COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS



Clustering and shrinkage methods in the portfolio optimization problems

MASTER THESIS

2019 Bc. Andrej Remák

COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS
DEPARTMENT OF APPLIED MATHEMATICS AND STATISTICS

Clustering and shrinkage methods in the portfolio optimization problems

MASTER THESIS

Study Programme:	Mathematical Economics, Finance and Modelling
Field of Study:	1114 Applied Mathematics
Department:	Department of Applied Mathematics and Statistics
Supervisor:	Mgr. Katarína Lučivjanská, PhD.

Bratislava 2019 **Bc. Andrej Remák**





Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Bc. Andrej Remák
Study programme:	Mathematical Economics, Finance and Modelling (Single degree study, master II. deg., full time form)
Field of Study:	Applied Mathematics
Type of Thesis:	Diploma Thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Clustering and shrinkage methods in the portfolio optimization problems

Annotation: With a large number of assets, investors face uncertainty that the theoretically optimal portfolio will not be superior in real-world applications. The set of input parameters is namely too large to be estimated with sufficient certainty. In the Diploma thesis, the student will analyze possible ways to deal with this issue and implement several suggested solutions on historical data. Furthermore, the student will provide a framework in which the approaches can be evaluated, compared and discuss their advantages and disadvantages.

Supervisor:	Mgr. Katarína Lučivjanská, PhD.
Department:	FMFI.KAMŠ - Department of Applied Mathematics and Statistics
Head of department:	prof. RNDr. Marek Fila, DrSc.
Assigned:	24.01.2018
Approved:	25.01.2018

prof. RNDr. Daniel Ševčovič, DrSc.
Guarantor of Study Programme

.....
Student

.....
Supervisor





Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Andrej Remák
Študijný program: ekonomicko-finančná matematika a modelovanie (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: aplikovaná matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Clustering and shrinkage methods in the portfolio optimization problems
Klastrovacie a zmršťovacie metódy v úlohe optimálneho portfólia

Anotácia: Pri veľkom počte aktív čelia investori neistote, že sa teoreticky optimálne portfólio v reálnych aplikáciách nebude správať najlepšie. Množina vstupných parametrov je totiž príliš veľká, aby sa dala odhadnúť s dostatočnou istotou. V diplomovej práci si študent naštuduje možné spôsoby, ako sa s týmto problémom vysporiadať a implementuje ich na reálnych dátach. Navyše, študent navrhne koncepciu, pomocou ktorej vie jednotlivé prístupy ohodnotiť, porovnať a zvážiť ich výhody a slabé stránky.

Vedúci: Mgr. Katarína Lučivjanská, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Marek Fila, DrSc.

Dátum zadania: 24.01.2018

Dátum schválenia: 25.01.2018 prof. RNDr. Daniel Ševčovič, DrSc.
garant študijného programu

.....
študent

.....
vedúci práce

Acknowledgement I would first like to thank my supervisor Mgr. Katarína Lučivjanská PhD, whose expertise was invaluable in the process of writing the thesis. Katarína Lučivjanská was always patient and available whenever I had a question about my research or writing. She consistently steered me in the right direction whenever she thought I needed it. It was a pleasure to work with her not only because she is an incredibly knowledgeable person in the field, but also because she is very kind and friendly.

I would also like to express my profound gratitude to my family and friends for providing me with continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Poďakovanie Rád by som sa touto cestou poďakoval mojej vedúcej diplomovej práce Mgr. Kataríne Ľučivjanskej PhD, ktorej skúsenosti pri písaní tejto práce boli neoceniteľné. Katarína Ľučivjanská bola vždy veľmi trpezlivá a ochotná zodpovedať všetky moje otázky. Keď sa jej zdalo potrebné ma naviesť na správnu cestu, vždy to urobila, čo si veľmi vážim. Bolo mi potešením s ňou pracovať, nielen preto, že je veľmi rozhľadená vo svojom obore, ale aj preto, že je veľmi milá a ľudská.

Takisto by som sa chcel veľmi poďakovať mojej rodine a kamarátom, ktorí mi boli oporou počas celého štúdia a podporovali ma počas písania tejto práce.

Abstract

REMÁK, Andrej: Clustering and shrinkage methods in the portfolio optimization problems [Master Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: Mgr. Katarína Lučivjanská, PhD., Bratislava, 2019, 60 p.

The aim of the thesis is to find the best performing asset allocation strategy, which addresses the parameter uncertainty problem. We evaluate the performance of various asset allocation models on thousands of portfolios. The finding is that none of the methods except the 1-norm-constrained portfolio performs superiorly in our hyperparameter environment. Moreover, methods seem to behave inconsistently in time, which should be taken into consideration. As a solution to this issue we provide a general framework using neural networks for choosing the superiorly performing asset allocation strategy for a given dataset. Thanks to the strategy switching we should be able to beat the static approach in all considered performance measures except turnover. Nevertheless, the strategy switching would be attractive only if the generated return more than offsets for the higher turnover expenses.

Keywords: portfolio optimization, parameter uncertainty, shrinkage, clustering, neural networks, LSTM

Abstrakt v štátnom jazyku

REMÁK, Andrej: Klastrovacie a zmršťovacie metódy v úlohe optimálneho portfólia [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: Mgr. Katarína Lučivjanská PhD, Bratislava, 2019, 60 s.

Cieľom našej práce je nájsť najlepšie sa správajúcu stratégiu optimálneho portfólia, ktorá si dokáže poradiť s neistotou v odhade parametrov. Výkonnosť viacerých metód overíme na tisíckach dátových setov. Zistili sme, že ani jedna z metód okrem 1-normou-ohraničeného portfólia nepreukazuje výrazne lepšie vlastnosti ako ostatné metódy. Navyše pozorujeme, že metódy sa v priebehu času správajú nekonzistentne, čo by malo byt zohľadnené pri investovaní. Ako riešenie tohto problému poskytujeme všeobecný nástroj využívajúci neurónové siete, ktorý pre daný dátový set predpovedá lepšie sa správajúcu stratégiu optimálneho portfólia. Vďaka takémuto dynamickému spôsobu zmeny stratégii, by sme mali oproti štandardnému statickému prístupu dosiahnuť lepšiu výkonnosť, až na obrat portfólia. Avšak, takýto prístup by bol atraktívny iba vtedy, ak by zvýšený výnos prevýšil výššie transakčné náklady vyplývajúce zo zvýšeného obratu.

Kľúčové slová: optimálne portfólio, neistota v parametroch, zmršťovanie, klastrovanie, neurónové siete, LSTM

Contents

Introduction **10**

1 General **12**

1.1 Basics 12

1.2 Performance measures 13

2 Asset Allocation Methods **16**

2.1 Naive Portfolio 16

2.2 Sample Tangency Portfolio 16

2.3 Suboptimal Weights 17

2.3.1 Plug-In Approach 19

2.3.2 Bayesian Approach 20

2.3.3 Optimal Two-Fund Rule 21

2.4 Norm-Constrained Portfolio 23

2.4.1 The 1-Norm-Constrained Portfolio 24

2.4.2 The 2-Norm-Constrained Portfolio 25

2.4.3 Cross - Validation 25

2.5 Partial Minimum-Variance Portfolios 26

2.6 Autoregressive models 27

2.6.1 Autoregression 28

2.6.2 Vector Autoregression 29

2.7 Shrinkage 30

2.8 Clustering 31

2.8.1 Agglomerative Clustering 31

2.8.2 Portfolio Optimal Granularity 33

3 Empirical overview **35**

3.1 Data 35

3.2 Analysis of methods 36

3.2.1 Hyperparameters setting 37

3.2.2 Results 39

3.3 Results - Clustering 43
3.4 Macroeconomic Factors 44

4 Dynamic Approach **48**
4.1 Benchmarks 48
4.1.1 Simple Moving Average 48
4.1.2 Autoregressive model 48
4.1.3 Seeing the future 49
4.2 Neural Networks 49
4.2.1 Feedforward Neural Networks 51
4.3 Long Short-Term Memory Network 51
4.4 Dynamic Approach - Comparison 52
4.5 LSTM - Further Improvements 54

Conclusion **56**

References **57**

Appendix **60**

Introduction

Markowitz's seminal work [21] shows that the investor who cares only about the mean and variance of portfolio returns should hold a portfolio on the efficient frontier. Conventionally, implementation of such portfolio via sample estimates is notorious for producing extreme weights and poor out-of-sample performance. The well known fundamental problem of the standard Markowitz model is the ignorance of the estimation risk. This is addressed by many researchers who propose their alternatives, built on the original model.

Kan and Zhou [16] focus on maximizing the expected out-of-sample performance, by optimally combining the sample tangency portfolio and a riskless asset. Researchers Brown [3], Klein and Bawa [17], and Stambaugh [27] take advantage of the Bayesian statistics. Wolf and Ledoit [19, 20] propose a shrinkage technique used to improve the covariance matrix estimation. DeMiguel, Garlappi, Nogales, Uppal [9] generalize this approach by solving the minimum-variance portfolio under norm-constrained asset weights. Independent of the chosen portfolio selection rule, as proposed by Bjerring, Ross and Weissensteiner [1], Branger, Ľučivjanská and Weissensteiner [2] and Fabrizio, Gallegati, Mantegna and Tola [11], dimension reduction technique such as clustering methods can also improve the out-of-sample performance.

Motivated by the work of DeMiguel, Garlappi, Uppal [8], where they find that none of the 14 asset allocation models available in the literature (2009) is consistently better than the naive $1/N$ portfolio, we set the objective to find the best performing asset allocation model. To achieve this we create randomly thousands of datasets based on real data, which distinguishes our work from the rest of the literature using either only a few datasets or simulated data. During the empirical overview we observe a varying performance of presented methods, i.e. the best performing method in one year is not necessarily the best performing method in another year. Thus we set a second objective, to develop a general framework for choosing the superiorly performing asset allocation strategy for given dataset.

The thesis consists of 4 chapters. In Chapter [1], we describe briefly the classical mean-variance portfolio optimization and performance measures used in the empirical overview. Chapter [2] presents clustering and shrinkage techniques, discuss various

asset allocation models and highlight their key features. In Chapter [3] we focus on the empirical comparison of all considered strategies evaluated on thousands of datasets, generated from real data, providing very robust results. In addition to the standard use of presented models we also apply them on the reduced data universe, which we achieve by utilizing clustering algorithms. We then focus on the inconsistent behaviour of the discussed methods, presumably driven by macroeconomic factors. Motivated by these findings, in the last Chapter [4] we propose a few strategy switching approaches based on the one step ahead prediction. The most sophisticated strategy switching approach is built on the Long Short-Term Memory Neural Network optimized over a multidimensional grid of hyperparameters.

1 General

Firstly we discuss the classical mean-variance portfolio optimization according to Markowitz [21] and then follow with performance measures which are essential for our empirical overview.

1.1 Basics

In our thesis we focus on the standard portfolio choice problem of an investor who chooses a portfolio in the universe of N risky assets and a riskless asset. Denote by R_t a vector of returns on the risky assets and R_{ft} for the riskless asset at time t. We work with excess returns that are defined as $r_t = R_t - R_{ft}1_N$, where 1_N is an N-vector of ones. It is assumed that excess returns r_t are independent and identically distributed over time following a multivariate normal distribution with mean μ and covariance matrix Σ.

The excess portfolio return at time t is $r_{pt} = w^T r_t$, where $w = (w_1, ..., w_N)$ are portfolio weights. Mean and variance of the portfolio are given by $\mu_p = w^T\mu$ and $\sigma_p^2 = w^T\Sigma w$. $1_N^T w$ is the amount of wealth invested into risky assets and $1 - 1_N^T w$ is invested into risk free. The objective of the investor is to maximize her mean-variance objective function

$$U(w) = \mu_p - \frac{\gamma}{2}\sigma_p^2, \tag{1}$$

where γ is the coefficient of relative risk aversion. When μ and Σ are known, the solution to the investor's optimal portfolio choice problem is

$$w^* = \frac{1}{\gamma}\Sigma^{-1}\mu, \tag{2}$$

and the resulting expected utility is

$$U(w^*) = \frac{1}{2\gamma}\mu'\Sigma^{-1}\mu = \frac{\theta^2}{2\gamma}, \tag{3}$$

where $\theta^2 = \mu'\Sigma^{-1}\mu$ is the squared Sharpe ratio of the ex ante tangency portfolio of the risky assets. It is important to notice that in practice, computing the optimal weights w^* is not possible because the true μ and Σ are unknown, i.e. an investor has to take

into account parameter uncertainty. Methods that address this problem are discussed in Chapter [2].

1.2 Performance measures

To compare the performance of the methods presented in Chapter [2] we use four criteria evaluated out-of-sample, namely; portfolio Sharpe Ratio, Certainty Equivalent Return, portfolio Turnover and Cumulative Return of portfolio. In the context of our work the term out-of-sample performance means the performance that is achieved in the real world application using the given portfolio model, i.e. to compute weights we use only the information that is available at the decision date. As we need to distinguish between individual portfolio rules and its performance measures, we need a more general notation than in Section [1.1], we therefore use index i for strategy allocation method i. The portfolio weights estimation is carried out using a rolling window procedure. To use this approach, the investor has to choose a window length to perform the estimation, denoted by $\tau < T$, where T is the total number of returns in the dataset. Using the data in the rolling window, we estimate the weights for various portfolios. Then we drop the earliest data in the dataset and add a new one, in our case, we work with monthly data, as additional information in daily data seems not to be relevant for improving the portfolio performance at the monthly investment horizon. Having this updated set of data, we estimate the weights of a portfolio again. We continue with this approach until we reach the end of our data. At the end of this procedure, we have generated $T - \tau$ portfolio weight vectors for each strategy i, that is, w_t^i for $t = \tau, \ldots, T-1$ for each strategy i.

Holding the portfolio weights w_t^i for one time period gives the out-of-sample return r_{t+1}^i at time $t+1$: $r_{t+1}^i = w_i^{i^T} r_{t+1}$, where r_{t+1} stands for the excess asset returns of the portfolio i. The time series of out-of-sample returns and weights for each strategy i is used to compute the out-of-sample Sharpe ratio $\hat{SR}^i$, Certainty Equivalent Return $\hat{CEQ}^i$, Turnover and Return. The out-of-sample variance and average return are calculated accordingly

$$(\hat{\sigma}^i)^2 = \frac{1}{T-\tau-1}\sum_{t=\tau}^{T-1}(w_t^{i^T} r_{t+1} - \hat{\mu}^i)^2, \tag{4}$$

$$\hat{\mu}^i = \frac{1}{T-\tau}\sum_{t=\tau}^{T-1} w_t^{i^T} r_{t+1}. \tag{5}$$

To have a fair comparison of strategies with different weights scaling adjusted to risk, Sharpe Ratio and Certainty Equivalent Return are used. Sharpe Ratio tells about the average return earned in excess of the risk-free rate per unit of volatility - risk. To implement investors risk preferences Certainty Equivalent Return is also discussed, giving information about the guaranteed return that the investor with risk aversion coefficient γ would accept instead of investing into the portfolio rule i. The main disadvantage of Sharpe Ratio is that it does not identify suboptimal weights in the portfolios, i.e. two portfolios with the same weights but scaled differently have the same Sharpe ratio, but Certainty Equivalent Return differs, as it is sensitive to scaling. The formulas for Sharpe Ratio and Certainty Equivalent Return are as follows

$$\hat{SR}^i = \frac{\hat{\mu}^i}{\hat{\sigma}^i}, \tag{6}$$

$$\hat{CEQ}^i = \hat{\mu}^i - \frac{\gamma}{2}(\hat{\sigma}^i)^2. \tag{7}$$

To have a feeling of how much the portfolio rule i rebalances assets on average, we measure its average Turnover. Such metric is very important for potential investors as it is an additional expense that needs to be accounted for. The problem with evaluating the real impact of Turnover to portfolio return is the level of commission fees that are quite complex. These days more and more services such as Robinhood, TradeZero and Revolut (launch soon) are bringing zero-commission stock trading. There are still some small obstacles to zero-commission trading, such as number of shares bought, limit orders or time of trade execution, but with growing market and competitive environment the trend is obvious, commission fees go down. Short selling is conditional by an opened margin account and additional interest rate fees, these issues are not taken into account in our Turnover formula, which treats short positions the same way as long positions. To calculate Return we use the classic discrete cumulative return formula,

an alternative of continuously compounded returns could also be used. The formulas for Turnover and Return are as follows

$$Turnover^i = \frac{1}{T-\tau-1}\sum_{t=\tau}^{T-1}\sum_{j=1}^{N}(\mid w^i_{j,t+1} - w^i_{j,t+} \mid), \tag{8}$$

$$Return^i = \left[\prod_{t=\tau}^{T-1} 1 + w_t^{i^T} r_{t+1}\right]^{1/(T-\tau)} - 1. \tag{9}$$

where in the definition of turnover, $w^i_{j,t}$ denotes the portfolio weight in asset j chosen at time t under strategy i, $w^i_{j,t+}$ the portfolio before rebalancing but at $t+1$, and $w^i_{j,t+1}$ the desired portfolio weight at time $t+1$ (after rebalancing), implying that turnover is equal to the sum of the absolute value of the rebalancing trades across the N available assets and over $T-\tau-1$ trading dates, normalized by the total number of trading dates.

2 Asset Allocation Methods

This chapter deals with the theory of various asset allocation methods that are essential for our further experiments. Optimal portfolio choice discussed in the first chapter is in mind of all mentioned methods, but the way how the problem is tackled varies a lot. According to comprehensive research, the main problem those methods have to address is the parameter uncertainty. If of interest, source code in the R language for all mentioned methods is attached on the CD, for more information see Appendix A.

2.1 Naive Portfolio

So simple and so difficult to outperform, especially if the estimation period is not sufficiently long, that is the $1/N$ strategy, which distributes wealth equally among all N risky assets, that is $w_{1/N} = (1/N, \ldots, 1/N)$. This approach is considered as a benchmark for all other methods as it is very easy to implement and surprisingly good performing. Extensive empirical research on this topic was done in the paper from DeMiguel, Garlappi and Uppal [8], in which they compare 14 optimal asset allocation models to the simple $1/N$ portfolio rule. They come to conclusion that none of the models are consistently better than the $1/N$ rule in terms of Sharpe Ratio, Certainty Equivalent Return or Turnover. This indicates that the error from estimating means and covariances diminish all the gains from the "optimal" diversification. For our further experiments, we also need and work with the $-1/N$ portfolio, that short sells all assets equally. This $-1/N$ portfolio gives us the possibility to bet on the decline in the market, when expected, e.g. when neural network predicts decline. Weights of those methods sums to 1 and -1 respectively.

2.2 Sample Tangency Portfolio

In this section we discuss the sample tangency portfolio of the famous Markowitz model [21] and its assumptions. As the out-of-sample performance of this portfolio rule is very disputable, a lot of research was done to understand and enhance its out-of-sample performance, it can be said that this portfolio rule is the backbone of our other discussed methods. Assuming that the returns r_t are independent and identically

distributed multivariate normal random variables, the sample mean and covariance matrix, computed as,

$$\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} r_t \tag{10}$$

$$\hat{\Sigma} = \frac{1}{T}\sum_{t=1}^{T}(r_t - \hat{\mu})(r_t - \hat{\mu})^T \tag{11}$$

are the sufficient statistics of the historical returns. To make notation easier to read, in the sections 2.2 and 2.3 we forget about rolling window length τ and work with all possible data T. In order to compute $\hat{\Sigma}^{-1}$, $T > N$ has to be fulfilled. From a statistical point of view, $\hat{\mu}$ and $\hat{\Sigma}$ are the maximum likelihood estimators of μ and Σ, therefore $\hat{w}$ is also maximum likelihood estimator of w^* (2), asymptotically the most efficient estimator of the unknown weights vector w^*. To compute weights of the Sample Tangency Portfolio we just replace μ and Σ in w^* by $\hat{\mu}$ and $\hat{\Sigma}$

$$\hat{w} = \frac{1}{\gamma}\hat{\Sigma}^{-1}\hat{\mu}. \tag{12}$$

this means that the method considers sample estimates as true parameters. This is called by Kan and Zhou [16] as plug-in approach, as we just plugged in $\hat{\mu}$ and $\hat{\Sigma}$ into w^*.

2.3 Suboptimal Weights

Methods in this section are based on the work of Kan and Zhou [16], in which authors aim to maximize the portfolios expected out-of-sample performance. Practically, these methods differ from the sample tangency portfolio $\hat{w}$ (12) by just a coefficient multiple of the weights vector, based on different assumptions. All of these portfolios have the same Sharpe Ratio as it is not affected by multiplying weights vector by a scalar.
In general, a portfolio rule is a function of the realizations of historical data returns Φ_T,

$$\hat{w} = f(r_1, ..., r_N). \tag{13}$$

Using such portfolio rule gives investor out-of-sample mean and variance given by

$$\tilde{\mu_p} = \hat{w}^T \mu, \tag{14}$$

$$\tilde{\sigma}_p^2 = \hat{w}^T \Sigma \hat{w}. \tag{15}$$

As $\hat{w}$ is a function of random variables, it is also a random variable, therefore $\tilde{\mu}_p$ and $\tilde{\sigma_p}^2$ are also random variables as functions of historical returns Φ_T. To be able to compare different portfolio rules, one needs to establish an objective function. Authors propose a measure that is consistent with the primary objective function (1) as well as a measure of Certainty Equivalent Return of the portfolio $\hat{w}$, which is

$$\tilde{U}(\hat{w}) = \tilde{\mu}_p - \frac{\gamma}{2}\tilde{\sigma_p}^2 = \hat{w}^T \mu - \frac{\gamma}{2}\hat{w}^T \Sigma \hat{w}. \tag{16}$$

It is important to notice that calculating $\tilde{U}(\hat{w})$ is not possible as μ and Σ are unknown to the investor. Hence authors focus on expected out-of-sample performance $E[\tilde{U}(\hat{w}]$, employing standard statistical decision theory with a loss function

$$L(w^*, \hat{w}) = U(w^*) - \tilde{U}(\hat{w}), \tag{17}$$

and expected loss function

$$\rho(w^*, \hat{w}) = E[L(w^*, \hat{w})] = U(w^*) - E[\tilde{U}(\hat{w})]. \tag{18}$$

This enables us to rank portfolios according to their expected out-of-sample performance $E[\tilde{U}(\hat{w})]$, which an investor can achieve on average under parameter uncertainty when using portfolio rule $\hat{w}$. As a motivation for the next section we compare the standard plug-in estimator $\hat{w}$ (12) with the optimal but unknown weights w^*. Under the normality assumption, the sample mean $\hat{\mu}$ and sample covariance matrix $\hat{\Sigma}$ are independently distributed with following distributions,

$$\hat{\mu} \quad \sim \quad N(\mu, \Sigma/T), \tag{19}$$

$$\hat{\Sigma} \quad \sim \quad W_N(T-1, \Sigma)/T, \tag{20}$$

where $W_N(T-1, \Sigma)$ denotes Wishart distribution with covariance matrix Σ and $T-1$ degrees of freedom. Given that $E[\hat{\Sigma}^{-1}] = T\ \Sigma^{-1}/(T-N-2)$ (page 97, [22]), we have

$$E[\hat{w}] = \frac{T}{T-N-2}w^*, \tag{21}$$

for $T > N+2$. As a consequence of this, weights are in ordered relationship as $|\hat{w}| > |w^*|$. This means that investors who do not know the true parameters and use portfolio rule $\hat{w}$, have tendency to take bigger positions in the risky assets than investors who know true parameters and use w^*.

2.3.1 Plug-In Approach

The first two approaches proposed by Kan and Zhou [16] are based on unbiased estimators of Σ and Σ^{-1}. Beginning with the unbiased estimator $\bar{\Sigma}$ of σ

$$\bar{\Sigma} = \frac{1}{T-1}\sum_{t=1}^{T}(r_t - \hat{\mu})(r_t - \hat{\mu})^T = \frac{T}{T-1}\hat{\Sigma}, \tag{22}$$

which is slightly greater than the sample covariance matrix $\hat{\Sigma}$, results into portfolio weights

$$\bar{w} = \frac{1}{\gamma}\bar{\Sigma}^{-1}\hat{u} = \frac{1}{\gamma}\left(\frac{T-1}{T}\right)\hat{\Sigma}^{-1}\hat{\mu} = \left(\frac{T-1}{T}\right)\hat{w}. \tag{23}$$

This portfolio rule $\bar{w}$ invests less extensively to the risky assets than the $\hat{w}$ does. Nonetheless, such portfolio rule still takes larger positions in the risky assets compared to the true optimal portfolio as

$$E[\bar{w}] = \frac{T-1}{T-N-2}w^*. \tag{24}$$

Moving on to the second approach, utilizing the biased estimator $\tilde{\Sigma}$ of Σ;

$$\tilde{\Sigma} = \frac{1}{T-N-2}\sum_{t=1}^{T}(r_t - \hat{\mu})(r_t - \hat{\mu})^T = \frac{T}{T-N-2}\hat{\Sigma}, \tag{25}$$

we get the unbiased estimator of $\tilde{\Sigma}^{-1}$. As a result, the plug-in estimator of the optimal portfolio weights is

$$\tilde{w} = \frac{1}{\gamma}\tilde{\Sigma}^{-1}\hat{\mu} = \frac{T-N-2}{T}\hat{w}, \tag{26}$$

which is an unbiased estimator of the unknown optimal portfolio w^*, specifically $E[\tilde{w}] = w^*$. Meaning that on average the investor who uses $\tilde{w}$ will invest the same amount of

money as with the true optimal portfolio w^*. Authors also derived the expected out-of-sample measure for both portfolio rules $\bar{w}$ and $\tilde{w}$, which are ordered as follows:

$$E[\tilde{U}(\tilde{w})] > E[\bar{U}(\bar{w})] > E[\hat{U}(\hat{w})]. \tag{27}$$

2.3.2 Bayesian Approach

A better method in regard to our expected out-of-sample performance can be achieved by using Bayesian approach based on the predictive distributions pioneered by Zellner and Chetty [28], which enables us to integrate estimation risk into the analysis directly. In contrast to the classical framework, where the utility is defined according to parameters μ and Σ, Bayesian approach assumes that the investor is concerned about the expected utility under the predictive distribution $p(r_{t+1} \mid \Phi_T)$, which is based on the prior distribution and realization of the historical data. If the prior is set very well and is centered around the true values, there is no dispute that such portfolio will dominate the classic plug-in rules $\hat{w}$, $\bar{w}$ and $\tilde{w}$. To make a fair comparison to the above mentioned plug-in methods, authors assume diffuse prior. Researchers Brown [3], Klein and Bawa [17], and Stambaugh [27] show that under standard diffuse prior $p_0(\mu, \Sigma)$ on μ and Σ,

$$p_0(\mu, \Sigma) \propto |\Sigma|^{-(N+1)/2}, \tag{28}$$

the derived Bayesian optimal portfolio rule has the same formula as w^* except the true unknown parameters are replaced by their predictive moments,

$$\hat{w}^{Bayes} = \frac{1}{\gamma}\left(\frac{T-N-2}{T+1}\right)\hat{\Sigma}^{-1}\hat{\mu} = \frac{T-N-2}{T+1}\hat{w}. \tag{29}$$

Examining the expected value of Bayesian solution weights vector $\hat{w}^{Bayes}$, we observe that it differs from the unbiased estimator $\tilde{w}$ only by a factor $T/(T+1)$;

$$E[\tilde{w}^{Bayes}] = \left(\frac{T}{T+1}\right)w^*, \tag{30}$$

which implies a more conservative behaviour than the rule with the true parameters known, as it advises investing less into risky assets. The simulations conducted by Brown [3] and Stambaugh [27] suggested that this portfolio rule outmatch the classic

plug-in methods. Kan and Zhou derived a closed-from formula for the expected ouf-of-sample performance $E[\tilde{U}(\hat{w}^{Bayes})]$ and showed

$$E[\tilde{U}(\hat{w}^{Bayes})] > E[\tilde{U}(\tilde{w})], \tag{31}$$

which proved the simulation findings of Stambaugh and Brown. This implies uniform superiority of the Bayesian solution to the classical plug-in rules $\hat{w}$, $\bar{w}$ and $\tilde{w}$, no matter what the true parameters are.

2.3.3 Optimal Two-Fund Rule

To improve the expected out-of-sample performance of the sample tangency portfolio $\hat{w}$, Kan and Zhou proposed three rules $\bar{w}$, $\tilde{w}$ and $\hat{w}^{Bayes}$ that one by one dominate its predecessors. Eventually, the work of Kan and Zhou did not end there, as they proposed a rule that uniformly dominates the Bayesian portfolio rule under diffuse prior. The next rule we briefly discuss is the Optimal Two-Fund rule, a rule that dominates the $\hat{w}^{Bayes}$, based on the work of Kan and Zhou [16]. In practice, a portfolio rule can be any function of the sufficient estimators $\hat{\mu}$ and $\hat{\Sigma}$, i.e.

$$\hat{w} = f(\hat{\mu}, \hat{\Sigma}). \tag{32}$$

Naturally, in our framework the investor is interested only in such function $f(\hat{\mu}, \hat{\Sigma})$ that maximizes the expected out-of-sample performance. Such function can be very complex and non-linear, with infinitely many ways to construct it. Initially, authors decide to limit their attention to class of portfolio rules that invest only in riskless asset and the sample tangency portfolio. In their further research they also add the minimum variance portfolio, and propose a Three-Fund rule, which in simulations tends to outperform the Two-Fund rule, but proof of this is missing. Expanding the formulas of plug-in rules $\bar{w}$, $\tilde{w}$ and Bayesian rules $\hat{w}^{Bayes}$, they consider the class of two-fund portfolio rules that are build on the following structure

$$\hat{w} = \frac{c}{\gamma}\, \hat{\Sigma}^{-1}\hat{\mu} \tag{33}$$

where c is an invariable scalar. Then they focus on deriving the formula for expected out-of-sample performance, which is

$$E[\tilde{U}(c\hat{\Sigma}^{-1}\hat{\mu}/\gamma)] = \frac{c\theta^2}{\gamma}\left(\frac{T}{T-N-2}\right) \tag{34}$$
$$-\frac{c^2}{2\gamma}\left(\theta^2+\frac{N}{T}\right)\left[\frac{T^2(T-2)}{(T-N-1)(T-N-2)(T-N-4)}\right],$$

assuming $T > N+4$, they differentiate with respect to c and obtain the optimal c^* .

$$c^* = \left[\frac{(T-N-1)(T-N-4)}{T(T-2)}\right]\left(\frac{\theta^2}{\theta^2+\frac{N}{T}}\right). \tag{35}$$

Pluging into (34) the optimal c^*, we get the expected out-of-sample performance

$$E[\tilde{U}(\hat{w}^*)] = \frac{\theta^2}{2\gamma}\left[\frac{(T-N-1)(T-N-4)}{(T-2)(T-N-2)}\right]\left(\frac{\theta^2}{\theta^2+\frac{N}{T}}\right). \tag{36}$$

Even though c^* is optimal, a practicable portfolio rule using c^* does not exist as ex-post Sharpe Ratio θ is unknown. In spite of such challenges, it is interesting to consider the following rule,

$$\hat{w}_* = \frac{c_3}{\gamma}\,\hat{\Sigma}^{-1}\hat{\mu}, \qquad c_3 = \frac{(T-N-1)(T-N-4)}{T(T-2)}. \tag{37}$$

which is optimal when $\theta^2 \to \infty$. Comparing this portfolio rule to the Bayesian rule $\hat{w}^{Bayes}$, authors found out that $\hat{w}_*$ dominates the Bayesian rule for all possible θ^2 values. The reason behind this is that $f(c) = E[\tilde{U}(c\hat{\Sigma}^{-1}\hat{\mu}/\gamma)]$ in (34) is a quadratic function of c, therefore the expected out-of-sample performance is decreasing function of c for $c \geq c^*$. They also show that $c^{Bayes} = \frac{T-N-2}{T+1} > c_3 > c^*$, when $T > N+4$. Thus the $\hat{w}_*$ dominates the Bayesian portfolio rule $\hat{w}^{Bayes}$ regardless of the value θ^2. This means that without any information about the value of θ^2, c_3 presents the best candidate for c that maximizes the expected performance. In practice, estimating θ^2 based on sample estimates is a natural choice

$$\hat{\theta}^2 = \hat{\mu}\hat{\Sigma}^{-1}\hat{\mu}. \tag{38}$$

On the other hand, the estimator $\hat{\theta}^2$ can be a heavily biased estimator of θ. For this reason, they derive the distribution of $\hat{\theta^2}$, which is

$$\hat{\theta}^2 \sim \left(\frac{N}{T-N}\right)F_{N,T-N}(T\theta^2). \tag{39}$$

where $F_{N,T-N}(T\theta^2)$ is a noncentral F-distribution with noncentrality parameter $T\theta^2$ and N and $T-N$ degrees of freedom. To decrease the bias of $\hat{\theta}^2$, they propose to use the unbiased estimator of θ^2, which is

$$\hat{\theta}_u^2 = \frac{(T-N-2)\hat{\theta}^2 - N}{T}. \tag{40}$$

The problem with this estimator is that it can take negative values which is undesirable for an estimator of θ^2. To solve this issue Kan and Zhou take advantage of the research by Ruhkin [24] and Kubokawa, Robert and Saleh [18], who propose a superior estimator of θ under the quadratic loss function, which can be rewritten as

$$\hat{\theta}_a^2 = \frac{(T-N-2)\hat{\theta}^2 - N}{T} + \frac{2(\hat{\theta}^2)^{\frac{N}{2}}(1+\hat{\theta}^2)^{-\frac{T-2}{2}}}{TB_{\hat{\theta}^2/(1+\hat{\theta}^2)}(N/2,(T-N)/2)}, \tag{41}$$

where

$$B_x(a,b) = \int_0^x y^{a-1}(1-y)^{b-1}dy \tag{42}$$

is the incomplete beta function. Utilizing the adjusted estimator of θ we get the optimal c^*

$$\hat{c}^* = c_3\left(\frac{\hat{\theta}_a^2}{\hat{\theta}_a^2 + \frac{N}{T}}\right) \tag{43}$$

and the associated Optimal Two-Fund rule weights are given as follows

$$\hat{w}^{\mathrm{II}} = \frac{1}{\gamma}\hat{c}^*\hat{\Sigma}^{-1}\hat{\mu}. \tag{44}$$

2.4 Norm-Constrained Portfolio

To implement Markowitz based portfolios in practice, investors need to estimate the means and covariances of asset returns. To achieve this the investors have been conventionally using the sample means and covariances. However the portfolios, relying on sample estimates, are known to perform poorly out-of-sample due to the estimation errors. Not only are the means more difficult to estimate but also have a larger impact on portfolio weights compared to the covariances of asset returns. This is supported by the work of Chopra and Ziemba [7], where authors quantify that in their setting the errors in expected returns are about ten times more significant than errors in covariances

and variances. As a solution to this problem, research focused on minimum-variance portfolios, which completely ignores the asset returns. The extensive empirical research on this topic, e.g. Jagannathan and Ma [15], shows that the minimum-variance portfolio usually performs better out-of-sample than other mean-variance portfolios, even when performance measures involving returns are considered, such as Sharpe Ratio.

A new approach for determining the optimal portfolio weights in the presence of estimation error has been proposed by DeMiguel, Garpali, Nogales and Uppal [9]. Using the following formulas,

$$\min_{w} w^T \hat{\Sigma} w, \tag{45}$$

$$w^T 1_N = 1, \tag{46}$$

$$\|w\|_p \leq \delta, \tag{47}$$

while

$$\|w\|_p = \left(\sum_{i=1}^{N} \mid w_i \mid^p \right)^{1/p}, \tag{48}$$

we get a class of portfolios that are called norm-constrained portfolios, which shrinks portfolio weights directly. In [9] they propose to use $p = 1$ and $p = 2$. The δ is found by using cross-validation where the minimum variance or maximum return is targeted. Details of the cross-validation algorithm are discussed in Section [2.4.3]. As this method ignores the expected returns and relies only on estimation of the covariance matrix of asset returns, it is less sensitive to estimation errors.

2.4.1 The 1-Norm-Constrained Portfolio

In case of $p = 1$ and $\delta > 1$, formulas (46) - (48) can be rewritten to provide a useful interpretation,

$$-\sum_{i \in N(w)} w_i < \frac{\delta - 1}{2}, \tag{49}$$

where $N(w) = \{i : w_i < 0\}$ is the set of asset indices for which portfolio weights are negative and $-\sum_{i \in N(w)} w_i$ is the amount of wealth sold short, where $\frac{\delta - 1}{2}$ is the shortsale budget which can be freely distributed among all of the assets.

2.4.2 The 2-Norm-Constrained Portfolio

Authors in [9] also provide an interesting interpretation of the 2-norm-constrained portfolio. Note that the solution to the problem (45) - (48) with a constraint on the 2-norm $\|w\|_p \leq \delta$ has the same solution as with constraint $\|w\|_p^2 \leq \delta^2$. Since the squared 2-norm is easier to analyze, authors define the 2-norm-constrained minimum variance portfolio as follows

$$\min_w w^T \hat{\Sigma} w, \tag{50}$$

$$w^T 1_N = 1, \tag{51}$$

$$\sum_{i=1}^{N} w_i^2 \leq \delta. \tag{52}$$

To gain a better intuition about the effect of the 2-norm constraint, the formula (52) can be equivalently rewritten as

$$\sum_{i=1}^{N} \left(w_i - \frac{1}{N} \right)^2 \leq \left(\delta - \frac{1}{N} \right). \tag{53}$$

This reformulated constraint means that the difference between this strategy and the $1/N$ strategy is bounded by $\delta - 1/N$. Note that for $\delta = 1/N$, we get the naive the $1/N$ portfolio.

2.4.3 Cross - Validation

In order to implement the norm-constrained portfolios (45) - (47) a parameter δ has to be set to some reasonable value. To obtain an appropriate value researchers DeMiguel, Garlappi, Nogales and Uppal in [9] propose the nonparametric technique known as cross validation.

First we discuss the approach to find δ that reduces out-of-sample variance. Having an estimation window composed of τ asset returns, for each t from 1 to τ perform following steps:

1. delete the tth sample return from the estimation window,

2. compute sample covariance matrix $\hat{\Sigma}_{(t)}$ on this reduced timeseries,

3. set $\theta = \delta$ and compute portfolio weights $(w_\theta)_{(t)}$ using (45) - (47) with $\hat{\Sigma}$ substituted as $\hat{\Sigma}_{(t)}$,

4. calculate the "out-of-sample" return of this portfolio on the tth asset return $(r_\theta)_{(t)} = ((w_\theta)_{(t)})^T r_t$

After doing this for all t, we can proceed to calculating out-of-sample variance of the τ out-of-sample returns $(r_\theta)_{(t)}$, that is

$$\hat{\sigma}_\theta^2 = \frac{\sum_{t=1}^{\tau}[(r_\theta)_{(t)} - (\bar{r}_\theta)_{(t)}]^2}{\tau - 1}, \tag{54}$$

in which $(\bar{r}_\theta)_{(t)} = \sum_{t=1}^{\tau}(r_\theta)_{(t)}/\tau$. Repeat this procedure for every considered δ and choose one that minimizes the "out-of-sample" return variance

$$\theta^* = \arg\min_\theta \hat{\sigma}_\theta^2. \tag{55}$$

This is a computationally very expensive operation as for one estimation period, the investor has to calculate $\tau \times |\delta|$ portfolio weights, only then the δ with the minimum out-of-sample variance (55) can be selected.

The other option is to choose δ that maximizes the portfolio return over the last period. The motivation behind this approach is based on the work by Campbell, Lo and MacKinley [6], in which they found out that the portfolio returns are strongly autocorrelated, in contrast to individual securities which are weakly negatively autocorrelated. As the autocorrelation in portfolio returns in the last period is the highest, we don't consider previous strategy returns. In this case, we choose δ^* that

$$\delta^* = \arg\max_\theta w_\theta^T r_\tau, \tag{56}$$

where r_t is the vector of asset returns for the last period of the estimation window and w_θ is the norm-constrained portfolio computed using all data in the estimation window with the parameter θ.

2.5 Partial Minimum-Variance Portfolios

In this section, we describe the approach from DeMiguel, Garpali, Nogales and Uppal [9], where the classical conjugate gradient method is applied to solve the minimum-

variance optimization problem. A sequence of portfolios is generated by this method. The first generated portfolio is the starting point (based on their work, we take the $1/N$ portfolio), the intermediate portfolio is the partial minimum-variance portfolio (PMP) and the last portfolio is the shortsale-unconstrained minimum-variance. The constructing procedure of these portfolios is discussed in more depth in their online appendix [10], while here we just summarize the necessary information to implement this algorithm.

Starting with a matrix $A \in R^N \times (N-1)$, that has ones on the diagonal and minus ones under diagonal, we follow with QR decomposition of the matrix A. According to notation from paper [9], we call Q the matrix Z. This matrix Z satisfies $1_N^T Z = 0$ and $Z^T Z = 1_N$, thus Z is an orthonormal basis for the subspace of all portfolios satisfying $w^T 1_N = 0$. We denote w_{PMP_k} as k^{th} partial minimum-variance portfolio. The algorithm is as follows:

$$w_{PMP_0} = 1_N/N, \quad k = 0, \tag{57}$$

$$\epsilon^k = -Z^T \hat{\Sigma} w_{PMP_k}, \tag{58}$$

$$w_{PMP_{k+1}} = w_{PMP_k} + \alpha_k w_{CG(k+1)}, \tag{59}$$

$$w_{CG(k+1)} = Z\epsilon^k + \beta^K w_{CG(k)}, \tag{60}$$

while for $k = 0$ we set $\beta^k = 0$ and for $k > 0$, we continue with:

$$\beta^k = \frac{(\epsilon^k)^T \epsilon^k}{(\epsilon^{k-1})^T \epsilon^{k-1}}, \tag{61}$$

$$\alpha_k = \frac{(\epsilon^k)^T \epsilon^k}{(w_{CG(k+1)})^T \hat{\Sigma} w_{CG(k+1)}}. \tag{62}$$

Using the cross-validation approach from Section [2.4.3], we can also focus on maximizing out-of-sample return or minimizing out-of-sample variance as in the norm-constrained portfolios. We just simply set $\theta = k \in \{1, ..., N-1\}$ and run the whole procedure without change.

2.6 Autoregressive models

Estimating expected returns of assets are notorious for their complex character and huge impact on the resulting portfolio. To have a broad range of reviewed methods

we also investigate two portfolio rules based on time series analysis, specifically autoregressive and vector autoregressive model of the first order, denoted as AR(1) and VAR(1). To see if this time series modelling approach is meaningful, we compare those two methods to standard mean variance Markowitz model with sample estimates of returns and covariance matrix. For all those three methods we forbid short selling and set sum of weights to 1. Using constraints like those is one of the easiest ways how to restrict the Markowitz model from its widely known extremal behaviour. Standard constrained Markowitz model with sample estimates is defined as

$$\max_{w} \quad w^T\hat{\mu} - \frac{\gamma}{2}w^T\hat{\Sigma}w, \tag{63}$$

$$w^Te = 1, \tag{64}$$

$$w \geq 0, \tag{65}$$

where $\hat{\mu}$ and $\hat{\Sigma}$ are calculated as sample estimates of returns as in (10) and (11), respectively.

2.6.1 Autoregression

In the first autoregression scenario, we model returns using standard AR(1) process, for each asset individually. Coefficient of AR(1) process are estimated based on the τ data returns in the rolling window. Fitting of coefficients and prediction is done for every asset in the dataset, therefore for each $i = 1, ..., N$ we have an optimization problem with three parameters δ_i, ϕ_i and variance of residuals ϵ_i. As we model assets one by one, and we are interested in the variance of the unexplained returns, the covariance matrix of AR(1) model is diagonal with variance of residuals for i-th asset on the position i, i and zeros elsewhere, i.e. we assume zero correlations between assets. AR(1) model is defined as follows

$$r_{i,t}^{AR(1)} = \delta_i + \phi_i r_{i,t-1} + \epsilon_{i,t}, \tag{66}$$

$$r_{i,pred}^{AR(1)} = \delta_i + \phi_i r_{i,t}, \tag{67}$$

$$\hat{\Sigma}_{i,i}^{AR(1)} = var(r_i^{AR(1)} - r_i) \tag{68}$$

To fit an AR(1) model we use R package *stats*, function *arima* and the maximum likelihood method of coefficient estimation, as it has proved to be more stable than the default method, which is conditional sum of squares - maximum likelihood. The difference between those two is that the initial state vector in case of maximum likelihood estimation is returned by Kalman filter whereas in the CSS-ML method, the initial state is produced by conditional sum of squares. In our experiments Kalman filter produced initial states that were good starting points for maximum likelihood methods as the optimization algorithm have always converged, which was not the case for CSS-ML. The standard mean-variance Markowitz model with returns predicted by AR(1), constraints on shorting and sum of weights is as follows

$$\min_{w} \quad w^T r_{pred}^{AR(1)} - \frac{\gamma}{2} w^T \hat{\Sigma}^{AR(1)} w, \tag{69}$$

$$w^T e = 1, \tag{70}$$

$$w \geq 0, \tag{71}$$

where $r_{pred}^{AR(1)} = (r_{1,pred}^{AR(1)}, ..., r_{N,pred}^{AR(1)})$ are one step ahead predicted returns.

2.6.2 Vector Autoregression

Our second time series model is VAR(1), which in contrast to AR(1) is fitted on all assets at once. The sample covariance matrix is based on the residuals of fitted values versus real returns in the estimation window, this produces covariance matrix $\hat{\Sigma}^{VAR(1)}$ with full structure. The main disadvantage of the VAR(1) model is the number of parameters that needs to be estimated, which is N parameters for intercept c, $N \times N$ parameters for matrix A, and $N \times N$ parameters for covariance matrix of error terms ϵ. To estimate those $N + 2N^2$ parameters the model has only $\tau \times N$ returns available. Finally, the VAR(1) model is defined in the following way

$$r_t^{VAR(1)} = c + Ar_{t-1} + \epsilon_t, \tag{72}$$

$$r_{pred}^{VAR(1)} = c + Ar_t, \tag{73}$$

$$\hat{\Sigma}^{VAR(1)} = cov(r^{VAR(1)} - r), \tag{74}$$

and the standard mean-variance Markowitz model with returns predicted by VAR(1), constraints on shorting and sum of weights is as follows

$$\max_{w} \quad w^T r_{pred}^{VAR(1)} - \frac{\gamma}{2} w^T \hat{\Sigma}^{VAR(1)} w, \tag{75}$$

$$w^T e = 1, \tag{76}$$

$$w \geq 0. \tag{77}$$

2.7 Shrinkage

Two fundamental ingredients to the mean-variance optimization problem discussed in Section [1.1] are the expected excess returns, which represent the ability to forecast future price movements, and the covariance matrix of stock returns, which allows for the risk control. Estimating the covariance matrix has always been one of the trickiest points, traditionally computed as sample covariance matrix. Ledoit and Wolf in the work [19, 20], propose a simple and effective improvement of the sample covariance matrix by using a statistical concept called shrinkage. The main idea of this approach is to pull the very positive and negative coefficients in the covariance matrix closer to zero, as these coefficients contain the most error. Ledoit and Wolf propose to calculate the linear combination of two extremes F and S,

$$\Sigma_{Shrink} = \delta F + (1 - \delta) S, \tag{78}$$

where δ is a shrinkage constant between 0 and 1, S is an estimator with low structure, in this case the sample covariance matrix, and the shrinkage target F is the highly structured estimator. The algorithm how to calculate shrinkage target F and the optimal shrinkage intensity $\hat{\delta}^*$ is discussed in the detail in the paper from Ledoit and Wolf [20], pp. 11-14. However, the optimal shrinkage intensity is quite straightforward to implement, it is rather tedious to write down, therefore we just give necessary equations for the shrinkage target F;

$$r_{ij} = \frac{s_{ij}}{\sqrt{s_{ii}s_{jj}}}, \tag{79}$$

$$\bar{r} = \frac{2}{(N-1)N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} r_{ij}, \tag{80}$$

$$f_{ii} = s_{ii} \quad and \quad f_{ij} = \bar{r}\sqrt{s_{ii}s_{jj}}, \tag{81}$$

where s_{ij} are the elements of the sample covariance matrix S.

2.8 Clustering

Parameter uncertainty has been detected as one major cause of the bad out-of-sample performance of the presented methods. Some literature addresses this problem by even more robust portfolio selection rules. Independent of the chosen portfolio selection rule, as proposed by Bjerring, Ross and Weissensteiner [1], Branger, Ľučivjanská and Weissensteiner [2] and Fabrizio, Gallegati, Mantegna and Tola [11], dimension reduction technique such as clustering methods are a viable solution to this problem. The portfolio optimization process is done over the equally weighted groups proposed by grouping strategy, as opposed to portfolio optimization over the whole asset menu.

2.8.1 Agglomerative Clustering

Origins of the following ideas are based on the work of Bjerring, Ross and Weissensteiner [1]. In order to take the best advantage from diversification when implementing portfolio selection rules, the reduced universe $n \in N$ should consist of the n assets with the lowest correlation with each other. Such sub-space can be interpreted as a problem of finding the longest path of $n \in N$ vertices in a simple cycle of an undirected graph. According to our target - to push similarly correlated assets into one group, the distance between the vertices of assets i and j can be represented by distance $d_{i,j} = 1 - cor(r_i, r_j)$. From an optimization point of view, such task is of a non-deterministic polynomial-time complexity and general approximation methods are not available. For this reason authors propose a heuristic method, an agglomerative hierarchical clustering, more specifically complete and single-linkage clustering, which are of $O(N^2)$ complexity.

In general, agglomerative hierarchical clustering starts by placing every object into in its own cluster and then groups these clusters into larger clusters until all object are merged into one cluster with all objects. The dissimilarity between objects is measured by a distance metrics (e.g. the Euclidean or Manhattan distance) and the dissimilarity between clusters is measured with a linkage criterion, the two most commonly being used are

$$\text{Complete linkage} \quad \max\{d(a,b) : a \in A, b \in B\}, \tag{82}$$

$$\text{Single linkage} \quad \min\{d(a,b) : a \in A, b \in B\}, \tag{83}$$

where d is a distance measure. Naturally, different metrics and linkage criterion produce diversely shaped and sized clusters, giving a lot of possibilities for experiments. The well known drawback of single linkage criterion is that it may force together clusters only due to a single elements being close to each other, despite the fact that the rest of the cluster may be very remote from each other. This drawback is answered with complete linkage that is inclined to create compact clusters of approximately similar diameters. Hence we stick to the complete linkage with Euclidean distance as proposed in [1]. For more information about comparison of single versus complete linkage see work of Hartigan [13].

As mentioned earlier, correlation is a reasonable distance measure. In order to ensure good quality of the clustering procedure, covariance matrix estimation is crucial. As proposed by Bjerring, Ross and Weissensteiner [1] and Ledoit and Wolf [19, 20], we use shrinkage technique to estimate the correlation matrix, as we discussed in Section [2.7]. Such estimator is typically much more efficient (lower mean squared error), positive definite and well conditioned.

To complete the clustering procedure, investor has to choose representatives of each cluster - medoids. A medoid in this understanding is some combination of assets returns in the group, forming a single time series of returns. We worked with three approaches how to choose a medoid. Namely naive, median and minimum-distance approach.

Naive medoid is represented as a simple average of asset returns. To find the median medoid investor has to calculate a sample mean for every single asset in the group and sort those means afterwards. In case of an odd number of assets in the cluster, medoid

is the asset with the middlemost sample mean return. Having an even number of assets in the group, we calculate the average between the two middlemost sample mean returns and use this as a median medoid. Lastly, the minimum distance medoid is the asset that has the lowest sum of distances to other assets in the group. After having assets separated into groups represented by medoids, we can use any of the methods mentioned in this chapter on this reduced universe. Other clustering approaches regarding distance measure and clustering method are thoroughly discussed in the work from Fabrizio, Gallegati, Mantegna and Tola [11].

2.8.2 Portfolio Optimal Granularity

The next question that has to be answered is the optimal number of clusters. In most numerical experiments we use a fixed number of clusters, so we can compare clustering approach with $N >> n$ assets and n clusters versus standard approach without clustering with just n assets. Branger, Lučivjanská, and Weissensteiner in the work [2] derive formula for optimal cluster count under strong assumptions on used portfolio rule (Sample Tangency portfolio $\hat{w}$, (12)) and equally sized clusters. This is problematic if the proposed optimal number of clusters is odd and number of assets N in the dataset is even. This approach also requires a clustering procedure that is capable of cluster size control. In our implementation we weaken the assumption on equally sized clusters and used portfolio rule, as we use Bayesian Approach - from Section [2.3], which is marginally different from standard Markowitz portfolio.

The proposed formula (84) is supported by Monte Carlo simulation, finding that optimal group size often beat standard approach of optimization over the whole asset menu and $1/N$ naive approach. Intuitively, contribution of clustering procedure to the portfolio optimization with parameter uncertainty is very dependent on the number of considered assets and the length of data, and the proposed formula to the optimal number of clusters g^* is as follows

$$g^* = \sqrt[3]{\frac{6\sigma_g^2(1-\rho_g)}{s_g^2 T}}. \tag{84}$$

Background of this approach can be found in the work from Branger, Lučivjanská, and Weissensteiner [2], on the other hand, following formulas are not explicitly stated in

the mentioned article. Right now we just briefly summarize this procedure without further details;

$$\sigma_g^2 = \frac{2}{N^2 - N} \sum_{i=2,j=1,i>j}^{N,N-1} cov(r_i, r_j), \tag{85}$$

$$\rho_g = \frac{2}{N^2 - N} \sum_{i=2,j=1,i>j}^{N,N-1} cor(r_i, r_j), \tag{86}$$

$$s_g = \frac{\max \hat{\mu} - \min \hat{\mu}}{N - 1}. \tag{87}$$

The sample mean $\hat{\mu}$ is calculated as in (10). It is possible that in some ill cases formula (84) proposes to use less than 1 or more than N clusters, in which case we set the optimal number of cluster g^* to 1 or N respectively.

3 Empirical overview

In this chapter, we empirically compare the performance of the optimal asset allocation strategies we discussed in Chapter [2].

3.1 Data

Our data consists of 1000 stocks that have been part of S&P 1500 index in the period from 01/01/2004 to 31/12/2015, these data come from Center for Research in Security Prices (CRSP). All the data have complete history of the monthly returns for all 144 months. As a risk-free rate we used 1-month US Treasury Bill from Ibbotson and Associates, Inc., available at Fama & French website [23].

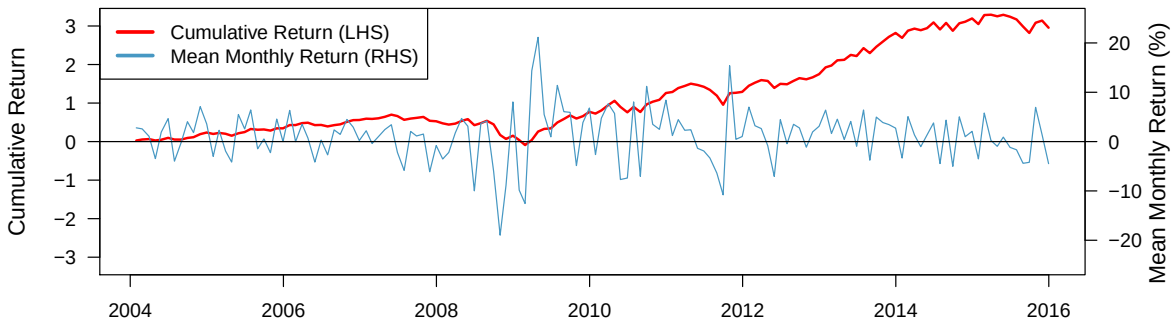

Figure 1: **Asset returns overview -1000 stocks;** Blue line reflects average monthly fluctuations in the asset returns and the red line shows the cumulative return of equally-weighed portfolio consisting of 1000 stocks. Data from Center for Research in Security Prices, observation period 01/01/2004 to 31/12/2015.

Simple overview of our risky assets data is provided in Figure 1. During the 2007 - 2009 global financial crisis, we observe negative return and high volatility. Another stressful period was in August 2011 when stock markets responded to the European sovereign debt crisis (Spain, Italy) and concerns over the slowing economic growth of the United States. In addition to this, in August 2011, Standard & Poor's downgraded credit rating of the United States from AAA to AA+ for the first time in the history.

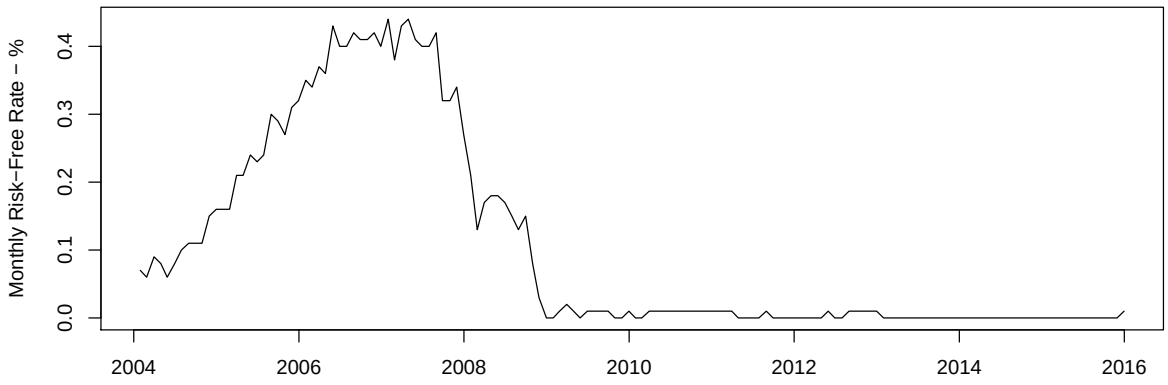

Figure 2: **1-month US Treasury Bill;** Observed period from 01/01/2004 to 31/12/2015. Data from Ibbotson and Associates, Inc.

In Figure 2 we observe the level of risk-free rate, which we used to calculate excess returns. We decided to use monthly rates as it corresponds to our monthly asset returns. The risk-free rate reached its peak at 0.44% in the January 2007. Following the global financial crisis, the risk-free rate has dropped to zero.

3.2 Analysis of methods

In order to compare presented methods discussed in Chapter [2] in a robust way, we created randomly 100 datasets out of 1000 stocks with size $N = 10$, $N = 20$ and $N = 30$ and calculated weights using rolling window procedure with 3 different rolling window lengths $\tau = 48$, $\tau = 60$ and $\tau = 72$ for 16 methods, all in all 9 different settings to compare 16 strategies. This means that for every dataset we have 16 time series of out-of-sample returns. For rolling window length $\tau = 48$ we get $144 - 48 = 96$ time series returns and likewise for other rolling window lengths. Portfolio rules settings that we used are summarized in Section [3.2.1]. The reason for calculating only 100 datasets and not more is the computational complexity of cross validation that targets minimal variance for norm-constrained portfolios. To calculate optimal δ for 100 datasets and one norm constrained portfolio considering 4 possible δ with $\tau = 48$ and $T - \tau = 96$, computer had to calculate $4 \times 48 \times 96 \times 100 \approx 1.8e6$ portfolio weights. Using parallel computation and 8 cores of Intel CPU i7-9700K running at 5 GHz it took over 5 minutes to finish one dataset. Therefore, to get the results that are presented in the section [3.2.2], for 9 different settings with 100 datasets each, it took all together approximately 75 hours.

3.2.1 Hyperparameters setting

Before we proceed to results and comparisons, we summarize the used methods and their parameters in Table 1. Additionally, we also include selected key features of individual methods.

Strategy	γ	μ	Σ	$\sum w_i$	**Short Pos.**	**Comment**
Naive Portfolio	-	-	-	1	No	- weights: $1/N$
Minus Naive Portfolio	-	-	-	-1	Yes	- weights: $-1/N$
Sample Tangency P.	5	$\hat{\mu}$	$\hat{\Sigma}$	-	Yes	- weights: $\hat{w}$ (12)
Sample T. P - Plug-in 1	5	$\hat{\mu}$	$\hat{\Sigma}$	-	Yes	- weights: $\bar{w}$ (23)
Sample T. P - Plug-in 2	5	$\hat{\mu}$	$\hat{\Sigma}$	-	Yes	- weights: $\tilde{w}$ (26)
Bayesian Porfolio	5	$\hat{\mu}$	$\hat{\Sigma}$	-	Yes	- weights: w^{Bayes} (29)
Optimal Two Fund rule	5	$\hat{\mu}$	sophis. estimator	-	Yes	- weights: $\hat{w}^{\text{II}}$ (44)
Norm Constrained - 1 - Variance	Min. Var.	-	$\hat{\Sigma}$	1	$\leq \frac{\delta-1}{2}$	- $\sum \mid w_i \mid \leq \delta$; - Cross-Val.: $\delta \in \{1, 1.5, 2, 2.5\}$ - Target: min. var
Norm Constrained - 1 - Return	Min. Var.	-	$\hat{\Sigma}$	1	$\leq \frac{\delta-1}{2}$	- $\sum \mid w_i \mid \leq \delta$; - Cross-Val.: $\delta \in \{1, 1.5, 2, 2.5\}$ - Target: max. ret
Norm Constrained - 2 - Variance	Min. Var.	-	$\hat{\Sigma}$	1	Yes	- $\sum w_i^2 \leq \delta$; - Cross-Val.: $\delta \in \{1, 1.5, 2, 2.5\}$ - Target: min. var
Norm Constrained - 2 - Return	Min. Var.	-	$\hat{\Sigma}$	1	Yes	- $\sum w_i^2 \leq \delta$; - Cross-Val: $\delta \in \{1, 1.5, 2, 2.5\}$ - Target: max. ret
Partial Minimum Portfolio	Part. Min. Var.	-	$\hat{\Sigma}$	-	Yes	- Conjugate Gradient optim; - Cross-Val: - $k \in \{1, ..., N\}$ - Target: max. ret
Opt. Gran. clustering - Bayesian Portfolio	5	$\hat{\mu}_{Clust}$	$\hat{\Sigma}_{Clust}$	-	Yes	- Aglomerative Clustering; - Distance: 1 - $cor(r_i, r_j)$ - Number of cluster: (84)
Markowitz - Constrained	5	$\hat{\mu}$	$\hat{\Sigma}$	1	No	-
Markowitz - AR1	5	$r_{pred}^{AR(1)}$	diag. with $var(resid)$	1	No	-
Markowitz - VAR1	5	$r_{pred}^{VAR(1)}$	full structure $cov(resid)$	1	No	-

Table 1: **Methods overview;** Parameters and methods that we used in our empirical Section [3.2.2].

The second column of Table 1 provides information about optimization procedure, if risk-aversion coefficient γ is given, then classical mean-variance optimization is used. The third and fourth column highlights the need and type of estimator used, where $\hat{\mu}$ and $\hat{\Sigma}$ are sample estimators. The fifth column is for the sum of weights, where we have just two options, unconstrained or 1, in which case portfolio rule does not invest into risk-free. Information about possibilities to use short positions are in the sixth column.

3.2.2 Results

This section summarizes the performance measures we collected from the numerical experiments with random datasets. Considered strategies were followed in the investment period from January 2010 to December 2015, giving us 72 out-of-sample monthly returns. Out of those 72 months we calculated four metrics for each dataset with its mean and standard deviation for each of those metrics, specifically Sharpe Ratio (6), Certainty Equivalent Return (7), Turnover (8) and Annualized Return

$$Annualized \quad Return^i = \left[\prod_{t=\tau}^{T-1} 1 + w_t^{iT} r_{t+1}\right]^{12/(T-\tau)} - 1. \tag{88}$$

Tables report means of the measures for individual strategies and standard deviations are provided in parenthesis. If a strategy encountered return in some month lower than -100% we called such a strategy a default, such dataset was excluded from metrics calculation. The important metrics that we focus mostly on are Sharpe Ratio, Certainty Equivalent Return and Turnover. Disadvantage of Annualized Return is that it can be manipulated by multiplying the weights vector by some arbitrary constant without increasing the Sharpe Ratio. Scaling all weights to 1 would not be beneficial, as we would loose the track of suboptimal weights, which are traceable with Certainty Equivalent Return. Values in Tables 2, 3, 4 are ordered according to the length of the rolling window τ; the lower position in a row means longer rolling window.

Firstly, we clearly observe a decreasing average turnover when the rolling window length increases. This is expected, as adding new information while loosing the last observation has smaller impact in the longer data series. Also if number of assets in the dataset is increased, average turnover has tendency to grow. Another apparent observation is that the Naive Portfolio Sharpe Ratio increases with larger datasets, beating Markowitz based approaches from Kan & Zhou, constrained mean-variance Markowitz VAR(1) and Optimal Granularity Clustering with Bayesian Approach. On top of this, unconstrained Markowitz based portfolios even had a nonnegligible probability of default, which makes it a very risky investment choice. Despite the enhancements made to the Sample Tangency Portfolio by Kan and Zhou, the outcomes of the analysis show inferior performance to other methods. The highest Sharp Ratio with very low Turnover and balanced performance across all settings combination showed 1-Norm Constrained portfolio, that offers the best performance with $N = 20$ and $\tau = 72$.

N = 10, 100 datasets	Annual. Return	Mean (%)	Sharpe Ratio	CEQ (%)	Turnover	Default (%)
Naive Portfolio	0.14 (0.04)	1.22 (0.28)	0.23 (0.06)	0.49 (0.38)	0.01 (0)	0
	0.14 (0.04)	1.22 (0.28)	0.23 (0.06)	0.49 (0.38)	0.01 (0)	0
	0.14 (0.04)	1.22 (0.28)	0.23 (0.06)	0.49 (0.38)	0.01 (0)	0
Minus Naive Portfolio	-0.15 (0.03)	-1.22 (0.28)	-0.23 (0.06)	-1.96 (0.31)	0.01 (0)	0
	-0.15 (0.03)	-1.22 (0.28)	-0.23 (0.06)	-1.96 (0.31)	0.01 (0)	0
	-0.15 (0.03)	-1.22 (0.28)	-0.23 (0.06)	-1.96 (0.31)	0.01 (0)	0
Sample Tangency Portfolio	0.22 (0.23)	2.75 (1.84)	0.18 (0.11)	-3.03 (1.83)	1.12 (0.18)	2
	0.17 (0.19)	1.95 (1.51)	0.16 (0.12)	-1.57 (1.23)	0.78 (0.12)	1
	0.14 (0.17)	1.55 (1.31)	0.15 (0.12)	-0.89 (1.03)	0.58 (0.08)	0
Sample T. P. - Plug-In - 1	0.22 (0.23)	2.69 (1.8)	0.18 (0.11)	-2.85 (1.76)	1.1 (0.18)	2
	0.17 (0.19)	1.92 (1.48)	0.16 (0.12)	-1.49 (1.20)	0.77 (0.11)	1
	0.14 (0.17)	1.53 (1.29)	0.15 (0.12)	-0.85 (1.01)	0.57 (0.08)	0
Sample T. P. - Plug-In - 2	0.19 (0.18)	2.06 (1.37)	0.18 (0.11)	-1.19 (1.15))	0.84 (0.14)	2
	0.15 (0.16)	1.55 (1.2)	0.16 (0.12)	-0.73 (1)	0.62 (0.09)	0
	0.13 (0.14)	1.29 (1.09)	0.15 (0.12)	-0.4 (0.84)	0.49 (0.07)	0
Bayesian Porfolio	0.19 (0.17)	2.02 (1.35)	0.18 (0.11)	-1.1 (1.12)	0.82 (0.13)	2
	0.15 (0.15)	1.53 (1.18)	0.16 (0.12)	-0.68(0.98))	0.61 (0.09)	0
	0.13 (0.14)	1.28 (1.08)	0.15 (0.12)	-0.37 (0.82)	0.48 (0.07)	0
Optimal Two Fund Portfolio	0.1 (0.09)	0.97 (0.74)	0.16 (0.11)	0.05 (0.64)	0.42 (0.1)	0
	0.08 (0.08)	0.75 (0.66)	0.14 (0.12)	0.09 (0.51)	0.33 (0.07)	0
	0.07 (0.08)	0.62 (0.62)	0.13 (0.12)	0.11 (0.46)	0.26 (0.05)	0
Norm Constrained - 1 - Variance	0.14 (0.05)	1.18 (0.4)	0.29 (0.1)	0.74 (0.42)	0.15 (0.04)	0
	0.14 (0.05)	1.19 (0.39)	0.29 (0.09)	0.75 (0.40)	0.12 (0.03)	0
	0.14 (0.05)	1.19 (0.38)	0.29 (0.1)	0.75 (0.41)	0.1 (0.03)	0
Norm Constrained - 1 - Return	0.14 (0.05)	1.21 (0.38)	0.29 (0.09)	0.74 (0.4)	0.36 (0.09)	0
	0.15 (0.05)	1.22 (0.39)	0.30 (0.1)	0.76 (0.41)	0.32 (0.08)	0
	0.14 (0.05)	1.21 (0.39)	0.30 (0.1)	0.75 (0.41)	0.28 (0.08)	0
Norm Constrained - 2 - Variance	0.14 (0.06)	1.19 (0.43)	0.28 (0.1)	0.72 (0.45)	0.17 (0.03)	0
	0.14 (0.05)	1.19 (0.41)	0.29 (0.1)	0.73 (0.42)	0.13 (0.02)	0
	0.14 (0.05)	1.18 (0.41)	0.29 (0.1)	0.73 (0.42)	0.1 (0.02)	0
Norm Constrained - 2 - Return	0.14 (0.06)	1.2 (0.43)	0.28 (0.1)	0.72 (0.45)	0.18 (0.03)	0
	0.14 (0.05)	1.19 (0.41)	0.29 (0.1)	0.73 (0.42)	0.13 (0.02)	0
	0.14 (0.05)	1.18 (0.41)	0.29 (0.1)	0.73 (0.42)	0.1 (0.02)	0
Partial Minimum Portfolio	0.14 (0.06)	1.21 (0.41)	0.29 (0.1)	0.73 (0.43)	0.58 (0.11)	0
	0.14 (0.05)	1.18 (0.41)	0.28 (0.1)	0.72 (0.43)	0.55 (0.12)	0
	0.14 (0.05)	1.19 (0.4)	0.28 (0.1)	0.71 (0.43)	0.53 (0.12)	0
Opt. Gran. Clustering - Bayesian porfolio	0.13 (0.08)	1.13 (0.65)	0.21 (0.1)	0.37 (0.52)	0.53 (0.16)	0
	0.09 (0.07)	0.83 (0.55)	0.17 (0.11)	0.25 (0.44)	0.4 (0.12)	0
	0.08 (0.06)	0.7 (0.49)	0.17 (0.1)	0.27 (0.4)	0.32 (0.12)	0
Markowitz - Constrained	0.15 (0.06)	1.25 (0.44)	0.28 (0.09)	0.73 (0.45)	0.19 (0.03)	0
	0.14 (0.06)	1.21 (0.44)	0.27 (0.09)	0.72 (0.41)	0.16 (0.02)	0
	0.14 (0.06)	1.2 (0.42)	0.28 (0.09)	0.69 (0.41)	0.13 (0.02)	0
Markowitz - AR1	0.16 (0.06)	1.36 (0.44)	0.28 (0.09)	0.18 (0.72)	0.87 (0.12)	0
	0.16 (0.06)	1.32 (0.42)	0.27 (0.09)	0.72 (0.43)	0.81 (0.12)	0
	0.15 (0.05)	1.28 (0.39)	0.27 (0.1)	0.69 (0.43)	0.75 (0.13)	0
Markowitz - VAR1	0.15 (0.09)	1.35 (0.68)	0.2 (0.1)	0.72 (0.42)	1.56 (0.07)	0
	0.15 (0.09)	1.33 (0.67)	0.21 (0.1)	0.3 (0.68)	1.52 (0.07)	0
	0.15 (0.08)	1.33 (0.61)	0.22 (0.1)	0.35 (0.63)	1.47 (0.08)	0

Table 2: **Performance overview for** $N = 10$**;** First row reports results for $\tau = 48$, second row for $\tau = 60$ and third row for $\tau = 72$. Investment period from January 2010 to December 2015, 72 months.

N = 20, 100 datasets	Annual. Return	Mean (%)	Sharpe Ratio	CEQ (%)	Turnover	Default (%)
Naive Portfolio	0.14 (0.03)	1.218 (0.2)	0.25 (0.05)	0.85 (0.27)	0.01 (0)	0
	0.14 (0.03)	1.218 (0.2)	0.25 (0.05)	0.85 (0.27)	0.01 (0)	0
	0.14 (0.03)	1.218 (0.2)	0.25 (0.05)	0.85 (0.27)	0.01 (0)	0
Minus Naive Portfolio	-0.15 (0.02)	-1.218 (0.2)	-0.25 (0.05)	-1.85 (0.19)	0.01 (0)	0
	-0.15 (0.02)	-1.218 (0.2)	-0.25 (0.05)	-1.85 (0.19)	0.01 (0)	0
	-0.15 (0.02)	-1.218 (0.2)	-0.25 (0.05)	-1.85 (0.19)	0.01 (0)	0
Sample Tangency Portfolio	0.13 (0.6)	5.25 (3.82)	0.17 (0.11)	-17.44 (5.58)	4.21 (0.52)	28
	0.3 (0.4)	4.26 (2.82)	0.19 (0.12)	-7.48 (2.58)	2.49 (0.29)	5
	0.29 (0.31)	3.37 (2.15)	0.19 (0.11)	-3.86 (1.62)	1.72 (0.18)	4
Sample T. P. - Plug In - 1	0.13 (0.59)	5.08 (3.75)	0.17 (0.11)	-16.64 (5.31)	4.13 (0.51)	27
	0.3 (0.39)	4.19 (2.77)	0.17 (0.11)	-7.16 (2.5)	2.45 (0.29)	5
	0.29 (0.31)	3.32 (2.12)	0.19 (0.11)	-3.72 (1.59)	1.69 (0.18)	44
Sample T. P. - Plug In - 2	0.21 (0.31)	2.91 (2.21)	0.16 (0.11)	-4.53 (2.39)	2.35 (0.3)	5
	0.26 (0.25)	2.72 (1.79)	0.19 (0.11)	-2.09 (1.47)	1.58 (0.2)	3
	0.23 (0.21)	2.35 (1.49)	0.19 (0.11)	-1.18 (1.06)	1.19 (0.13)	3
Bayesian Portfolio	0.21 (0.31)	2.85 (2.17)	0.16 (0.11)	-4.29 (2.30)	2.31 (0.3)	5
	0.25 (0.24)	2.67 (1.76)	0.19 (0.11)	-1.98 (1.44)	1.56 (0.19)	3
	0.23 (0.21)	2.31 (1.49)	0.19 (0.11)	-1.12 (1.04)	1.18 (0.13)	3
Optimal Two fund Portfolio	0.11 (0.11)	1.05 (0.88)	0.15 (0.11)	-0.1 (0.74)	0.87 (0.15)	0
	0.12 (0.1)	1.1 (0.79)	0.18 (0.11)	0.175 (0.71)	0.65 (0.11)	0
	0.11 (0.09)	0.99 (0.71)	0.18 (0.11)	0.22 (0.65)	0.52 (0.08)	0
Norm Constrained - 1 - Variance	0.15 (0.04)	1.2 (0.26)	0.34 (0.08)	0.87 (0.26)	0.26 (0.06)	0
	0.15 (0.03)	1.22 (0.24)	0.34 (0.07)	0.9 (0.23)	0.2 (0.06)	0
	0.15 (0.03)	1.24 (0.24)	0.36 (0.07)	0.93 (0.23)	0.16 (0.05)	0
Norm Constrained - 1 - Return	0.15 (0.04)	1.26 (0.29)	0.33 (0.08)	0.89 (0.28)	0.82 (0.1)	0
	0.15 (0.04)	1.26 (0.26)	0.34 (0.08)	0.91 (0.25)	0.68 (0.11)	0
	0.16 (0.03)	1.28 (0.26)	0.35 (0.07)	0.94 (0.25)	0.59 (0.11)	0
Norm Constrained - 2 - Variance	0.14 (0.04)	1.21 (0.34)	0.3 (0.09)	0.79 (0.34)	0.4 (0.05)	0
	0.15 (0.04)	1.23 (0.31)	0.32 (0.09)	0.86 (0.31)	0.28 (0.03)	0
	0.15 (0.04)	1.25 (0.27)	0.34 (0.08)	0.89 (0.26)	0.21 (0.03)	0
Norm Constrained - 2 - Return	0.14 (0.05)	1.21 (0.34)	0.3 (0.09)	0.79 (0.34)	0.4 (0.05)	0
	0.15 (0.05)	1.23 (0.31)	0.32 (0.09)	0.86 (0.31)	0.28 (0.03)	0
	0.15 (0.04)	1.25 (0.27)	0.34 (0.08)	0.89 (0.26)	0.21 (0.03)	0
Partial Minimum Portfolio	0.14 (0.04)	1.15 (0.29)	0.3 (0.08)	0.77 (0.3)	0.97 (0.14)	0
	0.14 (0.04)	1.19 (0.29)	0.32 (0.08)	0.82 (0.29)	0.86 (0.12)	0
	0.14 (0.04)	1.17 (0.27)	0.32 (0.08)	0.82 (0.27)	0.8 (0.13)	0
Opt. Gran. Clustering - Bayesian Portfolio	0.19 (0.11)	1.63 (0.75)	0.25 (0.08)	0.5 (0.53)	1.08 (0.24)	0
	0.15 (0.1)	1.3 (0.76)	0.22 (0.1)	0.46 (0.54)	0.87 (0.2)	0
	0.13 (0.09)	1.1 (0.73)	0.21 (0.11)	0.44 (0.54)	0.72 (0.18)	0
Markowitz - Constrained	0.16 (0.05)	1.31 (0.36)	0.3 (0.07)	0.83 (0.34)	0.26 (0.04)	0
	0.15 (0.05)	1.29 (0.37)	0.31 (0.08)	0.84 (0.34)	0.22 (0.03)	0
	0.15 (0.04)	1.28 (0.33)	0.31 (0.07)	0.85 (0.3)	0.21 (0.03)	0
Markowitz - AR1	0.17 (0.05)	1.42 (0.38)	0.28 (0.09)	0.76 (0.44)	1.13 (0.09)	0
	0.16 (0.05)	1.37 (0.36)	0.28 (0.08)	0.74 (0.42)	1.07 (0.1)	0
	0.16 (0.05)	1.32 (0.35)	0.27 (0.08)	0.71 (0.41)	1.02(0.1)	0
Markowitz - VAR1	0.11 (0.12)	1.26 (0.84)	0.15 (0.1)	-0.76 (1.23)	1.79 (0.06)	0
	0.12 (0.11)	1.22 (0.8)	0.16 (0.1)	-0.51 (1.39)	1.78 (0.05)	0
	0.12 (0.1)	1.26 (0.78)	0.17 (0.1)	-0.36 (1.73)	1.77 (0.05)	0

Table 3: **Performance overview for** $N = 20$**;** First row reports results for $\tau = 48$, second row for $\tau = 60$ and third row for $\tau = 72$. Investment period from January 2010 to December 2015, 72 months.

N = 30, 100 datasets	Annual. Return	Mean (%)	Sharpe Ratio	CEQ (%)	Turnover	Default (%)
Naive Portfolio	0.15 (0.02)	1.26 (0.17)	0.26 (0.04)	0.65 (0.19)	0.01 (0)	0
	0.15 (0.02)	1.26 (0.17)	0.26 (0.04)	0.65 (0.19)	0.01 (0)	0
	0.15 (0.02)	1.26 (0.17)	0.26 (0.04)	0.65 (0.19)	0.01 (0)	0
Minus Naive Portfolio	-0.15 (0.02)	-1.26 (0.17)	-0.26 (0.04)	-1.87 (0.2)	0.01 (0)	0
	-0.15 (0.02)	-1.26 (0.17)	-0.26 (0.04)	-1.87 (0.2)	0.01 (0)	0
	-0.15 (0.02)	-1.26 (0.17)	-0.26 (0.04)	-1.87 (0.2)	0.01 (0)	0
Sample Tangency Portfolio	x	x	x	x	x	99
	-0.08 (0.51)	7.03 (4.83)	0.17 (0.11)	-32.28 (7.77)	6.7 (0.78)	49
	0.05 (0.41)	4.46 (3.56)	0.15 (0.11)	-16.47 (4.41)	4.05 (0.41)	19
Sample T. P. - Plug In - 1	x	x	x	x	x	99
	-0.06 (0.52)	6.819 (4.75)	0.17 (0.11)	-31.05 (7.44)	6.58 (0.76)	48
	0.05 (0.42)	4.39 (3.49)	0.15 (0.11)	-16.08 (4.41)	3.99 (0.4)	18
Sample T. P. - Plug In - 2	0.07 (0.37)	3.69 (3.11)	0.14 (0.11)	-12.85 (5)	5.54 (0.97)	15
	0.14 (0.34)	2.83 (2.51)	0.14 (0.12)	-6.86 (2.87)	3.2 (0.41)	3
	0.15 (0.25)	2.38 (1.97)	0.14 (0.11)	-4.48 (1.98)	2.27 (0.25)	1
Bayesian Portfolio	0.07 (0.37)	3.56 (3.03)	0.14 (0.11)	-12.29 (4.76)	5.41 (0.95)	13
	0.14 (0.33)	2.81 (2.47)	0.14 (0.12)	-6.7 (2.96)	3.16 (0.42)	2
	0.15 (0.25)	2.35 (1.94)	0.14 (0.11)	-4.33 (1.94)	2.24 (0.24)	1
Optimal Two fund Portfolio	0.08 (0.11)	0.8 (0.85)	0.12 (0.12)	-0.32 (0.79)	1.39 (0.3)	1
	0.09 (0.11)	0.91 (0.85)	0.13 (0.12)	-0.18 (0.71)	1.04 (0.18)	0
	0.08 (0.09)	0.84 (0.76)	0.13 (0.11)	-0.14 (0.61)	0.83 (0.13)	0
Norm Constrained - 1 - Variance	0.14 (0.04)	1.15 (0.29)	0.33 (0.09)	0.83 (0.3)	0.33 (0.09)	0
	0.14 (0.04)	1.18 (0.27)	0.34 (0.08)	0.88 (0.26)	0.26 (0.07)	0
	0.15 (0.04)	1.20 (0.26)	0.35 (0.08)	0.9 (0.26)	0.21 (0.07)	0
Norm Constrained - 1 - Return	0.15 (0.04)	1.23 (0.32)	0.33 (0.1)	0.86 (0.34)	1.11 (0.12)	0
	0.15 (0.04)	1.25 (0.31)	0.34 (0.09)	0.89 (0.31)	1.01 (0.11)	0
	0.15 (0.04)	1.25 (0.33)	0.34 (0.1)	0.9 (0.33)	0.88 (0.11)	0
Norm Constrained - 2 - Variance	0.14 (0.07)	1.2 (0.5)	0.26 (0.11)	0.63 (0.54)	0.85 (0.09)	0
	0.15 (0.05)	1.24 (0.39)	0.3 (0.1)	0.79 (0.4)	0.52 (0.06)	0
	0.15 (0.05)	1.25 (0.37)	0.32 (0.1)	0.849 (0.37)	0.37 (0.04)	0
Norm Constrained - 2 - Return	0.14 (0.07)	1.2 (0.49)	0.26 (0.11)	0.63 (0.54)	0.87 (0.11)	0
	0.15 (0.05)	1.24 (0.39)	0.3 (0.1)	0.79 (0.4)	0.52 (0.06)	0
	0.15 (0.05)	1.25 (0.37)	0.32 (0.1)	0.849 (0.37)	0.37 (0.04)	0
Partial Minimum Portfolio	0.13 (0.06)	1.13 (0.43)	0.29 (0.11)	0.72 (0.47)	1.54 (0.19)	0
	0.14 (0.05)	1.15 (0.35)	0.31 (0.1)	0.78 (0.35)	1.26 (0.14)	0
	0.14 (0.05)	1.13 (0.34)	0.31 (0.1)	0.77 (0.35)	1.13 (0.14)	0
Opt. Gran. Clustering - Bayesian Portfolio	0.18 (0.11)	1.62 (0.84)	0.23 (0.1)	0.38 (0.66)	1.57 (0.35)	0
	0.14 (0.09)	1.25 (0.68)	0.21 (0.08)	0.35 (0.47)	1.26 (0.3)	0
	0.12 (0.08)	1.09 (0.62)	0.2 (0.09)	0.39 (0.43)	1.06 (0.29)	0
Markowitz - Constrained	0.16 (0.06)	1.3 (0.43)	0.3 (0.1)	0.8 (0.43)	0.29 (0.04)	0
	0.15 (0.05)	1.24 (0.39)	0.29 (0.08)	0.78 (0.37)	0.25 (0.03)	0
	0.15 (0.05)	1.22 (0.35))	0.29 (0.08)	0.77 (0.33)	0.21 (0.03)	0
Markowitz - AR1	0.17 (0.05)	1.48 (0.38)	0.28 (0.08)	0.76 (0.41)	1.25 (0.09)	0
	0.17 (0.05)	1.41 (0.35)	0.27 (0.07)	0.71 (0.36)	1.2 (0.1)	0
	0.16 (0.05)	1.36 (0.34)	0.27 (0.07)	0.7 (0.37)	1.15 (0.1)	0
Markowitz - VAR1	0.09 (0.17)	1.25 (1.18)	0.12 (0.11)	-1.74 (1.6)	1.85 (0.05)	0
	0.11 (0.15)	1.33 (1.04)	0.14 (0.11)	-1.1 (1.34)	1.85 (0.04)	0
	0.11 (0.12)	1.23 (0.87)	0.14 (0.1)	-0.84 (1.1)	1.84 (0.04)	0

Table 4: **Performance overview for** $N = 30$**;** First row reports results for $\tau = 48$, second row for $\tau = 60$ and third row for $\tau = 72$. Investment period from January 2010 to December 2015, 72 months.

3.3 Results - Clustering

In this section we examine the performance of agglomerative clustering procedure with naive medoids discussed in Section 2.8.1, combined with standard methods from the Chapter 2. Once again we generate 100 random datasets with $N = 200$ assets, and compute weights for all 16 methods, for clusters of size 10, 20 and 30. The parameters settings of the methods are the same as in Table 1. To make the situation more challenging, we choose the rolling window length $\tau = 48$, which is relatively short to the asset count. Measured metrics are the same as described in Section [3.2.2].

The standard approach without clustering and $N = 200$ assets is computationally not feasible due to $N > \tau$, resulting in the singular covariance matrix $\hat{\Sigma}$. Therefore we compare this clustering approach with 10, 20, 30 clusters to the results from Section [3.2.2], for $N = 10$, $N = 20$ and $N = 30$, with rolling window length $\tau = 48$. There are many other possible scenarios we could consider due to the high number of dimensions, in which we could move (e.g. 3 types of medoids, different asset count N, window length, cluster count and others), but these are not investigated further.

The results are given on the next page in Table 5. For 10 clusters the results are quite disappointing, as almost none of the strategies showed better performance than the standard approach with just 10 assets, except Markowitz - Constrained and Markowitz - VAR(1). The results are more optimistic for 20 and 30 clusters, where Plug-In rules, Bayesian Approach, Optimal Two Fund, Norm-Constrained and Partial minimum-variance portfolios show better or similar Sharpe Ratio performance than theirs counterpart without clustering. The $1/N$ rule also improved its Sharpe Ratio in case of 20 and 30 clusters. Overall, the best method in this setting is the 1-Norm-Constrained portfolio calculated over 20 and 30 clusters.

Clustering, N = 200, 100 datasets	Annual. Return	Mean (%)	Sharpe Ratio	CEQ (%)	Turnover	Default (%)
Naive Portfolio	0.15 (0.02)	1.37 (0.16)	0.22 (0.02)	0.37 (0.14)	0.61 (0.02)	0
	0.14 (0.01)	1.22 (0.11)	0.27 (0.03)	0.72 (0.12)	0.53 (0.01)	0
	0.15 (0.01)	1.24 (0.11)	0.28 (0.03)	0.73 (0.12)	0.45 (0.01)	0
Minus Naive Portfolio	-0.17 (0.01)	-1.37 (0.16)	-0.22 (0.02)	-2.38 (0.16)	0.61 (0.02)	0
	-0.15 (0.05)	-1.22 (0.11)	-0.27 (0.03)	-1.73 (0.11)	0.53 (0.01)	0
	-0.15 (0.05)	-1.24 (0.11)	-0.28 (0.03)	-1.74 (0.1)	0.45 (0.01)	0
Sample Tangency Portfolio	-0.04 (0.14)	0.84 (1.06)	0.06 (0.07)	-4.48 (1.61)	9.1 (0.81)	0
	0.55 (0.44)	7.43 (2.4)	0.27 (0.08)	-11.63 (3.75)	27.5 (2.26)	10
	x	x	x	x	x	96
Sample T. P. - Plug-In - 1	-0.03 (0.13)	0.82 (1.03)	0.06 (0.07)	-4.28 (1.56)	8.92 (0.79)	0
	0.54 (0.43)	7.28 (2.34)	0.27 (0.08)	-11.14 (3.67)	27.08 (2.37)	7
	x	x	x	x	x	96
Sample T. P. - Plug-In - 2	0 (0.1)	0.63 (0.79)	0.06 (0.07)	-2.36 (1.02)	6.83 (0.61)	0
	0.4 (0.23)	3.93 (1.34)	0.26 (0.08)	-1.83 (1.69)	15.03 (1.32)	1
	0.3 (0.36)	4.46 (2.32)	0.2 (0.1)	-7.54 (3.45)	26.84 (2.68)	7
Bayesian Porfolio	0 (0.09)	0.62 (0.78)	0.06 (0.07)	-2.26 (0.98)	6.69 (0.59)	0
	0.4 (0.22)	3.85 (1.37)	0.26 (0.08)	-1.68 (1.64)	14.73 (1.3)	1
	0.31 (0.35)	4.37 (2.28)	0.2 (0.1)	-7.14 (3.32)	26.29 (2.63)	0
Optimal Two Fund Portfolio	-0.02 (0.06)	-0.01 (0.53)	0.06 (0.07)	-0.83 (0.6)	3.56 (0.65)	0
	0.16 (0.08)	1.42 (0.55)	0.25 (0.08)	-0.55 (0.57)	5.44 (0.68)	0
	0.11 (0.08)	1.04 (0.61)	0.19 (0.1)	0.25 (0.57)	6.52 (0.84)	0
Norm Constrained - 1 - Variance	0.08 (0.03)	0.75 (0.24)	0.15 (0.05)	0.13 (0.24)	1.65 (0.2)	0
	0.14 (0.02)	1.13 (0.18)	0.34 (0.05)	0.85 (0.18)	0.93 (0.07)	0
	0.14 (0.02)	1.15 (0.18)	0.35 (0.05)	0.88 (0.18)	0.83 (0.08)	0
Norm Constrained - 1 - Return	0.08 (0.03)	0.78 (0.23)	0.16 (0.04)	0.14 (0.23)	1.72 (0.1)	0
	0.14 (0.02)	1.13 (0.18)	0.34 (0.05)	0.85 (0.18)	0.93 (0.07)	0
	0.14 (0.02)	1.15 (0.18)	0.35 (0.05)	0.88 (0.18)	0.83 (0.08)	0
Norm Constrained - 2 - Variance	0.07 (0.03)	0.7 (0.26)	0.14 (0.05)	0.1 (0.27)	2.19 (0.15)	0
	0.14 (0.03)	1.14 (0.25)	0.33 (0.08)	0.84 (0.25)	2.91 (0.15)	0
	0.14 (0.04)	1.35 (0.31)	0.3 (0.09)	0.78 (0.31)	4.02 (0.18)	0
Norm Constrained - 2 - Return	0.07 (0.03)	0.7 (0.26)	0.14 (0.05)	0.1 (0.27)	2.18 (0.15)	0
	0.14 (0.03)	1.14 (0.25)	0.33 (0.08)	0.84 (0.25)	2.91 (0.15)	0
	0.14 (0.04)	1.35 (0.31)	0.3 (0.09)	0.78 (0.31)	4.02 (0.18)	0
Partial Minimum Portfolio	0.07 (0.03)	0.71 (0.26)	0.14 (0.05)	0.08 (0.27)	1.84 (0.12)	0
	0.13 (0.03)	1.1 (0.25)	0.32 (0.07)	0.81 (0.24)	2.13 (0.15)	0
	0.13 (0.04)	1.11 (0.27)	0.32 (0.07)	0.8 (0.26)	2.65 (0.22)	0
Opt. Gran. Clust. - Bayes. Porfolio	0 (0.09)	0.62 (0.78)	0.06 (0.07)	-2.26 (0.98)	6.69 (0.59)	1
	0.4 (0.22)	3.85 (1.32)	0.26 (0.08)	-1.68 (1.64)	14.73 (1.3)	0
	0.31 (0.35)	4.37 (2.28)	0.2 (0.1)	-7.14 (3.32)	26.29 (2.63)	0
Markowitz - Constrained	0.09 (0.03)	0.89 (0.27)	0.3 (0.06)	0.17 (0.28)	1.21 (0.06)	0
	0.1 (0.03)	0.93 (0.25)	0.16 (0.06)	0.22 (0.27)	1.13 (0.07)	0
	0.09 (0.04)	0.89 (0.33)	0.17 (0.05)	0.14 (0.33)	1.02 (0.08)	0
Markowitz - AR1	0.13 (0.04)	1.25 (0.28)	0.2 (0.04)	0.21 (0.31)	1.37 (0.4)	0
	0.17 (0.04)	1.4 (0.27)	0.19 (0.09)	0.86 (0.29)	1.48 (0.04)	0
	0.16 (0.04)	1.4 (0.32)	0.28 (0.07)	0.78 (0.34)	1.49 (0.05)	0
Markowitz - VAR1	0.13 (0.07)	1.29 (0.49)	0.17 (0.07)	-0.14 (0.6)	1.76 (0.04)	0
	0.13 (0.08)	1.28 (0.55)	0.18 (0.05)	0.11 (0.65)	1.85 (0.03)	0
	0.11 (0.11)	1.25 (0.81)	0.15 (0.1)	-0.76 (1.09)	1.86 (0.05)	0

Table 5: **Performance overview for Clustering with naive medoids and** $N = 200$ **assets;** First row reports results for 10 clusters, second row for 20 clusters and third row for 30 clusters. Investment period from January 2010 to December 2015, 72 months.

3.4 Macroeconomic Factors

In this section we outline the motivation behind our strategy switching approach. To have even more robust results we decided to simplify Norm-Constrained and Partial minimum

variance portfolios by omitting cross validation step, by using fixed δ and k. Having this setting with reduced complexity we calculated 5000 datasets for $N = 10$ assets with rolling window length $\tau = 48$, whereas the computation of one dataset took only 5 seconds. The performance measures are given in Table 6. In contrast to the results in the first row of Table 2, where results for $N = 10$ and $\tau = 48$ are given, we have expanded the investment period in Table 6 to the maximum length which is 96 months with $\tau = 48$. The only two portfolios with positive Certainty Equivalent Ratio are 1-Norm-Constrained portfolio with fixed $\delta = 1$ (short selling budged is 0) and Markowitz-Constrained. These two portfolios are quite similar as both of them are restricted to short positions and their sum of weights is 1. The only difference is that the Norm-Constrained portfolio is minimum-variance portfolio, whereas the Markowitz-Constrained is mean-variance.

id	N = 10, 5000 datasets	Annual. Ret.	Mean (%)	Sharpe Ratio	CEQ (%)	Turnover	Default (%)
1.	Naive Portfolio	0.11 (0.06)	1.1 (0.48)	0.17 (0.07)	-0.11 (0.69)	0.01 (0)	0
2.	Minus Naive Portfolio	-0.15 (0.05)	-1.1 (0.48)	-0.17 (0.07)	-2.3 (0.54)	0.01 (0)	0
3.	Sample Tangency Portfolio	-0.06 (0.2)	1.18 (1.82)	0.06 (0.09)	-6.71 (3.18)	1.18 (0.19)	11.1
4.	Sample T. P. - Plug-In - 1	-0.05 (0.19)	1.16 (1.8)	0.06 (0.09)	-6.45 (3.15)	1.15 (0.19)	10.4
5.	Sample T. P. - Plug-In - 2	-0.01 (0.16)	0.91 (1.45)	0.06 (0.09)	-3.82 (2.23)	0.89 (0.15)	4.46
6.	Bayesian Porfolio	0 (0.16)	0.9 (1.42)	0.06 (0.09)	-3.66 (2.23)	0.87 (0.15)	4.22
7.	Optimal Two Fund Portfolio	0.03 (0.09)	0.52 (0.91)	0.06 (0.09)	-0.95 (1.65)	0.45 (0.12)	1.52
8.	1-Norm-Constrained - $\delta = 1$	0.09 (0.06)	0.82 (0.44)	0.17 (0.08)	0.1 (0.7)	0.11 (0.03)	0
9.	1-Norm-Constrained - $\delta = 2$	0.07 (0.07)	0.73 (0.48)	0.15 (0.09)	-0.06 (0.79)	0.19 (0.05)	0
10.	2-Norm-Constrained - $\delta = 1$	0.07 (0.07)	0.73 (0.48)	0.15 (0.09)	-0.07 (0.81)	0.19 (0.06)	0
11.	2-Norm-Constrained - $\delta = 2$	0.07 (0.07)	0.73 (0.48)	0.15 (0.09)	-0.07 (0.84)	0.19 (0.07)	0
12.	Partial Minimum Portfolio - $k = 4$	0.07 (0.07)	0.73 (0.47)	0.15 (0.09)	-0.07 (0.83)	0.18 (0.06)	0
13.	Opt. Gran. Clust. - Bayes. Porfolio	0.07 (0.1)	0.75 (1.11)	0.1 (0.09)	-0.57 (1.76)	0.5 (0.15)	0.7
14.	Markowitz - Constrained	0.09 (0.05)	0.86 (0.36)	0.16 (0.07)	0.08 (0.43)	0.2 (0.03)	0
15.	Markowitz - AR1	0.09 (0.08)	0.99 (0.62)	0.15 (0.09)	-0.24 (0.83)	0.96 (0.11)	0
16.	Markowitz - VAR1	0.09 (0.12)	1.15 (0.88)	0.12 (0.09)	-1.24 (1.4)	1.6 (0.06)	0.2

Table 6: **Performance overview for** $N = 10$**;** Simplified setup without cross-validation, affected methods are Norm-Constrained and Partial Minimum portfolios, the other methods are used as in Table 1. Rolling window length $\tau = 48$. Investment period from January 2008 to December 2015, 96 months.

In Figures 3 and 4 we focus on the analysis of inconsistent behaviour of presented methods during the year 2015, presumably driven by macroeconomic factors. Very similar unstable behaviour can be also noticed in other time periods. The first important observation in Figure3 is that the $-1/N$ portfolio (index $i = 2$) had the highest average return in 5 months; specifically in April, June, August, September and December. Secondly, a very similar pattern of average strategy returns in Figure 3 can be noticed in March, May, July and November. Another interesting observation in Figure3 can be noticed in May when market gained approximately on average 1% and strategies $8 - 14$ were in positive numbers, on other hand, in

July market was in decline with return bellow -1%, surprisingly strategies $8-14$ managed to stay again above zero.

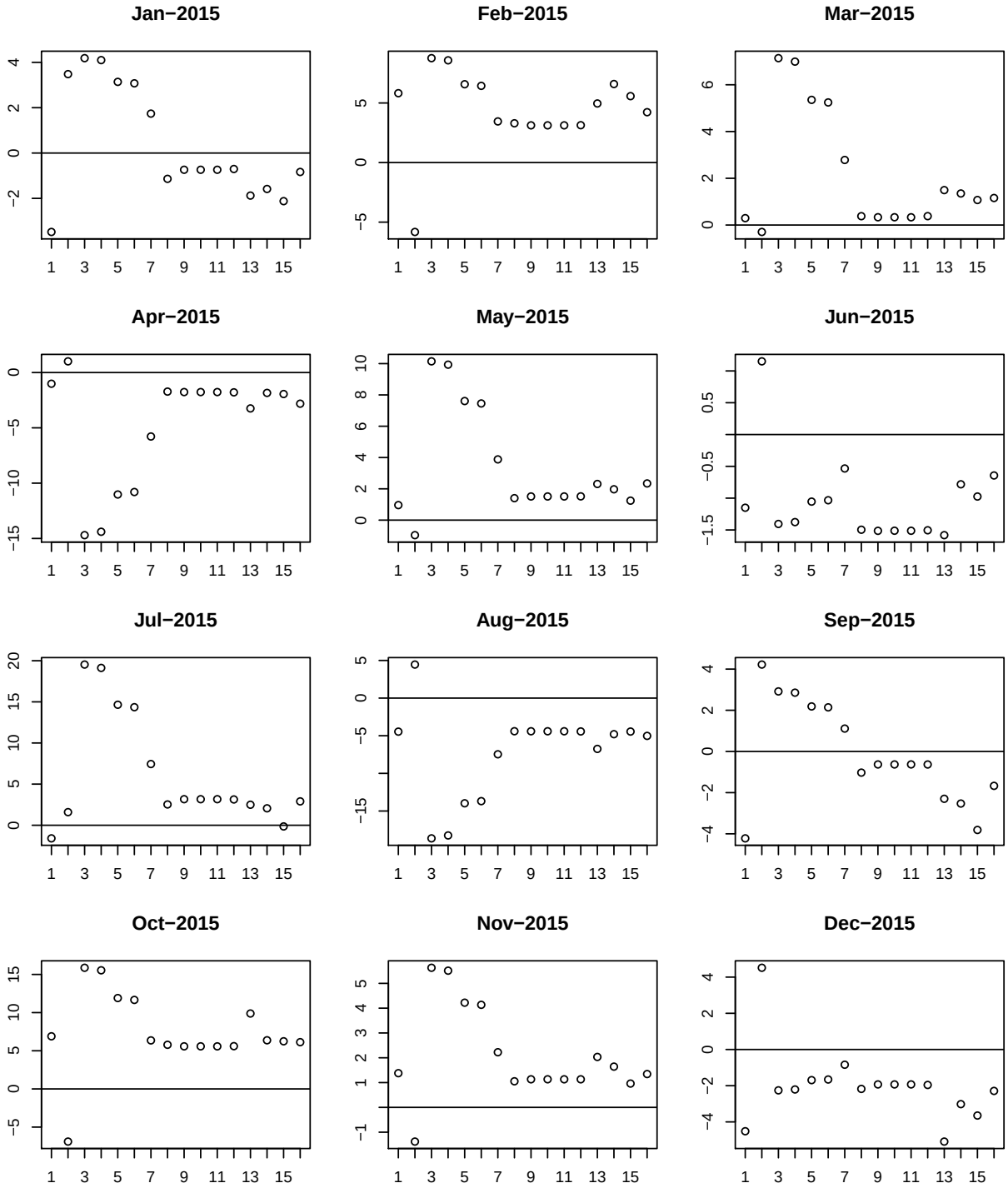

Figure 3: **Monthly Average Return by strategies (%);** The X-axis depicts the portfolio rule with index i according to the first column of Table 6. The Y-axis represents the monthly average return in the given month using the strategy i, calculated over 5000 datasets from Table 6.

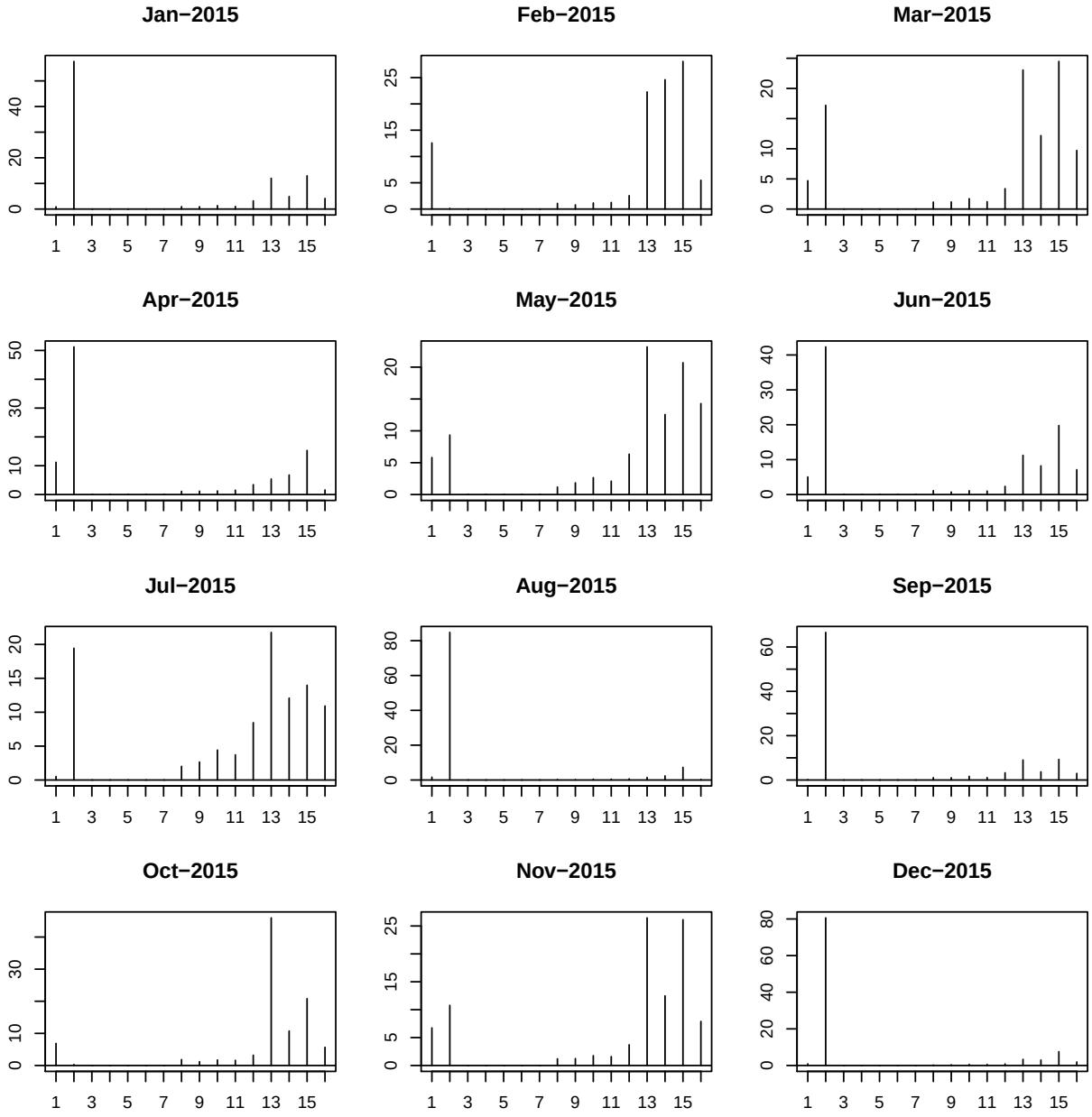

Figure 4: **Highest Return in the month (%);** The X-axis depicts the portfolio rule with index i according to the first column of Table 6. The Y-axis represents the percentage of datasets with the highest return in the given month using the strategy i, calculated over 5000 datasets from Table 6.

This inconsistent behaviour makes us think that timing the market, and using the appropriate strategy is very crucial, i.e. holding the same strategy for the whole investment period is not the wisest. The remaining problem is how to make the decision? How to choose the superior strategy for the next month?

4 Dynamic Approach

Motivated by brief analysis in Section [3.4], in this Chapter we propose a strategy switching approach for choosing the superior method in our asset allocation problem. Our target is to predict the strategy with the highest expected return in the next month, for every one of the 5000 datasets from Table 6 and for every strategy individually. From this analysis we exclude the strategies with non-zero probability of default (indices $2-7$, 13 and 16, in total 7 strategies out of 16), which we consider too risky and too volatile. Therefore, for 1 dataset and 1 month ahead prediction we predict 9 returns, from which we choose the one with the highest return and invest into such strategy.

The data we have available for this task is 5000 datasets with 96 monthly returns of 9 strategy returns with corresponding returns of $N = 10$ risky assets from which the weights were computed. To have a balance between the training and the out-of-sample period, we only predict and invest into the last 24 months (indices of months $73-96$), while the prediction for month $t+1 \in \{73, \ldots, 96\}$ uses only information to time t. Those last 24 months are used to calculate out-of-sample performance measures.

4.1 Benchmarks

Before describing the best performing approach, we start with the simpler ones.

4.1.1 Simple Moving Average

The simplest and the most straightforward approach is to choose the strategy with the highest average return in the last few months, for the following month. We consider approaches for the past $1-24$ months. If just one month is considered, this strategy invests in the portfolio with the highest return in the most recent observation (past month).

4.1.2 Autoregressive model

Another simple approach we consider is the standard AR(1), and the AR$(1,1)$ model, which are used to predict returns one month ahead. The model is trained using the expanding window technique, i.e. when predicting time $t+1$, the model has access to data from the month 1 to t.

4.1.3 Seeing the future

To see the upper limit which our proposed approach can reach is to use a method, that sees the future. Such method invests in the strategy that has the highest return in the invested month. As the investing decision is carried out in the first day of the month, information about end of the month return is seen from the future. Such strategy cannot be executed in a real-world environment. This method is used as the maximum benchmark, i.e. if we predict all returns in the correct order, we get the same returns as this methods. Important to notice is that this strategy offers only the highest return, not the highest Sharpe Ratio and Certainty Equivalent Return.

4.2 Neural Networks

Neural Networks have been successfully used in a number of various financial applications. In the asset allocation problem, Neural Networks are typically used to predict portfolio weights of the risky assets. When the portfolio weights are predicted by a Neural Network or another Machine Learning tool, the investor is not always able to see the intuition behind the prediction. In our use case we predict the portfolio returns, which is a time series forecasting problem, a special type of regression problem with ordered data. Such approach gives the investor an advice into which strategy to invest. As the investor knows the intuition behind every single method, it is easier to interpret the expected good performing method in contrast to when the weights are predicted. The network is trained using the expanding window technique, i.e. when predicting time $t + 1$, the model has access to data from onths 1 to t.

Our technical setup that we used to make these predictions is Python (v3.6.5), Keras (v2.2.4) running on top of TensorFlow GPU (v1.9.0) and cudatoolkit (v9.0). To keep the training-optimization process efficient, we use a graphic card Nvidia GeForce GTX 1080 Ti, the rest of the hardware is Intel CPU i7-9700K with 32GB RAM. Practical information regarding the implementation of the Neural Networks are in [4, 14].

Neural Networks can be denoted as a special type of function $y = f(x) + \epsilon$, where x and y are multidimensional tensors, ϵ is the error vector. For one dataset in our case, y is a two-dimensional tensor with dimensions 1×9 and x is either two or three dimensional, depending on the type of Neural Network used. Tensor x consists of some amount of past predictors, denoted by k. For one timestep and one dataset we have $N = 10$ asset returns, $p = 9$ strategies returns and a macroeconomic factor which is the monthly average return of all 1000 risky assets. The k denotes the number of past timesteps taken into account. Such

transformation of data is called the supervised learning; demonstrated on our data with 96 monthly returns in total, when $k = 12$ months, we use the first 12 months to predict the 13th month, 2nd to 13th months data predict the 14th month and it continues in this manner until the 96th month. This means that the Neural Network can work with 13th to 96th month, whereas the last 24 months (73rd to 96th month) are to be predicted using the expanding window starting in the 13th month. The inputs are scaled to be in the range -1 to 1, but it is important to set the scaler on the training set first, then remember the settings of the scaler (min and max values of the training set) and only then use this scaler on the validation set - the set from which we predict returns. This is the only way we can be sure that the information from the future is not used during the training. Another important note is that we do not shuffle datasets before training process, as the data are ordered monthly.

The training of the Neural Network is a process of finding the weights of function f that minimizes or maximizes the loss function, depending on the problem. The optimizers we use are Stochastic Gradient Descent and Adaptive Moment Estimation (Adam) with various parameters. Which optimize the Neural Network over some number of epochs and the number of objects that are optimized for at once is called a batch. We have experimented with minimizing various loss functions, including root mean squared error ($RMSE$), mean absolute error (MAE) and others, but mostly the MAE is used;

$$MAE = \frac{\sum_{i=1}^{n} \mid \hat{y}_i - y_i \mid}{n} = \frac{\sum_{i=1}^{n} \mid \epsilon_i \mid}{n}, \tag{89}$$

which can be calculated for one dataset ($n = 9$) or for all datasets ($n = 5000 \times 9$ (the indexing is simplified). The problem with such loss function is that it does not minimize exactly what is important. Therefore we also experiment with custom loss function, which is the MAE with added (+) penalization for the maximal returns being too far away. Thus the penalization is the MAE of maximal returns in predicted and maximal returns in true returns. Another important aspect of the Neural Networks construction is the width (number of neurons), the depth (number of layers) and the activation function used in the layers. Additionally some other features such as regularization techniques, e.g. some norm of weights or Dropout layer (randomly setting a fraction of input units to 0) can be used. This is a lot of decisions to be made before setting up a Neural Network. In our work we address this problem with a simple grid search and multiple runs of Neural Network training. Multiple runs are also recommended since we use stochastic optimization. The disadvantage of such approach is the large computational complexity. In our hyperparameter optimization we focus on the

following metrics (prediction $\hat{y}$ versus true values y):

- the percentage of the accurate sign prediction,
- the percentage of the highest return match,
- the MAE and our custom loss function.

4.2.1 Feedforward Neural Networks

Our starting point in the world of Neural Networks is the Feedforward Neural Network, where we use different widths and depths of the network with various combinations of Dense (sigmoid and tanh activation functions) and Dropout layers. In practice, this type of Neural Network is not the best known approach for time series problems. Therefore, we optimize hyperparameters only over a small grid. To utilize this type of Neural Network, we need to construct a two dimensional input x. In our case, for 5000 datasets and parameter $k = 10$, and when the month 80 is to be predicted, we use months 11 ($k = 10$) to 79 (predicted month is 80) for training. Those data depend on the last $k = 10$ data, as for one timestep we have 1 macro predictor, $N = 10$ asset returns and $p = 9$ portfolio rules, this gives us in total 20 predictors at one given time. Then we transform the data into supervised learning with $k = 10$, the Neural Network predicts 9 portfolio returns as a function of $k \times 20 = 200$ predictors. In our implementation use the online training approach. That means that we create a Neural Network when the first month (73rd) is predicted, when we predict one of the months $74 - 96$, we use the Neural Network from the past and train it on just some number of past months - also a hyperparameter that can be fine tuned. Thus means that we also need another set of parameters; number of epochs on which the first big Neural Network is trained, number of epochs on which the updating of the already trained Neural Network is done, and the batch size. An alternative approach is to create a new Neural Network 24 times for every predicted month $(73 - 96)$.

4.3 Long Short-Term Memory Network

Our final strategy switching approach is based on the Long Short-Term Memory Neural Network, one of the most commercially used tool of artificial intelligence, developed by Hochreiter and Schmidhuber [14]. Successfully used in the language modelling, image captioning, hand writing generation, image generation, question answering, processing video to text and even composing music. This is a special kind of a Recurrent Neural Network that is capable of

learning long-term dependencies, i.e. well suited to processing and making predictions based on a time series data. A standard LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. For thorough discussion and to gain a better the understanding of LSTM Neural Networks we recommend an online blog by Pitis [25], practical information can be found in the online books by Brownlee [4, 14]. In our implementation we use Keras CuDNNLSTM layer, which is a fast implementation of the standard LSTM layer using the NVIDIA CUDA® Deep Neural Network library (cuDNN). CuDNNLSTM layer can be only run on a NVIDIA GPU with the TensorFlow backend.

In practice, the main difference to the Feedforward Neural Network is the 3 dimensional input tensor x, where the first dimensions is the number of datasets, second dimension is the number of timesteps and the last stands for features, specifically in our case with 5000 datasets, $k = 5$ and 20 predictors (features), x has dimensions $5000 \times 5 \times 20$ or $5000 \times 1 \times 100$. To distinguish between those two input options, in our code we use a hyperparameter $timestep = multiple$ or $timestep = single$, according to which we reshape data. In the latter case, when $timestep = single$, the entire sequence is used as a one time step, in which case we have a Feedforward Neural Network with LSTM units. This is also considered a hyperparameter which we optimize.

4.4 Dynamic Approach - Comparison

In the first place, we describe the hyperparameters used in our Neural Networks. After a short hyperparameter optimization, we have the Feedforwardard Neural Network model with 2 Dense layers (90 neurons and sigmoid, 9 neurons and linear), custom loss function, $k = 6$ past predictors, batch size 1000, 10 epochs for the first training, then we follow with 3 epochs when updating. We consider last 5 months in the online training. The optimizer with index 0 was used, for more information see Appendix A. This Feedforward NN model correctly guessed 54.5% of signs and accurately predicted the strategy with the highest return with probability 19.9%.

The LSTM Neural Network model is composed of the first CuDNNLSTM layer (9 cells) and a Dense layer (9 neurons and linear), custom loss function, $k = 4$ past predictors, batch size 5000, 20 epochs for the first training, then we follow with 7 epochs when updating. We consider last 10 months in the online training. The optimizer with index 1 was used, for more information see Appendix A. This LSTM model correctly guessed 54.9% of signs and accurately predicted the strategy with the highest return with probability 24.5%.

N = 10, 5000 datasets	Annual. Ret.	Mean (%)	Sharpe Ratio	CEQ (%)	Turnover	Default (%)
AR(1)	0.06 (0.12)	0.62 (0.95)	0.11 (0.17)	-0.13 (0.92)	0.96 (0.39)	0
AR(2)	0.06 (0.12)	0.57 (0.99)	0.1 (0.17)	-0.2 (0.95)	1.09 (0.35)	0
AR(1, 1)	0.07 (0.12)	0.63 (0.99)	0.11 (0.17)	-0.14 (0.93)	1.04 (0.38)	0
Average Past Mean - 1 month	-0.12 (0.1)	-0.95 (0.94)	-0.19 (0.19)	-1.77 (0.99)	1.77 (0.32)	0
Average Past Mean - 3 months	-0.02 (0.13)	-0.05 (1.11)	-0.02 (0.19)	-0.93 (1.07)	1.24 (0.28)	0
Average Past Mean - 6 months	0.04 (0.14)	0.46 (1.09)	0.07 (0.17)	-0.49 (1)	1.01 (0.27)	0
Average Past Mean - 12 months	0.09 (0.14)	0.9 (1.1)	0.14 (0.15)	-0.15 (0.94)	0.86 (0.31)	0
Average Past Mean - 24 months	0.09 (0.14)	0.89 (1.1)	0.13 (0.15)	-0.19 (0.95)	0.77 (0.33)	0
Feedforward Neural Network	0.09 (0.13)	0.85 (1)	0.13 (0.15)	-0.2 (0.91)	1.6 (0.32)	0
LSTM	0.13 (0.13)	1.19 (1.01)	0.18 (0.14)	0.04 (0.94)	1.18 (0.34)	0
Seeing The Future - Benchmark	0.98 (0.22)	5.87 (0.97)	1.57 (0.33)	5.43 (0.8)	2.65 (0.74)	0

Table 7: **Performance overview for the Dynamic Approach** $N = 10$**;** The subject of combination are the methods from Table 6. Investment period from January 2014 to December 2015, 24 months.

N = 10, 5000 datasets	Annual. Ret.	Mean (%)	Sharpe Ratio	CEQ (%)	Turnover	Default (%)
Naive Portfolio	0.02 (0.08)	0.21 (0.63)	0.06 (0.14)	-0.3 (0.73)	0.04 (0)	0
Minus Naive Portfolio	-0.03 (0.07)	-0.21 (0.63)	-0.06 (0.14)	-0.72 (0.6)	0.04 (0)	0
1-Norm-Constrained - $\delta = 1$	0.09 (0.07)	0.77 (0.52)	0.21 (0.14)	0.4 (0.68)	0.15 (0.02)	0
1-Norm-Constrained - $\delta = 2$	0.1 (0.08)	0.87 (0.64)	0.22 (0.16)	0.43 (0.68)	0.24 (0.04)	0
2-Norm-Constrained - $\delta = 1$	0.1 (0.08)	0.87 (0.64)	0.22 (0.16)	0.43 (0.68)	0.24 (0.04)	0
2-Norm-Constrained - $\delta = 2$	0.1 (0.08)	0.87 (0.64)	0.22 (0.16)	0.43 (0.68)	0.24 (0.04)	0
Partial Minimum Portfolio - $k = 4$	0.1 (0.08)	0.87 (0.63)	0.22 (0.16)	0.44 (0.68)	0.23 (0.04)	0
Markowitz - Constrained	0.08 (0.09)	0.75 (0.66)	0.18 (0.16)	0.26 (0.68)	0.24 (0.05)	0
Markowitz - AR1	0.09 (0.1)	0.82 (0.79)	0.17 (0.16)	0.18 (0.83)	1.11 (0.17)	0

Table 8: **Performance overview for the Static Approach** $N = 10$**;** Investment period from January 2014 to December 2015, 24 months.

In Tables 7 and 8 we have the empirical comparison of the Dynamic and the Static Approach. The most promising Dynamic Approach using the LSTM model has not proved to be superior to the Static Approach. Other Dynamic Approaches have failed even more.

4.5 LSTM - Further Improvements

As we observed in Section [4.4], Dynamic Approach using the LSTM Neural Network has not proved to be a superior to the static methods, but still beating the Feedforward Neural Network and other strategy switching approaches. Here we discuss some issues which might be behind this and propose further improvements to the LSTM Neural Network.

1. Neural Networks are very data hungry, using more and longer datasets should be helpful. As the first improvement, we would generate more than 5000 datasets, generating weights for 5000 datasets takes about 24 hours.

2. We are of opinion, that the outputs from the asset allocation methods are affected by other factors mostly alongside macroeconomic factors such as interest rates, inflation, unemployment and economic growth. Adding more macroeconomic factors to the predictors base should improve the prediction power.

3. We built the hyperparameter optimization framework, for more information see Appendix A, but did not have enough time to take advantage of its power. Running the grid search for a few days, maybe a week would significantly improve the performance.

4. Define a new tailor made loss function made specifically for this problem. It should be based on minimizing the error which is created when the highest return is not the same as the highest return in the true vector of returns. Our custom loss function (MAE + MAE of highest returns) was introduced in the last days of this work and proper hyperparameter optimization was not carried out.

Conclusion

In this thesis we have discussed 16 asset allocation models, which tackle the parameter uncertainty problem in various ways. In the first part, we have described the technical aspect of construction of the asset allocation models with focus on the key features that represent each model. As a starting point we looked at the findings of DeMiguel, Garlappi and Uppal [8], who analysed 14 different asset allocation models. In their work, none of the 14 models was able to outperform the 1/N strategy across seven empirical datasets. In our work we used a unique and different approach compared to the rest of the available literature. The datasets we used were randomly generated from the real data, instead of the approach using the simulated data. Robust comparisons of 16 asset allocation models were conducted on thousands of datasets with the main focus on Sharpe Ratio, Certainty Equivalent Return and Turnover. Empirical Section [3] of this thesis have shown that 1-Norm-Constrained portfolio with cross-validation targeting the minimal variance proved to be the best among the models in our setting with the investment period spanning from 2010 to 2015. This model also demonstrates a very desirable quality of having a low Turnover.

Building up on the results described above, we have analysed the performance of using agglomerative clustering algorithms combined with the standard asset allocation models. This approach reduces the dimension of the problem and therefore should also decrease the error in parameter estimation. The analysis was conducted on a challenging example where we have examined out-of-sample returns of 100 datasets, each consisting of a sample of randomly picked 200 assets, with the rolling window of only 48 months. This setting has led to a singularity of the sample covariance matrix of returns which causes the standard methods to be unimplementable. For this reason we have presented a shrinkage technique, which is a simple and efficient improvement of the sample covariance matrix. Such estimator is typically much more efficient, well conditioned and demonstrates positive definiteness. Correlation matrix estimated with the shrinkage technique was used as an input for the agglomerative clustering algorithm, with the distance $1 - cor(r_i, r_j)$. After this reduction of the dimension of data, the standard asset allocation methods were applied. The best performing model in this analysis was again the 1-norm-constrained portfolio.

Our subsequent research focused on inconsistent returns of portfolio returns in time, where we observed unstable behaviour during the year 2015. The highest average return in as many as 5 months was produced by $-1/N$ strategy, which short sells all available assets. This behaviour is natural and expected, but we are of an opinion that it is important to take

this into consideration while using the given asset allocation methods and alternate these according to their expected performance in the given investment period.

In order to resolve this problem we test simple methods such as autoregressive model and simple moving average, where we predict returns one month in advance. Following the analysis we come to the conclusion that this approach does not bring any improvement compared to the static methods. Even a method, which invests in a strategy with the highest return in the previous month, delivers a negative return after observing this dynamic strategy for 24 months. We then experiment with a Feedforward Neural Network, which also does not bring very interesting results. As an additional and even stronger tool we select the Long Short-Term Memory Neural Network, which has been successfully used in a broad range of problems. We explain all the pitfalls considered for the construction of Neural Networks and then build hyperparameter optimisation framework, which serves for optimization of these parameters, although a very time consuming optimization. We also suggest a more optimal custom loss function, but our research in this area has not been completed yet. As an additional improvement we propose using longer data series, generating more datasets, new custom loss function so that it is more appropriate to the given problem and adding some macroeconomic factors to the predictors base. We think that it is mainly the macroeconomic factors together with the new custom loss function, which will be key for the success of the model. Considering the costly grid search, we do not provide a fine tuned model, but only a provisional product, which still needs to be enhanced by additional thorough hyperoptimalization. For the time being our optimized model cannot beat the static methods in the area of performance measures, but we assume it is just a question of time until the model is perfected. We have learnt that implementing a Neural Network on an efficient and liquid market is not a trivial problem and requires a tremendous amount of time and fine tuning.

References

[1] Bjerring, T.T., Ross, O, Weissensteiner, A.: Feature Selection for Portfolio Optimization, *Annals of Operations Research*, 2017, Vol. 256, No. 1, pp. 21–40

[2] Branger, N., Lučivjanská, K., Weissensteiner, A.: Optimal Granularity for Portfolio Choice, *The Review of Financial Studies*, 2016, Vol. 21, No. 4, pp. 1455–1508

[3] Brown, S. J.: Optimal Portfolio Choice under Uncertainty, *Ph.D. Diss.*, 1976, University of Chicago

[4] Brownlee, J.: Deep Learning with Python, *Machine Learning Mastery*, 2016

[5] Brownlee, J.: Long Short-Term Memory Networks With Python, *Machine Learning Mastery*, 2017

[6] Campbell, J. Y., Lo, A. W., MacKinley A. G.: The Econometrics of Financial Markets, *Princeton University Press*, New Jersey, 1997

[7] Chopra, V.K, Ziemba, W.T.: The Effect of Errors in Means, Variances and Covariances on Optimal Portfolio Choice, *The Journal of Portfolio Management*, 1993, Vol. 19, No. 2, pp. 6–11

[8] DeMiguel, V., Garlappi, L., Uppal, R.: Optimal Versus Naive Diversification: How Inefficient is the 1/N Portfolio Strategy?, *The Review of Financial Studies*, 2009, Vol. 22, No. 5, pp. 1915–1953

[9] DeMiguel, V., Garlappi, L., Nogales, F., Uppal, R.: A Generalized Approach to portfolio Optimization: Improving Performance by Constraining Portfolio Norms, *Management Science*, 2009, Vol. 76, No. 374, pp. 388–394

[10] DeMiguel, V., Garlappi, L., Nogales, F., Uppal, R.: Electronic Companion - A Generalized Approach to portfolio Optimization: Improving Performance by Constraining Portfolio Norms, available on the internet (07.04.2019): `http://faculty.london.edu/avmiguel/DeMiguelGarlappiNogalesUppalMSAppendix.pdf`

[11] Fabrizio, L., Gallegati M., Mantegna, R. N., Tola, V.: Cluster analysis for portfolio optimization, *Journal of Economic Dynamics & Control*, 2008, Vol. 32, No. 1, pp. 235–258

[12] Goyal, A., Welch, I.: A Comprehensive Look at The Empirical Perfomance of Equity Premium Prediction, *The Review of Financial Studies*, 2008, Vol. 21, No. 4, pp. 1455–1508

[13] Hartigan, J.A: Consistency of Single Linkage for High-Density Clusters, *Journal of the American statistical Association*, 1981, Vol. 21, No. 4, pp. 1455–1508

[14] Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory, *Neural Computation*, 1997, Vol. 9, No. 8, pp. 1735-1780

[15] Jagannathan, R., Ma, T.: Risk Reduction in Large Portfolios: Why Imposing the Wrong Constraints Helps, *Journal of Finance*, 2003, Vol. 58, pp. 1651–1684

[16] Kan, R., Zhou, G.: Optimal Portfolio Choice with Parameter Uncertainty, *Journal of Financial and Quantitative Analysis*, 2007, Vol. 42, No. 3, pp. 621–656

[17] Klein, R. W., Bawa V. S.: The Effect of Estimation Risk on Optimal Portfolio Choice, *Journal of Financial Economics*, 1976, Vol. 3, pp. 215–231

[18] Kubokawa, T. Robert C. P., Saleh, A. K. Md. E: Estimation of Noncentrality Parameters, *Canadian Journal of Statistics*, 1993, Vol. 21, pp. 45–57

[19] Ledoit, O., Wolf, M.: Improved Estimation of the Covariance Matrix of Stock Returns with an Application to Portfolio Selection, *Journal of Empirical Finance*, 2003, Vol. 10, No. 5, pp. 603–621

[20] Ledoit, O., Wolf, M.: Honey, I Shrunk the Sample Covariance Matrix, *UPF Economics and Business Working Paper*, 2003, Vol. 691

[21] Markowitz, H.: Portfolio Selection, *The Journal of Finance*, 1952, Vol. 7, No. 1, pp. 77–91

[22] Muirhead, R. J.: Aspects of Multivariate Statistical Theory, *Wiley*, New York, 2008

[23] Risk-Free Data, Ibbotson and Associates, Inc., available on the internet (18.04.2019): `http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`

[24] Rukhin, A. L.: Estimation of the Noncentrality parameter of an F-Distribution, *Journal of Statistical Planning and Inference*, 1993, Vol. 35, pp. 201–211

[25] Pitis, S.: Written Memories: Understanding, Deriving and Extending the LSTM, *R2RT*, 2016, available on the internet (11.05.2019): `https://r2rt.com/written-memories-understanding-deriving-and-extending-the-lstm.html`

[26] Schäfer, J., Strimmer, K.: A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics, *Statistical Applications in Genetics and Molecular Biology*, 2005, Vol. 4, No. 1, Article 32

[27] Stambaugh, R. F.: Analyzing Investments Whose Histories Differ in Length, *Journal of Financial Economics*, 1997, , Vol. 45, pp. 285–331

[28] Zellner, A., Chetty, K. V.: Prediction and Decision Problems in Regression Models from the Bayesian Point of View, *Journal of the American Statistical Association*, 1965, Vol. 60, No. 311, pp. 608–616

Appendix A

At the last page of this Thesis we attach a CD where a source code, used in some parts of our work, is available. These files are not meant to be run, use them only to gain an intuition about the implemented procedure. For more information do not hesitate to contact me at *andrej.remak@gmail.com*.

1. **Weights.R;** source code of the asset allocation methods from Chapter [2].
2. **Datasets_Calculations.R;** computing the weights function in parallel and saving them into datasets.
3. **Results.R;** a file which interprets the computed results from the Static and Dynamic Approach.
4. **Results_Helpfun.R;** help functions for the results interpretation, plotting histograms and calculating performance measures.
5. **Hyperparameter_Optimization.py;** hyperparameter optimization framework used in Section 4.2.
6. **Neural_Networks.py;** file used to run the prediction of the last 24 months, hyper parameters are set in the beginning of the file.
7. **NN_Models.py;** definition of LSTM Neural Networks and Feedforward Neural Networks which can be used in the hyperparameter optimization process.
8. **NN_Helpfun.py** help functions that are needed for the hyperparameter optimization process, such as; setting the optimizer, reshaping data, converting time series to supervised learning, merging and sorting data to according to months, spliting data into training and validation set, scaling the data, import data from *.mat* files generated by R, plotting functions, NumpyEncoder
9. **Table_Hyperparameter_Optimization.JPG** an example of the hyperparameter optimization output.