and covariance matrix of historical returns. On the other hand, the same optimization model can be applied to the forecasted, rather than historical returns (see e.g. [32]), which can be viewed as a much more complex procedure for the information extraction. This example illustrates the reasoning for de-coupling of information extraction and portfolio optimization tasks within the proposed framework: the same optimization models can be directly applied to different kinds of inputs and vice versa – different optimization techniques can be applied to the same inputs.

Another big role of information extraction is the possibility to utilize novel types of data for more efficient portfolio management. The clearest example nowadays is provided by the possibility to utilize textual data for assets price forecasting and portfolio optimization, made possible by the advances in the field of natural language processing. Market sentiment, which is the attitude of market participants towards specific assets, industries or market in general, has been studied for a long time within the field of Behavioral Finance. [2] However, it is only recently that modelling techniques became developed enough to automate the sentiment extraction and text-based inference, thus making the direct usage of news, analytical reports and social media feeds as a source of information in automated portfolio management possible. The topics of sentiment analysis and financial text mining are too broad and diverse for detailed discussion in the current work, so the interested readers are referred to the reviews on the subject provided, among others, by [56], [41] (sections on Financial Sentiment Analysis and Financial Text mining) and [2]. For our purposes, it is sufficient to highlight that information extraction as a part of portfolio management framework deals with the issues of how various and possibly qualitative raw information sources can be processed and utilized for asset preselection or portfolio optimization. One simplistic example of such application could be a text-based classification model that classifies market sentiment regarding particular assets as positive, neutral or negative and only assets with positive market sentiment are preselected for portfolio optimization.

Information extraction approaches can be broadly classified into two categories: forecasting and encoding. Forecasting is the most widely used type of information extraction, which deals with predicting the future characteristics of the assets, the most common choices being price return forecasting. While predicting asset price is the most natural challenge for modelling, some researchers attempt forecasting other features such as volatility or correlations between assets. We refer the reader interested in more detailed discussion of the issue to the corresponding literature reviews provided among others by [46], and [28], who specifically focus on Deep Learning methods for forecasting, and [17], who take a bit wider Machine Learning perspective. All the three recent reviews suggest that Deep Learning is actively changing the field of forecasting, with Long-Short Term Memory (LSTM) networks (a subclass of Recurrent Neural Networks (RNN)) being the most widely used modelling solution. Essentially, any kind of optimization approach can be adapted to use the forecasted features of the assets for optimization and, hence, the quality of the overall portfolio will largely depend on the accuracy of forecasting. For example, [32] tested various Machine Learning techniques for return forecasting and optimized the portfolio using Markowitz MVO model, but with forecasted returns, concluding that forecast-based portfolio optimization is superior to the one based on historical returns and identifying Random Forest as the best forecasting model in their experiments. [5] used a hybrid CNN-BiLSTM model for asset returns forecasting and adapted the classical Black-Litterman model for portfolio optimization. [53] introduced a model named Deep Responsible Investment Portfolio, which includes a Multivariate Bidirectional Long Short-Term Memory neural network predicting stock returns, and a Reinforcement Learning model optimizing portfolio weights. More classical approach was followed by [44], who proposed modelling individual asset returns with ARMA-GARCH and the joint distribution of the residuals with an Archimedean Copula model with marginals defined according to Extreme Value Theory. Subsequently, they used Monte-Carlo simulation to forecast many possible realizations of the future market and thus estimate the expectation. A unique approach was consistently developed by [40, 4, 3, 39, 18], who approximate portfolio returns with a Markov Chain and forecast the future wealth distribution by evaluating all the possible realizations of the estimated Markov Process. This approach is discussed in greater detail in Section 2.1 of this thesis.

The encoding type of information extraction strives to obtain a new representation of the separate assets or even market in general, in order to extract and encode the most essential information about the assets or market in a lower-dimensional and therefore cost-efficient representation. Thus, instead of the human-interpretable information such as future price forecast, trend classification or asset quality ranking, encoding type of information extraction obtains non-interpretable, but supposedly optimal representation of all the available information. Then, the specific portfolio optimization techniques can be trained to interpret the resulting representations and optimize the portfolio. An advantage of the encoding approach is that encoders can be constructed to combine various types of input information, such as analytical texts and price data (See e.g. [49, 10]). Moreover, it is possible to encode not only the available information about the specific assets, but also their interrelations, which is the target of Graph Neural Networks (GNNs). [10] proposed a novel high performing multimodal deep learning architecture labelled Trans-DiCE for stock price prediction, which utilizes two types of data: price data and textual news data. Trans-DiCE extracts features from price data using a variation of Convolutional Neural Network (Att-DCNN) and from news data using Event Transformer Encoder (ETE) model, then the features extracted from both data types are stacked and combined by another encoder. Using Trans-DiCE as a base for Reinforcement Learning agent, the authors effectively combined the tasks of information extraction and portfolio optimization, thus developing an end-to-end portfolio management model. [49] developed a model named DeepPocket, which is based on Graph Convolution Network, thus encoding both the separate asset features and interrelations between assets in the form of a graph, forecasting the future state of the market using graph convolution and optimizing the portfolio by the means of Reinforcement Learning.

1.2.3 Asset preselection

Asset preselection is a subtask of reducing the dimensionality of the otherwise prohibitively large space of assets available for inclusion in the portfolio. Overall, there are two fundamentally different approaches that we distinguish based on the conducted literature review: approximation and reduction. Approximation refers to modelling all the asset returns with a smaller number of factors or components. Then the portfolio weights are being assigned to these components rather than actual assets, however, one can later estimate the corresponding asset weights, which would provide the optimal component exposure. There are two primary ways these components can be estimated: fundamental (factor analysis) and data-driven (principal components analysis). Fundamental approach is based on the idea that the expected returns and risk of the firms on the stock market are dependent on and can be explained by their fundamental characteristics. This idea lies at the core of the classical works of Fama and French [13], [14] on the empirical asset pricing models, who empirically investigated and identified such factors. An overview of such approaches can be found in [31], who discusses factor-based mean variance optimization (FBMVO). Originally, FBMVO was aiming to solve one of the limitations of the Markowitz's MVO approach, which is the difficulty of estimation of the large-scale portfolio covariance matrix. To solve this issue, the expected returns could be modeled by the fundamental risk factors. Thus, instead of estimation of $N \times N$ (where N is the number of assets) covariance matrix to be used for MVO, it would be sufficient to estimate each asset's return variance, loadings on the factors and the covariance between factors. Given that the number of factors is much smaller than the number of firms considered for portfolio, this approach significantly reduces the number of estimates that need to be obtained. Following the same logic, one might take more straightforward pathway of Principal Components Analysis (PCA). Using PCA one might estimate few components explaining most of the variance in underlying asset returns and then use these component series in optimization instead of the actual asset return series. [18, 3]

The second group of approaches, which we label reduction, focuses on picking the smaller set of actual assets out of a wider pool. Such selection inherently requires some sort of measure of asset quality, which can be absolute, meaning that asset is either good or bad, or relative implying that there is a ranking of assets and several best are selected. Moreover, we may distinguish between approaches using historical, expected and statistical properties of the assets for selection. For example, [55] and [38] train several models for asset return forecasting and select the assets with best performance forecasted for portfolio optimization. Other researchers adapt Meta-Heuristic algorithms widely used for portfolio optimization to asset preselection. For instance, [58] and [43] employ the meta-heuristic Genetic Algorithm (GA) for asset preselection. In the first case, [58] use the GA to find the best combination of 4 fundamental stock characteristics that corresponds to the highest expected earnings. In the second, [43] use non-dominated sorting algorithm to rank single assets and pairs of negatively correlated assets based on their historical returns and volatility. [21] use perhaps the most simplistic method of selecting the top N assets with the largest trading volume in recent periods. [54] select the assets based on statistical tests of their returns for stationarity, normality and independence.

A combined perspective is employed by [29] and [18], who consider both historical performance of the assets and the expectation of the future wealth, estimated by modelling each asset's returns as a Markov Chain. [29] consider only desirable properties of the assets, i.e. those that rational investor would like to maximize, and select the fixed number of assets based on direct ranking. On the other hand, [18] consider both desirable and undesirable properties of the assets and use slack-based model of Data Envelopment Analysis (DEA) [52] to obtain the overall efficiency score ranging from 0 to 1 for each asset. Thus, they select only the "fully efficient" assets with the score of 1. As highlighted by [15], there is also an adjusted robust DEA version of the model, originally proposed by [42], which incorporates the uncertainty of the metric estimates.

Besides the practical task of reducing the scale of optimization problem to be solved, asset preselection might also be grounded on more solid theoretical foundations. For instance, pairs trading strategy is based on identification of fundamentally related assets, and making investment decisions based on current spread between them, rather than individual assets performance, can be considered a form of asset preselection, e.g. [33, 11]. In fact, pairs trading can be considered an example of portfolio management strategy built around asset preselection, since all the other parts of the process have to be adapted to this strategy correspondingly. Thus, it shall be viewed as a special case and not the universal and flexible tool within the proposed framework.

1.2.4 Online regulation

A separate class of portfolio management approaches was originally labeled online portfolio selection strategies [27]. In order to avoid the potential confusion, within the proposed framework we refer to these approaches as online regulation, thus they directly formulate the parametric models for weights adjustment in time, which typically take the current weights and recent performance of the assets or strategies as inputs. Hence, the core distinctive feature of these approaches is that they are only concerned with how the weights should be reallocated at each time step, rather than how the weight should be distributed initially. A large survey of such strategies was provided by [27], who identified the following four main classes of strategies: Follow-the-Winner, Follow-the-Loser, Pattern Matching and Meta-Learning.

Follow-the-Winner (FW) strategies follow the paradigm that more weight should be given to the stocks or experts performing best. The notion of "experts" here refers to the fact that the same strategies are considered applicable to the case of redistribution of weights between several base portfolios, thus creating a sort of ensemble in which the portion of total wealth is allocated between several portfolios experts strategies, which in its term allocate their portion among a set of assets. In the simplest case, however, the weights are being reallocated between assets in a single portfolio. Thus, FW strategies define the models for weights reallocation at each time step, such that more weight is given to the best-performing assets experts, but avoiding dramatic changes in portfolio composition.

Follow-the-Loser (FL) strategies follow the opposite and somewhat counter-intuitive approach of redistributing the weights to the worst performing assets. This approach is rooted in mean-reversion trading paradigm, which suggests that the assets performing good (bad) in the past are expected to perform bad (good) in the future. Essentially, mean-reversion strategy assumes some kind of stationarity in the market and tries to arbitrage the short-term deviations from the long-term mean. Alternative relatively broad strategy that is based on mean-reversion paradigm is pairs-trading, which is focused on identifying cointegrated assets and trading them in pair, thus trading the spread between them, which is expected to be stationary. More details on this approach, as well as its modern forms, can be found in [11] and [33]. Thus, FL strategies formulate the parametric models for redistribution of weights towards the assets that performed poorly in the recent past.

Unlike FW and FL, pattern matching (PM) strategies are typically non-parametric approaches that follow a 2-step process. In the first step, the price-relative set (pattern) similar to the currently observed one is identified in historical data. In the second step, the portfolio weights adjustment is performed based on the current weights and the identified similarity set. Essentially, PM strategies are based on the assumption that markets repeat themselves, thus by identifying the similar pattern in the past and knowing how the market developed further, one can optimize the current portfolio.

Finally, meta-learning algorithms (MLA) are a set of strategies directly aimed at aggregating several experts, i.e. several portfolio weights vectors proposed by different strategies, into a single global portfolio. Thus, they are directly related to the concept of ensembles, were the predictions of several independently trained models are combined in order to arrive at final prediction. Typically, such approaches increase the stability of the final predictions relative to any single base model used in the ensemble.

2 End-to-end portfolio management algorithm: Markovian approach

In this Section we briefly describe the approach consistently developed in [40, 4, 3, 39, 18], who follow the idea of modelling the portfolio returns and associated wealth as Markov Processes. Thus, at the core of this approach lies the Markovian assumption suggesting that the return of either a single asset or a portfolio of assets at time t only depends on the corresponding return at time $t-1$. Despite the fact that it is hardly possible to evaluate to what extent this assumption is realistic in a pure sense, it allows the formulation of neat mathematical modeling framework presented further, which was consistently demonstrated to yield good performance by the original authors over the recent years. Perhaps, this is due to the fact that Markov Process assumption presents a good compromise. On one hand, the returns are treated as a stochastic process in accordance with the well known Efficient Market Hypothesis, suggesting that financial markets follow a random walk. On the other hand, the assumption of some memory present in the process seems reasonable from behavioral finance perspective: people trading on the stock market clearly take the recent performance of the assets into account making their decisions and it is their choices that ultimately produce the market price.

The choice of this specific modelling approach was based on the following reasons. First of all, it incorporates all the parts of the portfolio management modeling defined within the proposed framework as described in Section 1.1. Secondly, there was no software implementation available for this rather complex approach, thus we contribute to the literature and practical research by presenting a working implementation of the model. Finally, the approach is quite unique in the general field of portfolio management, warranting for a deeper investigation, with certain extensions established in the thesis that are deemed to expand both theoretical and practical understanding of it. In the following sections, we present the approach in details following the proposed framework structure and highlighting the current implementation features.

2.1 Information extraction

In the proposed framework, information extraction part of portfolio management modeling corresponds to estimation and prediction of the certain features of the portfolio. Under the considered approach, the main feature that has to be estimated based on available data is the Markov process of future wealth, which is expected to be obtained by investing in a given portfolio. The approach is based purely on historical price data.

Define a $N \times T$ matrix Φ, consisting of historical price data, where N is the number of assets and T is the number of observations. Furthermore, define historical returns as a $N \times (K-1)$ matrix $\mathbf{Z} = (Z_{n,t})$, where $Z_{n,t} = \Phi_{n,t}/\Phi_{n,t-1}$ and the portfolio return as $z^* = x'\mathbf{Z}$, where $x \in \mathbb{R}^N$ is a vector of portfolio weights attributed to each asset. Since the concept of portfolio weights is practically just a representation of allocation of all the available funds, we assume that $\sum_{i=1}^{N} x_i = 1$. In the current modeling approach short selling is not allowed, so weights are further constrained by $x_i \in [0,1]$. We assume that the returns of either a single asset z_i or portfolio of assets z^* follow a univariate Markov Process. Such process is fully defined by $s \in \mathbb{R}^S$, $Q \in \mathbb{R}^{S\times S}$ and $u \in \mathbb{R}^S$, where S is the number of states of the process, s is the vector of these states, $Q = (q_{i,k})$ is a transition matrix of probabilities such that $q_{i,k}$ is a probability to transit from state s_i to state s_k and u is the vector of probabilities to be in each state at the start of the process.

In order to estimate the discrete states of the process, we need to discretize the observed continuous returns. Following the methodology by [29], the vector of states s can be estimated using the following algorithm.

1. Find the range of returns $r = [\max_t(z_t), \min_t(z_t)]$.

2. Divide the range of returns r into S intervals, with the boundaries given by

$$r_i = \left(\frac{\min_t(z_t)}{\max_t(z_t)}\right)^{\frac{i}{S}} \max_t(z_t), \quad i = 0, \ldots, S.$$

3. Estimate the states as geometric average of the corresponding interval boundaries,

$$s_i = \sqrt{r_i r_{i-1}}, \quad i = 1, \ldots, S.$$

Consequently, each state s_i can be represented as follows:

$$s_i = s_1 d^{i-1}, \quad \text{where} \quad d = \left(\frac{\min_t(z_t)}{\max_t(z_t)}\right)^{\frac{1}{S}}, \quad i = 1, \ldots, S. \tag{2}$$

Note, that (2) helps simplifying the computation of subsequent features. Next, the elements of the transition matrix Q are estimated as

$$\begin{aligned} q_{i,k} &= \mathbb{P}\big(z_{t+1} \in (r_k, r_{k-1}) \mid z_t \in (r_i, r_{i-1})\big), \\ &= \frac{\sum_{t=1}^{T-1} \mathbb{1}\big(z_{t+1} \in (r_k, r_{k-1}) \mid z_t \in (r_i, r_{i-1})\big)}{\sum_{t=1}^{T-1} \mathbb{1}\big(z_t \in (r_i, r_{i-1})\big)}, \quad i, k = 1, \ldots, S. \end{aligned} \tag{3}$$

where $\mathbb{1}(A)$ denotes an indicator function for an event A.

The situation is interesting with the initial probabilities u. [29] suggest taking them as unconditional probabilities to be in each state, which can then be estimated as

$$u_i = T^{-1} \sum_{t=1}^{T} \mathbb{1}\big(z_t \in (r_i, r_{i-1})\big), \quad i = 1, \ldots, S. \tag{4}$$

While this proposition seems completely reasonable for the purely theoretical consideration of Markov processes, for our particular use case it seems somewhat counterintuitive. Since ultimately our goal is to estimate the distribution of the future wealth generated by this process at time $T + h$, $h > 0$, assuming that we are currently at time T and we use the $1, \ldots, T$ returns to estimate this process. Thus, we know the state at previous step $s_i^{(T)}$, notice that we use the $^{(T)}$ style notation to represent the otherwise implicit time dimension, but do not know the initial future state $s^{(T+1)}$ and the vector u is supposed to represent the probabilities $u_k = \mathbb{P}(s^{T+1} = s_k)$. Hence, it seems much more intuitive to estimate u as $u_k = q_{i,k}, \forall k \in 1, ..., S$, in other words as transition probabilities from the last known state.

The estimated parameters of $s \in \mathbb{R}^S$, $Q \in \mathbb{R}^{S\times S}$ and $u \in \mathbb{R}^S$ thus fully describe the Markov process of a portfolio return. Based on it, we can estimate the distribution of wealth at time $T+h$, in the future that a portfolio can reach. For any $t = T, \ldots, T+h$, the wealth process can be described as $W_{t+1} = s_i W_t$, $i = 1, \ldots, S$, where s_i is the state of the return process at time t and W_t is portfolio wealth at time t. This definition has a very important implication for the Markovian case, as it suggests that attainable future wealth at time $T+h$ corresponds to the unique combination of the states traversed by the Markov process of the return of length h. In other words, the order of the states does not matter for final wealth, only which states were visited. Assuming that the wealth process starts from $W_T = 1$, there are $(S-1)h+1$ possible wealth values that can be reached in h steps, where S is the number of states in the return process. This follows from the fact that W_{t+1} can take S possible values given by $s_i W_t$, $i = 1, \ldots, S$ and the recombination effect of the Markov chain. Moreover, given the states definition provided above, these wealth values can be computed as $w_i^{(t)} = s_1^t d^{1-i}$, where $w^{(t)}$ is a vector of all the possible wealth values the process can reach at time t > T. The probabilities of attaining each of these wealth can be computed using the computational scheme suggested by [29], via a sequence of matrices

$$G^{(t)} \quad = \quad (g_{i,j}^{(t)})_{1\le i\le (S-1)t+1, 1\le j\le S}, \quad t = T, \ldots, T+h,$$

where $g_{i,j}^{(t)}$ is a probability to attain the wealth i by time t and in the state j. This sequence of matrices can then be computed recursively, starting from $G^{(T+1)} = u$, as $G^{(t)} = \text{diagM}(G^{t-1}Q)$. diagM is a an operation of shifting each column of any given matrix $A \in R^{(m,n)}$ down by j-1 rows, where $j \in 1, \ldots, n$, thus transforming A into $\tilde{A} \in R^{(m+n-1,n)}$, where all the non-zero elements on the left from main diagonal, as described by [40]. Finally, the vector of probabilities to attain the wealth values $w^{(t)}$ is given by $p^{(t)} = G^{(t)} I_S$, where I_S is a unit vector. Thus, we define W_{T+h} as a portfolio wealth distribution at time $T+h$ in the future considering Markovian behavior of portfolio returns and it is fully described by $w^{(T+h)}$ vector of wealth values and $p^{(T+h)}$ vector of probabilities to attain them. From it we can compute multiple statistics described further, which can be used for asset preselection and portfolio optimization, thus we refer to this particular algorithm of estimating the Markovian distribution of future wealth as information extraction. Furthermore, the approach introduced here may be extended to bivariate case, in order to evaluate the joint behavior of two wealth processes, as was originally proposed in [3] and is described in Appendix B.

2.2 Asset preselection

In the current research we consider the general class of asset preselection algorithms based on comparative evaluation of the assets relative to several statistics, which can be viewed as either desirable or undesirable. We first formulate the abstract approach and then provide the details on specific algorithms and measures used for evaluation.

The general problem of asset preselection can be stated as follows: given a (typically) large set $\mathcal{I}$, containing the available assets, select a subset $\mathcal{J} \subset \mathcal{I}$. Throughout the Section we assume that $|\mathcal{I}| = N$, $|\mathcal{J}| = M$, $M \le N$. As an input for such preselection we use the $N \times K - 1$ matrix $\mathbf{Z}$, where K is the number of historical price observations, consisting of historical returns, although without the loss of generality the abstract version of the approach formulated here can be applied to any data representing the assets as long as some statistics can be formulated for each asset. For example, some technical indicators used by traders use not only daily closing prices, but also consider the volume of trade and daily ranges of prices. We then evaluate the individual assets by computing a series of desirable statistics (throughout the Section denoted by $\mathcal{H}$) and a set of undesirable statistics ($\mathcal{L}$), where the term desirable (undesirable) refers to the fact whether rational investor would want to maximize (minimize) the given statistic of asset data. We denote these statistics by $l_{n,i}(z_n) \forall n \in 1, ..., N, i \in 1, ..., L$ for undesirable and $h_{n,j}(z_n) \forall n \in 1, ..., N, j \in 1, ..., H$ for desirable. Since we are not limiting our analysis to a single measure of asset goodness, an algorithm is needed for comparative evaluation of the assets based on the estimated statistics. In the current work, we consider two such algorithms: a Ranking algorithm, proposed in Section 2.2.1 and Slack-based Data Envelopment Analysis (SBM DEA), originally proposed by [52] and applied to asset preselection by [18].

2.2.1 Ranking algorithm

The main idea behind our ranking algorithm is to simply combine all the statistics computed into a single score, by which the assets can be appropriately ordered, with only the and top $q \leq N$ being preselected. Note, that various statistics can have very different scales and that some of them are desirable and some are undesirable. For this reason, we standardize all the statistics and compute the final score $\mathrm{A_n}$ as

$$\begin{aligned}
\tilde{l}_{n,i} &= \frac{l_{n,i} - \hat{\mu}(l_{n,i})}{\hat{\sigma}(l_{n,i})} \forall n \in 1, \ldots, N, i \in 1, \ldots, L, \\
\tilde{h}_{n,j} &= \frac{h_{n,j} - \hat{\mu}(h_{n,j})}{\hat{\sigma}(h_{n,j})} \forall n \in 1, \ldots, N, j \in 1, \ldots, H \\
\mathrm{A_n} &= \sum_{i=1}^{H} \tilde{h}_{n,i} - \sum_{i=1}^{L} \tilde{l}_{n,i}
\end{aligned} \tag{5}$$

Finally, we rank the assets based on the final score and pick the top M as our final preselection.

While this algorithm is really simple, it is introduced to investigate whether the much more complex and computationally intensive DEA approach presented further provides a substantial improvement over it. Thus, we propose it as a sort of preselection benchmark.

2.2.2 Slack-based Data Envelopment Analysis

The SBM DEA model used in the current work for asset preselection is defined as follows for a single asset $n = 1, \dots, N$.

$$\begin{aligned}
& \min \left(\phi_n = q - \frac{1}{L} \sum_{i=1}^{L} \frac{s_i^-}{l_{n,i}} \right), \\
\text{s. t.} \quad & q + \frac{1}{H} \sum_{j=1}^{H} \frac{s_h^+}{h_{n,j}} = 1, \\
& \sum_{n=1}^{N} \delta_n h_{n,j} - s_j^+ \;=\; q h_{n,j}, \quad j = 1, \dots, H, \\
& \sum_{n=1}^{N} \delta_n l_{n,i} + s_i^- \;=\; q l_{n,i}, \quad i = 1, \dots, L, \\
& \delta \geq 0, \quad s^- \geq 0, \quad s^+ \geq 0, \quad q > 0.
\end{aligned} \tag{6}$$

Slacks s_j^- and s_i^+ represent the excess of inputs (undesirable statistics) and shortage of outputs (desirable statistics) respectively.q is a non-zero constant introduced by [52] in order to linearize the optimization problem. The asset is deemed efficient if $\phi_n^* = 1$, which corresponds to $s_j^- = 0$ and $s_i^+ = 0$, in other words when it has no excess in undesirable statistics and no shortage of desirable.

It is important to highlight that in this work we use the original version of SBM DEA as formulated in [52] and not the modified one from [18]. This is done for two primary reasons. First of all, the formulation in [18] according to the authors is fully based on [52], however, we noticed few differences, for which no justification was provided. For instance, one may notice, that in [52], statement 9, which formulates the LP form, the summation is done over input indices in the minimization operation. At the same time, following their definitions, [18], statement 1, sum over output indices in the same minimization operation. The second reason is purely technical. As the modified formulation would require full implementation to be done from scratch, it was assessed as infeasible within the scope of the current work, especially given that the original implementation would need to be not only correct, but also highly performant due to the nature of the end-to-end algorithm, which requires it to be evaluated for hundreds of assets multiple times. Therefore, from the practical side, we use the implementation provided in "additiveDEA"[1] R package, which is based on [52] and optimized for performance.

2.2.3 Statistical measures

For the preselection of assets we employ a set of statistical measures as suggested by [18]. They can be split into 2 categories: historical (HS) and dynamic (MSG). The former are estimated directly from the historical asset returns, while the latter are estimated by considering the Markovian behavior of asset returns as described in Section 2.1. In particular, we consider the following undesirable statistics, that rational investor would like to minimize:

- Standard deviation of the asset returns, denoted by $\hat{\sigma}(z_n)$

[1]https://cran.r-project.org/package=additiveDEA

- Kendall correlation between asset returns and market lower stochastic bound, denoted by $\tau^l(z_n, \underline{b})$, where $\underline{b}_t = \min_n(Z_{:,t}) \forall t \in 1, ..., T$ and τ denotes Kendall correlation.
- Standard deviation of future wealth at time T, $\hat{\sigma}(W_{T+h}(z_n))$, where W_{T+h} is the future wealth distribution estimates as discussed in Section 2.1.
- Expected amount of time for investor to reach certain level of wealth ($win \geq 1$) before time T, $\mathbb{E}\left(\pi_{win}(z_n)\right)$, where π is a distribution of time it takes portfolio wealth to reach certain level under the Markovian assumption estimated as discussed in Appendix C. Naturally, rational investor would like to minimize the time to realize gains. In the current work, we use expectation of time to first gain the wealth of 1.05.
- Conditional Value at Risk (CVaR) of the estimated alpha-stable distribution of the future log wealth at time $T + h$ – $CVaR_{0.05}(\ln(W_{T+h}(z_n)))$. The precise formulation of $CVaR$ is provided in Appendix D.

Moreover, we consider the following desirable statistics that rational investor would like to maximize.

- Cumulative wealth obtained over the last K periods - $\prod_{t=T-K}^{T} z_{n,t}$
- Empirical mean of the asset returns - $\frac{\sum_{t=1}^{T} z_{n,t}}{T}$
- Kendall correlation between asset returns and market upper stochastic bound - $\tau^u(z_n, \bar{b})$, where $\bar{b}_t = max_n(Z_{:,t}) \forall t \in 1, ..., T$
- Expected future wealth at time $T + h$, $\mathbb{E}\left(W_{T+h}(z_n)\right)$
- Expected square root utility function of future wealth at time T+h - $E(2\sqrt{W_{T+h}(z_n)})$
- Location parameter $\delta_{W_{T+h}(z_n)}$ of the alpha-stable distribution, estimated for future wealth. *(desirable)*
- Expected amount of time for investor to lose certain level of wealth $loss \leq 1$ before time T - $E(\pi_{loss}(z_n))$. In the current work, we use the value of $loss = 0.95$.

2.3 Portfolio optimization

Portfolio optimization is a stage in the algorithm evaluation where we allocate the weights among the assets earlier preselected by one of the algorithms described in the previous subsection. In general form it might be formulated as follows.

$$\max_x f(x), \tag{7}$$

$$\begin{aligned} \text{s. t.} \quad & \sum_{n=1}^{N} x_n = 1, \\ & 0 \leq x_n \leq 1, \quad n \in 1, ..., N. \end{aligned} \tag{8}$$

Where $f(x)$ is some objective function of the portfolio weights x, with several examples used in the current work described further. Portfolio weight allocation is thus a constrained optimization problem, with the two basic constraints being non-negativity of weights and the total wealth available for distribution equal to 1. The non-negativity constraint can be relaxed, which would then implicitly assume allowing short-selling, when one essentially bets on the drop of asset price. In this work we stick to the standard case with no short sales.

2.3.1 Objective functions

In this thesis we consider four functions $f(x)$ in (7), each one representing a separate trading strategy and objective. We consider the 3 objective functions presented in [18], namely the MSG Sharpe Ratio, MSG Stable Ratio and MSG Pearson Ratio, and extend this list by proposing the MSG Omega Ratio statistic, which is essentially an adaptation of the well known financial performance measure Omega Ratio to Markovian context. Notice that we follow the [18] manner of naming the functions with MSG in the beginning, in order to highlight the fact that we refer specifically Markovian versions and to maintain terminology comparability with previous research.

MSG Sharpe Ratio is an extension of the classical risk measure – Sharpe Ratio – originally proposed in [48] to the Markovian case. It is defined as follows:

$$\text{MSG Sharpe Ratio} \quad = \quad \frac{\mathbb{E}\,(W_{T+h}) - 1}{\sigma(W_{T+h})}. \tag{9}$$

The MSG Stable Ratio of the portfolio is given by:

$$\text{MSG Stable Ratio} = \frac{\delta(W_{T+h})}{CVaR_{0.05}(W_{T+h} - E(W_{T+h})) + 1} \tag{10}$$

Where $\delta(W_{T+h})$ is the location parameter of stable distribution, which is considered the best parametric approximation of the W_{T+h} discrete distribution of future wealth. $CVaR_{0.05}(W_{T+h} - E(W_{T+h}))$ is respectively the Conditional Value at Risk of the same stable distribution. Note that it is being calculated from $W_{T+h} - E(W_{T+h})$, which refers to the same stable distribution of W_{T+h} with 0 mean. More details regarding the procedure of estimating the stable distribution parameters and CVaR are provided in Appendix D. MSG Pearson Ratio is defined as:

$$\text{MSG Pearson Ratio} = \frac{1 + \rho(W_{T+h}, \overline{b})}{1 + \rho(W_{T+h}, \underline{b})} \tag{11}$$

Where $\rho(W_{T+h}, \overline{b})$ and $\rho(W_T, \underline{b})$ are Pearson correlations between the portfolio wealth process and upper ($\overline{b}$) and lower ($\underline{b}$) market stochastic bounds processes. In order to estimate these correlations, we need to consider the joint behavior of the 2 processes. This is being done by estimating the bivariate Markov process, the procedure for which is outlined in Appendix B. Finally, we also consider the extension of the Omega Ratio (OR) to Markovian case. Classical Omega Ratio was originally proposed in [23] as an alternative to Sharpe Ratio, since by definition Sharpe Ratio takes into account only the first 2 moments of the return (wealth in Markovian case) distribution, while Omega Ratio by definition uses all the moments. The MSG Omega Ratio is then given as follows:

$$\text{MSG Omega Ratio} = \frac{\int_{\theta}^{\infty}(1 - F_{W_{T+h}}(x))\mathrm{d}x}{\int_{-\infty}^{\theta} F_{W_T}(x)\mathrm{d}x} \tag{12}$$

Where $F_{W_{T+h}}$ is a cumulative density function of the stable distribution best approximating the distribution of W_{T+h} and θ by definition is a threshold parameter, which in classical case represents the required return level and in MSG case represents the required future wealth level. The formulation of the stable distribution CDF used in the current work is based on [8] and is outlined in Appendix D. It should be mentioned that for the classical Omega Ratio, which is a statistic of portfolio returns, there was developed a linear form, which is more suitable for optimization [22], however, this formulation cannot be directly adapted to the MSG case as it utilizes the property of portfolio return being a linear function of individual asset returns, which cannot be directly extrapolated to MSG case. Therefore, we use the standard form presented above, which is being estimated using numeric integration.

The role of θ is interesting in this adaptation and shall be discussed separately. First of all, notice that since the denominator of OR is the integral from -infinity to θ of the cumulative distribution function, it tends to 0 when $\theta \leq \mu$ and decreases with μ being the mean of the corresponding stable distribution, which might create computational issues as OR then goes to infinity. In the classical OR θ is a hyperparameter specified by the user. Since in the classical case OR is a function of portfolio returns, the expectation of which varies rather modestly with portfolio weights, it is not that problematic to set it even for OR optimization. On the other hand, future wealth expectation might vary dramatically with portfolio weights. Consider a simplistic example to illustrate this idea. Even without the use of Markov chain for future wealth forecasting, the most naive way to form expectations about it is given by

$$\mathbb{E}\left(\hat{W}_{T+h}\right) = \hat{\mu}(\tilde{x}'Z)^h. \tag{13}$$

Where $E(\hat{W_{T+h}})$ is the expected future wealth at time T+h and $\hat{\mu}(\tilde{x}'Z)$ is the empirical mean of equally-weighted portfolio returns. This holds since starting from $\hat{W}_0 = 1$, future wealth is essentially a cumulative product of returns at each time step. Therefore, the first thing that we should account for is that the magnitude of future wealth depends on forecasting horizon h and so shall our required level of it – $\theta = \theta(h)$. Even then forming an expectation remains problematic for the user, as the space of future wealth distributions attainable by different portfolios is vast. Therefore, we use the following heuristic to set it.

$$\begin{aligned} &\theta(h) = (\tilde{\theta}\hat{\mu}(\tilde{x}'Z))^h, \\ &\tilde{x}_n = \frac{1}{N}, \text{ for } n \in 1, ..., N. \end{aligned} \tag{14}$$

We still leave a hyperparameter notion of $\tilde{\theta}$, which shall be specified by user. But now it becomes essentially a scale on the equally weighted portfolio return, which is taken to the power of h in order to account for forecasting horizon. We believe that this hyperparameter is more easy to set, as the meaningful values naturally lie around 1. Moreover, thus we protect the optimization from the numerical issues associated with potential division by 0.

2.3.2 Simplex optimization heuristic

It was suggested by [18, 4] that standard out-of-the box optimization algorithms do not perform very well with the MSG functions and our empirical experiments confirm it. Thus, in order to optimize the given functions we use the following heuristic algorithm described in [4]. Given the objective function $f(x)$, where w is an N-dimensional vector of weights, and initial feasible solution x_{init}, the algorithm tries to find the best solution x_{opt}, by iteratively increasing and decreasing the value of a single weight and adjusting all the other weights accordingly. Initially we set $x_{opt} = x_{init}$ and the improvement by increasing is performed first. For each asset $i \in 1, ..., N$, M - 1 alternative portfolios are evaluated, defined as

$$
\begin{aligned}
x^*_m &= (1-\gamma_m)x_{opt} + \gamma_m e^i \\
e^i_n &= \begin{cases} 0, \text{ if } n \neq i \\ 1, \text{ if } n = i \end{cases} \\
\gamma_m &= (\frac{m}{M})^p \forall m \in 1...M-1
\end{aligned} \tag{15}
$$

If a new optimum is found on a search direction i, the current optimum weights are being updated by $x_{opt} = x^*_m$. Note that optimization takes two parameters M and p. M defines how many points on the interval $(x^{opt}_i, 1)$ are evaluated, while p defines the distribution of these points: larger p means that the evaluated points will concentrate closer to the x^{opt}_i value. If for any i better solution is found - it becomes the new optimum, and the subsequent evaluation starts from there. If increasing evaluation manages to improve the value of $f(x)$, the algorithm tries to further improve it by decreasing the weight of a single asset. Again, for each weight $i \in 1, ..., N$, M - 1 alternative portfolios are evaluated, defined as

$$
\begin{aligned}
w^*_x &= (1-\gamma_m)x_{opt} + \gamma_x d^i \\
d^i &= \frac{(x - x_i e^i)}{1 - x_i} \\
e^i_n &= \begin{cases} 0, \text{ if } n \neq i \\ 1, \text{ if } n = i \end{cases} \\
\gamma_m &= (\frac{m}{M})^p \forall m \in 1...M-1 \\
0 &< x_i < 1
\end{aligned} \tag{16}
$$

If the evaluation by decreasing manages to find a new optimum - the increasing is tried again. The process is repeated until no improvement is made. It should be mentioned that this formulation of the algorithm slightly differs from [4] in a way, that we evaluate M - 1 alternative portfolios on each direction, instead of M as technically proposed in [4]. We suppose that the original formulation was somewhat incorrect, since the Mth alternative portfolio is by definition the portfolio with i'th weight equal to 1. If it is to be found optimal by the increasing evaluation, than it would become the initial one for evaluation by decreasing. However, the evaluation by decreasing includes a constraint of $x_i < 1$, as for $x_i = 1$ the proposed formulation of d^i yields division by 0. Moreover, a portfolio of a single asset is undesirable from the realistic point of view, as it provides no diversification of risks. Thus, we resolve this inconsistency of the original formulation by simply excluding the Mth portfolio from the search space.

2.4 Online regulation

The final part of the end-to-end algorithm according to the proposed framework is online regulation, which deals with how portfolio is being adjusted in time. The original approach features one notion, which we interpret as an online regulation part of the algorithm. Since all the MSG statistics are dependent on the forecasting horizon h, which corresponds to the number of steps in the future for which we forecast wealth distribution, it is suggested in [18, 4, 40, 3] that the full portfolio selection cycle, meaning asset preselection and subsequent optimization, shall be performed only each h'th step respectively. In the mean time, at steps $t \mod h \neq 0$, the weights should be maintained constant. By maintenance of weights, we refer to the fact that the real weights in the portfolio are naturally changing in time as underlying asset prices change. Consider the simplified example, where there are only 2 assets and at step t_0 we assign equal 0.5 weights to them. Suppose that the realized returns at step t_0 were $z_{1,t_0} = 2$ and $z_{2,t_0} = 0.5$, meaning that the price of the first asset increased 2-fold and of the second decreased. Then, at the beginning of step t_1:

$$
\begin{aligned}
W_{t_1} &= 0.5 z_{1,t_0} + 0.5 z_{2,t_0} = 1.25, \\
\tilde{x}_{1,t_1} &= \frac{0.5 z_{1,t_0}}{W_{t_1}} = 0.8, \\
\tilde{x}_{2,t_1} &= \frac{0.5 z_{2,t_0}}{W_{t_1}} = 0.2
\end{aligned}
\tag{17}
$$

Therefore, in order to set equal weights again for step t_1, we need to redistribute 0.3 of our wealth, which becomes important and shall be kept in mind if one wants to account for transaction costs. If transaction costs are not accounted for in the model, the notion of maintaining weights simply implies that the adjustment presented above shall not be performed.

Besides the simplistic online regulation approach used in the original approach formulations, which essentially implies not applying the otherwise necessary adjustment, we propose two other algorithms, which pursue the goal of regularizing the portfolio allocation in time. [18] found that the weight turnover, defined as a portion of portfolio weights being changed between redistribution was rather high for the proposed method. High turnover in general is undesirable, as in real-life environment it would yield large transaction costs, which might outweight the gains provided by more efficient distribution at a given time. We shall further distinguish between two potential sources of the turnover, the first being asset preselection, where the overall set of assets changes and portfolio optimization turnover, where the weights are being redistributed between assets.

2.4.1 MSG optimization regulation (R1)

The first regularization method we propose is associated with the very nature of the Markovian approach described in this work. The penalty for weight redistribution can be integrated in the proposed approach by adjusting the starting wealth for the forecasting. By default, for the estimation of future wealth distribution, the starting wealth is always taken as $W_T = 1$. However, very naturally, we can consider the notion of transaction costs as a penalty on the redistribution of weights. Suppose we have two vectors of weights, x_c and $\tilde{x}$, where x_c are current weights at the beginning of the period and $\tilde{x}$ are new weights that are considered by the optimizer. The turnover (TO), representing what portion of wealth is being redistributed, is then given by:

$$\kappa_{\text{TO}} \quad = \quad \frac{\sum |x_c - \tilde{x}|}{2}. \tag{18}$$

Assuming that transaction costs take a form of certain % commission on either sale or purchase of assets, given by commission rate c the portion of wealth we would lose by making the considered redistribution is given by $c\kappa_{\text{TO}}$. Then, we can modify the starting wealth value used for evaluation of future wealth distribution as $W_T = 1 - c\kappa_{\text{TO}}$, since the default 1 is essentially the base multiplier representing "full" wealth whatever it is, now we adjust it so that we incorporate the expected initial loss from redistribution. This would ultimately result in future wealth distribution mean being a function of turnover, thus for higher turnover the values of all the objective functions that are based on expectation of W_{T+h}, including MSG Sharpe Ratio, MSG Omega Ratio and MSG Stable Ratio will be lower. Note that such modification will have no effect on MSG Pearson Ratio, which is purely based on concordance between portfolio wealth and marker stochastic bounds and is thus independent of the mean of the future wealth. Moreover, this method does not anyhow affect the turnover coming from asset preselection, as it's impact is limited to optimization. Nonetheless, we find it interesting and appealing for it very natural interpretation.

2.4.2 General purpose regulation (R2)

The second regularization method we consider is more simple and universal, as it is not directly associated with the Markovian approach, but can be used as a general online regulation tool. It also demonstrates more straightforwardly the whole idea behind the class of methods that we call online regulation. Instead of taking the new weights found by the whole portfolio allocation procedure, we take some function of the current weights that we have in portfolio and the new ones proposed by the algorithm. The simplest function to use in this case is weighted average. Thus, we can define the final weights allocated at each redistribution step as

$$x_t = \nu x_{t-1} + (1 - \nu)x_t^* \tag{19}$$

Where ν is the weight parameter controlling how much redistribution we allow and x_t^* are weights proposed by the allocation procedure. Notice that this approach handles both sources of the turnover, as weights for the assets that were not selected for portfolio have weight 0 in x_t^* and vice verse - the newly selected assets have weight 0 in x_{t-1}.

3 Empirical results

In this Section we present and discuss the empirical results of testing the current implementation of the proposed framework on real life data. First, we introduce the general set up for experiments, then provide some technical analysis for each of the end-to-end algorithm constituents, finally we close this Section by reporting and analyzing the best strategy configurations discovered so far.

3.1 Experimental setup

For the empirical tests, we use the dataset covering stock price data on S&P500[2] constituents obtained from open source. This dataset covers 10 years of daily price history for 500 stocks included in the S&P500 index, which features 500 leading U.S. publicly traded companies, with a primary emphasis on market capitalization. It is one of the most widely used benchmarks in the investment field and a common choice for testing purposes, e.g. [18, 11, 55].

Due to computational complexity of the considered algorithms, the long-term testing is extremely time-consuming, therefore we select 2 specific subsets of 1 calendar year or 252 trading days each for the tests. The first subset, labeled Growth, covers the year of 2016, when the global economic environment was stable and S&P500 index was growing steadily as shown in Figure 1. The second subset, labeled Recession, covers the current year from 2021-12-09 to 2022-12-09, during which there was a substantial drop in S&P500 index valuation, as shown in Figure 2, and the global economy was in the state of turbulence caused by pandemic of Covid-19 and Russian invasion to Ukraine. Thus, we try to grasp some insights on how different models behave under different market conditions.

The standard algorithm back testing pipeline consists of the following steps, which are repeated for each trading day in the testing set:

- Take some period of historical returns prior to assumed decision making time as training set. In the current work we use 5 calendar years or approximately 1260 trading days.
- Fit the information extraction model on the training data.
- Select the initial subset of assets to be considered for portfolio using preselection algorithm.
- Allocate weights among the preselected assets using optimization algorithm.
- Evaluate portfolio performance based on the asset returns in the next day not included in the training set.
- Shift the training set by 1 trading day into the future.
- Apply online regulation rules to get the new weights.

Recall from Section 2.4 that Online Regulation is the part of the algorithm that controls strategy behavior in time. In the simplest case applied if not specified otherwise, it simply tracks the number of algorithm evaluations (time steps) and initiates the new fit–preselect–optimize cycle each k steps. By default, we use k=5, meaning that full portfolio redistribution is performed each 5 trading days or weekly and use the same forecasting horizon for Markovian information extraction model. Moreover, we use $S = 9$ states for the Markov model estimation, same as in [18], and MSG Omega Ratio $\tilde{\theta} = 1.01$, unless specified otherwise. The hyperparameters of the metrics used for preselection are set as defined in Section 2.2.

[2] https://www.kaggle.com/datasets/andrewmvd/sp-500-stocks

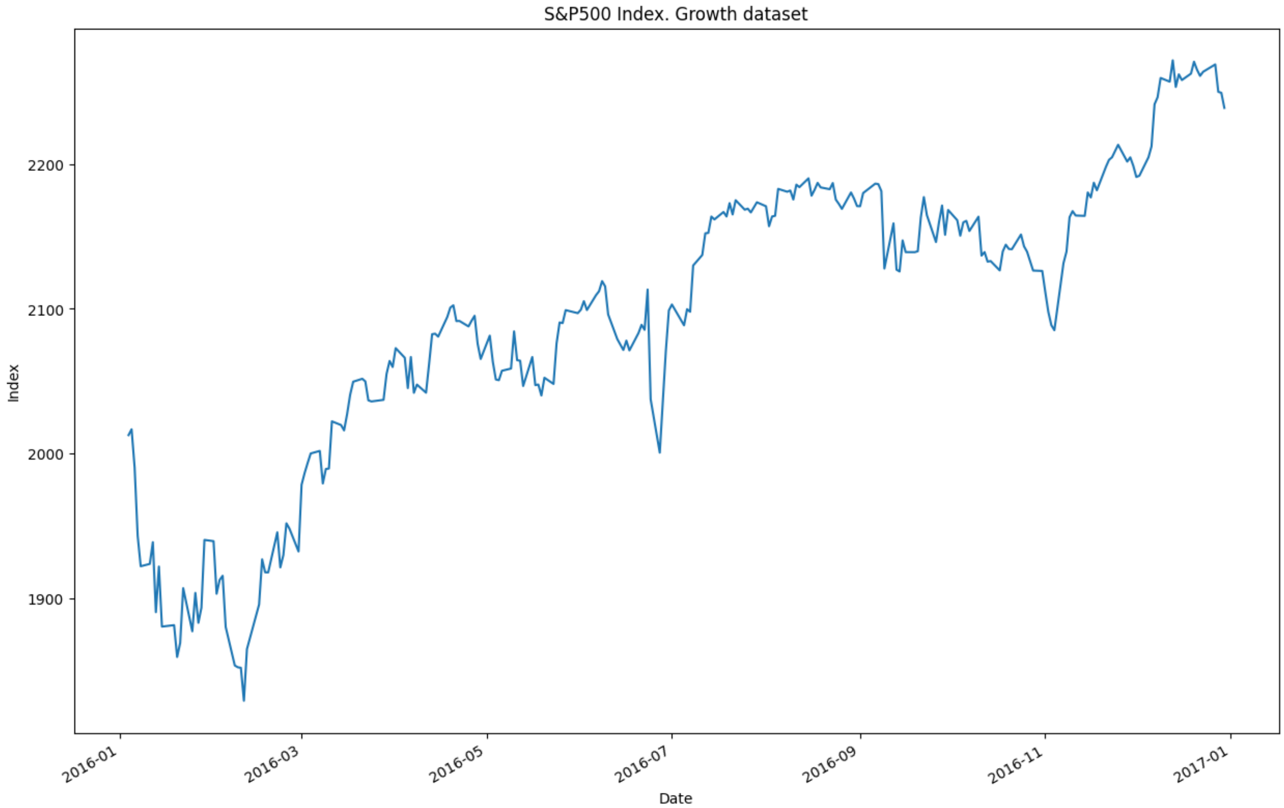

Figure 1: S&P500 Index performance on the Growth dataset

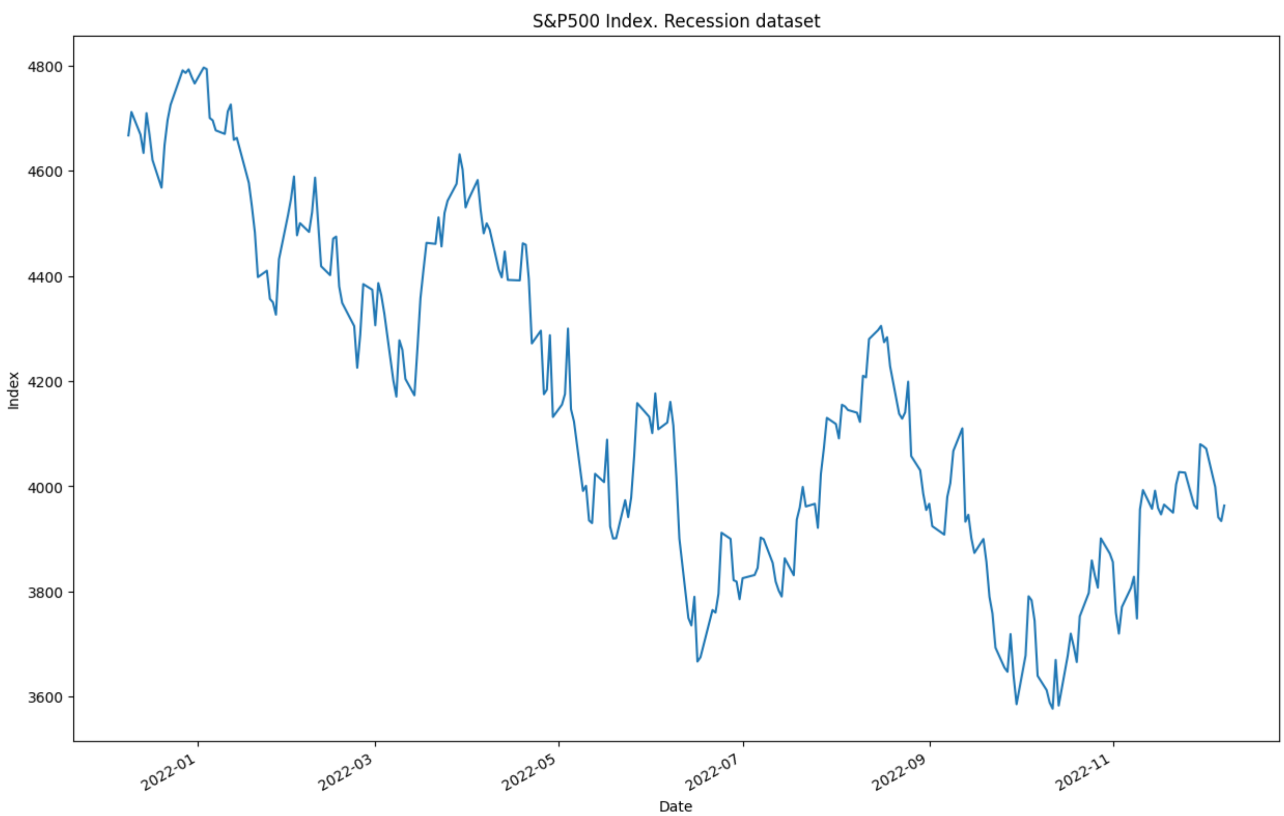

Figure 2: S&P500 Index performance on the Recession dataset

3.2 Strategy evaluation

We evaluate the performance of the tested strategies using multiple metrics characterizing different sides of the strategy performance. Notice that in order to distinguish between theoretical and empirical variables, we stick to $\hat{}$ notation in this Section. Thus $\hat{x}_t$ is the vector of weights allocated by the algorithm for step t, $\hat{z}_t$ is the vector of observed asset returns at step t, $\hat{x}_t^{'}\hat{z}_t$ is the realized portfolio return, $\hat{W}_t$ is wealth achieved by portfolio at step t and $t \in [1, \ldots, T]$ is the evaluation step. Return metrics quantify the positive side of portfolio performance, i.e. how much return and wealth the strategy managed to generate over the testing period.

- Mean portfolio return - empirical average of daily returns realized by the strategy on the testing data.

$$\hat{\mu}(\hat{x}^{'}\hat{z}) = \frac{\sum_{t=1}^{T}\hat{x}_t^{'}\hat{z}_t}{T}. \tag{20}$$

- Final wealth - the cumulative wealth achieved by the strategy on the testing dataset.

$$\hat{W}_T = \prod_{t=1}^{T}\hat{x}_t^{'}\hat{z}_t \tag{21}$$

Risk metrics quantify the stability of portfolio performance. They measure how volatile the strategy returns are and quantify the expected and realized losses.

- Standard deviation of returns - empirical portfolio volatility.

$$\hat{\sigma}(\hat{x}^{'}\hat{z}) = \sqrt{\frac{\sum_{t=1}^{T}(\hat{x}_t^{'}\hat{z}_t - \hat{\mu}(\hat{x}^{'}\hat{z}))^2}{T}}. \tag{22}$$

- Maximum Drawdown (MDD) - is the maximum observed loss from a peak to a trough of a portfolio, before a new peak is attained. With *trough* being the maximum realized consecutive loss and *peak* being the previous peak value of wealth, MDD is defined as follows:

$$MDD = \frac{trough - peak}{peak} \tag{23}$$

- Value-at-Risk (VaR) - the minimum expected loss at some probability level. Empirical VaR is essentially a quantile of the distribution of losses, given by $\hat{x}^{'}\hat{z} - 1$. We use standard 5% VaR.

$$P(\hat{x}^{'}\hat{z} - 1 < -VaR_{0.05}) = 0.05 \tag{24}$$

- Conditional Value-at-Risk (CVaR) - expected value of loss at a given probability level. Empirically it can be approximated as the average of losses below certain quantile.

$$CVaR_{0.05} = -\frac{\sum_{(\hat{x}^{'}\hat{z}-1)<-VaR_{0.05}}(\hat{x}^{'}\hat{z} - 1)}{\sum_{t=1}^{T} I((\hat{x}^{'}\hat{z} - 1) < -VaR_{0.05})} \tag{25}$$

Integrated portfolio performance measures attempt to provide a more robust view on portfolio performance by considering both return and risk simultaneously.

- Sharpe Ratio (SR). Empirical version of the standard Sharpe Ratio.

$$SR = \frac{\hat{\mu}(\hat{x}'\hat{z}) - 1}{\hat{\sigma}(\hat{x}'\hat{z})} \tag{26}$$

- Win Ratio (WR). Ratio of the number of times portfolio realizes gains over number of times it loses.

$$WR = \frac{\sum_{t=1}^{T} \mathbb{1}(\hat{x}'\hat{z} > 1)}{\sum_{t=1}^{T} \mathbb{1}(\hat{x}'\hat{z} < 1)} \tag{27}$$

Turnover measures quantify how much portfolio composition changes in time. Following the discussion in Section 2.4, we distinguish two types of turnover: asset turnover resulting from preselection algorithm selecting different assets and weight turnover which is the ultimate measure of final weights redistribution.

- Asset turnover. Asset turnover is defined as an average number of assets that are being changed by the preselector on each time step. Recall from Section 2.2 that at each redistribution step preselection algorithm selects a subset $\mathcal{J}$ of assets out of the set of all available assets $\mathcal{I}$. Denote the subset of assets selected at time t as $\mathcal{J}^{(t)}$. Then asset turnover can be estimated as follows

$$AT = (T-1)^{-1} \sum_{t=2}^{T} (1 - \frac{|\mathcal{J}^{(t)} \cap \mathcal{J}^{(t-1)}|}{|\mathcal{J}^{(t-1)}|}) \tag{28}$$

- Weights turnover. Weights turnover is defined same as in Section 2.4 and incorporates the concept of natural weights change over time discussed therein captured by $\tilde{x}$. Moreover, we consider the turnover in the most practical sense, as the portion of portfolio that has to be redistributed, hence the division by 2 based on the assumption that any value of sold assets is being immediately reinvested into the portfolio by purchasing some other assets.

$$WT = \frac{\sum_{t=2}^{T} |\hat{x}_t - \tilde{x}_{t-1}|}{2(T-1)} \tag{29}$$

Finally, we use a single measure of Mean sum of squared weights (MSSW) to quantify the sparsity of the resulting portfolios. In other words, we want to quantify whether the strategy tends to select just a few assets with high weights assigned to them as opposed to spreading the weights thinly over all the available assets.

$$MSSW = T^{-1} \sum_{t=1}^{T} \sum_{n=1}^{N} \hat{x}_{t,n}^2 \tag{30}$$

Notice, that theoretically this measure takes values in the interval $(0, 1]$, however for specific dataset the lower bound is given by $N(\frac{1}{N})^2$, where N is the total number of assets in the dataset.

3.3 Information extraction analysis

In this Section, we provide technical analysis of the Markov Chains model of future wealth used as information extraction tool in current work. By doing so, we pursue two goals. First we would like to illustrate the caveats of the method identified empirically. Secondly, we strive to test the suggestion provided in Section 2.1 that it might be more meaningful to evaluate the space of Markov Process realizations starting with initial state probabilities conditional on the last state in the data used for fitting the model, rather than unconditional ones as proposed in [18].

The main weak spot of the Markov chain model identified within current research is its susceptibility to outliers in the modelled series. Recall from Section 2.1, that in order to estimate Markov process states the range of returns is being discretized into S intervals. Consequently, dramatic outliers in data series lead to poor discretization as the intervals become too wide. This results in sparse transition matrix being estimated, as most of the empirical returns are being attributed to the same interval. In the worst-case scenario of a very small return variance except for a single dramatic outlier, the resulting transition matrix might essentially contain a single state with transition probabilities significantly different from 0. Such a transition matrix results in forecasted wealth distribution being extremely narrow, to the extent that stable distribution approximation elaborated in Appendix D becomes impossible both from theoretical and practical perspectives, as it becomes impossible to estimate distinct quantiles of the future wealth distribution.

While running empirical experiments, we identified 11 SNP500 index constituents for which stable distribution approximation of future wealth was not possible in standard setting: BIIB, DRI, EPAM, HIG, OKE, OXY, PEP, SWK, TGT, TRGP, WMB. Besides simply removing the assets from consideration, we identified two possible remedies for this problem: increasing the number of states or somehow dealing with outliers. Number of chain states is a model hyperparameter that directly controls the granularity of discretization. However, it directly affects the computational complexity of the algorithm, so increasing it would adversely affect the computation time. Moreover, identifying the number of states that would suit all the assets in the market then becomes manual and cumbersome procedure, which goes against our philosophy of algorithmization. Therefore, we suggest that it might be simpler to treat the outliers themselves. Outlier removal is not considered an option in the given case as the outliers are not anyhow erroneous data points, on the opposite, they carry important information about potential extreme behavior of the asset returns. Therefore, we chose to suppress outliers closer to the center of distribution by using the rule-of-thumb Tukey's fences method. Tukey's fences suggest considering values above $q_{0.75} + 1.5IQR$ and below $q_{0.25} - 1.5IQR$ as outliers and values above $q_{0.75} + 3IQR$ and below $q_{0.25} - 3IQR$ as extreme outliers, where q_p is a p^{th} quantile of the sample distribution and IQR is the inter-quantile range given by $IQR = q_{0.75} - q_{0.25}$. Thus, we suggest suppressing the outliers found according to these definitions by setting their values to the threshold levels. Using such approach, we still retain part of the information regarding the extreme behavior, while at the same time improving the stability of the model. Suppressing just the severe outliers was sufficient for stabilizing stable distribution parameters estimation for 11 stocks we identified as problematic.

In order to illustrate the idea, consider the example of EPAM stock during the 6 year period of 2016-12-09 – 2022-12-09. Figure 3 demonstrates original EPAM stock return series together with the series obtained by suppressing severe outliers and all outliers according to Tukey's fences definition.

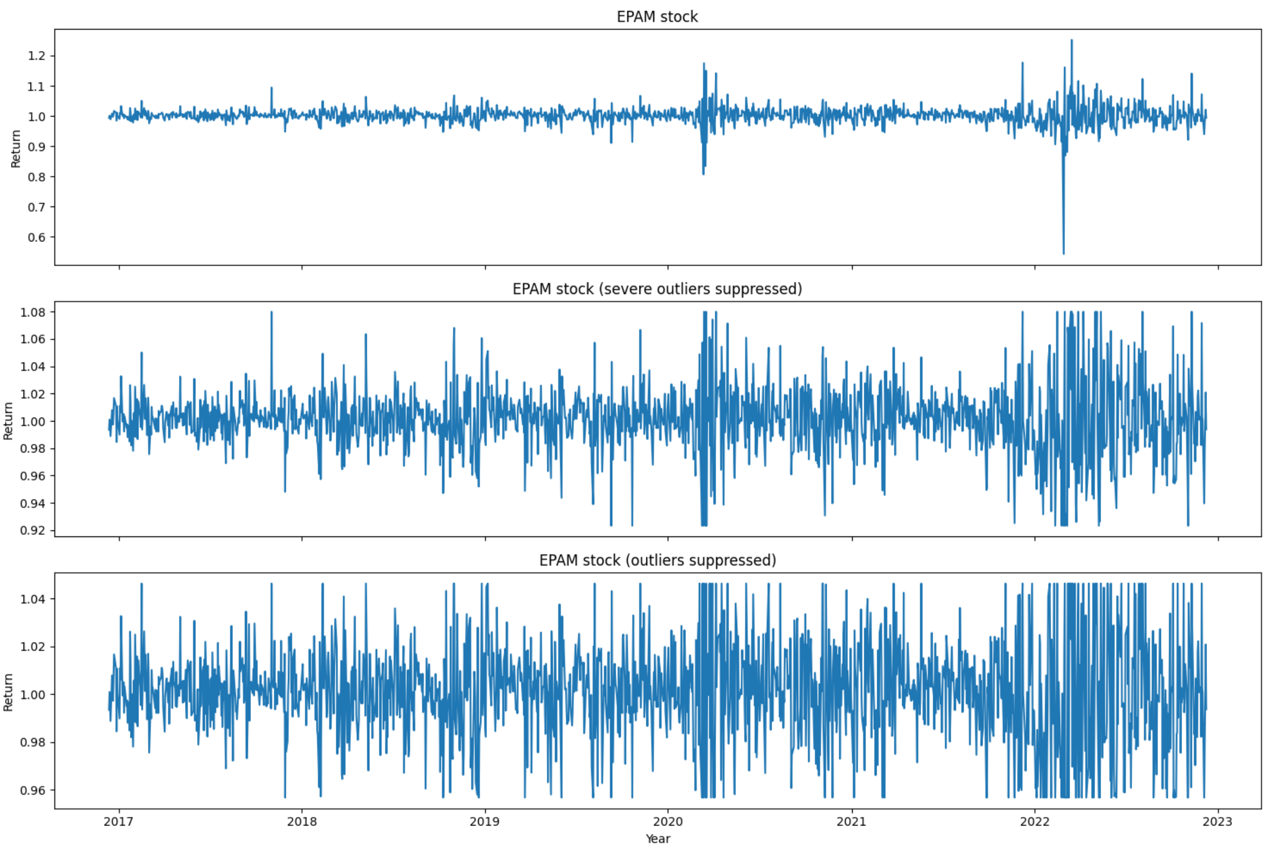

Figure 3: EPAM stock returns under different outlier suppression regimes

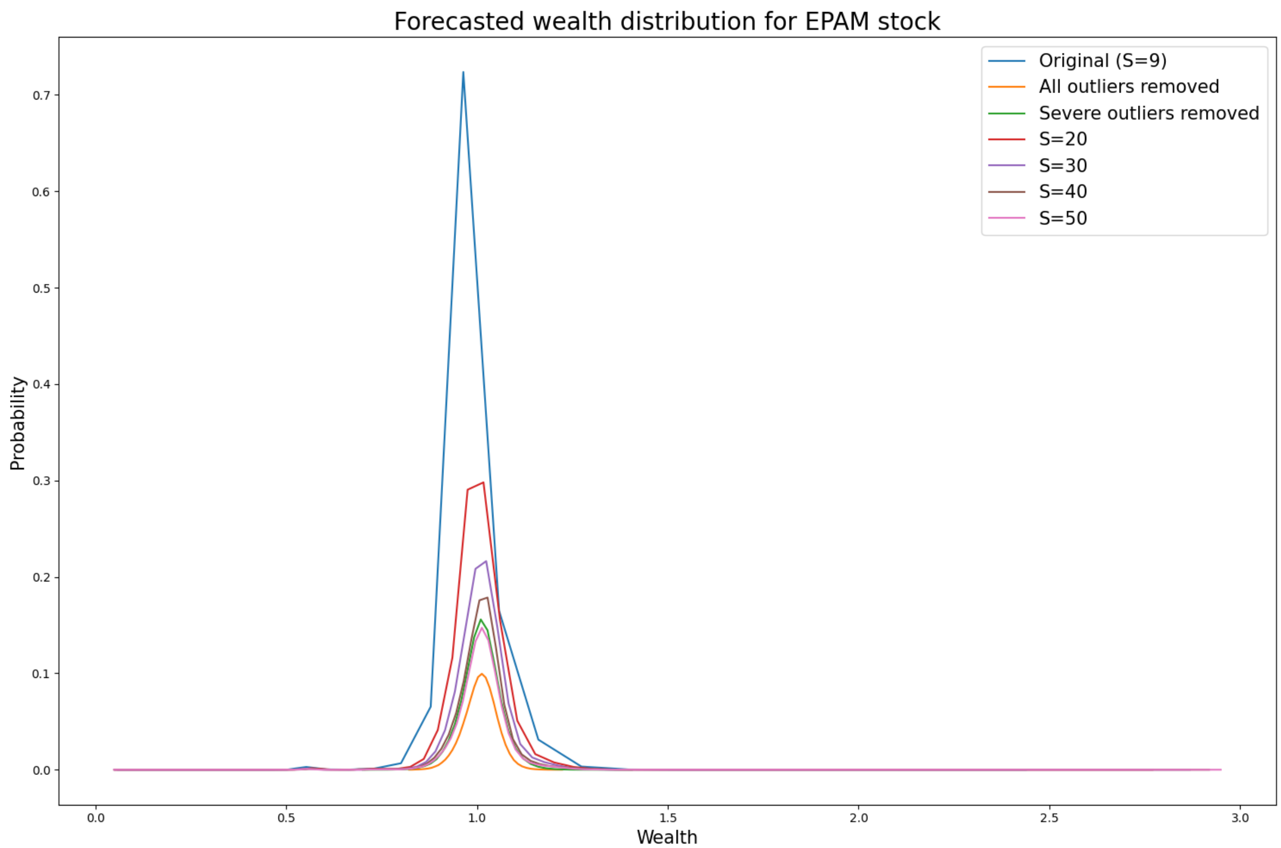

Figure 4: Wealth distribution forecasted for EPAM stock 5 days in the future under different number of states and outlier removal

Figure 4 demonstrates the effect of outlier suppression on the wealth distribution forecast for 5 steps into future produced by the Markov chain model, as well as the effect of increasing the number of states. Notice the seeming convergence of the center of distribution as the number of states increases with the distribution obtained with outliers suppression, suggesting that at least in terms of the first moments, outlier suppression has similar effect to finer discretization. Obviously, the tails are considerably shorter in the distribution forecasted under outlier suppression, but we suppose that this information loss is outweighed by the reduced complexity of the method fine-tuning and computational time improvement as long as it is not too dramatic.

In order to test the effect of the outlier suppression on the model performance, test the proposition on the use of conditional probabilities and gain some additional insights about the model, we need a way to evaluate it's alignment with the empirical, out-of-sample data. Since standard metrics such as Mean Squared Error or Mean Absolute Percent Error, cannot be used to evaluate the goodness of fit for Markov Chain model as we forecast the distributions of future wealth values, rather than a specific value, we propose the following metric to assess how well the actual realized wealth values align with the forecasted distributions.

$$\omega = \frac{(\hat{W}_t - E(W_t))^2}{\sigma(W_t)} \tag{31}$$

Configuration	Mean	SD	Median	Q75	Q25	Max	Min
Unconditional	0.016738	0.039029	0.006212	0.016246	0.001301	0.547281	2e-6
Conditional	0.016784	0.039044	0.006248	0.016246	0.001301	0.547273	6e-6
Unconditional (Out 3)	0.016711	0.033896	0.006148	0.016798	0.001320	0.375054	2e-6
Conditional (Out 3)	0.016693	0.033882	0.006190	0.016788	0.001335	0.375747	7e-6
Unconditional (Out 1.5)	0.019617	0.039803	0.007375	0.019598	0.001637	0.456352	4e-6
Conditional (Out 1.5)	0.019583	0.039775	0.007330	0.019448	0.001634	0.456187	5e-6

Table 1: Exploratory statistics of the ω-statistic for 490 asset return series

Where $\hat{W}_t$ is the realized wealth at time T. The proposed ω-statistic can be interpreted as a sort of realized standard deviation, since the numerator is effectively a variance between forecasted model expectation and empirical values and we divide by standard deviation of the forecasted distribution to account for expected amount of deviation. Naturally, higher values of ω correspond to less alignment between the forecast and reality. Despite the fact that it is difficult to interpret the absolute values of this statistic, it can be utilized in a relative sense to compare how well the model performs for different series or with varying configuration.

In order to grasp the details of model performance and carry out all the comparisons of interest, we embrace the following testing procedure. For each of the 490 stocks that were consistently included in the SNP500 index over the period from 2016-12-09 to 2022-12-09, we apply the following iterative procedure starting from 2021-12-09:

- Fit the Markov chain model on 5 years period preceding the current date.
- Obtain the forecast for wealth distribution 5 days into future.
- Evaluate the realized wealth over the next 5 days as a cumulative product of 5 subsequent returns.
- Shift the current date and the bounds of the training period by 1 day.
- Repeat the procedure until no data left.

Thus, for each of the 490 assets we obtain 253 (number of trading days between 2021-12-09 and 2022-12-09) forecasts for 5 days into future together with realized wealth values over the same time frame. We repeat this procedure under 6 different conditions, defined by whether the conditional or unconditional starting probabilities are used for wealth forecast and whether outliers are being suppressed and under what criteria (severe only or all). For each of the 253 forecasts we compute the ω-statistic and then average it to get the aggregated measure of the model performance under different regimes for each of the 490 asset series. The resulting statistic distribution over asset series is presented by boxplot chart in Figure 5, while the Table 1 summarizes the exploratory statistics over asset series. We indicate the statistics referring to severe outliers suppression as Out 3 and all outliers suppression as Out 1.5, corresponding to the respective IQR multipliers.

From these results, we derive several conclusions. The most apparent one is that there are dramatic differences in how well are the different assets are being modelled by the Markov chain, as suggested by our daily average of the ω-statistic. The boxplot in Figure 5 clearly shows that there are many outliers among the asset series, for which the performance is considerably worse compared to the majority of the

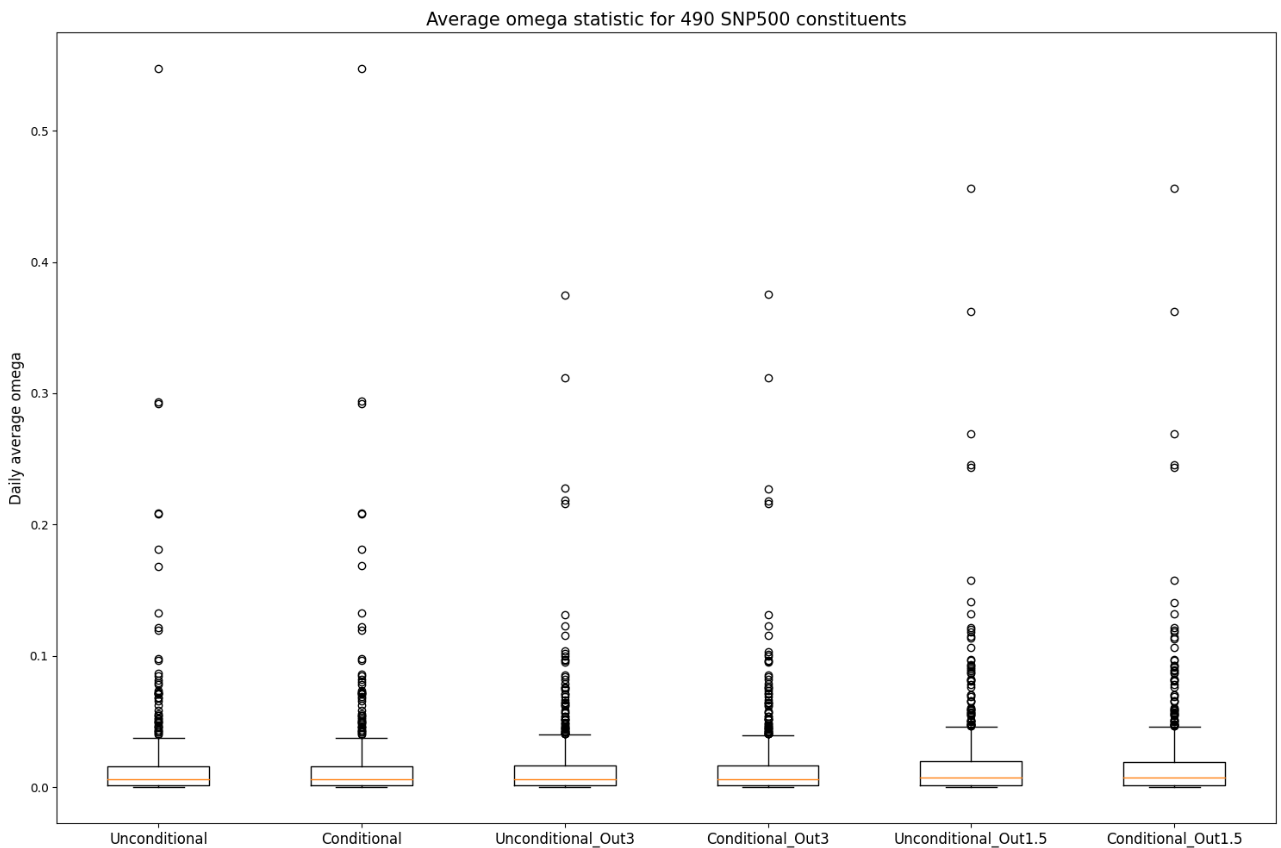

Figure 5: Daily average omega statistic for 490 SNP500 constituents on the Recession dataset

assets. Although it was not utilized this way in the current work, this analysis might serve as another asset preselection technique, suggesting that the outlying assets which do not seem to be modelled well by the Markov chain shall not be considered for portfolio as our ability to get reasonable forecasts for their wealth distribution is doubtful.

Secondly, there seems to be barely any difference between the model performance under conditional and unconditional starting probabilities. Therefore, we found our proposed modification from Section 2.1 invalid, and conclude that unconditional start proposed in [40] should be preferred, as it is less likely to produce any unexpected behavior and there are no performance gains achieved with the conditional version.

Finally, what concerns outlier suppression we observe an interesting pattern where suppressing just the severe outliers by Tukey's definition yields some performance improvement for the most problematic assets, while suppressing all the outliers results in noticeable average asset performance deterioration. We thus interpret it as a confirmation of the aforementioned trade-off between improving model stability and losing some information through the outlier suppression. Therefore, we conclude that severe outliers suppression is a useful heuristic to avoid the estimation stability issues outlined in the beginning of this Section.

3.4 Asset preselection analysis

Recall from Section 2.2 that in the current work we consider two preselection algorithms, which both use a set of statistics to identify most efficient assets which should be considered for the portfolio – Ranking and DEA. In this Section we analyze and compare the considered algorithms from two perspectives: asset turnover, which can be considered as algorithm stability, and selected sets intersection, which can be considered as a measure of how different the results obtained by each of the algorithms are. For both algorithms we use the same set of preselection metrics with the same hyperparameters, therefore any performance differences shall be purely attributed to the preselection algorithm itself. Moreover, we investigate how preselection is affected by outlier suppression heuristic introduced in Section 3.3, following the conclusions made in that Section, we consider only severe outlier suppression. In order to obtain the necessary results, we consider the preselections made by both of the algorithms with and without outlier suppression on the Recession dataset following the standard back-testing procedure.

Figure 6 demonstrates asset turnover in time for the 4 considered configurations. Since asset preselection is made only each 5 trading days, the Moving Average(20) smoothing was applied for visual comeliness. We may clearly see that DEA method exhibits much higher asset turnover compared to our Ranking algorithm, thus we deem it to be more unstable in time and more susceptible to transaction costs. Moreover, suppressing the severe outliers (annotated by "_out" postfix) clearly increases asset turnover for both algorithms. From more detailed analysis, we observed that the most unstable assets tend to be preselected consistently due to unrealistic expectations resulting from poor Markov chain fit when outliers are not suppressed.

Figure 7 demonstrates the % of intersection between the preselections made by each pair of configurations. The intersection is calculated as a number of assets included in both sets over the total number of unique assets in both sets. It was also smoothed with Moving Average (20) for visual clarity. This chart demonstrates that DEA method is more affected by outlier suppression compared to Rank-

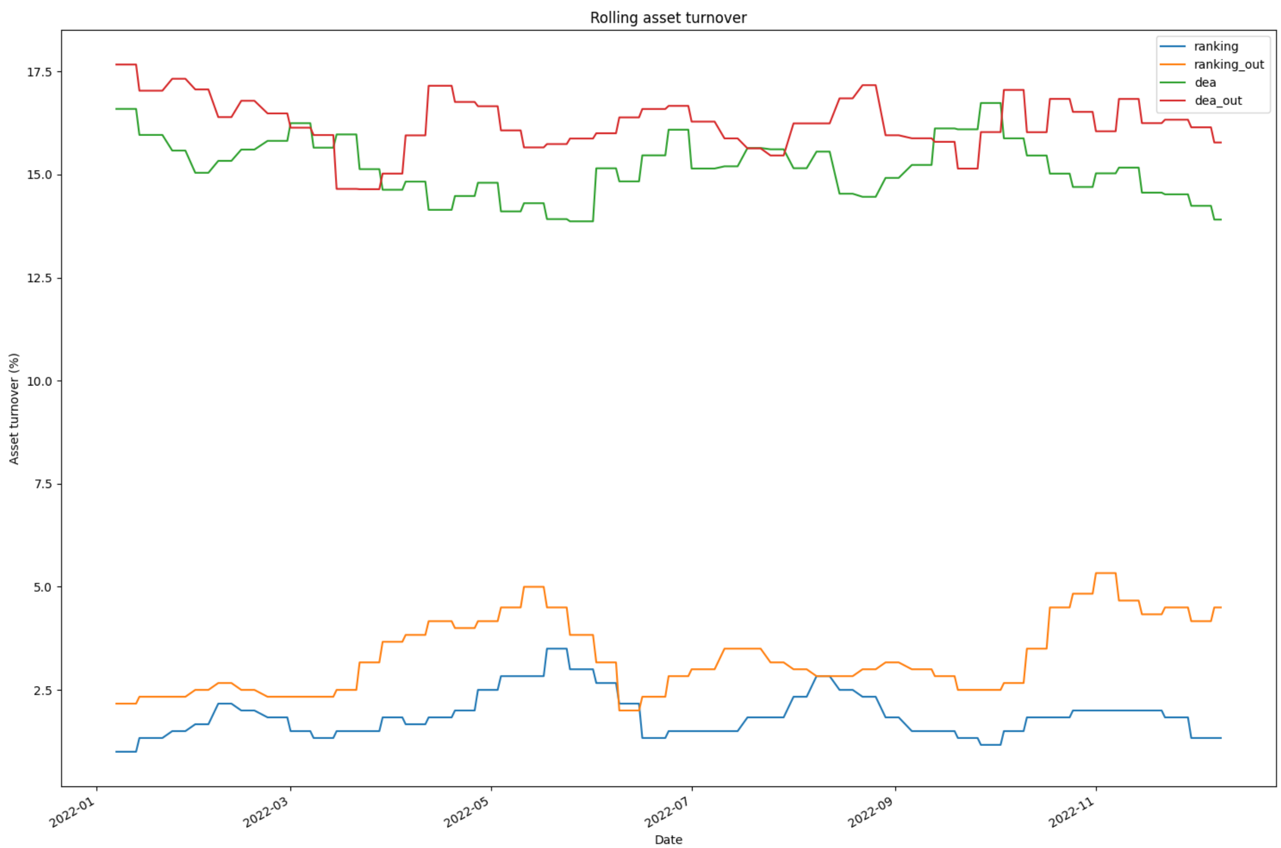

Figure 6: Rolling average daily asset turnover of Ranking and DEA algorithms with and without outlier suppression

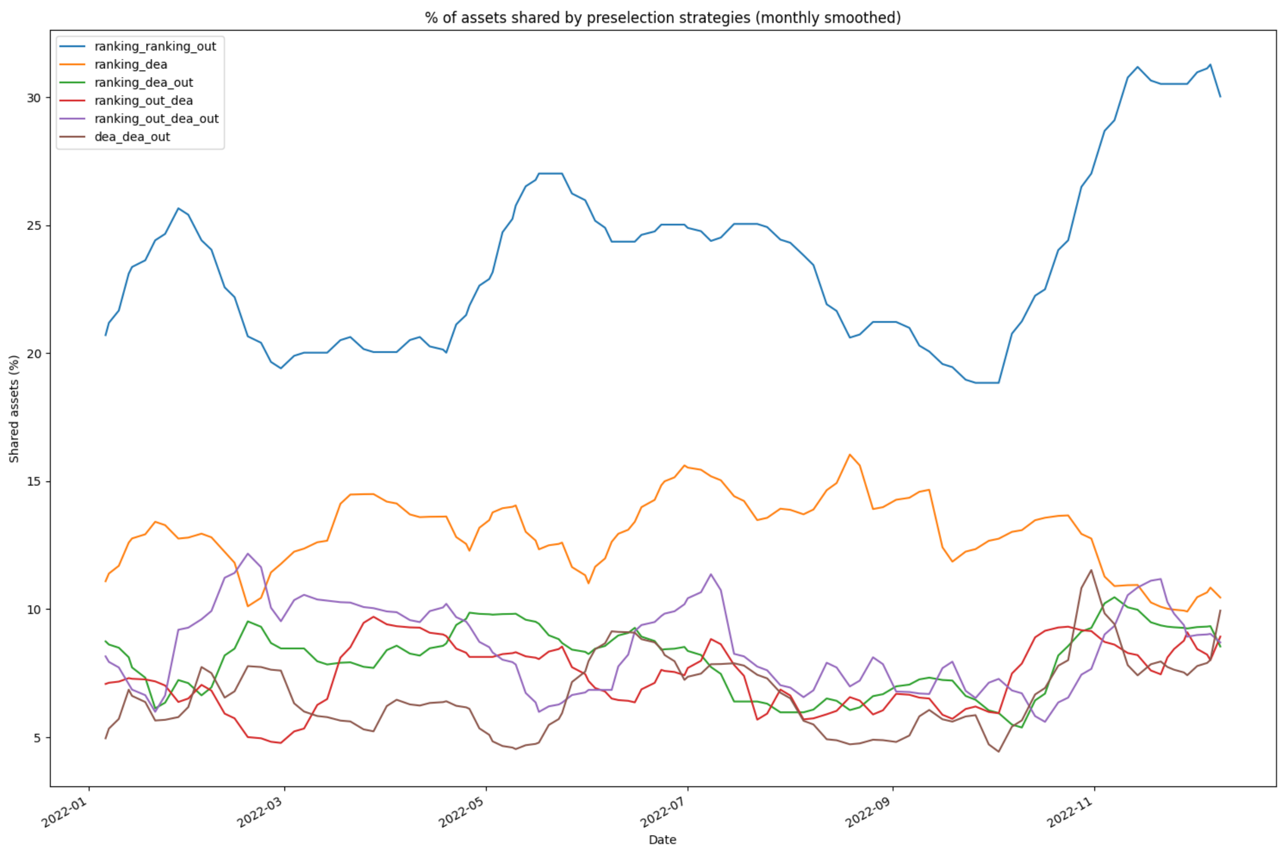

Figure 7: Rolling average daily % intersection in assets preselected by Ranking and DEA algorithms with and without outlier suppression

ing. In fact, for the Ranking algorithm the preselection decisions changed only slightly with outlier suppression, while DEA and DEA with outliers suppressed show the lowest intersection out of all pairs. Moreover, we can see that when outliers are not suppressed, the intersection between DEA and Ranking is considerably higher, supporting the argument that both algorithms are being affected in a similar manner by distorted Markov chain fit, although DEA is affected much more significantly. These results confirm that the two considered algorithms, while very similar in nature and using the exactly same inputs, produce considerably different results.

Finally, analyzing the performance of the assets preselected by each of the algorithms, we observed that assets preselected with Ranking performed much worse on the testing Recession dataset on average as compared to DEA and benchmarks. At the same time, equally weighted portfolio of DEA preselected assets outperformed both S&P index and global equal-weight portfolio used as benchmarks on both datasets (see Table 4). Moreover, in contrast to our stability considerations, both Ranking and DEA preselected assets with outlier suppression showed worse performance than the sets obtained with no outlier suppression.

3.5 Portfolio optimization analysis

In this Section we conduct empirical analysis of the optimization procedure and optimized functions. In particular, we provide analysis of the optimized functions themselves, compare the optimization performance of the heuristic approach outlined in Section 2.3 with Genetic Algorithm that is widely used in the field [37, 16] and, finally, demonstrate the impact of outlier suppression discussed in Section 3.3 on the optimization.

In order to grasp the understanding of how optimized functions presented in Section 2.3 look like and what issues they might present we developed the following procedure. Same as in Section 3.3 we take a single time period of 6 years from 2016-12-09 to 2022-12-09, corresponding to Recession dataset with all the training data. In order to reduce dimensionality of the weight space, we use Principal Component Analysis (PCA) to extract just the first two components, PC1 and PC2, explaining the maximum portion of variance in the underlying assets - 46.1% in our case. We then interpret the scores of these components as the two series best representing the underlying market. Similar interpretation of PCA components as market driving factors can be found in [18]. Notice that originally extracted components have mean of 0, which is problematic for Markov chain estimation procedure designed for return series non-negative by definition. Therefore, we shift both score series up by the absolute value of the minimum among both series plus a small constant. Since it is relation between the series that affects optimization results, shifting location of both series by the same value does not cause any distortion to results. Finally, we evaluate each of the 4 optimized functions on 100 portfolios of these 2 components obtained by increasing the weight of the first component x_{PC1} from 0 to 1 with the step of 0.01. The weight of the second component, by definition, is given by $x_{PC2} = 1 - x_{PC1}$. The results of this exploration are displayed in Figure 8. This analysis yields several observations.

First of all, we clearly see that none of the functions is smooth, even though Sharpe Ratio looks rather close to it, so each of them presents several local maximums. Secondly, we may clearly see that the 4 observed global maximums suggest different weight allocations, thus confirming that each of the objective functions utilizes different market information and conveys unique trading signal. Moreover,

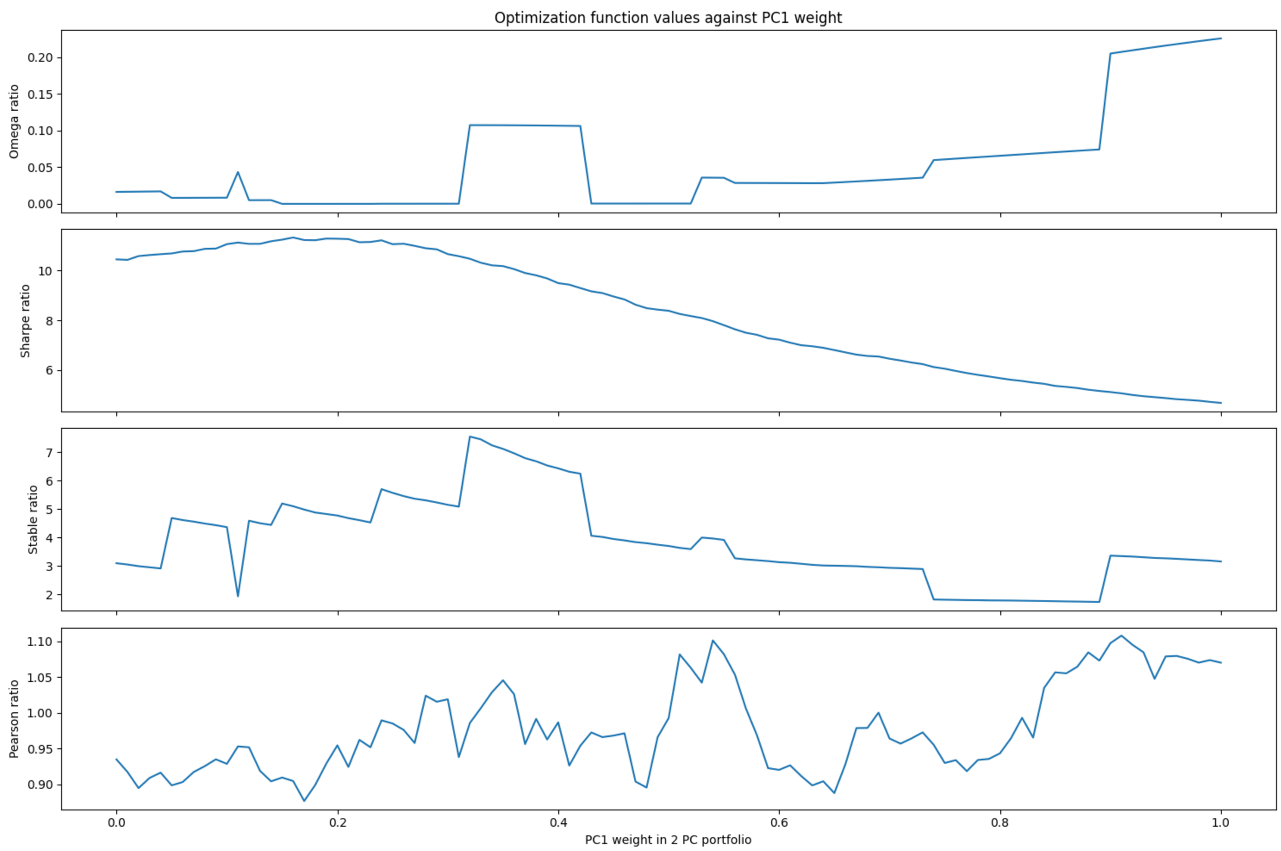

Figure 8: Optimized functions evaluation on the space of portfolios including first two principal components of the overall market

since the first principal component captures the biggest portion of variance, the function favoring it might be considered as more risk-taking. From this analysis it happens to be MSG Omega Ratio proposed in the current work. Finally, as a more technical observation, the shape of the MSG Omega Ratio and MSG Stable Ratio is interesting as they both present very sharp step-function pattern. The most reasonable explanation we have is that it is associated with the way stable distribution parameters are estimated, as both of the functions utilize it. Perhaps minor adjustments in weights result in the same quantiles estimated for future wealth distribution as the currently used method described in Appendix D is rather rough.

Next, we consider the optimization performance of the heuristic algorithm described in Section 2.3, which we refer to as Simplex for convenience, in comparison with the widely used Genetic Algorithm. To do so, we use the DEA method for asset preselection and run the simulated trading evaluation on both Growth and Recession datasets with preselection and weights redistribution done each 5 trading days. In such a manner we evaluate 12 strategies, given by all the combinations of the following configurations:

- Severe outliers, either suppressed or not suppressed.
- Optimized function: MSG Sharpe Ratio, MSG Omega Ratio, MSG Stable Ratio
- Optimization algorithm: GA or simplex exploration heuristic

Figure 9 presents the optimization times for 3 optimized functions without outlier suppression optimized with GA and Simplex algorithms. The tests were performed on the same machine. One may notice that Genetic Algorithm is typically slower than the discussed Simplex algorithm, however it's optimization times are much more stable. The instability of the Simplex optimization times is determined by 2 reasons. First, exploration is being done steadily, attending any single parameter individually, hence optimization time depends on the number of parameters (weights), which differs in time using DEA preselection. Secondly, the number of objective function evaluations is not fixed, it keeps searching as long as the improvements within the given tolerance level are achieved, as opposed to GA, which uses the notion of maximum number of generations a limit for the search. In our testing setup, GA was configured to make no more than 2000 function evaluations as otherwise computational time would become prohibitive.

Table 2 summarizes the performance of the two algorithms over different strategy configurations. In particular, we are interested in the Win rate and VT (Value per Time) columns, which correspond to the number of times optimal function value found by a given algorithm exceeds the result of its counterparts and the average improvement per second respectively. This summary clearly shows the superiority of the Simplex algorithm in terms of the improvement per time. However, this only holds as long as the number of weights for optimization stays within reasonable limits. The next thing we observe is that the extremely high values of MSG Omega Ratio and MSG Stable Ratio recorded when outliers are not being suppressed. They highlight the instability of the whole algorithm performance in the presence of severe outliers. Recall from Section 2.3, Eq. 12, that denominator of the ratio is the integral of future wealth stable distribution cumulative density function from minus infinity to threshold. Thus, as estimated location of the stable distribution increases and moves further from the

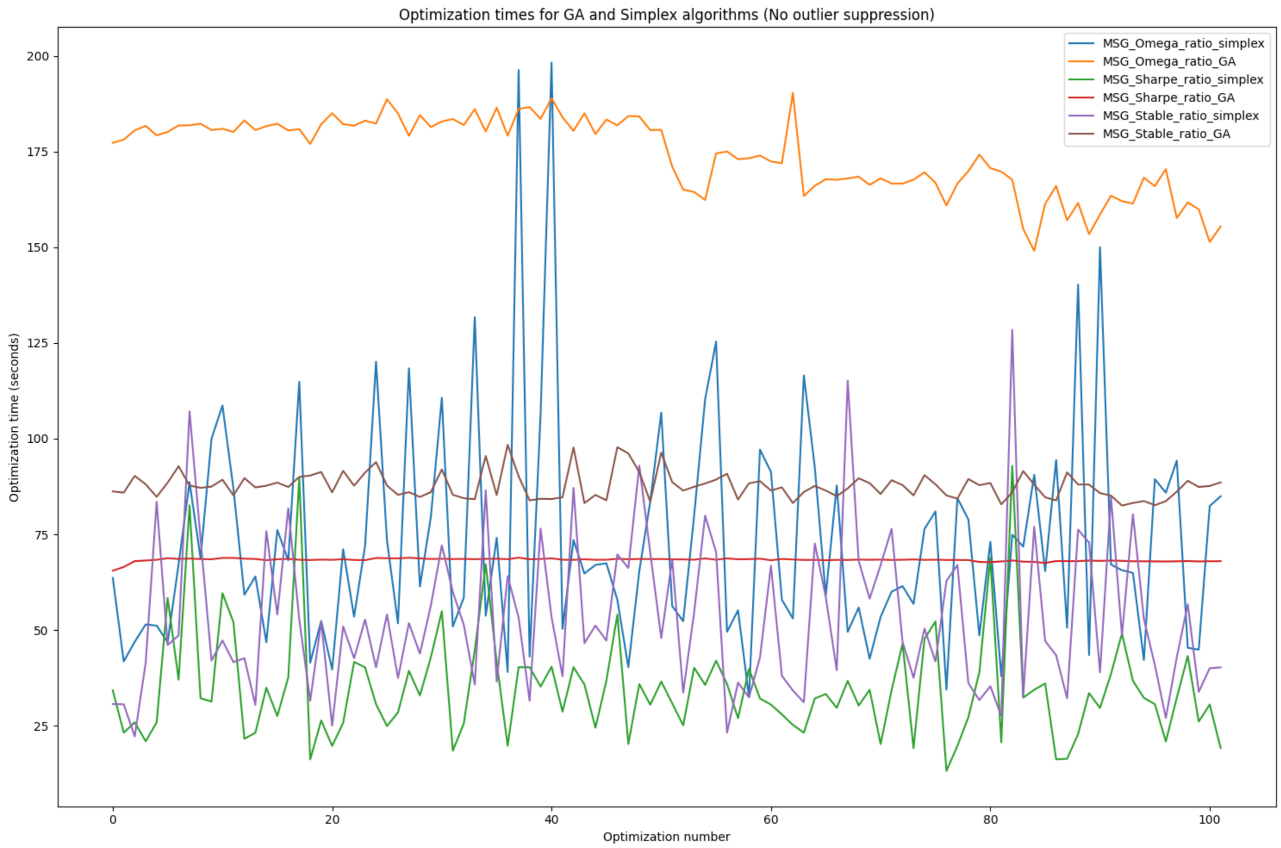

Figure 9: Aggregated optimization times on both Recession and Growth datasets for GA and Simplex optimization algorithms

Function	No outliers	Optimizer	Mean time	Mean value	Win rate	V/T
MSG Sharpe Ratio	yes	Simplex	27.18	0.28	0.48	0.011
MSG Sharpe Ratio	yes	GA	54.03	0.28	0.52	0.005
MSG Sharpe Ratio	no	Simplex	34.85	0.84	0.80	0.028
MSG Sharpe Ratio	no	GA	68.32	0.50	0.20	0.0074
MSG Omega Ratio	yes	Simplex	300.84	0.74	0.98	0.0046
MSG Omega Ratio	yes	GA	176.34	0.20	0.02	0.0007
MSG Omega Ratio	no	Simplex	73.38	$5.29 \cdot 10^{19}$	1	$9.54 \cdot 10^{17}$
MSG Omega Ratio	no	GA	174.04	1.06	0	0.0059
MSG Stable Ratio	yes	Simplex	47.48	35.00	0.58	0.85
MSG Stable Ratio	yes	GA	70.15	35.16	0.42	0.50
MSG Stable Ratio	no	Simplex	53.26	323519	0.76	6346
MSG Stable Ratio	no	GA	87.68	101968	0.24	1168

Table 2: Optimization summary, where win rate refers to the number of times optimization algorithm finds superior value, V/T is Value per Time

Strategy	$\hat{\mu}$	$\hat{\sigma}$	W_T	SR	WR	MDD	VaR	CVaR	AT	WT	MSSW
Sharpe	0.999	0.021	0.705	-0.056	0.969	-0.088	0.033	0.053	0.151	0.167	0.615
Sharpe R1	1	0.025	0.708	-0.041	1.032	-0.148	0.044	0.061	0.151	0.147	0.870
Sharpe R2	1	0.016	0.954	-0.004	0.992	-0.068	0.028	0.035	0.037	0.089	0.282
Stable NO	1	0.012	1.102	0.039	1.181	-0.084	0.020	0.027	0.161	0.179	0.321
Stable NO R1	1	0.012	1.104	0.039	1.074	-0.119	0.020	0.028	0.161	0.155	0.349
Omega NO	1	0.036	0.765	-0.011	0.895	-0.193	0.056	0.082	0.161	0.197	0.695
Omega NO R1	1	0.034	0.793	-0.009	1.016	-0.129	0.052	0.082	0.161	0.175	0.717

Table 3: Strategies performance with and without online regulation

threshold, the denominator tends to 0. In order to avoid division by 0, in our implementation we set the limit for MSG Omega Ratio to equal 1^{20} when denominator is 0. Thus, the results for data without filtering the outliers suggest that unrealistically high location parameter is being forecasted resulting in MSG Omega Ratio hitting this cut off point.

The resulting strategy performance further confirms the superiority of the Simplex algorithm as under all the configurations the realized Sharpe Ratio for the strategies using Simplex optimization outperformed the GA version. This result additionally supports the overall idea that the considered target functions carry relevant trading signals, thus portfolio with a higher historical value of the given function is expected to demonstrate better performance, which we observe. As a side observation, due to internal mechanics of the algorithms, GA tends to produce much more sparse portfolios compared to Simplex (see Tables 4 and 5).

3.6 Online regulation analysis

In this section we demonstrate the results of testing the regularization technique proposed in Section 2.4. Recall, that we suggested regularizing weight redistribution in time by adjusting the initial wealth value used as a stating point for evaluation of the Markov process realizations by the ratio of the considered weight redistribution. In the empirical results presentation, we refer to this regulation approach as R1, while notation R2 corresponds to the second regulation method discussed in the same section. In order to test this approach, we test 6 strategies defined by 3 objective function to optimize: MSG Omega Ratio, MSG Sharpe Ratio and MSG Stable Ratio and whether R1-type regulation is applied or not. For the tests, we set the R1-type regularization parameter to $c = 0.1$. Recall that we do not consider MSG Pearson Ratio, since starting wealth adjustment has no effect on correlations. Since R2-type of regulation is much more straightforward, we only provide its results for MSG Sharpe Ratio optimizing strategy for illustrative purposes. For it we use smoothing parameter $\nu = 0.5$.

Figure 10 shows the weight turnover in time for all the considered strategies. We apply Moving Average (20) smoothing for visualization purposes, as weights are being redistributed only each 5 trading days. We can clearly see, that our proposed method works for all the considered objective functions optimization, resulting in considerably lower weight redistribution in time. Notice, that unlike R2-type regulation which straightforwardly reduces weight turnover proportional to the value of ν (2 times in this case), R1-type regulation effect varies in magnitude from no difference at all to dramatic decrease, which can be seen for MSG Sharpe Ratio optimizing strategy in the beginning of the testing period.

Table 3 summarizes the final performance of the strategies with and without regularization in

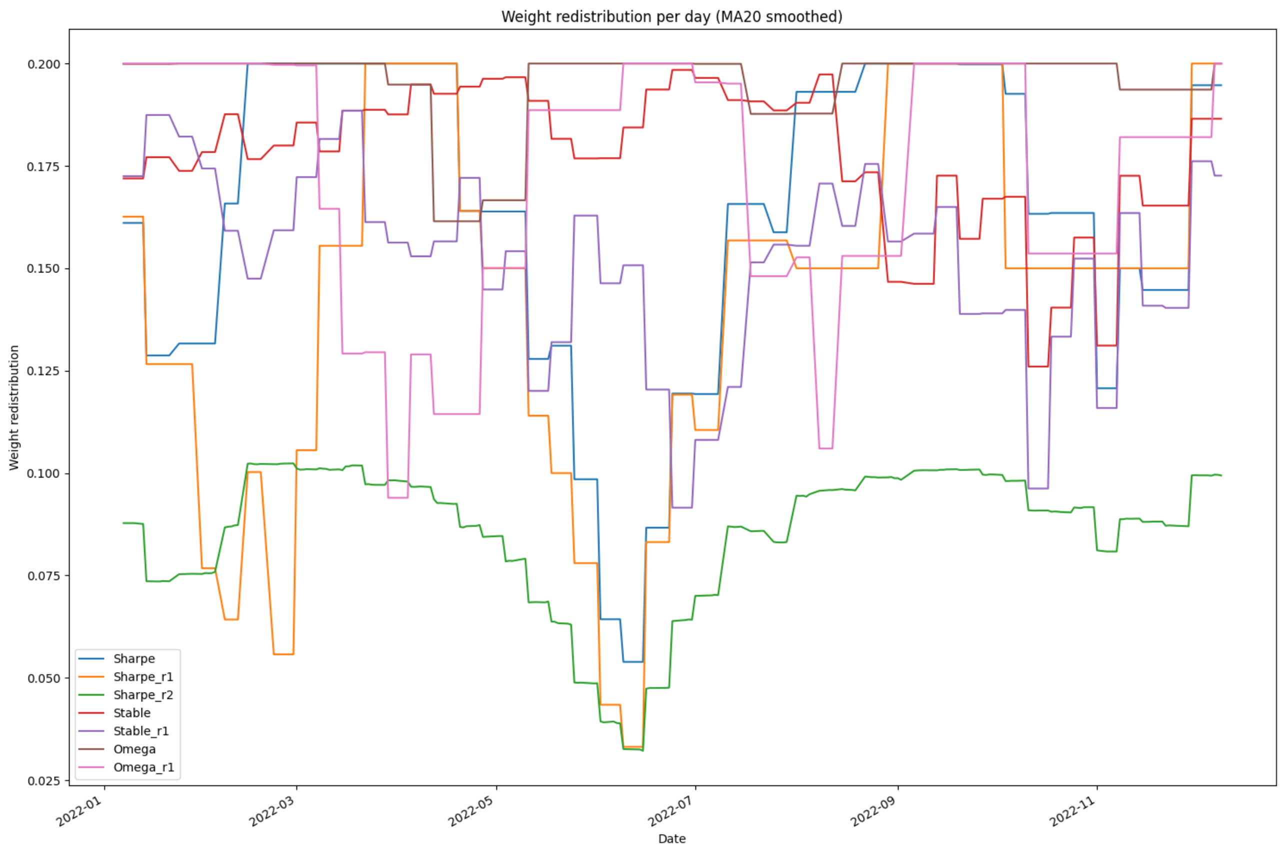

Figure 10: Averaged daily weight redistribution for strategies with and without regulation applied

numeric terms. Notice that we use NO (No Outliers) notation to indicate the outlier suppression applied for Omega and Stable strategies, as they are considered to be more susceptible to them as indicated in Section 3.5. Comparing the average weight turnover realized by the strategies we see, that while in time the effect of R1 regulation varies, on average it is rather similar for all the optimized functions and even close to the value of regularization parameter set. For MSG Sharpe Ratio the average weight turnover reduction of around 12% was achieved, while for MSG Stable Ratio and MSG Omega Ratio – 13.4% and 11% respectively. Another interesting observation is that R1-type regulated strategies show higher weight sparsity, as opposed to R2-type regulation, which naturally results in more dense portfolios through maintaining some weights even for assets not preselected in a given evaluation. The increase is very substantial for the Sharpe Ratio optimizing strategy, although not that big for Stable and Omega. Perhaps, it is related to the fact that for the 3 Sharpe strategies, outliers where not suppressed, unlike for Omega and Stable strategies. What concerns the overall performance, we may see that at least in this test regulated strategies realized slightly better Sharpe Ratios than their unregulated counterparts.

3.7 Final results

In this Section we report the final results of the top 10 best performing strategies identified from the multitude of experiments conducted on both Growth and Recession datasets. The selection was performed based on the realized Sharpe ratio of the strategies and in each table they are sorted by it in decreasing order. The notation used to identify different strategy configurations is the following:

- "NO" indicates that severe outliers were suppressed prior to model fitting.
- Omega (x) indicates the $\tilde{\theta} = x$ hyperparameter value, if several versions are presented in the same table. Otherwise the default value of 1.01 is implied.
- 1/N indicates equal weight distribution
- R1 indicates R1-type regulation method applied
- T refers to the redistribution frequency used in the experiment
- GA refers to Genetic Algorithm used for optimization

The default setting is considered to be DEA preselection, no outlier suppression, Simplex optimization and no regulation, in these cases we indicate only the objective function optimized. It is necessary to elaborate, that while no full scale cross-validation of the hyperparameters was performed due to computational complexity of the end-to-end approach testing, two hyperparameter values were tested for Omega ratio - 1.01 and 1.05. Moreover, we tested two redistribution frequencies T=5 and T=20, corresponding to weekly and monthly redistribution respectively.

Table 4 presents the results of the evaluation of the selected strategies on the Growth dataset. The corresponding wealth paths of the strategies are displayed in Figure 11. On this dataset, all of the top performing strategies realized Sharpe Ratio between 2 and more than 4 times higher than the S&P500 index benchmark, thus validating the considered approach. 5 out of 10 strategies optimized MSG Omega

Strategy	$\hat{\mu}$	$\hat{\sigma}$	W_T	SR	WR	MDD	VaR	CVaR	AT	WT	MSSW
Omega (1.05)	1.004	0.022	2.775	0.196	1.312	-0.048	0.026	0.040	0.152	0.201	0.711
Stable	1.002	0.013	1.811	0.185	1.471	-0.029	0.014	0.019	0.152	0.181	0.312
Omega (1.01) NO R1	1.005	0.030	2.843	0.153	1.135	-0.110	0.037	0.053	0.157	0.167	0.798
Omega (1.01)	1.002	0.019	1.666	0.119	1.312	-0.103	0.026	0.036	0.147	0.187	0.732
Pearson	1.002	0.019	1.679	0.118	1.066	-0.049	0.021	0.031	0.152	0.194	0.535
DEA 1/N	1.001	0.008	1.253	0.110	1.377	-0.030	0.014	0.019	0.152	0.159	0.032
Omega (1.01) NO GA	1.001	0.010	1.304	0.109	1.312	-0.039	0.016	0.022	0.156	0.182	0.071
Sharpe	1.002	0.019	1.541	0.101	1.049	-0.053	0.026	0.033	0.152	0.189	0.736
DEA 1/N NO	1.001	0.008	1.220	0.097	1.333	-0.043	0.012	0.020	0.157	0.164	0.030
Sharpe NO R1	1.003	0.029	1.809	0.096	1.118	-0.138	0.046	0.055	0.157	0.147	0.999
S&P500	1	0.008	1.095	0.048	1.082	-0.039	0.013	0.020	-	-	-
1/N	1.001	0.009	1.163	0.069	1.355	-0.048	0.015	0.023	0	0.005	0.002

Table 4: Growth Dataset Results

Strategy	$\hat{\mu}$	$\hat{\sigma}$	W_T	SR	WR	MDD	VaR	CVaR	AT	WT	MSSW
Stable (T5)	1.001	0.015	1.340	0.083	1.144	-0.077	0.020	0.029	0.150	0.188	0.587
Omega (T5)	1.002	0.027	1.386	0.061	0.961	-0.155	0.035	0.044	0.150	0.186	0.694
Sharpe (T5)	1.001	0.023	1.296	0.055	1.049	-0.103	0.030	0.038	0.150	0.182	0.560
Stable (T20)	1.001	0.028	1.332	0.052	1.126	-0.094	0.028	0.037	0.036	0.047	0.559
Omega (T20)	1.002	0.032	1.360	0.052	0.902	-0.120	0.030	0.041	0.018	0.050	0.700
Stable NO (T5)	1	0.012	1.102	0.039	1.181	-0.084	0.020	0.027	0.161	0.179	0.321
Stable NO R1 (T5)	1	0.012	1.104	0.039	1.074	-0.119	0.020	0.028	0.161	0.155	0.349
Ranking Sharpe (T5)	1	0.013	1.069	0.026	1.202	-0.050	0.022	0.030	0.018	0.002	0.976
Pearson NO (T5)	1	0.016	1.013	0.011	0.820	-0.082	0.023	0.029	0.161	0.196	0.276
Stable GA (T5)	1	0.013	0.995	0.005	1.007	-0.092	0.022	0.028	0.150	0.174	0.051
S&P500	0.999	0.015	0.843	-0.037	0.762	-0.089	0.026	0.033	-	-	-
1/N	1	0.015	0.913	-0.017	0.917	-0.098	0.024	0.032	0	0.006	0.002

Table 5: Recession Dataset Results

Ratio proposed in current work, thus confirming the validity of such optimization target and indicating its superior performance on the growing market, as the strategy optimizing this metric showed the best overall performance judging by Sharpe Ratio. The difference in results between experiments with different hyperparameter values for MSG Omega Ratio indicate the need for more detailed analysis of its effect on strategy performance. Due to scope of this thesis, only two values of 1.05 and 1.01 were used in experiments, however in future research, it would be interesting to consider a more granular and deeper inspection of the optimal hyperparameter choice.

Clearly, the large impact on the achieved results is made by preselection. Notice that equal-weight strategy with DEA preselection scores the top 5 Sharpe Ratio, suggesting that even without any optimization the sets of assets preselected by it outperform the general market. In this regard it is also important to mention that MSG Sharpe Ratio optimizing strategies, while present among the top, performed worse than the equal weight strategy for the same asset preselection. This result suggests that Sharpe Ratio might under-perform, at least on the growing market.

Table 5 presents the results of the evaluation of the selected strategies on the Recession dataset. The corresponding wealth paths of the strategies are displayed in Figure 12. The results demonstrate that even during turbulent economic conditions, the considered strategies still outperform the S%P500 index benchmark given both by index itself and equally-weighted portfolio of its constituents. Moreover,

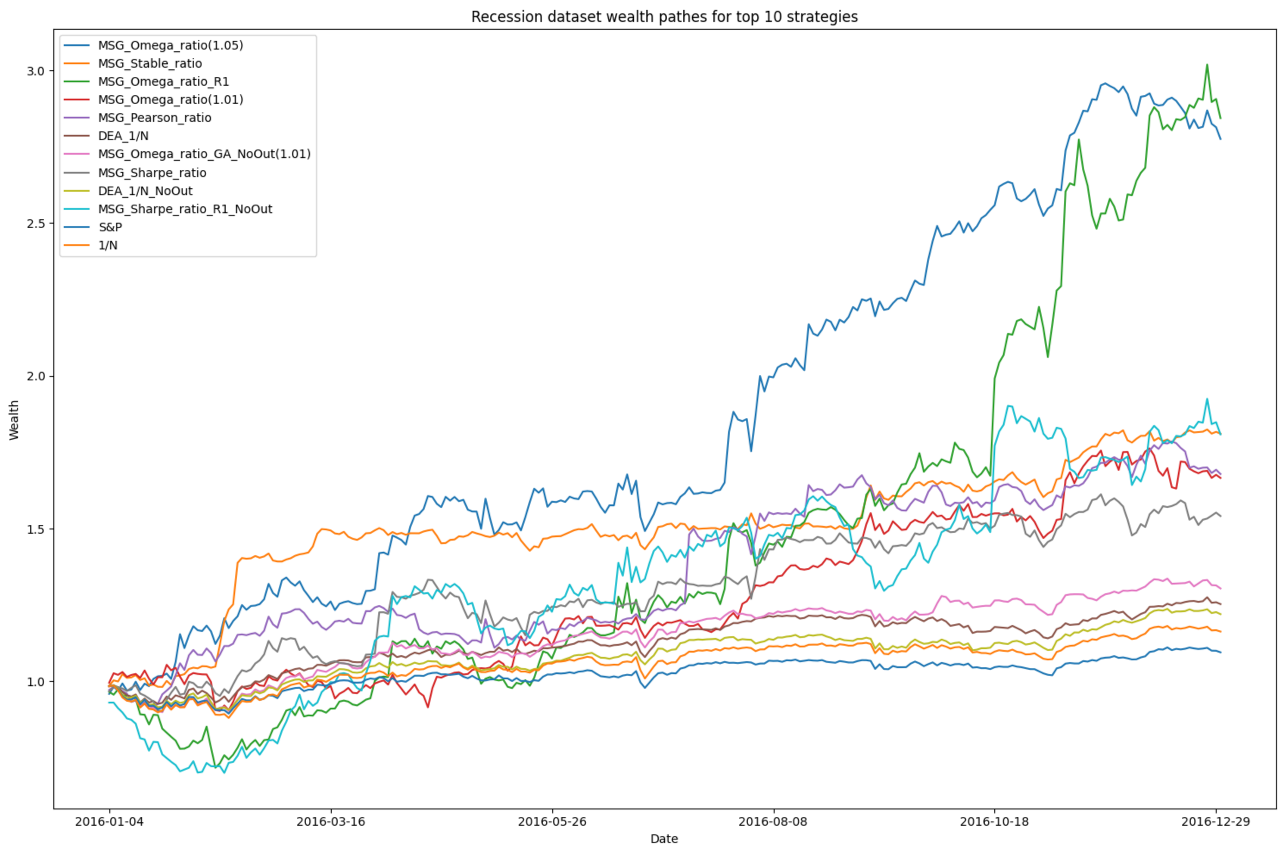

Figure 11: Wealth pathes of strategies on Growth dataset

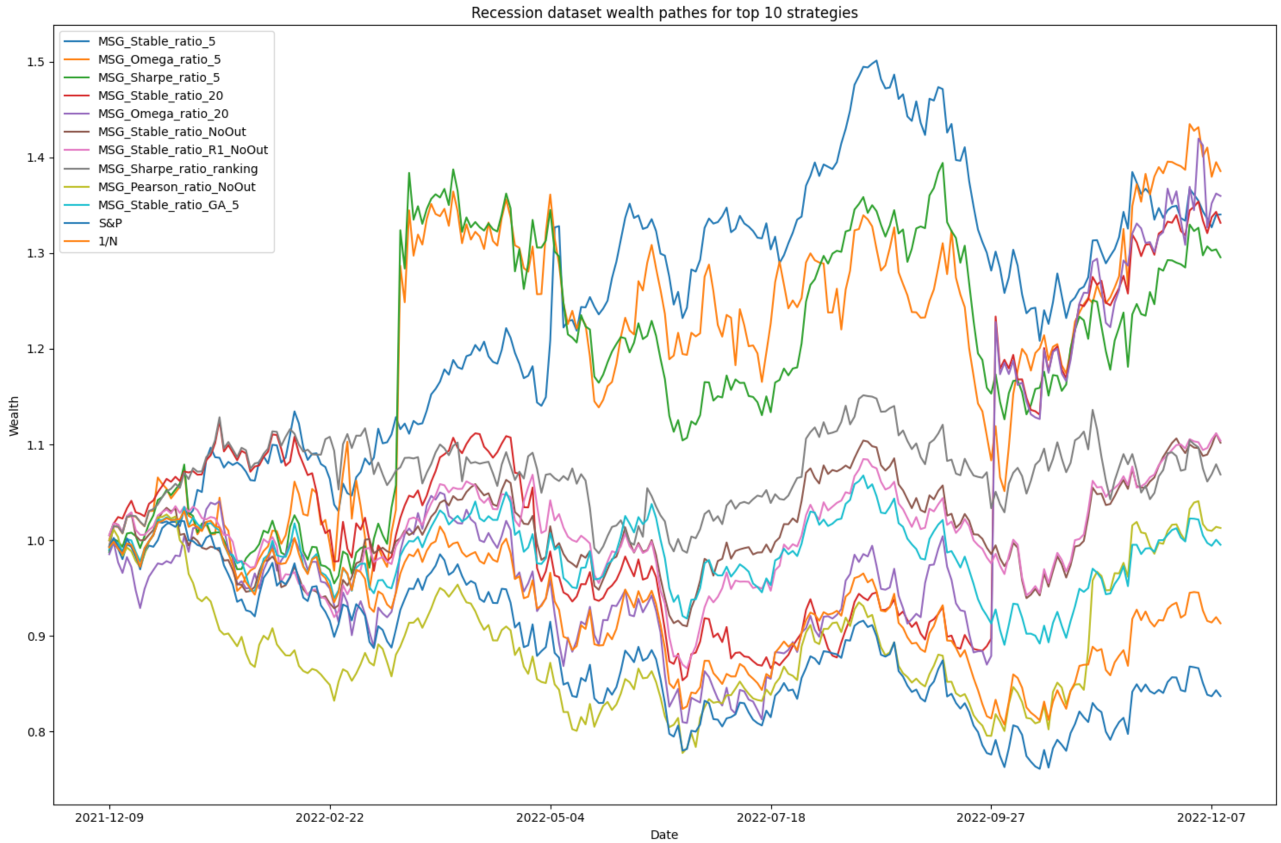

Figure 12: Wealth pathes of strategies on Recession dataset

unlike on the Growth dataset, Omega and Stable ratio optimizing strategies with monthly redistribution (T=20) show similar results to their weekly redistributing counterparts. This result is interesting, since naturally less frequent redistribution results in lower weight turnover and, thus, is less susceptible to transaction costs in the real world environment.

Interestingly enough, one configuration with Ranking preselection appears among top performers. However, one might notice that extremely low weights turnover together with MSSW being reasonably close to 1 suggest that this strategy essentially was holding a single asset over the whole test period. It is likely that the sole asset might be a proxy to some growth signal of the market that we would want to maximize. This suggests an avenue for future research, in the direction of sparse recovery of leading market signals.

Considering the results from both datasets, we may see that different optimization objectives tend to produce the final portfolios of certain sparsity. Thus, Sharpe and Omega ratios tend to produce more sparse portfolios as compared to Pearson and Stable ratios. While sparsity is not synonymous to diversification, the empirical results show that Omega and Sharpe ratio tend to be more risky as compared to Stable and Pearson ratio maximizing strategies. Essentially, the main impact of it may be seen in the fact that Omega-based configuration realized the top performance on the growing market, while Stable-based won during the recession, realizing approximately 33% higher Sharpe ratio.

Despite the previous results indicating that outlier suppression improves numerical stability of the considered methods, from the final results we may see that strategies with no outlier suppression managed to achieve better results. However, we suspect that these results might be sporadic, therefore, in order to make firm conclusions, a larger comparison study may be needed, covering a wider range of diverse datasets.

4 Conclusions

In this paper we proposed a conceptual framework for algorithmic portfolio management, disassembling this complex problem into 4 subtasks: information extraction, asset preselection, portfolio optimization and online regulation. On the theoretical level, we demonstrated that this representation helps unifying the otherwise fragmented approaches used both in literature and in practice. Moreover, this representation was designed with programmatic implementation in mind and was put in the foundation of the Algoport[3] Python package that was developed during this work, which is presented in more details in Appendix A. We believe that this implementation demonstrates in practice, the applicability of the proposed framework concept, as it allows configuration, tuning and testing of multiple end-to-end algorithms in efficient manner.

As a case study for the proposed framework, we implemented the Markovian portfolio management approach consistently developed by [29, 40, 18] and proposed several extensions. Considering the Markov Chain model itself, we identified the following issues. The state discretization procedure described in Section 2.1 is highly susceptible to outliers in return series, which can be addressed by either fine tuning the number of states or treating outliers themselves. Since increasing the number of states directly affects computational time of the algorithm, which was found to be challenging even in the initial setting, we explored the second option. Since outliers in return series carry important information about the assets, we suggest that they should not be removed, but can be suppressed thus achieving a trade-off between model stability and information loss. Using the proposed ω-statistic of our own design, in Section 3.3 we demonstrated that suppressing severe outliers using Tukey's fences method does not deteriorate, but even improves model fitness for the most problematic assets, while solving the stability issues. On the other hand, our final results suggest that in general strategies run without outlier suppression performed better already on the asset preselection stage. Thus, at this stage we conclude that results of this method are mixed and require further investigation.

The same ω-statistic analysis rejected idea formulated in Section 2.1 expecting that we might get more accurate predictions by evaluating the Markov Process of wealth realizations starting with probabilities conditional on the last observed state, rather than unconditional ones. Since no prediction accuracy gains were achieved in the conducted experiment, we conclude that unconditional probabilities should be preferred as such approach is likely to be more numerically robust. Moreover, we identified that Markov Chain model performance differs substantially among assets, thus suggesting that assets might be preselected based on their concordance with Markov Chains model. We leave the additional tests of this proposal for future research as the validity of the proposed ω-statistic itself should be first tested more thoroughly, as it was developed for comparative and analytical purposes and not to be used in the actual algorithm.

Considering preselection part of the framework, we formulated a simple ranking algorithm, which

[3] https://github.com/astekas/algoport

was supposed to serve as a benchmark for the more sophisticated DEA approach. The empirical results provided in Section 3.4 suggest that while the two algorithms indeed generate alternative asset sets and our ranking algorithm was found to be more stable in time, DEA algorithm managed to preselect much better assets in terms of their performance. This confirms the applicability of DEA as a useful tool in algorithmic portfolio management, which not only reduces the dimensionality of the optimization problem, but is alone sufficient to outperform the benchmark index by equal-size investment in preselected assets.

In Section 3.5 we demonstrated the inherent challenge of semi-parametric function optimization by evaluating the space of portfolios constructed out of first two principal components of asset returns. Thus, we justified the use of heuristic Simplex optimization procedure introduced in Section 2.3. We further benchmarked it against the popular Genetic Algorithm, showing the superiority of Simplex both in terms of computational time and achieved results. Moreover, in the thesis we introduced the extension of Omega Ratio to the Markovian case as a new objective function to optimize. The empirical results confirm that such optimization target is indeed valid and demonstrates extremely good performance, especially under favorable market conditions.

In Section 2.4 we also introduced a novel regulation method, which successfully regularizes weight redistribution done by the model in time as was demonstrated in Section 3.6. Since in the real world weight redistribution is associated with transaction costs, the proposed method helps adjusting the algorithm performance dynamically, based on the current transaction cost level. As an unanticipated side effect, the empirical results suggest that this method also tends to increase sparsity of the final portfolios. A clear limitation of the proposed method is that it deals only with weight turnover, while part of it is irreducible due to asset turnover. Therefore, we also introduced the so-called R2 regulation method, which is more straightforward in it's nature and is able to handle both asset and weight turnover.

Concluding this thesis, we would like to briefly mention several approaches and ideas that were considered during the current research, but were not fully developed due to the scope and time limitations, therefore we leave them as pathways for the future research. On the information extraction part, ARMA-GARCH-Copula family of approaches (e.g. [44]) can be very straightforwardly integrated into the developed framework with the initial rough implementation already available in the software implementation, developed alongside this thesis, "Algoport". As opposed to the Markov Chains model used in the current work, this approach is fully parametric and can be briefly summarized as modelling individual return series with ARMA-GARCH family of models and capturing their joint distribution with a copula model of ARMA-GARCH residuals. Then, a Monte Carlo simulation of possible realizations for future time can be drawn in order to forecast the distribution of the whole market returns, similarly to how we do it for a single series with a Markov Chain. We deem that such approach might be a viable and more flexible alternative to Markov Chains, able to capture more complex dependencies between the assets and, thus, possibly resulting in more stable and better performing strategies.

During the current study, we also formulated a novel approach for asset preselection, yet did not have sufficient time to test it. Following the idea of factor-based portfolios found in literature, we suggest that Non-negative Matrix Factorization (NMF) [19] might be a more suitable tool than Principal Component Analysis, widely used for this purpose. The problem with PCA, which we slightly touch upon in Section 3.5, is that while the component scores can be interpreted as latent factors, based on the underlying asset returns given certain loading weights, which implies that the factors themselves can be optimized in-line with the approaches, discussed within this thesis, the conversion back from the weights in latent space to asset weights is not straightforward. Component loadings cannot be directly interpreted as such "asset weight" in the component, as they are not constrained to be non-negative. [6] On the other hand, NMF strives to find an optimal representation of a non-negative matrix as a product of 2 non-negative matrices, which can be interpreted as scores and loading in PCA terminology. As returns are non-negative by their nature, the first condition is satisfied. Then, the non-negative loadings produced by the method can be normalized to sum up to one, essentially making each NMF "factor" a valid portfolio. From our brief tests, it seemed that such procedure allows representing the whole market as a several valid portfolios, which represent approximately same percentage of variance as the same number of principal components. However, the certain advantage of this method is that once we allocate the weights to the representative portfolios, conversion to actual asset weights is as simple as matrix multiplication of factor weights by the normalized loadings. Clearly, further research is necessary to validate this method, but we find it very promising and thus considered for inclusion in "Algoport".

References

[1] Amine Mohamed Aboussalah, Ziyun Xu, and Chi Guhn Lee. What is the value of the cross-sectional approach to deep reinforcement learning? *Quantitative Finance*, 22(6):1091–1111, 2022.

[2] Divya Aggarwal. Defining and measuring market sentiments: a review of the literature. *Qualitative Research in Financial Markets*, 14(2):270–288, 2022.

[3] Enrico Angelelli, Sergio Ortobelli, and Anna Bianchi. Financial applications of bivariate markov processes. *Mathematical Problems in Engineering*, 2011:15 pages, 01 2011.

[4] Enrico Angelelli, Sergio Ortobelli, and Gaetano Iaquinta. Garch type portfolio selection models with the markovian approach. *International Journal of Mathematical Models and Methods in Applied Sciences*, 5:308–315, 01 2011.

[5] Ronil Barua and Anil K. Sharma. Dynamic Black Litterman portfolios with views derived via CNN-BiLSTM predictions. *Finance Research Letters*, 49(June):103111, 2022.

[6] Robert Bass, Scott Gladstone, and Andrew Ang. Total portfolio factor, not just asset, allocation. *The Journal of Portfolio Management*, 43:38–53, 03 2017.

[7] Szymon Borak, Wolfgang Härdle, and Rafał Weron. Stable distributions. *Statistical Tools for Finance and Insurance*, pages 21–44, 2005.

[8] Szymon Borak, Wolfgang Karl Härdle, and Rafał Weron. Stable distributions. *Statistical Tools for Finance and Insurance*, 04 2005.

[9] Michael W. Brandt, Pedro Santa-Clara, and Rossen Valkanov. Parametric Portfolio Policies: Exploiting Characteristics in the Cross-Section of Equity Returns. *Review of Financial Studies*, 22(9):3411–3447, September 2009.

[10] Divyanshu Daiya and Che Lin. Stock movement prediction and portfolio management via multimodal learning with transformer. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3305–3309, 2021.

[11] Juan Du. Mean–variance portfolio optimization with deep learning based-forecasts for cointegrated stocks. *Expert Systems with Applications*, 201(March):117005, 2022.

[12] Francesco A Fabozzi, Joseph Simonian, and Frank J Fabozzi. Risk parity: The democratization of risk in asset allocation. *The Journal of Portfolio Management*, 47(5):41–50, 2021.

[13] Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, 1993.

[14] Eugene F. Fama and Kenneth R. French. A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22, 2015.

[15] Alireza Ghahtarani, Ahmed Saif, and Alireza Ghasemi. *Robust portfolio selection problems: a comprehensive review*, volume 22. Springer Berlin Heidelberg, 2022.

[16] Abhishek Gunjan and Siddhartha Bhattacharyya. *A brief review of portfolio optimization techniques*. Number 0123456789. Springer Netherlands, 2022.

[17] Bruno Miranda Henrique, Vinicius Amorim Sobreiro, and Herbert Kimura. Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, 124:226–251, 2019.

[18] Mohammad Mehdi Hosseinzadeh, Sergio Ortobelli Lozza, Farhad Hosseinzadeh Lotfi, and Vittorio Moriggia. Portfolio optimization with asset preselection using data envelopment analysis. *Central European Journal of Operations Research*, 2022.

[19] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5:1457–1469, dec 2004.

[20] Szu Hao Huang, Yu Hsiang Miao, and Yi Ting Hsiao. Novel Deep Reinforcement Algorithm with Adaptive Sampling Strategy for Continuous Portfolio Optimization. *IEEE Access*, 9:77371–77385, 2021.

[21] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem. pages 1–31, 2017.

[22] Michalis Kapsos, Steve Zymler, Nicos Christofides, and Berç Rustem. Optimizing the omega ratio using linear programming. *Journal of Operational Risk*, 9, 06 2014.

[23] Con Keating and William Shadwick. A universal performance measure. *Journal of Performance Measurement*, 6, 01 2002.

[24] J. L. Kelly. A new interpretation of information rate. *The Bell System Technical Journal*, 35(4):917–926, 1956.

[25] Yongjae Lee, Min Jeong Kim, Jang Ho Kim, Ju Ri Jang, and Woo Chang Kim. Sparse and robust portfolio selection via semi-definite relaxation. *Journal of the Operational Research Society*, 71(5):687–699, 2020.

[26] Bin Li. OLPS - On-Line Portfolio Selection via Machine Learning. 17:1–5, 2016.

[27] Bin Li and Steven C.H. Hoi. Online portfolio selection: A survey. *ACM Computing Surveys*, 46(3), 2014.

[28] Zhixi Li and Vincent Tam. A Novel Meta-Heuristic Optimization Algorithm Inspired by the Spread of Viruses. (Zhixi Li), 2020.

[29] Sergio Ortobelli Lozza, Enrico Angelelli, and Daniele Toninelli. Set-portfolio selection with the use of market stochastic bounds. *Emerging Markets Finance and Trade*, 47(SUPPL. 5):5–24, 2011.

[30] Giorgio Lucarelli and Matteo Borrotti. A deep Q-learning portfolio management framework for the cryptocurrency market. *Neural Computing and Applications*, 32(23):17229–17244, 2020.

[31] Matthew R. Lyle and Teri Lombardi Yohn. Fundamental Analysis and Mean-Variance Optimal Portfolios. *Accounting Review*, 96(6):303–327, 2022.

[32] Yilin Ma, Ruizhu Han, and Weizhong Wang. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, 165(June 2020):113973, 2021.

[33] Babak Mahdavi-Damghani, Konul Mustafayeva, Cristin Buescu, and Stephen Roberts. Portfolio optimization for cointelated pairs: SDEs vs Machine learning. *Algorithmic Finance*, 8(3-4):101–125, 2020.

[34] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, March 1952.

[35] Harry Markowitz. The optimization of a quadratic function subject to linear constraints. *Naval Research Logistics Quarterly*, 3(1-2):111–133, 1956.

[36] J. Huston McCulloch. Simple consistent estimators of stable distribution parameters. *Communications in Statistics - Simulation and Computation*, 15(4):1109–1136, 1986.

[37] K. Metaxiotis and K. Liagkouras. Multiobjective Evolutionary Algorithms for Portfolio Management: A comprehensive literature review. *Expert Systems with Applications*, 39(14):11685–11698, 2012.

[38] Liangyu Min, Dewen Liu, Xiaohong Huang, and Jiawei Dong. Worst-case Mean-VaR Portfolio Optimization with Higher-Order Moments. 30(1), 2022.

[39] Sergio Ortobelli, Enrico Angelelli, and Alda Ndoci. Timing portfolio strategies with exponential lévy processes. *Computational Management Science*, 16, 02 2019.

[40] Sergio Ortobelli Lozza, Enrico Angelelli, and Annamaria Bianchi. Financial applications of bivariate Markov processes. *Mathematical Problems in Engineering*, 2011, 2011.

[41] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications: A survey. *Applied Soft Computing Journal*, 93, 2020.

[42] Pejman Peykani, Emran Mohammadi, Armin Jabbarzadeh, and Alireza Jandaghian. Utilizing robust data envelopment analysis model for measuring efficiency of stock, a case study: Tehran stock exchange. 1:15–24, 01 2016.

[43] B. Y. Qu, Q. Zhou, J. M. Xiao, J. J. Liang, and P. N. Suganthan. Large-Scale Portfolio Optimization Using Multiobjective Evolutionary Algorithms and Preselection Methods. *Mathematical Problems in Engineering*, 2017, 2017.

[44] Maziar Sahamkhadam, Andreas Stephan, and Ralf Östermark. Portfolio optimization based on GARCH-EVT-Copula forecasting models. *International Journal of Forecasting*, 34(3):497–506, 2018.

[45] Yoshiharu Sato. Model-Free Reinforcement Learning for Financial Portfolios: A Brief Survey. pages 1–20, 2019.

[46] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing Journal*, 90:106181, 2020.

[47] Meeta Sharma and Hardayal Singh Shekhawat. Intelligent portfolio asset prediction enabled by hybrid Jaya-based spotted hyena optimization algorithm. *Kybernetes*, 50(12):3331–3366, 2021.

[48] William F. Sharpe. Mutual fund performance. *The Journal of Business*, 39(1):119–138, 1966.

[49] Farzan Soleymani and Eric Paquet. Deep graph convolutional reinforcement learning for financial portfolio management – DeepPocket. *Expert Systems with Applications*, 182(April):115127, 2021.

[50] Stoyan Stoyanov, Gennady Samorodnitsky, Svetlozar Rachev, and Sergio Ortobelli. Computing the portfolio conditional value-at-risk in the alpha-stable case. *Probability and Mathematical Statistics*, 26, 08 2005.

[51] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[52] Kaoru Tone. A slacks-based measure of efficiency in data envelopment analysis. *European Journal of Operational Research*, 130(3):498–509, 2001.

[53] Nhi N.Y. Vo, Xuezhong He, Shaowu Liu, and Guandong Xu. Deep learning for decision making and the optimization of socially responsible investments and portfolio. *Decision Support Systems*, 124(July):113097, 2019.

[54] Elaine Wah, Yi Mei, and Benjamin W. Wah. Portfolio optimization through data conditioning and aggregation. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 253–260, 2011.

[55] Wuyu Wang, Weizi Li, Ning Zhang, and Kecheng Liu. Portfolio formation with preselection using deep learning from long-term financial data. *Expert Systems with Applications*, 143:113042, 2020.

[56] Frank Z. Xing, Erik Cambria, and Roy E. Welsch. Natural language based financial forecasting: a survey. *Artificial Intelligence Review*, 50(1):49–73, 2018.

[57] Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. Qlib: An AI-oriented Quantitative Investment Platform. 2020.

[58] Lean Yu, Shouyang Wang, and Kin Keung Lai. Multi-attribute portfolio selection with genetic optimization algorithms. *Infor*, 47(1):23–30, 2009.

APPENDICES

A Algoport package

Algoport[4] is a Python package implementation of the framework proposed in this work. As of version 0.1.2 it consists of the following modules:

- Markov. Contains a single class of MarkovChainModel, which is our original Python implementation of the information extraction model described in this paper.

- AssetSelection. Defines the class hierarchy for asset preselection algorithms. DEA_AS is a wrapper around R "additiveDEA" package, accessed via rpy2 bridge. Ranking_AS is our pure Python implementation of the Ranking preselection algorithm.

- PortfolioOptimization. Defines the class hierarchy for weight optimization algorithms. Currently available optimizers include: SimplexOptimizer, which is our original implementation of the Simplex optimization algorithm described in Section 2.3; SciPy, providing a wrapper around SciPy optimization algorithms, which handle constraints necessary for weight optimization; PyMoo providing a wrapper around GeneticAlgorithm implementation in PyMoo package. In future versions it will be extended to the multitude of algorithms available therein; MVO, which is a classical Mean-Variance Optimization algorithm, solved by the means of quadratic programming.

- Strategy is the main class, which serves as a decision making hub. It collects the history of portfolio performance over subsequent evaluations and provides the endpoint for implementing various regulation approaches.

- Backtesting. A technical module, defining Backtest class, which iteratively passes data to Strategy, computes and records the final test run metrics, stores and plots the results.

[4] https://github.com/astekas/algoport

- Data. Technical module with a single Dataset, which class manages the datasets included in the package. At the moment, only SNP500 is available.

More detailed usage guidelines will be consistently added to git repository.

B Bivariate Markov process estimation

The Markov process of portfolio wealth estimation procedure, described in section 2.1 for univariate case can be further extended to bivariate case using the following method. Suppose that we have two series of returns $z_{x,t}, z_{y,t}$, each following a homogeneous Markov process. Then, we can approximate their joint behavior with a bivariate Markov process $Z_t = (z_{x,t}, z_{y,t})$. In order to describe it we need to estimate the states of this process, transition matrix and the initial state. We assume that both individual processes have the same number of states S. In fact, the formulation given in [3] does not impose such constraint, but the implementation can be made more efficient by assuming it. Then the bivariate process has S^2 states $s^{(i)} = (s_x^{(i_x)}, s_y^{(i_y)}) \forall i \in 1, ..., N^2$. In order to estimate them we consider the range of K past returns given by

$$(\min_{t \in -K,...,0} z_{x,t}, \max_{t \in -K,...,0} z_{x,t}) \text{ x } (\min_{t \in -K,...,0} z_{y,t}, \max_{t \in -K,...,0} z_{y,t}) \tag{32}$$

This range is then divided into S^2 bidimensional intervals (a_i, a_{i-1}) x (b_j, b_{j-1}), where a_i and b_j are given by:

$$\begin{aligned} a_i &:= (\frac{\min_t z_{x,t}}{\max_t z_{x,t}})^{\frac{i-1}{S}} \max_t z_{x,t}, i = 1, ..., S+1 \\ b_j &:= (\frac{\min_t z_{y,t}}{\max_t z_{y,t}})^{\frac{j-1}{S}} \max_t z_{y,t}, j = 1, ..., S+1 \end{aligned} \tag{33}$$

Similarly to the univariate case, the bivariate states are then estimated by the geometric average of the interval bounds.

$$\begin{aligned} s_x^{(i_x)} &:= \sqrt{a_{i_x} a_{i_x - 1}}, i_x = 1, ..., S \\ s_y^{(i_y)} &:= \sqrt{b_{i_y} b_{i_y - 1}}, i_y = 1, ..., S \end{aligned} \tag{34}$$

Respectively, we can introduce $d = (d_x, d_y)$ as a universal multiplier for alternative state representation given by:

$$\begin{aligned} s_x^{(i_x)} &:= s_x^{(1)} d_x^{1-i_x}, i_x = 1, ..., S \\ s_y^{(i_y)} &:= s_y^{(1)} d_y^{1-i_y}, i_y = 1, ..., S \\ d_x &:= (\frac{\max_t z_{x,t}}{\min_t z_{x,t}})^{\frac{1}{S}}; d_y := (\frac{\max_t z_{y,t}}{\min_t z_{y,t}})^{\frac{1}{S}} \end{aligned} \tag{35}$$

Next, we estimate the transition matrix $Q = \{q(i_x, i_y, j_x, j_y)\}_{i_x, i_y, j_x, j_y \in 1, ..., S}$. Notice that since transition matrix is a matrix of probabilities to transit from the state $s^{(i)}$ to the state $s^{(j)}$ and the states are 2-dimensional for bivariate case - transition matrix becomes 4-dimensional. More formally it is given by

$$q_{i,j} = P(Z_{t+1} = s^{(j)} | Z_t = s^{(i)}) \tag{36}$$

It can be estimated non-parametrically, by counting the number of times empirical returns fall within the interval corresponding to a given state $s^{(i)}$ and how many times transitions to each other state $s^{(j)}$

occurred from it.

$$\begin{aligned}
\tilde{q}_{i,j} &= \frac{\pi_{i,j}}{\pi_i} \\
\pi_{i,j} &= \sum_{t=-K}^{0} I(z_x \in (a_x^{(j_x)}, a_x^{(j_x-1)}) \cap z_y \in (b_y^{(j_y)}, b_y^{(j_y-1)}) | z_x \in (a_x^{(i_x)}, a_x^{(i_x-1)}) \cap z_y \in (b_y^{(i_y)}, b_y^{(i_y-1)})) \\
\pi_i &= \sum_{t=-K}^{0} I(z_x \in (a_x^{(i_x)}, a_x^{(i_x-1)}) \cap z_y \in (b_y^{(i_y)}, b_y^{(i_y-1)}))
\end{aligned} \tag{37}$$

Where I is an indicator operator taking value of 1 if condition is true and 0 otherwise. Next the bivariate wealth process can be evaluated similarly to univariate case. Recall from section 2.1 that a univariate return process generates (S-1)h + 1 possible wealth values at time T+h in the future. Respectively, the bivariate wealth process can reach $((S-1)h+1)^2$ values of wealth at time T+h in the future. They are given by

$$\begin{aligned}
& w^{l,h} = (w_x^{(l_x,h)}, w_y^{(l_y,h)}) \\
& w_x^{(l_x,h)} = w_x^{(l_x,0)} (s_x^{(1)})^t d_x^{1-l_x} \\
& w_{xy}^{(l_y,h)} = w_y^{(l_y,0)} (s_y^{(1)})^t d_y^{1-l_y} \\
& l = (l_x, l_y) \in L_h := \{(l_x, l_y) : 1 \leq l_x, l_y \leq 1 + h(S-1)\}
\end{aligned} \tag{38}$$

In order to evaluate the probability of attaining each of these wealth values, we need to consider the evolution of such process, which can be represented by a sequence of matrices $G_t^{(l,i)}$. Notice, that since the evolution matrix by definition contains the probabilities to attain wealth w^l, while being in the state s^i, it is a 4-dimensional matrix with dimensions given by:

$$\begin{aligned}
& G_t^{(l,i)} = \{g_t(l,i)\}, \\
& l = (l_x, l_y) \in L_t := \{(l_x, l_y) : 1 \leq l_x, l_y \leq 1 + h(S-1)\} \\
& i = (i_x, i_y) \in I := \{(i_x, i_y) : 1 \leq i_x, i_y \leq S\}
\end{aligned} \tag{39}$$

In order to initiate the process, we need to fix the starting wealth $w^{l,0} = (1,1)$ and the starting state probabilities u, where $u^i = \frac{\pi_i}{T}$, which essentially represents the unconditional probability of being in each state. Then, the sequence of matrices $G_t^{(l,i)}$ can be computed iteratively from the following scheme:

$$g_t(l,i) = \begin{cases} u^i, & t = T, l = (1,1), \\ \sum_h g_{t-1}(l-k+1,k) q(k,i), & \begin{array}{l} 1 \leq l_x - k_x + 1 \leq (1+(h-1)(S-1))^2, \\ 1 \leq l_y - k_y + 1 \leq (1+(h-1)(S-1))^2 \end{array} \end{cases} \tag{40}$$

This computation scheme differs from the one formulated in [3], statement 2.15, rather dramatically. The reason for this is that we believe the original formulation to be incorrect and somewhat misleading. First of all, the original formulation did not feature the upper constraints on the indices from g_{t-1} that are to be used, making the direct implementation of it infeasible, as it required taking values of g_{t-1} which simply do not exist as dimensionality expends with time. Secondly, the original formulation imposed constraints on the values of i, rather than h to be taken, which again can be proved wrong very easily, as it essentially implies that transition probabilities to some states in G_t are to be set to 0, while in the Markov chain, with starting probabilities $u^i \neq 0 \forall i \in I$ and transition probabilities $q_{i,j} \neq 0, \forall i, j \in I$ there will always be non-zero probability to be in any state for any attained wealth. Finally, the probabilities $P^l = \{p(l)\}$ to obtain the wealth w^l at time T+h are given by:

$$p_t(l) = \begin{cases} 1, & t = 0, l = (1,1), \\ \sum_{k \in I} g_t(l,k) & \end{cases} \tag{41}$$

Thus the probabilities P^l together with w^l fully define the distribution of future wealth at time T W_T, which is two-dimensional for bivariate case.

C Time to gain or lose distribution approximation

In this appendix we outline the approach used for estimation of the distribution of time to gain/lose certain portion of wealth (W_T) before the forecasting horizon T, assuming that it follows Markov process estimated as described in Section 2.1. We implement the method as described in [39].

We denote by stopping time $\tau_T = \arg\min_t W_t \in A$ as the shortest passage of time, such that portfolio wealth following Markov process reaches the value below (loss) or above (gain) certain threshold (denoted as condition A) before the stopping time T. Respectively, we want to evaluate the distribution of it $\pi(\tau_T) = \mathbb{P}(\tau = t | W_t \in A)$, $t = 1, \ldots, T$. Consider two sequences of matrices $\hat{Q}^{(t)} = \{\hat{q}^{(t)}_{l,s}\}_{1 \leq l \leq (S-1)t+1, 1 \leq s \leq S}$ and $\tilde{Q}^{(t)} = \{\tilde{q}^{(t)}\}_{1 \leq l \leq (S-1)t+1, 1 \leq s \leq S}$, where $t = 0, 1, ..., T$ is the step of the Markov process and S is the number of states. $\tilde{Q}^{(t)}$ represents the matrix of probabilities of obtaining, at time t, a wealth $w^{(l)}$, while being in the state s and never obtaining wealth satisfying the condition A at previous steps $k = 0, \ldots, t-1$. At the same time, $\hat{Q}^{(t)}$ represents the matrix of probabilities of obtaining at time t wealth $w^{(l)} \notin A$, while being in the state s and never obtaining wealth belonging to A at previous steps $k = 0, ..., t-1$. Thus, $\tilde{q}^{(t)}_{l,s} = 0$ if $w^{(l)} \in A$.

These matrices can be computed sequentially, starting with $\tilde{Q}^{(0)} = u$ and $\hat{Q}^{(t)} = u$, where u is the vector of unconditional probabilities to be in each of the states at time 0. We suppose that the starting wealth $w^{(1)}_0 = 1 \notin A$ at time $t = 0$. Then, the sequence of matrices is estimated iteratively as follows:

$$\begin{aligned} \tilde{Q}^{(t)} &= \text{diagM}(\hat{Q}^{(t-1)} P), \\ \hat{Q}^{(t)} &= \text{ZeroA}(\tilde{Q}^{(t)}), \end{aligned} \tag{42}$$

for $t = 1, ..., T$, where P is the transition matrix of the Markov process estimated as described in Section 2.1, diagM is an operation shifting each column j of the given matrix down by $j - 1$ and ZeroA denotes the operation where we set to 0 all the rows $l = 1, \ldots, 1 + t(S-1)$ of the matrix $\tilde{Q}^{(t)}$ corresponding to the wealth $w^{(l)} \in A$. Finally, the distribution of probabilities is given by

$$P(\tau = t) = \begin{cases} e^{(t)'} \tilde{Q}^{(t)} \mathbf{1}_S, \text{ for } t = 1, ..., T-1, \\ \mathbf{1}'_{(S-1)t+1} \tilde{Q}^{(t)} \mathbf{1}_S, \text{ for } t = T, \end{cases} \quad e^{(t)}_s = \begin{cases} 1, \text{ if } w^{(l,t)} \in A, \\ 0, \text{ otherwise} \end{cases} \tag{43}$$

D Stable distribution of W_T: parameters estimation, CVaR, CDF

Several statistics used in the current work rely on the stable distribution of future wealth W_T, so we believe that it is necessary to outline all the details of its estimation. First of all, recall that we denote by W_T the distribution of the future wealth which can be attained by a portfolio following Markov process in T time steps starting from t = 0, $W_0 = 1$. This distribution is described by 2 vectors: w_t of attainable wealth values and p_t of probabilities to attain them and it's estimation is described in section 2.1. It is suggested by [40] that this discrete distribution is best approximated by a stable non-Gaussian law. Thus, we need to estimate 4 parameters: α, β, σ and μ to fully characterize it. μ is a standard location parameter, which simply shifts the distribution to left or right. σ is a scale parameter, which compresses or extends the distribution relative to μ. β defined on the $[-1, 1]$ interval is also known as skewness and is a measure of asymmetry of the distribution. Finally, α is known as characterizing parameter and determines the heaviness of distribution tails. It should be mentioned that in general α is defined as lying in the range $(0, 2]$, however for $\alpha \leq 1, \mu$ is undefined. Therefore, both we and original authors [18] assume that $\alpha \in (1, 2]$ in our particular case. To estimate these parameters we utilize the quantiles method proposed by [36], which allows estimating stable distribution parameters from 5 quantiles of the given observed distribution: $q_{0.05}, q_{0.25}, q_{0.5}, q_{0.75}, q_{0.95}$. We do not describe the full method here as it was implemented without any modification, just mention our approach to computing the quantiles, since we need to estimate quantiles of W_t, rather than some observed sample. Assuming that w_t and p_t are sorted in increasing order, we first compute the empirical cumulative distribution $\tilde{C}_{W_t} = \{c_{W_t}(i)\}_{i \in 1,\ldots,1+t(S-1)}$ as

$$c_{W_t}(i) = \sum_{n=1}^{i} p_t(n) \tag{44}$$

Then any quantile $q_m(W_t)$ can be found as

$$q_m(W_t) = w_t(argmax_i(c_{W_t}(i) \leq m)) \tag{45}$$

From the estimates of 5 quantiles we then find the estimates of the stable distribution parameters $\tilde{\alpha}, \tilde{\beta}, \tilde{\sigma}$ and $\tilde{\mu}$.

Conditional Value at Risk (CVaR) is a common risk measure quantifying the expected loss in case of the most negative scenario realization. [50] suggested the following formulation of CVaR for stable distributions, which we use in this work:

$$\begin{aligned}
CVaR_\epsilon(X) &= \frac{\alpha|VaR_\epsilon(X)|}{(1-\alpha)\pi\epsilon} \int_{-\tilde{\theta}_0}^{\frac{\pi}{2}} g(\theta) exp(-|VaR_\epsilon(X)|^{\frac{\alpha}{(\alpha-1)}\nu(\theta)}) d\theta \\
g(\theta) &= \frac{\sin(\alpha(\tilde{\theta}_0+\theta)-2\theta)}{\sin(\alpha(\tilde{\theta}_0+\theta))} - \frac{\alpha\cos^2(\theta)}{\sin^2(\alpha(\tilde{\theta}_0+\theta))}, \\
\nu(\theta) &= (\cos(\alpha\tilde{\theta}_0))^{\frac{1}{(\alpha-1)}} (\frac{\cos(\theta)}{\sin(\alpha(\tilde{\theta}_0+\theta))})^{\frac{\alpha}{(\alpha-1)}} \frac{\cos(\alpha\tilde{\theta}_0+(\alpha-1)\theta)}{\cos(\theta)}, \\
\tilde{\theta}_0 &= \frac{1}{\alpha}\arctan(\tilde{\beta}\tan(\frac{\pi\alpha}{2})), \\
\tilde{\beta} &= -sign(VaR_\epsilon(X))\beta
\end{aligned} \tag{46}$$

$VaR_\epsilon(X)$ is defined by $P(X \leq -VaR_\epsilon(X)) = \epsilon$, which makes $-VaR_\epsilon(X)$ an ϵ quantile of the distribution of X, or, in our case the estimated stable distribution. This formulation is given for the standardized case, however given the translation and scale invariance properties of it, they are incorporated back by:

$$CVaR_\epsilon(\sigma X + \mu) = \sigma CVaR_\epsilon(X) - \mu \tag{47}$$

Finally, this work also utilizes the cumulative density function of the estimated stable distribution (CDF). For it we use the formulation originally suggested in [7]. As we assume that $\alpha \in (1, 2]$, we use only the corresponding cases.

$$\begin{aligned} F(X,\alpha,\beta) &= \begin{cases} 1 - \frac{1}{\pi}\int_{-\tilde{\theta}_0}^{\frac{\pi}{2}} exp(-(X-\zeta)^{\frac{\alpha}{\alpha-1}})\nu(\theta)d\theta, \text{ if } X > \zeta, \\ \frac{1}{\pi}(\frac{\pi}{2} - \tilde{\theta}_0), \text{ if } X = \zeta, \\ 1 - F(-X;\alpha,-\beta), \text{ if } X < \zeta \end{cases} \\ \zeta &= -\beta\tan(\frac{\pi\alpha}{2}) \\ \tilde{\theta}_0 &= \frac{1}{\alpha}\arctan(-\zeta) \end{aligned} \tag{48}$$