CONVEX OPTIMIZATION IN QUANTITATIVE FINANCE

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Kasper Johansson
June 2025

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Stephen Boyd, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Sanjay Lall

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Markus Pelger

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format.

Preface

Over the past 70 years, quantitative investing has evolved from a niche discipline guided by heuristics and rules of thumb into a mainstream approach to portfolio management. This transformation has been propelled in large part by advances in mathematical optimization, especially convex optimization, and remarkable growth in computational power.

Convex optimization problems, which exhibit certain structural properties, can be solved reliably and efficiently, even when involving large-scale, nonlinear, or nondifferentiable functions. These methods are particularly well-suited for quantitative finance, where models often balance complexity, interpretability, and scalability.

This thesis explores the application of convex optimization to three central challenges in quantitative investing: predicting the covariance matrix of asset returns, generating and combining asset signals, and constructing portfolios. Building on foundational work by Markowitz (1952) and Engle (2002), we develop methods that are effective, interpretable, and computationally efficient. Together, these contributions address the core steps of the investment process: risk estimation, return prediction, and portfolio optimization.

Acknowledgments

I'll be honest, for me this is the most important part of the thesis. The math was tricky, the writing was slow, but the people made it all worthwhile. Before any equations or figures, I owe a few heartfelt thank-yous. The past few years have been an incredible ride, and I've been lucky enough to learn from some of the most inspiring, brilliant, and occasionally eccentric people I could imagine. There are far too many people I owe thanks to, but I'll try my best, and promise to only get overly sentimental once (maybe twice).

First and foremost, I want to thank my advisor, Professor Stephen Boyd. I think if I tried to describe how much I have learned from you, and what an inspiring space your lab has been, I'd need another 189 pages. So I'll just say this: It's been a true privilege to learn from you, watch you work, and occasionally pretend I understood everything you said on the first try.

To the rest of my defense committee, Mert Pilanci, Markus Pelger, Sanjay Lall, and Ron Kahn, thank you for reading my work, for asking hard questions, and for not asking too many. Your feedback helped shape this thesis more than you know.

This work wouldn't exist without the collaboration of some amazing minds (and even better humans). Thank you to Giray Ogut, Philippe Schiele, Thomas Schmelzer, Ron Kahn, and Markus Pelger for pushing my thinking, sharing your time, and making the work more enjoyable than I ever expected.

I want to thank cvxgrp for making the past few years enjoyable and an incredible learning experience. A special thank you to Daniel, Max, and Giray for the countless hours you have spent with me in and out of 243. I definitely would have gone insane without you to keep me company in our windowless office...

To all my friends outside of the lab who have shaped me into the person I am today, I want to thank you for making it easy to stay in touch, no matter where I am, and for always welcoming me back home with open arms. Thanks for reminding me there's life beyond convex optimization.

And Maya P, thank you for being the best distraction I could've ever asked for. Thank you for getting me out of bed every morning and pushing me through those 6am workouts, even though you're a much worse loser than I am (and I'm pretty bad). In all seriousness, thank you for making my life about a thousand times better.

I owe most of who I am to my mom and dad. Dad, thank you for teaching me to be curious, to never stop learning, and to always take on the toughest challenges. Mom, thank you for caring for me, dragging me to practices and games for some 10+ years, and for traveling across the world when times were tough for me. I hope that one day I can pay it forward.

I also want to thank two very, very special people: my grandparents. Thank you for picking up the phone every Sunday, even when I keep pushing our calls past bedtime. Most importantly, thank you for being nothing but supportive, encouraging, and unbelievably positive, regardless of what I'm pursuing, whether it's spending my days in a lab on the other side of the world, or biking in the middle of nowhere. I'm so lucky to have two people like you in my life, and I really hope you understand how much your encouragement has meant over the years.

Finally, to my little brother and lifelong best friend, thank you. I couldn't have asked for a better person to share a childhood, a family, and uncountably many memories with. I can't wait to see where our journeys are headed!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis covers five applications of convex optimization in quantitative investing, addressing core challenges in risk estimation, signal generation, and portfolio construction. Each chapter is based on a published or forthcoming paper and represents a collaborative effort to develop practical, interpretable, and computationally efficient methods for real-world financial decision-making.

Chapter 2 is based on the paper [163], co-authored with Giray Ogut, Thomas Schmelzer, Markus Pelger, and Stephen Boyd. In this chapter we consider the well-studied problem of predicting the time-varying covariance matrix of a vector of financial returns. Popular methods range from simple predictors like rolling window or exponentially weighted moving average (EWMA) to more sophisticated predictors such as generalized autoregressive conditional heteroscedastic (GARCH) type methods. Building on a specific covariance estimator suggested by Engle in 2002, we propose a relatively simple extension that requires little or no tuning or fitting, is interpretable, and produces results at least as good as MGARCH, a popular extension of GARCH that handles multiple assets. To evaluate predictors we introduce a novel approach, evaluating the regret of the log-likelihood over a time period such as a quarter. This metric allows us to see not only how well a covariance predictor does over all, but also how quickly it reacts to changes in market conditions. Our simple predictor slightly outperforms MGARCH in terms of regret. We also test covariance predictors on downstream applications such as portfolio optimization methods that depend on the covariance matrix. For these applications our simple covariance predictor and MGARCH perform similarly.

Chapter 3 is based on the paper [45], co-authored with Stephen Boyd, Ronald Kahn, Philipp Schiele, and Thomas Schmelzer. In this chapter we study the Markowitz portfolio construction problem. More than seventy years ago Harry Markowitz formulated portfolio construction as an optimization problem that trades off expected return and risk, defined as the standard deviation of the portfolio returns. Since then the method has been extended to include many practical constraints and objective terms, such as transaction cost or leverage limits. Despite several criticisms of Markowitz's method, for example its sensitivity to poor forecasts of the return statistics, it has become the dominant quantitative method for portfolio construction in practice. In this article we describe an extension of Markowitz's method that addresses many practical effects and gracefully handles the uncertainty inherent in return statistics forecasting. Like Markowitz's original formulation, the extension is also a convex optimization problem, which can be solved with high reliability and speed.

Chapter 4 is based on the paper [165], co-authored with Thomas Schmelzer and Stephen Boyd. This chapter proposes a new method for finding statistical arbitrages (stat-arbs) that can contain more assets than just the traditional pair. We formulate the problem as seeking a portfolio with the highest volatility, subject to its price remaining in a band and a leverage limit. This optimization problem is not convex, but can be approximately solved using the convex-concave procedure, a specific sequential convex programming method. We show how the method generalizes to finding moving-band stat-arbs, where the price band midpoint varies over time.

Chapter 5 is based on the paper [166], co-authored with Thomas Schmelzer and Stephen Boyd. This chapter develops a method for managing a dynamic basket of moving-band stat-arbs, leveraging the convex-concave optimization method introduced in the previous chapter. We consider illustrate the method on recent historical data, showing that it can perform very well in terms of risk-adjusted return, essentially uncorrelated with the market.

Lastly, chapter 6 is based on the paper [167], co-authored with Stephen Boyd. In this chapter we consider the problem of constructing a portfolio that combines traditional financial assets with crypto assets. We show that despite the documented attributes of crypto assets, such as high volatility, heavy tails, excess kurtosis and skewness, a simple extension of traditional risk allocation provides robust solutions for integrating these emerging assets into broader investment strategies. Examination of the risk allocation holdings suggests an even simpler method, analogous to the traditional 60/40 stocks/bonds allocation, involving a fixed allocation to crypto and traditional assets, dynamically diluted with cash to achieve a target risk level.

Chapter 2

A simple covariance predictor for financial returns

2.1 Introduction

2.1.1 Covariance prediction

We consider cross-sections, *e.g.*, a vector time series of n financial returns, denoted $r_t \in \mathbf{R}^n$, $t = 1, 2, \ldots$, where $(r_t)_i$ is the return of asset i from $t-1$ to t. We focus on the case where the mean $\mathbf{E}\, r_t$ is small enough that the second moment $\mathbf{E}\, r_t r_t^T \in \mathbf{R}^{n \times n}$ is a good approximation of the covariance matrix $\mathbf{cov}(r_t) = \mathbf{E}\, r_t r_t^T - (\mathbf{E}\, r_t)(\mathbf{E}\, r_t)^T$, where $\mathbf{E}$ denotes expectation. This is the case for most daily, weekly, or monthly stock, bond, and futures returns, factor returns, and index returns. We start by focussing on the case where the number of assets n is modest, say, on the order 10–100 or so; in section 2.8 we explain how to extend the method to much larger universes using ideas such as factor models.

We model the returns r_t as independent random variables with zero mean and covariance $\Sigma_t \in \mathbf{S}^n_{++}$ (the set of symmetric positive definite matrices). We focus on the problem of predicting or estimating Σ_t, based on knowledge of $r_1, \ldots, r_{t-1}$. The prediction is denoted as $\hat{\Sigma}_t \in \mathbf{S}^n_{++}$. The predicted volatilities of assets are given by

$$\hat{\sigma}_t = \mathbf{diag}(\hat{\Sigma}_t)^{1/2} \in \mathbf{R}^n,$$

where **diag** with a matrix argument is the vector of diagonal entries of the matrix, and the squareroot of a vector above is elementwise. We denote the predicted correlations as

$$\hat{R}_t = \mathbf{diag}(\hat{\sigma}_t)^{-1} \hat{\Sigma}_t\, \mathbf{diag}(\hat{\sigma}_t)^{-1},$$

where **diag** with a vector argument is the diagonal matrix with entries from the vector argument.

Covariance estimation comes up in several areas of finance, including Markowitz portfolio construction [208, 129], risk management [224], and asset pricing [280]. Much attention has been devoted to this problem, and a Nobel Memorial Prize in Economic Sciences was awarded for work directly related to volatility estimation [90].

While it is well known that the tails of financial returns are poorly modeled by a Gaussian distribution, our focus here is on the bulk of the distribution, where the Gaussian assumption is reasonable. For future use, we note that the log-likelihood of an observed return r_t, under the Gaussian distribution $r_t \sim \mathcal{N}(0, \hat{\Sigma}_t)$, is

$$l_t(\hat{\Sigma}_t) = \frac{1}{2}\left(-n\log(2\pi) - \log\det\hat{\Sigma}_t - r_t^T\hat{\Sigma}_t^{-1}r_t\right). \tag{2.1}$$

The Gaussian log-likelihood is closely related to a popular metric for evaluating covariance predictors in econometrics, called the (Gaussian) quasi-likelihood (QLIKE) [249, 250, 183]. QLIKE is the negative log-likelihood, under the Gaussian assumption, up to an additive constant and a positive scale factor. Roughly speaking, we seek covariance predictors that achieve large values of log-likelihood, or small values of QLIKE, on realized returns. We will describe evaluation of covariance predictors in detail in section 2.4.

2.1.2 Contributions

This monograph makes three contributions. First, we propose a new method for predicting the time-varying covariance matrix of a vector of financial returns, building on a specific covariance estimator suggested by Engle in 2002. Our method is a relatively simple extension that requires very little tuning and is readily interpretable. It relies on solving a small convex optimization problem, which can be carried out very quickly and reliably [48]. Our method performs as well as much more complex methods, as measured by several metrics.

Our second contribution is to propose a new method for evaluating a covariance predictor, by considering the regret of the log-likelihood over some time period such as a quarter. This approach allows us to evaluate how quickly a covariance estimator reacts to changes in market conditions.

Our third contribution is an extensive empirical study of covariance predictors. We compare our new method to other popular predictors, including rolling window, exponentially weighted moving average (EWMA), and generalized autoregressive conditional heteroscedastic (GARCH) type methods. We find that our method performs slightly better than other predictors. However, even the simplest predictors perform well for practical problems like portfolio optimization.

Everything needed to reproduce our results, together with an open source implementation of our proposed covariance predictor, is available online at

`https://github.com/cvxgrp/cov_pred_finance`.

2.1.3 Outline

In section 2.2 we describe some common predictors, including the one that our method builds on. We introduce our proposed covariance predictor in section 2.3. In section 2.4 we discuss methods for validating covariance predictors that measure both overall performance and reactivity to market changes. We describe the data we use in our first empirical studies in section 2.5, and give the results in section 2.6.

In the next sections we discuss some extensions of and variations on our method, including realized covariance prediction (section 2.7), handling large universes via factor models (section 2.8), obtaining smooth covariance estimates (section 2.9), and using our covariance model to generate simulated returns (section 2.10).

2.2 Some common covariance predictors

In this section we review some common covariance predictors, ranging from simple to complex, with the goal of giving context and fixing our notation. To simplify some formulas, we take $r_\tau = 0$ for $\tau \leq 0$.

2.2.1 Rolling window

The rolling window predictor with window length or memory M is the average of the last $M \geq n$ outer products,

$$\hat{\Sigma}_t = \alpha_t \sum_{\tau=t-M}^{t-1} r_\tau r_\tau^T, \quad t = 2, 3, \ldots,$$

where $\alpha_t = 1/\min\{t-1, M\}$ is the normalization constant. The rolling window predictor can be evaluated via the recursion

$$\hat{\Sigma}_{t+1} = \frac{\alpha_{t+1}}{\alpha_t}\hat{\Sigma}_t + \alpha_{t+1}(r_t r_t^T - r_{t-M} r_{t-M}^T), \quad t = 1, 2, \ldots,$$

with initialization $\hat{\Sigma}_1 = 0$.

For $t < n$, the rolling window covariance estimate is not full rank. To handle this, as well as to improve the quality of the prediction, we can add regularization or shrinkage, for example by adding a positive multiple of $\mathbf{diag}(\hat{\Sigma}_t)$ to our estimate [185, 184], or approximating the predicted covariance matrix by a diagonal plus low rank matrix, as described in section 2.8.

2.2.2 EWMA

The exponentially weighted moving average (EWMA) estimator, with forgetting factor $\beta \in (0,1)$, is

$$\hat{\Sigma}_t = \alpha_t \sum_{\tau=1}^{t-1} \beta^{t-1-\tau} r_\tau r_\tau^T, \quad t = 2, 3, \ldots, \tag{2.2}$$

where

$$\alpha_t = \left(\sum_{\tau=1}^{t-1} \beta^{t-1-\tau} \right)^{-1} = \frac{1-\beta}{1-\beta^{t-1}}$$

is the normalization constant. The forgetting factor β is usually expressed in terms of the half-life $H = -\log 2/\log \beta$, for which $\beta^H = 1/2$. The half-life H is the number of periods when the exponential weight has decreased by a factor of two. For example, for a half-life of one year, the current observed return has twice the impact on our covariance prediction as the return observed one year ago. The EWMA predictor is widely used in practice; for example RiskMetrics suggests the forgetting factor $\beta = 0.94$, which corresponds to a half-life of around 11 days [225, 201].

The EWMA covariance predictor can be computed recursively as

$$\hat{\Sigma}_{t+1} = \frac{\beta - \beta^t}{1-\beta^t} \hat{\Sigma}_t + \frac{1-\beta}{1-\beta^t} r_t r_t^T, \quad t = 1, 2, \ldots,$$

with initialization $\hat{\Sigma}_1 = 0$. Like the rolling window predictor, the EWMA predictor is singular for $t < n$, which can be handled using the same regularization methods described above.

2.2.3 GARCH and MGARCH

GARCH. The generalized autoregressive conditional heteroscedastic (GARCH) predictor decomposes the return of a single asset as

$$r_t = \mu + \epsilon_t,$$

where μ is the mean return and ϵ_t is the innovation, and models the innovation as

$$\epsilon_t = \sigma_t z_t, \qquad \sigma_t^2 = \omega + \sum_{\tau=1}^{q} a_\tau \epsilon_{t-\tau}^2 + \sum_{\tau=1}^{p} b_\tau \sigma_{t-\tau}^2,$$

where σ_t is the asset volatility, z_t are independent $\mathcal{N}(0,1)$, and q and p (often both set to one in practice) determine the GARCH order [36]. (Recall that we assume zero mean.) The model parameters are ω, $a_1, \ldots, a_q$, and $b_1, \ldots, b_p$. Estimating the model parameters requires solving a nonconvex optimization problem [18].

With $p = 0$ we recover the autoregressive conditional heteroscedastic (ARCH) predictor, introduced in the seminal paper by [90]. This paper set the foundation for a wide variety of popular volatility and correlation predictors and earned him the 2003 Nobel Memorial Prize in Economic Sciences.

MGARCH. There are several ways of extending the GARCH predictor to a multivariate or vector setting. The most popular is the dynamic conditional correlation (DCC) predictor [92], which is a two-step approach described below.

Many other MGARCH predictors have been proposed. The most straightforward generalization from the univariate to multivariate predictors is the VEC predictor, where the covariance matrix is vectorized and each element is modeled as a GARCH process with dependencies on all other elements [38]. However, this extension requires estimating $n(n+1)(n(n+1)+1)/2 \approx n^4/2$ parameters, which can be impractical even for modest values of n.

Following the VEC extension of GARCH, multivariate GARCH (MGARCH) predictors have been proposed in two lines of development [285]. The first line involves models that impose restrictions on the parameters of the VEC predictor, including DVEC [36], BEKK [96], FF-MGARCH [304], O-GARCH [4], and GO-GARCH [299], to name some. However, these predictors have been shown to be hard to fit and can yield inconsistent estimates [57]. (These inconsistencies may not have much practical impact.) For detailed reviews of MGARCH predictors we refer the reader to [285, 20].

2.2.4 DCC GARCH

The second line of extensions of GARCH to vector time series models conditional covariances through separate estimates of conditional variances and correlations [92, 98]. In [37] Bollerslev introduced the constant conditional correlation predictor (CCC) where the individual asset volatilities are modeled as separate GARCH processes, while the correlation matrix is assumed constant and equal to the unconditional correlation matrix. This predictor was later extended to the dynamic conditional correlation (DCC) predictor where the correlation matrix is allowed to change over time [92]. The DCC model has the form

$$\Sigma_t = D_t R_t D_t,$$

where D_t is the diagonal matrix of standard deviations, *i.e.*, $(D_t)_{ii} = (\Sigma_t)_{ii}^{1/2}$, and R_t is the correlation matrix associated with Σ_t.

DCC GARCH models the diagonal elements of D_t as separate univariate GARCH processes as described above. The correlation matrix R_t is then modeled as a constrained multivariate GARCH (MGARCH) process, *e.g.*, as

$$\begin{aligned} R_t &= \mathbf{diag}(\mathbf{diag}(Q_t))^{-1/2} Q_t \, \mathbf{diag}(\mathbf{diag}(Q_t))^{-1/2}, \\ Q_t &= \bar{Q}(1-a-b) + a\tilde{r}_t\tilde{r}_t^T + bQ_{t-1}, \end{aligned}$$

where $\bar{Q}$ is the unconditional correlation matrix, a and b are the MGARCH parameters, and $\tilde{r}_t$ are the volatility adjusted returns defined as

$$\tilde{r}_t = D_t^{-1} r_t.$$

The parameters can be estimated in two steps via (quasi) maximum likelihood, but requires solving non-convex optimization problems [92]. This predictor has become a popular choice amongst MGARCH predictors due to its interpretability. Variants of the DCC predictor are widely used in finance, where it is also often used in combination with EWMA estimates. Conditional correlation predictors are easier to estimate than other multivariate GARCH predictors, and their parameters are more interpretable.

Iterated covariance estimation. DCC, which separately estimates the volatilities and correlations, is closely related to the idea of iterated covariance predictors [18]. Iterated covariance predictors estimate the covariance matrix in multiple iterations. In a two-step iteration we first form a first covariance estimate $\hat{\Sigma}_t^{(1)}$ of the returns r_t, at each time t, and form the whitened returns

$$\tilde{r}_t = \left(\hat{\Sigma}_t^{(1)}\right)^{-1/2} r_t.$$

In the second iteration we form the covariance estimate $\hat{\Sigma}_t^{(2)}$ of the whitened returns $\tilde{r}_t$. The final covariance estimate (of the returns r_t) is then formed as

$$\hat{\Sigma}_t = \left(\hat{\Sigma}_t^{(1)}\right)^{1/2} \hat{\Sigma}_t^{(2)} \left(\hat{\Sigma}_t^{(1)}\right)^{1/2}.$$

This procedure can be iterated further, and has been shown empirically to improve the quality of the covariance estimate; see [18] for details. In DCC, $\hat{\Sigma}^{(1)}$ is diagonal and models the volatilities; $\hat{\Sigma}^{(2)}$ is a correlation matrix.

2.2.5 Iterated EWMA

Iterated EWMA (IEWMA) was proposed by [92] and is analogous to DCC GARCH but with EWMA estimates of the volatilities and correlations instead of GARCH. Engle proposed IEWMA as an efficient alternative to the DCC GARCH predictor, although he did not refer to it as IEWMA; we use this term to emphasize its connection to iterated whitening, as proposed in [18]. Specifically, IEWMA can be viewed as an iterated whitener, where we first use a diagonal whitener (which estimates the volatilities) and then a full matrix whitener (which estimates the correlations). This is analogous to the two-step iterated covariance predictor where $\hat{\Sigma}_t^{(1)}$ is the diagonal matrix of squared volatility estimates and $\hat{\Sigma}_t^{(2)}$ estimates the correlation matrix of the volatility adjusted returns.

First we form an estimate of the volatilities $\hat{\sigma}_t = \mathbf{diag}(\hat{\Sigma}_t)^{1/2}$ using EWMA predictors for each asset. We denote the half-life of these volatility estimates as H^{vol}. We then form the marginally standardized returns as

$$\tilde{r}_t = \hat{D}_t^{-1} r_t, \tag{2.3}$$

where $\hat{D}_t = \mathbf{diag}(\hat{\sigma}_t)$. These vectors should have entries with standard deviation near one. It is common practice to winsorize the standardized returns; a good rule of thumb is to clip $\tilde{r}_t$ at ± 4.2, which corresponds to clipping r_t at $\pm 4.2\hat{\sigma}_t$.

Then we form a EWMA estimate of the covariance of $\tilde{r}_t$, which we denote as $\tilde{R}_t$, using half-life H^{cor} for this EWMA estimate. (We use the superscript 'cor' since the diagonal entries of $\tilde{R}_t$ should be near one, so $\tilde{R}_t$ is close to a correlation matrix.) From $\tilde{R}_t$ we form its associated correlation matrix $\hat{R}_t$, *i.e.*, we scale $\tilde{R}_t$ on the left and right by a diagonal matrix with entries $(\tilde{R}_t)_{ii}^{-1/2}$. Since the diagonal entries of $\tilde{R}_t$ should be near one, $\tilde{R}_t$ and $\hat{R}_t$ are not too different.

Our IEWMA covariance predictor is

$$\hat{\Sigma}_t = \hat{D}_t \hat{R}_t \hat{D}_t, \quad t = 2, 3, \ldots.$$

This is the covariance predictor proposed in [92]; replacing $\hat{R}_t$ with $\tilde{R}_t$ we obtain the iterated whitener proposed by Barratt and Boyd in [18]. As mentioned above, they are typically quite close.

It is common to choose the volatility half-life H^{vol} to be smaller than the correlation half-life H^{cor}. The intuition here is that we can average over fewer past samples when we predict the n volatilities $\hat{\sigma}_t$, but need more past samples to reliably estimate the $n(n-1)/2$ off-diagonal entries of $\hat{R}_t$. Empirical studies on real return data confirm that choosing a faster volatility half-life than correlation half-life yields better estimates.

2.3 Combined multiple iterated EWMAs

In this section we introduce a novel covariance predictor, which we call combined multiple iterated EWMAs, for which we use the acronym CM-IEWMA. The CM-IEWMA predictor is constructed from a modest number of IEWMA predictors, with different pairs of half-lives, which are combined using dynamically varying weights that are based on recent performance.

The CM-IEWMA predictor is motivated by the idea that different pairs of half-lives may work better for different market conditions. For example, short half-lives perform better in volatile markets, while long half-lives perform better for calm markets where conditions are changing slowly.

2.3.1 Dynamically weighted prediction combiner

We first describe the idea in a general setting. We start with K different covariance predictors, denoted $\hat{\Sigma}_t^{(k)}$, $k = 1, \ldots, K$. These could be any of the predictors described above, or predictors of the same type with different parameter values, *e.g.*, half-lives (for EWMA) or pairs of half-lives (for IEWMA). In some contexts these different predictors are referred to as a set of K experts [141, 168].

We denote the Cholesky factorizations of the associated precision matrices $(\hat{\Sigma}_t^{(k)})^{-1}$ as $\hat{L}_t^{(k)}$, *i.e.*,

$$\left(\hat{\Sigma}_t^{(k)}\right)^{-1} = \hat{L}_t^{(k)}(\hat{L}_t^{(k)})^T, \quad k = 1, \ldots, K,$$

where $\hat{L}_t^{(k)}$ are lower triangular with positive diagonal entries. We will combine these Cholesky factors with nonnegative weights $\pi_1, \ldots, \pi_K$ that sum to one, to obtain

$$\hat{L}_t = \sum_{k=1}^{K} \pi_k \hat{L}_t^{(k)}. \tag{2.4}$$

From this we recover the weighted combined predictor

$$\hat{\Sigma}_t = \left(\hat{L}_t \hat{L}_t^T\right)^{-1}. \tag{2.5}$$

We will see below why we combine the Cholesky factors of the precision matrices, and not the covariance or precision matrices themselves.

2.3.2 Choosing the weights via convex optimization

The log-likelihood (2.1) can be expressed in terms of the Cholesky factor of the precision matrix $\hat{L}_t$ as

$$l_t(\hat{\Sigma}_t) = -(n/2)\log(2\pi) + \sum_{i=1}^{n} \log \hat{L}_{t,ii} - (1/2)\|\hat{L}_t^T r_t\|_2^2,$$

where $\|\cdot\|_2$ denotes the Euclidean norm. This is a concave function of the weights $\pi \in \mathbf{R}_+^K$ [48].

We choose the weights at time t as the solution of the convex optimization problem

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{N} \left(\sum_{i=1}^{n} \log \hat{L}_{t-j,ii} - (1/2)\|\hat{L}_{t-j}^T r_{t-j}\|_2^2 \right) \\ \text{subject to} & \hat{L}_\tau = \sum_{j=1}^{K} \pi_j \hat{L}_\tau^{(j)}, \quad \tau = t-1, \ldots, t-N \\ & \pi \geq 0, \quad \mathbf{1}^T \pi = 1, \end{array} \tag{2.6}$$

with variables $\pi_1, \ldots, \pi_K$, where N is the look-back, $\mathbf{1}$ denotes the vector with entries one, and $\geq$ between vectors means entrywise. In words: we choose the (mixture) weights in each period so as to maximize the average log-likelihood of the combined prediction over the trailing N periods. The problem (2.6) is convex, and can be solved very quickly and reliably by many methods [48]. The covariance predictor is then recovered using (2.4) and (2.5).

The look-back N is a parameter that can be adjusted to give good performance. Numerical experiments suggest that the predictor is not very sensitive to the choice of N, and that a choice $N = 10$ seems to work well for asset universes up to a few hundred assets.

We mention several extensions of the weight problem (2.6). First, we can add one prediction which is diagonal, using any estimates of the volatilities (including constant). This gives us shrinkage, automatically chosen. We can also add a constraint or objective term that encourages the weights to vary smoothly over time, as discussed more in section 2.9.

The CM-IEWMA predictor is a special case of the dynamically weighted prediction combiner described above, where the K predictions are each IEWMA, with different pairs of half-lives H^{vol} and H^{cor}.

2.4 Evaluating covariance predictors

There are several ways of evaluating a covariance predictor, often divided into two categories, direct and indirect [250], [7, §7]. Direct methods use a proxy for the true covariance matrix to evaluate the predictor, while indirect methods use the covariance predictor on tasks of interest, such as portfolio construction or portfolio tracking.

Popular direct methods are the Mincer-Zarnowitz (MZ) regression and its variants, based on statistical tests of the regression coefficients of a predicted variable on an observed variable (or in the case of variance and covariance, a proxy for the observed variable) [230, 292]. Direct methods also include the comparison between different predictors in terms of some loss function. Common loss functions are the mean squared error (MSE) and quasi-likelihood (QLIKE) [249, 250]. To select good models, the model confidence set (MCS) is usually used [139], or the Ledoit–Wolf test [187] to compare Sharpe ratios.

Indirect methods use applications to rank covariance predictors, and include the minimum variance and mean-variance portfolios, as well as portfolio tracking tasks.

The difference in performance between various predictors can also be evaluated using statistical tests. For a more detailed discussion of both direct and indirect methods, we refer the reader to [250].

In this section we discuss several evaluation metrics for covariance predictors. The first three metrics are direct, and include the mean squared error and two metrics based on a statistical measure, the log-likelihood under a Gaussian distribution. The remaining metrics judge a covariance predictor by the performance of a portfolio using a method that depends on a covariance matrix. We are mainly interested in illustrating how simple methods can perform just as well as or better than more complex ones, rather than finding optimal predictors in a statistical sense. Therefore we look at the absolute performance of covariance predictors on these metrics.

2.4.1 Mean squared error

The mean squared error (MSE) is a common metric for evaluating a covariance predictor $\hat{\Sigma}_t$, defined as

$$\frac{1}{T}\sum_{t=1}^{T}\|r_t r_t^T - \hat{\Sigma}_t\|_F^2,$$

i.e., the average squared Frobenius norm of the difference between the realized (rank one) covariance matrix $r_t r_t^T$ and the covariance predictor $\hat{\Sigma}_t$. Lower values of MSE are better. One variation on the MSE error assumes that $\hat{\Sigma}_t$ is constant over some number of time periods and replaces the rank one realized covariance $r_t r_t^T$ with an average of the rank one terms over the periods, *i.e.*, the realized empirical covariance.

2.4.2 Log-likelihood

A natural way of judging a covariance predictor is via its average log-likelihood on realized returns,

$$\frac{1}{2T}\sum_{t=1}^{T}\Big(-n\log(2\pi) - \log\det\hat{\Sigma}_t - r_t^T\hat{\Sigma}_t^{-1}r_t\Big),$$

with larger values being better. This metric can be used to compare different predictors.

To understand the performance of a covariance predictor over time and changing market conditions, we can examine the average log-likelihood over periods such as quarters, and look at the distribution of quarterly average log-likelihood values. We are particularly interested in poor, *i.e.*, low values.

2.4.3 Log-likelihood regret

Recall that the best constant predictor, in terms of the log-likelihood, is the empirical sample covariance

$$\Sigma^{\text{emp}} = \frac{1}{T}\sum_{t=1}^{T} r_t r_t^T,$$

with value

$$\frac{1}{2}\big(-n(\log(2\pi)+1) - \log\det\Sigma^{\text{emp}}\big).$$

For any other constant $\Sigma \in \mathbf{S}_{++}^n$, the log-likelihood is lower than the log-likelihood of Σ^{emp}. We define the *average log-likelihood regret* as the average log-likelihood of the (constant) empirical covariance, minus the average log-likelihood of the covariance predictor. The regret is a measure of how much the covariance predictor $\hat{\Sigma}_t$, $t = 1, \ldots, T$, underperforms the best possible constant covariance predictor (*i.e.*, the sample covariance matrix). The term regret comes from the field of online optimization; see, *e.g.*, [321, 233, 144, 143].

We want our covariance predictor to have small regret. The regret is typically positive, but it can be negative, *i.e.*, our time-varying covariance can have higher log-likelihood than the best constant one. The regret is not any more useful than the log-likelihood when comparing predictors over one time interval, since it simply adds a constant and switches the sign. But it is interesting when we compute the regret over multiple periods, like months or quarters. The regret over multiple quarters removes the effect of the log-likelihood of the empirical covariance varying due to changing market conditions, and allows us to assess how well the covariance predictor adapts.

2.4.4 Portfolio performance

We can also judge the performance of a covariance predictor by the investment performance of portfolio construction methods that depend on the estimated covariance matrix. As with log-likelihood or log-likelihood regret, we can examine the portfolio performance in periods such as quarters, to see how evenly the performance is spread over time.

One obvious metric of interest is how close the ex-ante and realized portfolio volatilities are. The metrics described above, MSE, log-likelihood, and log-likelihood regret, are agnostic to the portfolio; with specific real portfolios we can see how well our covariance predictors predict portfolio volatility.

We will assess a covariance predictor using five simple portfolio construction methods. The first is an equally weighted (or $1/n$) portfolio, which does not by itself depend on the covariance, but does when we adjust it with cash to achieve a given ex-ante risk. The second, third, and fourth portfolios depend only on the covariance matrix. They are minimum variance, risk parity, and maximum diversification portfolios. For an in depth discussion of these portfolios, see [53]. The last portfolio we consider is a mean-variance portfolio, using a very simple mean estimator.

For each portfolio we look at four metrics: realized return, volatility, Sharpe ratio, and maximum drawdown. The returns, volatilities, and Sharpe ratios are reported in annualized values. The Sharpe ratio is defined as the ratio of the excess return (over the risk-free rate), divided by the volatility of the excess return,

$$\frac{\frac{1}{T}\sum_{\tau=1}^{T}(r_t^p - r_t^{\mathrm{rf}})}{\left(\frac{1}{T}\sum_{\tau=1}^{T}\left(r_t^p - \frac{1}{T}\sum_{\tau=1}^{T} r_t^p\right)^2\right)^{1/2}},$$

where r_t^p and r_t^{rf} are the portfolio and risk-free returns at time t. The maximum drawdown is defined as

$$\max_{1\leq t_1<t_2\leq T}\left(1-\frac{V_{t_2}^p}{V_{t_1}^p}\right),$$

where

$$V_t^p = V_0(1+r_1^p)(1+r_2^p)\cdots(1+r_t^p)$$

is the portfolio value at time t (with returns re-invested), starting with value $V_0 > 0$.

In addition to portfolio performance, we can also examine how well the covariance prediction predicts the portfolio volatility. We compare the realized or ex-post portfolio volatility

$$\left(\frac{1}{T}\sum_{t=1}^{T}(r_t^T w_t)^2\right)^{1/2},$$

to the predicted or ex-ante portfolio volatility

$$\left(\frac{1}{T}\sum_{t=1}^{T} w_t^T \hat{\Sigma}_t w_t\right)^{1/2},$$

where $w_t \in \mathbf{R}^n$ are the portfolio weights. This directly measures the ability of the estimated covariance matrix to predict portfolio risk.

Equal weight portfolio. We take the equal weight or $1/n$ portfolio with $w = (1/n)\mathbf{1}$. This portfolio does not depend on the covariance $\hat{\Sigma}_t$, but when we mix it with cash, as described below, it will.

Minimum variance portfolio. The (constrained) minimum variance portfolio is the solution of the convex optimization problem

$$\begin{array}{ll} \text{minimize} & w^T \hat{\Sigma}_t w \\ \text{subject to} & w^T \mathbf{1} = 1, \quad \|w\|_1 \leq L_{\max}, \quad w_{\min} \leq w \leq w_{\max} \end{array}$$

with variable $w \in \mathbf{R}^n$, where $L_{\max} \geq 1$ is a leverage limit, and $w_{\min}$ and $w_{\max}$ are lower and upper bounds on the weights, respectively.

Risk-parity portfolio. The portfolio return volatility $\sigma(w) = (w^T \hat{\Sigma}_t w)^{1/2}$ can be broken down into a sum of volatilities (risks) associated with each asset as

$$\frac{\partial \log \sigma(w)}{\partial w_i} = \frac{\partial \sigma(w)}{\sigma(w)} \frac{w_i}{\partial w_i} = \frac{w_i (\hat{\Sigma}_t w)_i}{w^T \hat{\Sigma}_t w}, \quad i = 1, \ldots, n.$$

The risk parity portfolio is the one for which these volatility attributions are equal [263]. This portfolio can be found by solving the convex optimization problem [52],

$$\text{minimize} \quad (1/2) x^T \hat{\Sigma}_t x - \sum_{i=1}^n (1/n) \log x_i,$$

with variable x, and then taking $w = x^\star / (\mathbf{1}^T x^\star)$.

Maximum diversification portfolio. The diversification ratio of a long-only portfolio (*i.e.*, one with $w \geq 0$) is defined as

$$D(w) = \frac{\hat{\sigma}_t^T w}{(w^T \hat{\Sigma}_t w)^{1/2}}.$$

The diversification ratio tells us how much higher the portfolio volatility would be if all assets were perfectly correlated. The maximum diversification portfolio is the portfolio w that maximizes $D(w)$, possibly subject to constraints [66]. Like the risk-parity portfolio, the maximum diversification portfolio can be found via convex optimization. We let $x^\star$ denote the solution of the convex optimization problem [52]

$$\begin{array}{ll} \text{minimize} & x^T \hat{\Sigma}_t x \\ \text{subject to} & \hat{\sigma}_t^T x = 1, \quad x \geq 0, \end{array}$$

with variable x. The maximum diversification portfolio is $w = x^\star / \mathbf{1}^T x^\star$.

Volatility control with cash. We mix each of the four portfolios described above with cash to achieve a target value of ex-ante volatility σ^{tar}. To do this we start with the portfolio weight vector w_t, and compute its ex-ante volatility $\sigma_t = (w_t^T \hat{\Sigma}_t w_t)^{1/2}$. Then we add a cash component so that the overall ex-ante volatility equals our target, *i.e.*, we use the $(n+1)$ weights (with the last component denoting cash)

$$\left[\begin{array}{c} \theta w_t \\ (1-\theta) \end{array} \right], \qquad \theta = \frac{\sigma^{\text{tar}}}{\sigma_t}.$$

This portfolio will have ex-ante volatility σ^{tar}. Note that the cash weight can be either positive (when it dilutes the portfolio volatility) or negative (when it leverages the portfolio volatility to the desired level). The target volatility σ^{tar} should be chosen so as to avoid portfolios that are either too diluted or too leveraged.

Mean variance portfolio. The last portfolio we consider is a basic mean-variance portfolio, defined as the solution of the convex optimization problem

$$\begin{array}{ll} \text{maximize} & \hat{r}_t^T w \\ \text{subject to} & \|\hat{\Sigma}_t^{1/2} w\|_2 \leq \sigma^{\text{tar}} \\ & \mathbf{1}^T w + c = 1, \quad \|w\|_1 \leq L_{\max}, \\ & w_{\min} \leq w \leq w_{\max}, \quad c_{\min} \leq c \leq c_{\max} \end{array}$$

with variable w, where $\hat{r}_t$ is the predicted mean return vector at time t. The vector w gives the weights of the non-cash assets and c denotes the cash weight. The non-cash and cash weights are limited by $w_{\min}, w_{\max}$ and $c_{\min}, c_{\max}$, respectively. This portfolio does not need cash dilution, since it includes cash in its construction. (If σ^{tar} is chosen appropriately, it will have ex-ante risk σ^{tar}.) The mean-variance portfolio depends not only on a covariance estimate, but also a return estimate. For this we use one of the simplest possible return estimates, a EWMA of the realized returns.

2.5 Data sets and experimental setup

We illustrate our method on three different data sets: a set of 49 industry portfolios, a set of 25 stocks, and a set of 5 factor returns, each augmented with cash (with the historical risk-free interest rate). For each data set we show results for six covariance predictors. Everything needed to reproduce the results is available online at

Table 2.1: Industry portfolios.

Agriculture	Food products
Candy & soda	Beer & liquor
Tobacco products	Recreation
Entertainment	Printing and publishing
Consumer goods	Apparel
Healthcare	Medical equipment
Pharmaceutical products	Chemicals
Rubber and plastic products	Textiles
Construction materials	Construction
Steel works etc.	Fabricated products
Machinery	Electrical equipment
Automobiles and trucks	Aircraft
Shipbuilding, railroad equipment	Defense
Precious metals	Non-metallic and industrial metal mining
Coal	Petroleum and natural gas
Utilities	Communication
Personal services	Business services
Computers	Computer software
Electronic equipment	Measuring and control equipment
Business supplies	Shipping containers
Transportation	Wholesale
Retail	Restaurants, hotels, motels
Banking	Insurance
Real estate	Trading
Other	

https://github.com/cvxgrp/cov_pred_finance.

2.5.1 Data sets

Industry portfolios. The first data set consists of the daily returns of a universe of $n = 49$ daily traded industry portfolios, shown in table 2.1, along with cash. The data set spans July 1st 1969 to December 30th, 2022, for a total of 13496 (trading) days. The data was obtained from the Kenneth French Data Library [110].

Stocks. The second data set consists of the daily returns of $n = 25$ stocks and cash. The stocks were chosen to be the 25 largest stocks in the S&P 500 at the beginning of 2010, listed in table 2.2. This data set spans January 4th 2010 to December 30th, 2022, for a total of 3272 (trading) days. The stock data was attained through the Wharton Research Data Services (WRDS) portal [308].

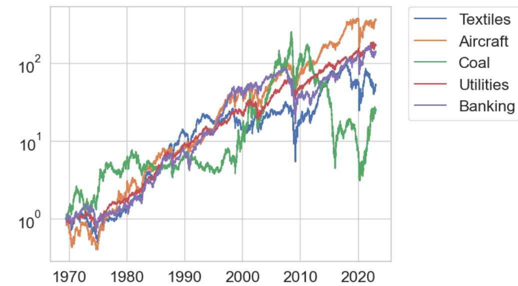
Factor returns. The third data set consists of daily returns of the five Fama-French factors taken from the Kenneth French Data Library [110], shown in table 2.3. The data set spans July 1st 1963 to December 30th, 2022, for a total of 14979 (trading) days.

Table 2.2: List of companies and their tickers.

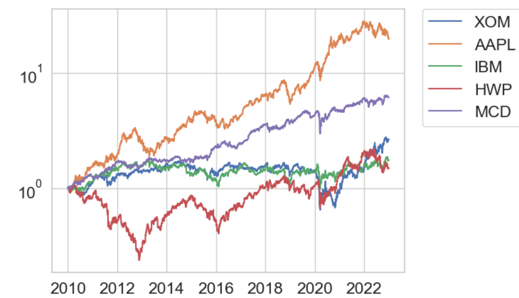
Ticker	Company Name
XOM	Exxon Mobil
WMT	Walmart
AAPL	Apple Inc.
PG	Procter & Gamble
JNJ	Johnson & Johnson
CHL	China Mobile
IBM	IBM
SBC	AT&T
GE	General Electric
CHV	Chevron
PFE	Pfizer
NOB	Noble
NCB	NCR
KO	Coca-Cola
ORCL	Oracle Corporation
HWP	Hewlett-Packard
INTC	Intel Corporation
MRK	Merck & Co.
PEP	PepsiCo
BEL	Becton, Dickinson and Company
ABT	Abbott Laboratories
SLB	Schlumberger
P	Pandora Media
PA	Pan American Silver
MCD	McDonald's

Table 2.3: The five Fama-French factors.

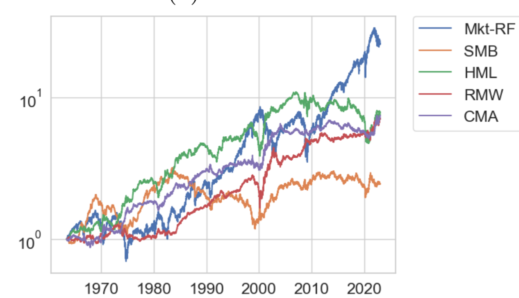
Factor	**Description**
MKT-Rf	market excess return over risk-free rate
SMB	small stocks minus big stocks
HML	high book-to-market stocks minus low book-to-market stocks
RMW	stocks with high operating profitability minus stocks with low operating profitability
CMA	stocks with conservative investment policies minus stocks with aggressive investment policies



(a) Industry data set.



(b) Stock data set.



(c) Factor data set.

Figure 2.1: Cumulative returns of five assets from each data set.

Cumulative returns. In figure 2.1 we show the cumulative returns of the five factors, and the cumulative returns of five assets chosen from each of the industry and stock data sets.

Table 2.4: Half-lives for CM-IEWMA predictors, given as $H^{\mathrm{vol}}/H^{\mathrm{cor}}$, in days.

Data set	**Half-lives**				
Industries	21/63	63/125	125/250	250/500	500/1000
Stocks	10/21	21/63	63/125	125/250	250/500
Factors	5/10	10/21	21/63	63/125	125/250

2.5.2 Six covariance predictors

For each data set we evaluate six covariance predictors, described below.

- Rolling window estimates with 500-, 250-, and, 125-day windows for the industry, stock, and factor data sets, respectively, denoted RW in plots and tables.
- EWMA predictors with 250-, 125-, and, 63-day half-lives, for the industry, stock, and factor data sets, respectively, denoted EWMA.
- IEWMA predictors with half-lives (in days) $H^{\mathrm{vol}}/H^{\mathrm{cor}}$ of 125/250, 63/125, and 21/63 for the three data sets, respectively, denoted IEWMA.
- DCC GARCH predictor, denoted MGARCH, with parameters re-estimated annually using the `rmgarch` package in R [118].
- CM-IEWMA predictor with $K = 5$ IEWMA predictors and a lookback of $N = 10$ days, with half-lives shown in table 2.4. For each of the fastest IEWMA predictors we regularize the covariance estimate by increasing the diagonal entries by 5%.
- Prescient predictor, *i.e.*, the empirical covariance for the quarter the day is in. This predictor maximizes log-likelihood for each quarter, and achieves zero regret. It is of course not implementable, and meant only to show a bound on performance with which to compare our implementable predictors.

All the parameters above (*e.g.*, half-lives) are chosen as reasonable values that give good overall performance for each predictor. The results are not sensitive to these choices.

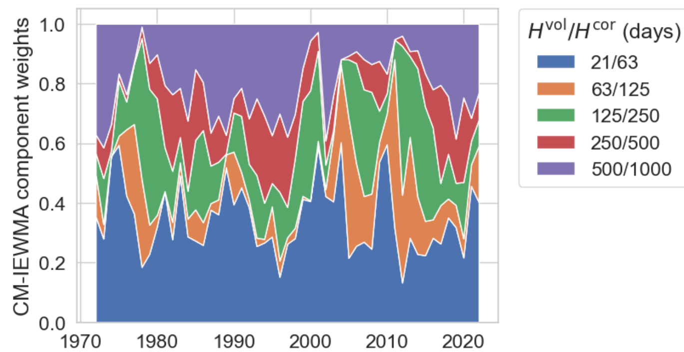
For our experiments we use the first two years (500 data points) of each data set to fit the MGARCH predictor and initialize the other predictors. (After this initial MGARCH fit, we re-estimate its parameters annually.) Hence, the evaluation period for our experiments below ranges from June 24th 1971 to December 30th, 2022, for the industry portfolios, from December 28th, 2011, to December 30th, 2022, for the stock portfolios, and from June 28th 1965 to December 30, 2022, for the factor portfolios.

2.6 Results

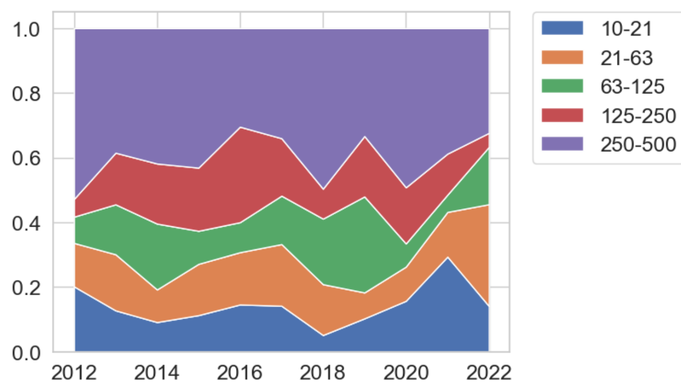
2.6.1 CM-IEWMA component weights

Figure 2.2 shows the weights for each of the five components of the CM-IEWMA predictors, averaged yearly, for the three data sets.

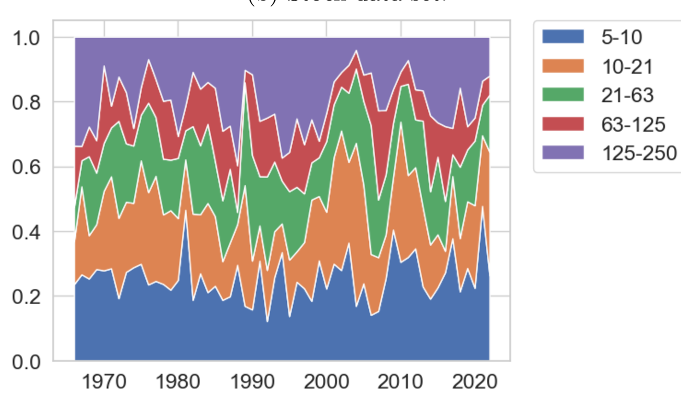
We can see how the predictor adapts the weights depending on market conditions. Substantial weight is put on the slower (longer half-life) IEWMAs most years. During and following volatile periods like the 2000 dot.com bubble or 2008 market crash, we see a big increase in weight on the faster IEWMAs. We can illustrate these changes in weights in response to market conditions via the effective half-life of the CM-IEWMA, defined as the weighted average of the five (longer) half-lives, shown in figure 2.3, averaged yearly.



(a) Industry data set.

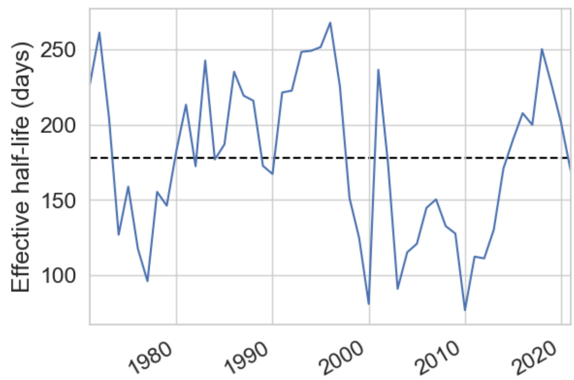


(b) Stock data set.

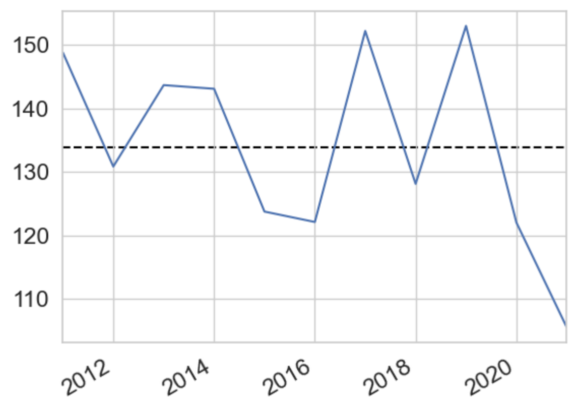


(c) Factor data set.

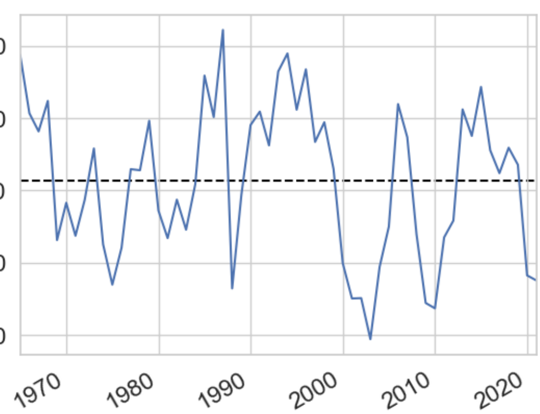
Figure 2.2: Weights of the various IEWMA components in the CM-IEWMA predictors on three data sets. The IEWMA components are represented as $H^{\mathrm{vol}}/H^{\mathrm{cor}}$ for the volatility and correlation half-lives, respectively.



(a) Industry data set.



(b) Stock data set.



(c) Factor data set.

Figure 2.3: Effective half-lives of the CM-IEWMA predictor on three data sets.

2.6.2 Mean squared error

Table 2.5 shows the average, standard deviation, and maximum of the MSE computed over distinct quarters for the six covariance predictors on the three data sets (with lower being better for all three metrics).

CM-IEWMA and MGARCH do better than the other predictors on all metrics over all data sets, with MGARCH doing slightly better on the industry data and CM-IEWMA slightly better on the stock data. Interestingly, on the factor data set, the CM-IEWMA predictor does better than the prescient predictor.

2.6.3 Log-likelihood and log-likelihood regret

Figure 2.4 shows the average quarterly log-likelihood for the different covariance predictors over the evaluation period. Not surprisingly, the prescient predictor does substantially better than the others. The different predictors follow similar trends, with even the prescient predictor experiencing a drop in log-likelihood during market turbulence. Close inspection shows that the CM-IEWMA and MGARCH predictors almost always have the highest log-likelihood in each quarter.

Figure 2.5 shows the average quarterly log-likelihood regret for the different covariance predictors over the evaluation period. Clearly, CM-IEWMA and MGARCH perform best in volatile markets. Figure 2.6 illustrates the difference between CM-IEWMA and MGARCH. As seen, CM-IEWMA consistently has lower regret on the industry and stock data sets, while they perform similar on the factor data. More precisely, CM-IEWMA has lower regret than MGARCH in 87% of the quarters for the industry data, 71% for the stock data, and 51% for the factor data.

Table 2.6 illustrates the differences in regret further, by showing the average, standard deviation, and the maximum of the average quarterly regret. As we can see, the average quarterly regret is lower for CM-IEWMA than for the other predictors. The regret is also more stable for CM-IEWMA, as the standard deviation is lower. Finally, the maximum average quarterly regret is also lower for CM-IEWMA than for the other predictors. These results are most prominent on the industry and stock data, while MGARCH does similar on the factor data.

Figure 2.7 gives a final illustration of these results, by showing the cumulative distribution functions of the average quarterly regret for the different covariance predictors.

Clearly, CM-IEWMA has the lowest regret on the industry and stock data set, and MGARCH does similar on the factor data.

Table 2.5: Metrics on the MSE, computed over distinct quarters, for six covariance predictors on three data sets.

Predictor	Average/10^{-4}	Std. Dev./10^{-3}	Max/10^{-2}
RW	7.6	4.0	3.9
EWMA	7.5	4.0	3.9
IEWMA	7.4	3.9	3.9
MGARCH	**6.8**	**3.6**	**3.8**
CM-IEWMA	6.9	**3.6**	**3.8**
Prescient	6.6	3.5	3.7

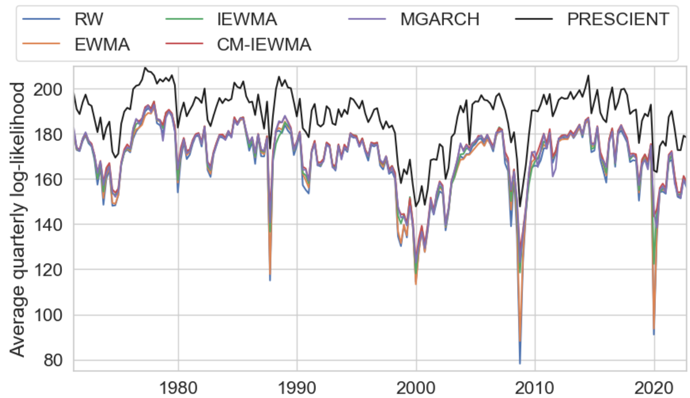
(a) Industry data set.

Predictor	Average/10^{-7}	Std. Dev./10^{-6}	Max/10^{-5}
RW	3.4	1.9	2.4
EWMA	3.4	1.9	2.4
IEWMA	3.3	1.8	2.4
MGARCH	**3.2**	**1.8**	2.4
CM-IEWMA	**3.2**	**1.8**	2.4
Prescient	3.1	1.8	2.3

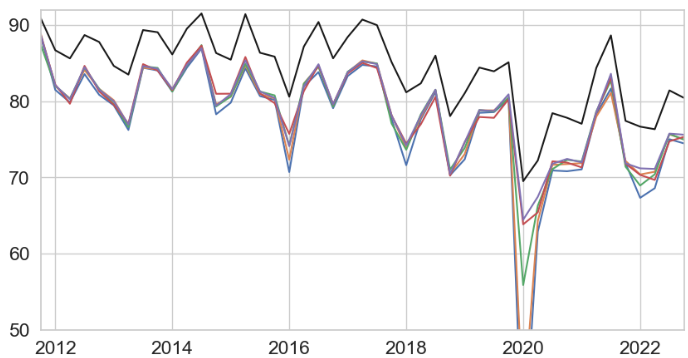
(b) Stock data set.

Predictor	Average/10^{-4}	Std. Dev./10^{-3}	Max/10^{-2}
RW	3.4	1.6	1.1
EWMA	3.3	1.6	1.1
IEWMA	3.2	1.6	1.1
MGARCH	3.0	**1.4**	1.0
CM-IEWMA	**2.9**	**1.4**	**0.9**
Prescient	3.0	1.5	1.0

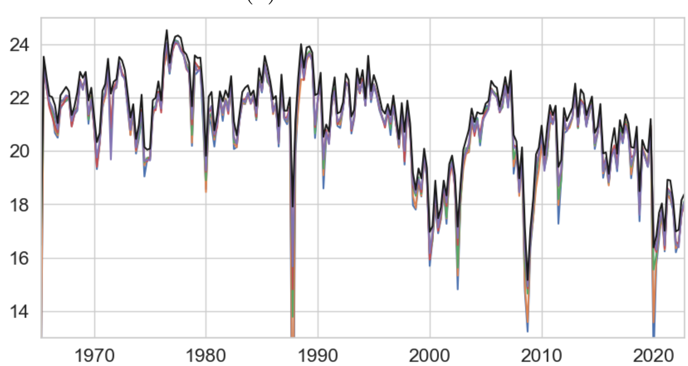
(c) Factor data set.



(a) Industry data set.

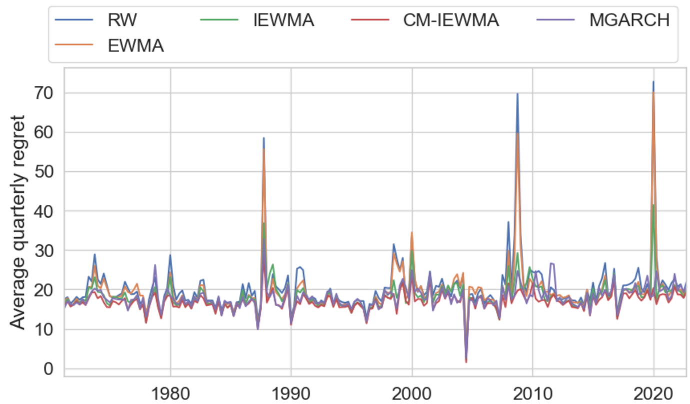


(b) Stock data set.

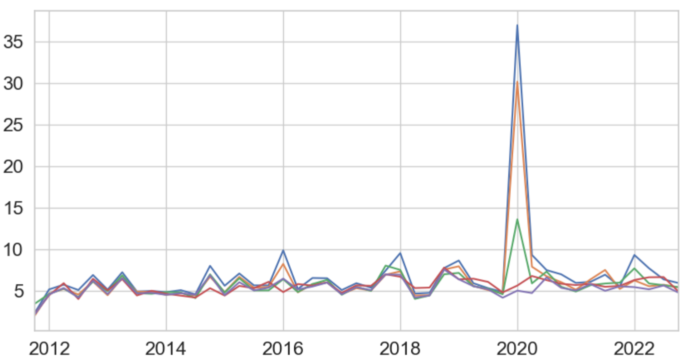


(c) Factor data set.

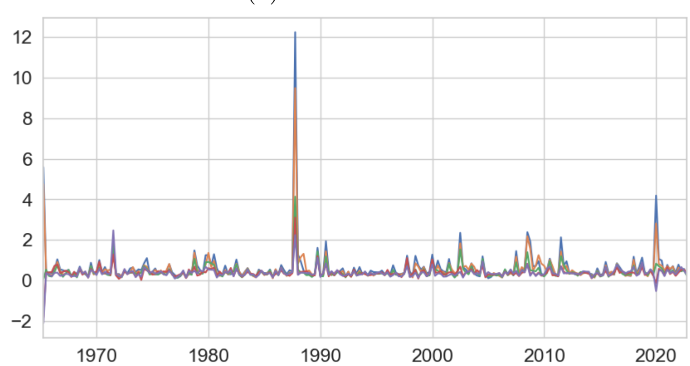
Figure 2.4: The log-likelihood, averaged quarterly, for six covariance predictors and three data sets.



(a) Industry data set.

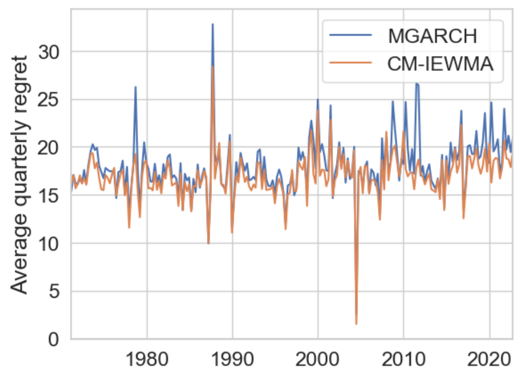


(b) Stock data set.

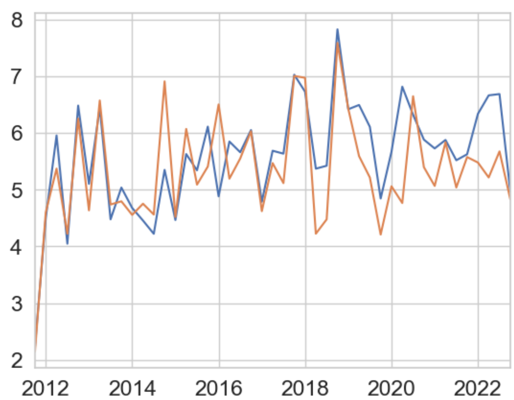


(c) Factor data set.

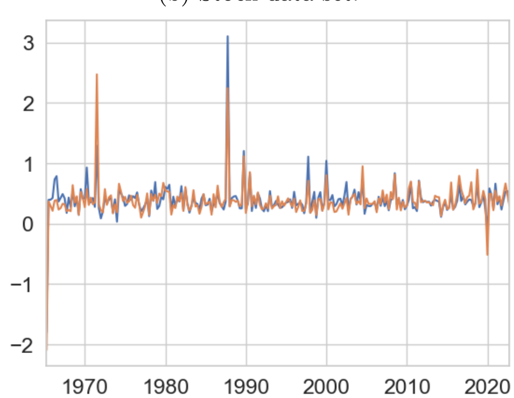
Figure 2.5: The regret, averaged quarterly, for five covariance predictors over the evaluation periods for three data sets.



(a) Industry data set.



(b) Stock data set.



(c) Factor data set.

Figure 2.6: The regret for MGARCH and CM-IEWMA, averaged quarterly over the evaluation periods for three data sets.

Table 2.6: Metrics on the average quarterly regret for six covariance predictors on three data sets.

Predictor	Average	Std. dev.	Max
RW	20.4	6.9	72.8
EWMA	19.4	6.2	70.1
IEWMA	18.2	3.6	41.4
MGARCH	17.9	3.0	32.8
CM-IEWMA	**16.9**	**2.4**	**28.4**
PRESCIENT	0.0	0.0	0.0

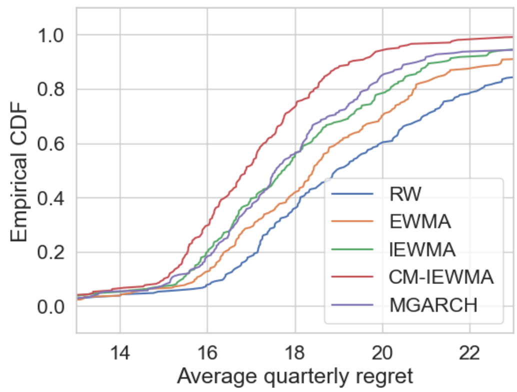
(a) Industry data set.

Predictor	Average	Std. dev.	Max
RW	7.0	4.8	37.0
EWMA	6.2	3.8	30.2
IEWMA	5.8	1.6	13.6
MGARCH	5.6	**1.0**	7.8
CM-IEWMA	**5.3**	**1.0**	**7.6**
PRESCIENT	0.0	0.0	0.0

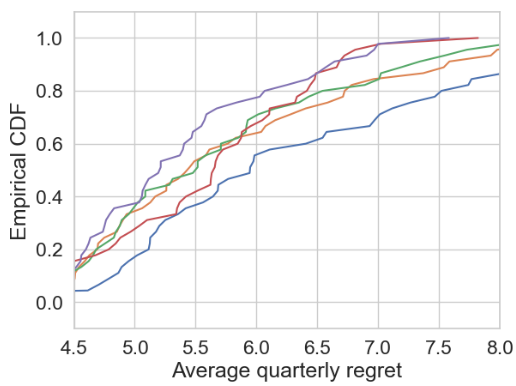
(b) Stock data set.

Predictor	Average	Std. dev.	Max
RW	0.6	0.9	12.2
EWMA	0.6	0.7	9.5
IEWMA	**0.4**	**0.3**	4.1
MGARCH	**0.4**	**0.3**	3.1
CM-IEWMA	**0.4**	**0.3**	**2.9**
PRESCIENT	0.0	0.0	0.0

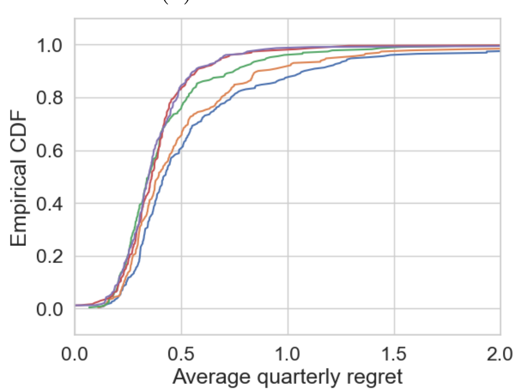
(c) Factor data set.



(a) Industry data set.



(b) Stock data set.



(c) Factor data set.

Figure 2.7: Cumulative distribution functions of average quarterly regret for five covariance predictors on three data sets.

2.6.4 Portfolio performance

In this section we evaluate the covariance predictors on the portfolios described in §2.4.4. In the minimum variance and mean-variance portfolios, we use $L_{\text{max}} = 1.6$ (which corresponds to 130:30 long:short), $w_{\text{min}} = -0.1$ and $w_{\text{max}} = 0.15$ for the industry and stock return portfolios, and $w_{\text{min}} = -0.3$ and $w_{\text{max}} = 0.4$ for the factor return portfolio. We use target (annualized) volatilities of 5%, 10%, and 2% for the industry, stock, and factor return portfolios, respectively.

For the mean-variance portfolio, our estimated returns are EWMAs of the trailing realized returns. For the industry and stock data we use 250-day half-life EWMAs, winsorized at the 40th and 60th percentiles (cross-sectionally), and for the factor data a 63-day half-life EWMA (not winsorized).

Equal weight portfolio. Table 2.7 shows the metrics for the equal weight portfolio. All predictors track the volatility targets well. MGARCH attains the highest Sharpe ratios, although the results are very close. The drawdowns are also very similar for all predictors, but MGARCH and CM-IEWMA seem slightly better than the rest.

Minimum variance portfolio. Table 2.8 shows the metrics for the minimum variance portfolio. For the factor data set, MGARCH does best. On the industry and stock data sets, the three EWMA-based predictors track the volatility target fairly well, while RW and MGARCH underestimate volatility. CM-IEWMA and MGARCH both attain a high Sharpe ratio. However, we note that the high Sharpe ratio for MGARCH, as compared to the other predictors, is a consequence of the high volatility. Finally, CM-IEWMA seems to consistently attain a lower drawdown than the other predictors, although the other EWMA-based approaches also do well.

To illustrate how the minimum variance trading strategy has evolved over time, we show the yearly annualized Sharpe ratios for the CM-IEWMA predictor in figure 2.8. We can see that the Sharpe ratio achieved by the minimum variance portfolio decreases over time for the industry and stock data sets, with a small upward trend for the factor data set.

Risk parity portfolio. The results for the risk-parity portfolio are shown in table 2.9. Overall the results are similar for the various predictors. There is very little that separates the predictors on the industry data set. On the stock data, CM-IEWMA and MGARCH attain the highest Sharpe ratios and lowest drawdowns. On the factor data set, MGARCH has the best overall performance.

Maximum diversification portfolio. The maximum diversification portfolio results are illustrated in table 2.10. On the industry and stock data sets, CM-IEWMA and MGARCH do best in terms of Sharpe ratio, drawdown, and tracking the volatility target. On the factor data set, MGARCH does best overall.

Table 2.7: Metrics for the equal weight portfolio performance for six covariance predictors over the evaluation periods on three data sets.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	2.2	5.4	0.4	16
EWMA	2.2	5.1	0.4	15
IEWMA	2.2	5.1	0.4	15
MGARCH	2.4	5.1	**0.5**	14
CM-IEWMA	2.3	5.0	**0.5**	**13**
PRESCIENT	4.3	4.9	0.9	8

(a) Industry data set.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	6.8	10.6	0.6	23
EWMA	6.4	10.0	0.6	21
IEWMA	6.7	10.1	0.7	20
MGARCH	7.2	9.4	**0.8**	**15**
CM-IEWMA	6.8	9.6	0.7	17
PRESCIENT	12.8	9.9	1.3	10

(b) Stock data set.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	2.9	2.1	1.4	15
EWMA	2.9	2.0	1.4	15
IEWMA	3.0	2.0	1.5	14
MGARCH	3.2	2.0	**1.6**	**12**
CM-IEWMA	2.9	2.1	1.4	15
PRESCIENT	3.3	2.0	1.7	12

(c) Factor data set.

Table 2.8: Metrics for the minimum variance portfolio performance for six covariance predictors over the evaluation periods on three data sets.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	3.1	5.8	0.5	23
EWMA	3.1	5.4	0.6	**19**
IEWMA	3.3	5.5	0.6	**19**
MGARCH	4.3	6.1	**0.7**	20
CM-IEWMA	3.5	5.3	**0.7**	20
PRESCIENT	3.8	5.0	0.8	13

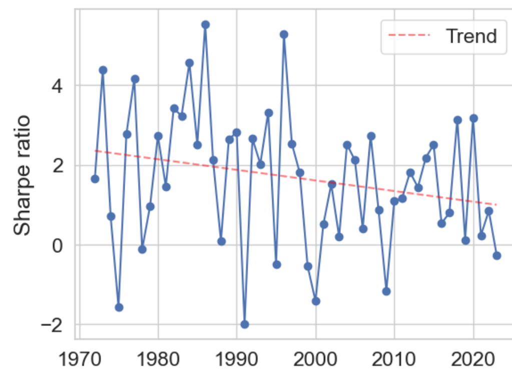
(a) Industry data set.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	9.7	12.0	0.8	23
EWMA	8.9	11.1	0.8	20
IEWMA	9.7	11.3	**0.9**	19
MGARCH	11.3	12.3	**0.9**	18
CM-IEWMA	9.1	11.0	0.8	**15**
PRESCIENT	15.6	10.0	1.6	10

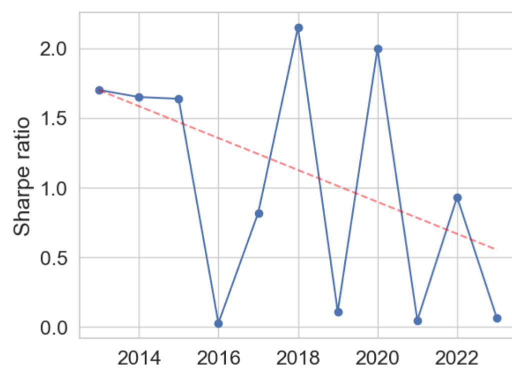
(b) Stock data set.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	1.3	2.2	0.6	20
EWMA	1.4	2.1	0.7	18
IEWMA	1.2	2.1	0.6	17
MGARCH	1.8	2.1	**0.9**	**15**
CM-IEWMA	1.2	2.1	0.5	21
PRESCIENT	1.0	2.0	0.5	22

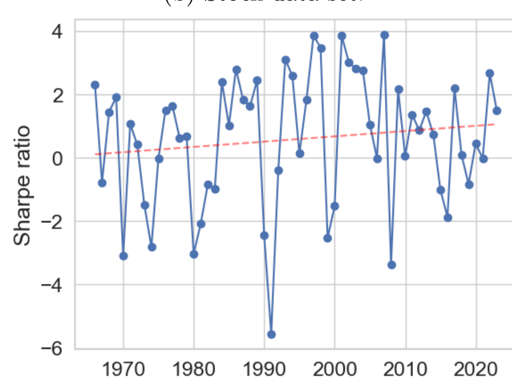
(c) Factor data set.



(a) Industry data.



(b) Stock data set.



(c) Factor data set.

Figure 2.8: Yearly annualized Sharpe ratios together with the linear trend for minimum variance portfolios on three data sets.

Table 2.9: Metrics for the risk parity portfolio performance for six covariance predictors over the evaluation periods on three data sets.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	2.4	5.4	**0.5**	16
EWMA	2.4	5.1	**0.5**	15
IEWMA	2.5	5.1	**0.5**	14
MGARCH	2.7	5.1	**0.5**	14
CM-IEWMA	2.5	5.0	**0.5**	**13**
PRESCIENT	4.7	4.9	1.0	8

(a) Industry data set.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	7.4	10.8	0.7	22
EWMA	6.8	10.1	0.7	21
IEWMA	7.2	10.2	0.7	20
MGARCH	7.9	9.7	**0.8**	**15**
CM-IEWMA	7.4	9.7	**0.8**	16
PRESCIENT	14.3	9.9	1.5	9

(b) Stock data set.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	1.6	2.1	0.7	19
EWMA	1.7	2.1	0.8	18
IEWMA	1.6	2.1	0.8	18
MGARCH	2.0	2.1	**1.0**	**16**
CM-IEWMA	1.5	2.1	0.7	17
PRESCIENT	1.4	2.0	0.7	17

(c) Factor data set.

Table 2.10: Metrics for the maximum diversification portfolio performance for six covariance predictors over the evaluation periods on three data sets.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	2.1	5.5	0.4	16
EWMA	2.1	5.1	0.4	16
IEWMA	2.2	5.2	0.4	14
MGARCH	2.5	5.1	**0.5**	**12**
CM-IEWMA	2.3	5.0	**0.5**	**12**
PRESCIENT	3.8	5.0	0.8	10

(a) Industry data set.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	8.4	11.2	0.8	22
EWMA	7.9	10.4	0.8	21
IEWMA	8.2	10.4	0.8	20
MGARCH	10.0	9.8	**1.0**	**15**
CM-IEWMA	8.8	10.0	0.9	16
PRESCIENT	13.5	9.9	1.4	11

(b) Stock data set.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	1.4	2.2	0.7	19
EWMA	1.5	2.1	0.7	19
IEWMA	1.4	2.1	0.7	19
MGARCH	2.0	2.1	**1.0**	**16**
CM-IEWMA	1.4	2.1	0.7	18
PRESCIENT	1.3	2.0	0.7	18

(c) Factor data set.

Mean variance portfolio. The results for the mean-variance portfolio are given in table 2.11. On the industry data set all predictors underestimate volatility. The results are similar across predictors, with CM-IEWMA and MGARCH performing slightly better than the rest in terms of Sharpe ratio and drawdown. On the stock data set, CM-IEWMA seems to do best overall. On the factor data set, the results are almost identical between predictors.

Since we use simple EWMA return predictors, we can expect the mean variance portfolio performance to vary over time. Intuitively it should be better on historical data than more recent data. To illustrate this, figure 2.9 shows the yearly annualized Sharpe ratios for the three portfolios.

There is a clear downward trend in the Sharpe ratios for the industry and factor data sets, illustrating the difficulty of predicting returns in recent years. This can be compared to the minimum variance portfolios (figure 2.8) that have a more stable performance over time, and notably do not depend on a mean estimate.

Table 2.11: Metrics for the mean variance portfolio performance for six covariance predictors over the evaluation periods on three data sets.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	5.6	6.2	0.9	16
EWMA	5.6	5.8	1.0	15
IEWMA	5.9	5.7	1.0	14
MGARCH	6.7	6.4	1.0	14
CM-IEWMA	6.1	5.6	**1.1**	**13**
PRESCIENT	4.6	5.0	0.9	10

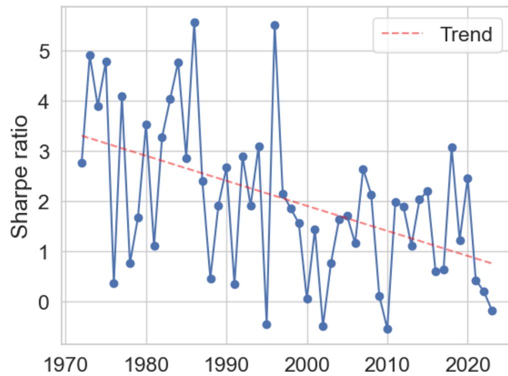
(a) Industry data set.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	6.1	11.9	0.5	26
EWMA	5.9	11.0	0.5	20
IEWMA	7.9	11.1	**0.7**	15
MGARCH	8.3	11.9	**0.7**	18
CM-IEWMA	7.3	10.9	**0.7**	**13**
PRESCIENT	14.3	9.9	1.4	9

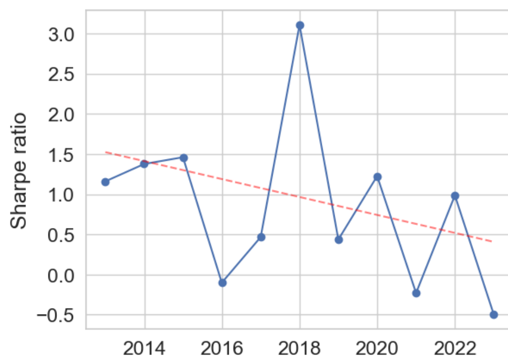
(b) Stock data set.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
RW	7.5	2.2	3.3	4
EWMA	7.2	2.1	**3.4**	4
IEWMA	7.1	2.1	3.3	4
MGARCH	7.3	2.2	3.3	**3**
CM-IEWMA	6.9	2.2	3.2	4
PRESCIENT	6.5	1.9	3.3	4

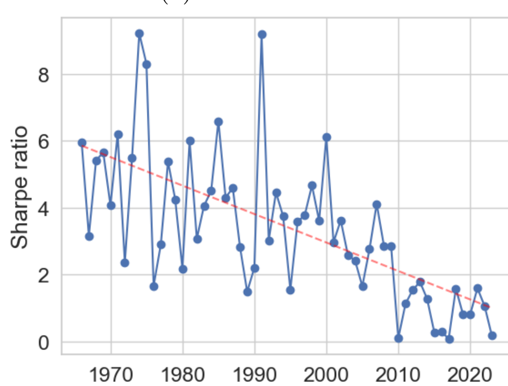
(c) Factor data set.



(a) Industry data.



(b) Stock data set.



(c) Factor data set.

Figure 2.9: Yearly annualized Sharpe ratios together with the linear trend for mean variance portfolios on three data sets.

2.6.5 Summary

In terms of log-likelihood and regret, CM-IEWMA performs best, followed by MGARCH, which performs better than the simpler covariance predictors. In downstream portfolio optimization experiments, CM-IEWMA and MGARCH again perform better than the other predictors, although in many cases not by much. In these experiments there is more variation in the results, partly explained by the difference between our prediction (of a covariance matrix) and our metrics (such as return, risk, drawdown). Even the simplest covariance predictors do a reasonable job of predicting the portfolio risk.

2.7 Realized covariance

We have so far focused on predicting the covariance matrix of asset returns using historical return data, *i.e.*, we predict $\hat{\Sigma}_t$ from $r_1, \ldots, r_{t-1}$. In this section we consider the use of additional data, specifically, intraperiod returns. As an example, suppose the period is (trading) days. The methods described in previous sections predict the covariance of the daily return from previous daily returns. In so-called realized covariance, we predict the daily return covariance using intraday returns. Instead of single period returns r_t, we have multiple returns associated with period t. It is not surprising that using multiple realized returns for each period, instead of just one, can improve our covariance estimates.

Recent literature has shown that realized volatility and correlation measurements (based on high-frequency intraperiod data) can improve performance over traditional predictors that rely on a single realization per period. [138] extend the univariate GARCH model to the joint modeling of returns and realized measures of volatility, and show empirically that this improves performance over the standard GARCH model. In [22] a multivariate realized GARCH model is proposed. More recently, [39] propose a realized semicovariance GARCH model to allow for nuanced responses to positive and negative return shocks.

In this section we show that the dynamically weighted prediction combiner of §2.3.1 readily handles multiple realized returns per period. For simplicity we will assume each period has the same number of intraperiod returns, equally spaced in time. We redefine the return vector to be a return matrix $r_t \in \mathbf{R}^{n \times m}$ with columns that are the m intraperiod return vectors, for times $t = 1, \ldots, T$. The realized covariance at time t is defined as

$$C_t = r_t r_t^T,$$

the same formula for the realized return when r_t is a single (vector) return. The realized covariance matrix C_t has rank m when the m return vectors are linearly independent and $m \leq n$; this can be compared to the realized covariance when we do not have intraperiod returns, which is rank one.

2.7.1 Combined multiple realized EWMAs

The dynamically weighted prediction combiner of §2.3.1 readily handles multiple realized covariance predictors.

Realized EWMA. We define the realized EWMA (REWMA) predictor as

$$\hat{\Sigma}_t = \alpha_t \sum_{\tau=1}^{t-1} \beta^{t-1-\tau} C_\tau, \quad t = 2, 3, \ldots,$$

where C_τ is the realized covariance at time τ, $\beta \in (0, 1)$ is the forgetting factor, and α_t is the normalizing constant; see §2.2.2 for details. This is the same formula as the usual EWMA covariance, with one return per period, given in (2.2), with r_t extended to be a matrix of multiple returns.

Combined multiple realized EWMAs. The combined multiple realized EWMA (CM-REWMA) predictor starts with a set of K REWMA predictors $\hat{\Sigma}_t^{(k)}$ with half-lives $H^{(k)}$, $k = 1, \ldots, K$, and combines them using the dynamically weighted prediction combiner of §2.3.1.

2.7.2 Data and experimental setup

Data set. We consider a universe of $n = 39$ assets with five-minute intraday returns corresponding to $m = 77$. The assets were taken as a subset of those used by [252], and are available at [253]. The data set spans January 2nd 2004 to December 30th 2016, for a total of 252021 data points over 3273 trading days. We list the assets in table 2.12.

Four covariance predictors. We evaluate four covariance predictors, described below.

- The CM-IEWMA predictor used for the stock data from §2.5.2. This predictor only uses daily returns, and is not a realized covariance predictor.
- An REWMA predictor with a half-life of $H = 10$ days, denoted REWMA-10.
- A CM-REMWA predictor with five components with half-lives of $1, 5, 10, 21,$ and 63 days, respectively.
- Prescient predictor, *i.e.*, the empirical covariance for the quarter the day is in. As with the CM-IEWMA predictor, this predictor uses daily return data. It is of course not implementable, and meant only to show a bound on performance with which to compare our implementable predictors.

Ticker	Company Name
JPM	JPMorgan Chase
GS	Goldman Sachs
KO	The Coca-Cola Company
IBM	International Business Machines Corporation
CAT	Caterpillar Inc.
CVX	Chevron Corporation
XOM	Exxon Mobil Corporation
GE	General Electric Company
MRK	Merck & Co., Inc.
VZ	Verizon Communications Inc.
PFE	Pfizer Inc.
WMT	Walmart Inc.
C	Citigroup Inc.
HD	The Home Depot, Inc.
BA	The Boeing Company
MMM	3M Company
MCD	McDonald's Corporation
NKE	NIKE, Inc.
JNJ	Johnson & Johnson
INTC	Intel Corporation
MSFT	Microsoft Corporation
AAPL	Apple Inc.
AMZN	Amazon.com Inc.
CSCO	Cisco Systems, Inc.
PG	Procter & Gamble Co.
ABT	Abbott Laboratories
VLO	Valero Energy Corporation
HON	Honeywell International Inc.
LMT	Lockheed Martin Corporation
TXN	Texas Instruments Inc.
COST	Costco Wholesale Corporation
PEP	PepsiCo, Inc.
UNP	Union Pacific Corporation
WFC	Wells Fargo & Co.
CVS	CVS Health Corporation
ORCL	Oracle Corporation
XRX	Xerox Corporation
TMO	Thermo Fisher Scientific Inc.
NSC	Norfolk Southern Corporation

Table 2.12: Assets used in realized covariance study.

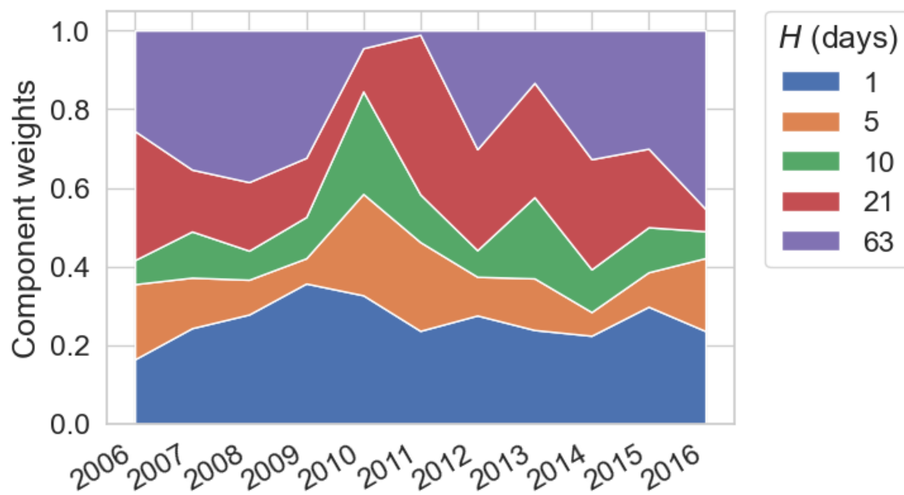


Figure 2.10: CM-REWMA component weights, averaged annually.

2.7.3 Empirical results

CM-REWMA component weights. Figure 2.10 shows the component weights for the CM-REWMA predictor, averaged annually. The weights are fairly stable over time but a weight shift toward faster changing EWMAs is seen in 2008, during the financial crisis.

MSE. The average, standard deviation, and maximum MSEs, computed over distinct quarters for the four covariance predictors, are given in table 2.13. The REWMA and CM-REWMA do

Predictor	Average/10^{-4}	Std. Dev./10^{-3}	Max/10^{-3}
CM-IEWMA	3.1	1.2	7.6
REWMA	**3.0**	**1.1**	7.3
CM-REWMA	**3.0**	**1.1**	**7.2**
PRESCIENT	3.0	1.1	7.1

Table 2.13: Metrics on the MSE, computed over distinct quarters.

slightly better than the CM-IEWMA predictor, but overall there is not a big difference between the predictors.

Regret. Figure 2.11 shows the average regret over distinct quarters for the CM-IEWMA, REWMA, and CM-REWMA predictors. The CM-REWMA predictor has the lowest regret in almost all quarters. It has lower regret than the REWMA predictor in 41 out of the 50 quarters, and lower regret than the CM-IEWMA predictor in 39 out of the 50 quarters.

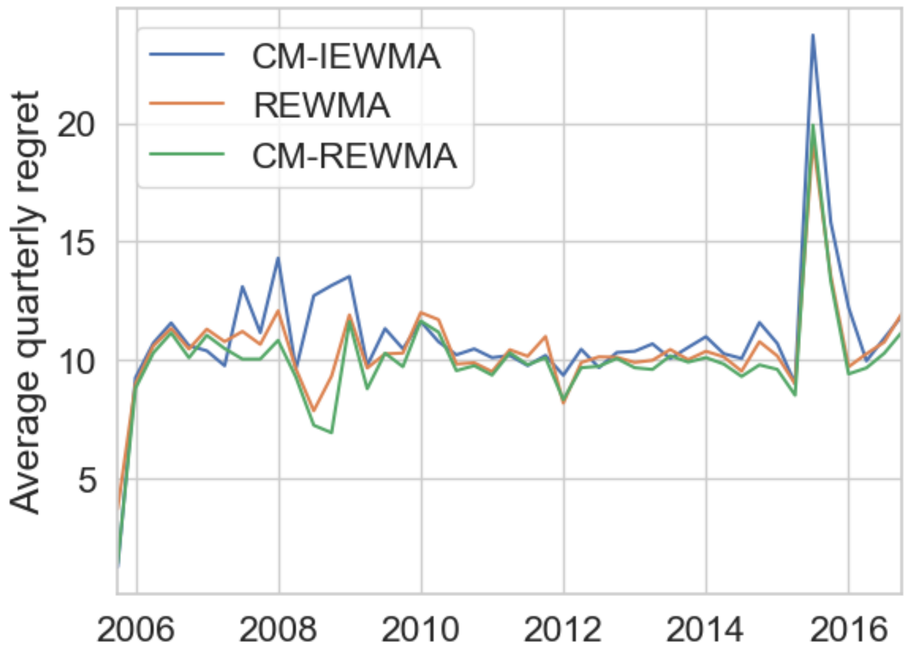


Figure 2.11: Average regret over distinct quarters for three covariance predictors.

Finally, figure 2.12 shows the cumulative distribution functions of the average quarterly regret for the different covariance predictors. CM-REWMA has lower regret than both the CM-IEWMA and REWMA predictors, while REWMA has lower regret than CM-IEWMA.

Portfolio performance. Table 2.14 shows the portfolio metrics for five different portfolio construction methods. CM-REWMA does better than, or as well as, REWMA on almost all metrics, and better than CM-IEWMA for all portfolios. However, the difference on portfolio tasks is not large.

Summary. The results above show that using realized covariance, *i.e.*, intraperiod returns instead of just one return per period, gives covariance estimates that are a bit better than those obtained using only one return per period.

2.8 Large universes

In a practical setting we often encounter a larger number of assets than considered in the previous sections, which has led to extensive research in high-dimensional covariance estimation. One challenge in large dimensions is ensuring positive definiteness of the covariance matrix, in particular with model-based approaches such as MGARCH [20]. Several techniques have been proposed for estimating MGARCH models in large dimensions; see, *e.g.*, [97, 77, 76]. Others have focused on estimating

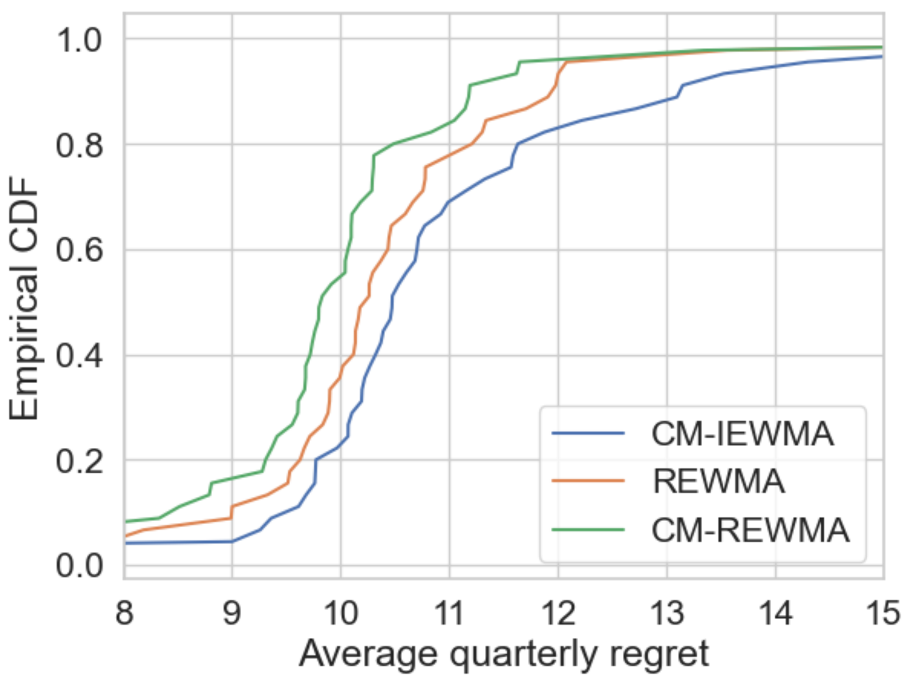


Figure 2.12: Cumulative distribution functions of the average quarterly regret for three covariance predictors.

Predictor	Return/%	Risk/%	Sharpe	Drawdown/%
Equal weight				
CM-IEWMA	3.2	9.9	0.3	18
REWMA	3.7	10.3	**0.4**	**16**
CM-REWMA	4.4	10.6	**0.4**	**16**
PRESCIENT	6.7	9.9	0.7	13
Minimum variance				
CM-IEWMA	10.7	11.0	1.0	25
REWMA	10.7	10.5	1.0	**21**
CM-REWMA	12.0	10.7	**1.1**	**21**
PRESCIENT	11.7	10.0	1.2	12
Risk parity				
CM-IEWMA	4.1	10.0	0.4	18
REWMA	4.7	10.3	**0.5**	**17**
CM-REWMA	5.5	10.6	**0.5**	**17**
PRESCIENT	8.0	9.9	0.8	12
Maximum diversification				
CM-IEWMA	3.6	10.2	0.4	25
REWMA	4.3	10.5	0.4	21
CM-REWMA	5.1	10.8	**0.5**	**19**
PRESCIENT	7.8	9.9	0.8	16
Mean variance				
CM-IEWMA	8.6	10.5	0.8	22
REWMA	8.5	10.3	0.8	**16**
CM-REWMA	9.3	10.5	**0.9**	19
PRESCIENT	10.9	9.8	1.1	21

Table 2.14: Metrics for five different portfolio construction methods, using four covariance predictors.

realized covariance matrices in high dimensions; see, *e.g.*, [244, 301, 142, 79, 103, 3]. For a detailed review of recent developments in high-dimensional covariance estimation, we recommend [21, §6].

The methods described in previous sections can be adapted to handle large universes of assets, say n larger than 100 or so. In this section we describe two closely related methods for improving the performance with large n. Both methods end up modeling $\hat{\Sigma}_t$ as a low rank plus diagonal matrix, in so-called factor form. Before describing these methods, we mention that evaluating log-likelihood regret is complicated with large n. For the empirical covariance to be nonsingular (which is needed to evaluate the regret), we need at least n periods; for daily returns with $n = 1000$, this amounts to four years. Even if we have n periods of data, we would only be able to evaluate the regret a few times. For example, with $n = 1000$ (four years) we need at least 40 years of data to compute the average regret over 10 distinct periods. The log-likelihood, however, can still be evaluated over fewer than n periods.

2.8.1 Traditional factor model

In practice most return covariance matrices for large universes are constructed from factors, with the model

$$r_t = F_t f_t + z_t, \quad t = 1, 2, \ldots,$$

where $F_t \in \mathbf{R}^{n \times k}$ is the factor exposure matrix, $f_t \in \mathbf{R}^k$ is the factor return vector, $z_t \in \mathbf{R}^n$ is the idiosyncratic return, and k is the number of factors, typically much smaller than n. The factor returns are constructed or found by several methods, such as principal component analysis (PCA), or by hand; see, *e.g.*, [14, 13, 190, 191, 255, 254, 102, 101]. Thus we assume that the factor returns are known. Given the factor returns, the rows of the factor exposure matrix are typically found by least squares regression over a rolling or exponentially weighted window [68]. The idiosyncratic returns z_t are then found as the residuals in this least squares fit. The factor returns f_t are modeled as $\mathcal{N}(0, \Sigma_t^{\mathrm{f}})$, and the idiosyncratic returns z_t are modeled as $\mathcal{N}(0, E_t)$, where E_t is diagonal. It is also assumed that the factor returns and idiosyncratic returns are independent across time and of each other.

We end up with a covariance matrix in factor form, *i.e.*, rank k plus diagonal,

$$\Sigma_t = F_t \Sigma_t^{\mathrm{f}} F_t^T + E_t. \tag{2.7}$$

We can easily use the methods described above with a factor model. Simply predict the factor return covariance $\hat{\Sigma}_t^{\mathrm{f}}$ (using the factor returns f_t) and the idiosyncratic variances $\hat{E}_t$ (using the entries of z_t), using the methods described in this monograph, and then form the covariance estimate

$$\hat{\Sigma}_t = F_t \hat{\Sigma}_t^{\mathrm{f}} F_t^T + \hat{E}_t.$$

The factor model (2.7) can be written in a simpler form as

$$\Sigma_t = \tilde{F}_t \tilde{F}_t^T + E_t, \tag{2.8}$$

with $\tilde{F}_t = F_t(\Sigma_t^{\mathrm{f}})^{1/2}$. This form does not include a factor covariance Σ^{f}, or equivalently, assumes $\Sigma_t^{\mathrm{f}} = I$, *i.e.*, the factors are independent with standard deviation one. (The associated factors are called whitened factors.) We will use the factor model form (2.8) in the sequel.

The factor model (2.8) has parameters $\tilde{F}_t$ and E_t, which all together include $nk+n$ scalar parameters. (Some of these are redundant; for example we can insist without loss of generality that F is lower triangular.) The factor model contains substantially fewer scalar parameters than a generic $n \times n$ covariance matrix, which contains $n(n+1)/2$ scalar parameters.

The smaller number of parameters is not the only reason for using a factor model. Another is that it often gives better covariance estimates. We can think of the low rank plus diagonal structure as regularization, which can improve out-of-sample performance. In addition, the low rank plus diagonal structure can be exploited in portfolio construction, bringing the computational complexity down from $O(n^3)$ to $O(nk^2)$ operations [48]. This makes portfolio optimization with $n = 1000$ assets and $k = 50$ factors extremely fast, and makes possible optimization of portfolios with much larger values of n.

2.8.2 Fitting a factor model to a covariance matrix

In this section we consider the problem of fitting a given covariance matrix Σ by one in factor form, $\hat{\Sigma} = FF^T + E$, where $F \in \mathbf{R}^{n \times k}$. This corresponds to the model $r = Ff + z$, with (factor return) $f \sim \mathcal{N}(0, I)$, and (idiosyncratic return) $r \sim \mathcal{N}(0, E)$, with E diagonal. We let $\theta = (F, E)$ denote the parameters of our factor form model.

We seek $F \in \mathbf{R}^{n \times k}$ and diagonal $E \in \mathbf{R}^{n \times n}$ (with positive diagonal entries) that minimize the Kullback-Leibler (KL) divergence between $\mathcal{N}(0, \Sigma)$ and $\mathcal{N}(0, \hat{\Sigma})$,

$$\mathcal{K}(\Sigma, \hat{\Sigma}) = \frac{1}{2}\left(\log \frac{\det \hat{\Sigma}}{\det \Sigma} - n + \mathbf{Tr}\, \hat{\Sigma}^{-1}\Sigma\right). \tag{2.9}$$

The KL divergence can be expressed in terms of the average log-likelihood of $\mathcal{N}(0, \hat{\Sigma})$ under $\mathcal{N}(0, \Sigma)$ as

$$\mathop{\mathbf{E}}_{r \sim \mathcal{N}(0,\Sigma)} \ell_{\hat{\Sigma}}(r) = -\mathcal{K}(\Sigma, \hat{\Sigma}) - (1/2)(n \log 2\pi + n + \log\det\Sigma), \tag{2.10}$$

where $\ell_{\hat{\Sigma}}(r)$ is the log-likelihood of r under $\mathcal{N}(0, \hat{\Sigma})$. Hence minimizing the KL-divergence (2.9) is equivalent to maximizing the expected log-likelihood (2.10) of r under the model $\mathcal{N}(0, \hat{\Sigma})$.

Solution via EM. We can use the expectation-maximization (EM) algorithm to approximately minimize $\mathcal{K}(\Sigma, FF^T + E)$ over F and E [81, 268]. Usually EM is used to fit a factor model to data, *i.e.*, samples; here we use it to fit a given Gaussian distribution $\mathcal{N}(0, \Sigma)$. The method described below was suggested and derived by Emmanuel Candès. We are not aware of its appearance in prior literature. A forthcoming paper on this method will include more detail and applications.

The EM algorithm is an iterative method for maximizing (2.10). Each iteration consists of two steps: the expectation or E-step, and the maximization or M-step. We use the conventional symbols used to describe EM, and use subscript $j = 1, 2, \ldots$ to denote iteration number. (A good method for initializing the EM algorithm is provided below.)

E-step. In the E-step, we find the expected log-likelihood under the current estimate of the parameters $\theta_j = (F_j, E_j)$, over the true distribution of r:

$$Q(\theta \,||\, \theta_j) = \mathop{\mathbf{E}}_{r\sim\mathcal{N}(0,\Sigma)} \mathop{\mathbf{E}}_{p_{\theta_j}(f|r)} \ell_\theta(r, f) \tag{2.11}$$

where $p_{\theta_j}(f \mid r)$ is the density of the conditional distribution of f under the parameter estimates at iteration j, and $\ell_\theta(r, f)$ is the log likelihood of the joint distribution with variable $\theta = (F, E)$.

With our factor model the complete log-likelihood of (r, f) is

$$\begin{aligned} \ell_\theta(r, f) \quad &= \quad -\frac{1}{2}\left((r - Ff)^T E^{-1}(r - Ff) + f^T f + \log\det E\right) \\ &\quad + \frac{1}{(2\pi)^{n/2}} + \frac{1}{(2\pi)^{k/2}} - k/2. \end{aligned}$$

The conditional distribution of $f \mid r$ under θ_j is [31]

$$f \mid r \;\sim\; \mathcal{N}(B_j r, G_j),$$

where

$$B_j = G_j F_j^T E_j^{-1}, \quad G_j^{-1} = F_j^T E_j^{-1} F_j + I. \tag{2.12}$$

Hence, (2.11) becomes, up to an additive constant,

$$-\frac{1}{2}\,\mathbf{Tr}(E^{-1}\,(C_{rr} - 2C_{rf}F^T + FC_{ss}F^T)) - \frac{1}{2}\log\det E, \tag{2.13}$$

where

$$C_{rr} = \Sigma, \quad C_{rf} = \Sigma B_j^T, \quad C_{ff} = B_j \Sigma B_j^T + G_j. \tag{2.14}$$

M-step. In the M-step (2.11) is maximized with respect to θ to obtain the updated parameters:

$$\theta_{j+1} = \underset{\theta}{\text{argmax}}\ Q(\theta \,||\, \theta_j).$$

The maximizer of (2.13) is [268]

$$\begin{aligned} F_{j+1} &= C_{rf} C_{ff}^{-1}, \\ E_{j+1} &= \mathbf{diag}(\mathbf{diag}(C_{rr} - 2C_{rf} F_{j+1}^T + F_{j+1} C_{ff} F_{j+1}^T)), \end{aligned}$$

where the inner **diag** extracts the diagonal of its (matrix) argument, and the outer **diag** creates a diagonal matrix from its (vector) argument.

EM iteration. The EM iteration has the form

$$\begin{aligned} F_{j+1} &= C_{rf} C_{ff}^{-1}, \\ E_{j+1} &= \mathbf{diag}(\mathbf{diag}(C_{rr} - 2C_{rf} F_{j+1}^T + F_{j+1} C_{ff} F_{j+1}^T)), \end{aligned}$$

where C_{rr}, C_{rf}, and C_{ff} come from (2.12) and (2.14).

Initialization. To initialize the EM algorithm we use the following method, based on low rank approximation via eigendecomposition. We work with the correlation matrix of Σ, denoted

$$R = \mathbf{diag}(\sigma)^{-1} \Sigma \, \mathbf{diag}(\sigma)^{-1},$$

where $\sigma = \mathbf{diag}(\Sigma)^{1/2}$ (entrywise). First we express R in its eigendecomposition $R = \sum_{i=1}^n \lambda_i q_i q_i^T$, with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. We then form the rank k approximation

$$\widehat{R} = \sum_{i=1}^k \lambda_i q_i q_i^T.$$

We only need to compute the k dominant eigenvectors and eigenvalues, which can be done efficiently using for example the Lanczos algorithm [123]. Let

$$\widehat{E} = \mathbf{diag}\left(\mathbf{diag}(R - \widehat{R})\right),$$

which can be shown to have positive diagonal entries. Our low-rank plus diagonal approximation of R is then $\widehat{R} + \widehat{E}$. It is also a correlation matrix, *i.e.*, has diagonal entries one. Our final factor approximation of Σ is given by

$$\mathbf{diag}(\sigma)(\widehat{R} + \widehat{E})\, \mathbf{diag}(\sigma) = FF^T + E,$$

with

$$F = \mathbf{diag}(\sigma)[\sqrt{\lambda_1}q_1 \cdots \sqrt{\lambda_k}q_k], \qquad E = \mathbf{diag}(e \circ \sigma^2),$$

where $\circ$ denotes the elementwise (Hadamard) product, and σ^2 means elementwise.

This initialization alone can serve as a basic method to fit a factor model to a given covariance matrix. We will see below that in terms of portfolio optimization, it serves just as well as a factor model fit using the EM method.

2.8.3 Data and experimental setup

Data set. We gather the 500 largest NASDAQ stocks (by market capitalization) at the beginning of 2000 from the WRDS portal [308], compute the daily returns of these stocks from January 3rd 2000 to December 30th 2022, and remove any stocks with missing return values during this period. This gives us 238 stocks over 5787 (trading) days. We acknowledge that we induce a survivor bias, but the purpose of this empirical study is solely to demonstrate the benefit of regularization in large universes, and not to backtest a trading strategy.

Traditional factor model. We create a factor model using PCA as follows. Every year, the k principal components of largest explanatory power are computed, using the past two years of returns. These define the columns of the factor exposure matrix F_t for the following year, and the factor returns f_t are the projections of the returns onto these principal components. The idiosyncratic returns z_t are the residuals. We leverage the CM-IEWMA predictor to compute the factor covariance, using three IEWMA components with half-lives (in days) $H^{\text{vol}}/H^{\text{cor}}$ of $\lceil k/2 \rceil/k, k/3k$, and $3k/6k$, where k denotes the number of factors. To estimate the idiosyncratic variances a 21-day EWMA is used. We evaluate the factor models on the average log-likelihood over the evaluation period.

Fitting a factor model to the covariance matrix. We use a CM-IEWMA covariance predictor with four IEWMA components with half-lives $63/125, 125/250, 250/500$, and $500/1000$ days, respectively, given as $H^{\text{vol}}/H^{\text{cor}}$. Given the CM-IEWMA estimate $\hat{\Sigma}_t$ at time t, we approximate it using a factor model as described in §2.8.2.

To evaluate the factor models, we look at the average log-likelihood over the evaluation period and several performance metrics for a minimum variance portfolio with $L_{\max} = 1.6$, $w_{\min} = -0.1$, and $w_{\max} = 0.15$, diluted to a target risk of 10%.

2.8.4 Empirical results

Traditional factor model. Figure 2.13 shows the log-likelihood versus the number of factors k for k between 2 and 75. A large increase in log-likelihood is attained with around 20 factors, as

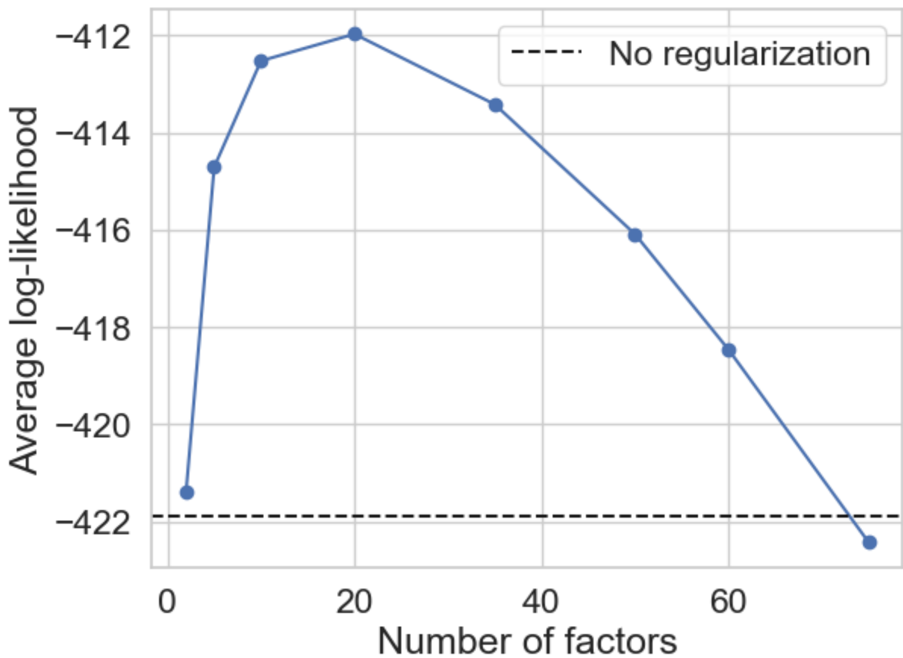


Figure 2.13: Log-likelihood versus the number of factors, using a conventional factor model.

compared to using the full covariance matrix. Thus using a traditional factor model and applying our covariance estimation method to the factor returns improves our overall covariance prediction.

Fitting a factor model to the covariance matrix. Figure 2.14 shows the log-likelihood versus the number of factors (*i.e.*, the rank of the low-rank component) k for various k between 2 and 75, using the eigendecomposition initialization and the EM algorithm. We see that a rank of about $r = 20$ seems optimal for this data set, and achieves a noticeably higher log-likelihood than using the full-rank covariance. Moreover, the EM algorithm does better than just computing the eigendecomposition.

Figure 2.15 shows the portfolio metrics for the minimum variance portfolios. We can see that with roughly 10 factors or more, the performance is essentially identical to that obtained using the full covariance matrix. For these experiments we observed no notable difference between the two factor model fitting methods, *i.e.*, the simple eigendecomposition based initialization and the more sophisticated EM method. While using the factor model does not improve portfolio performance, it greatly speeds up the computation of the portfolio optimization problems.

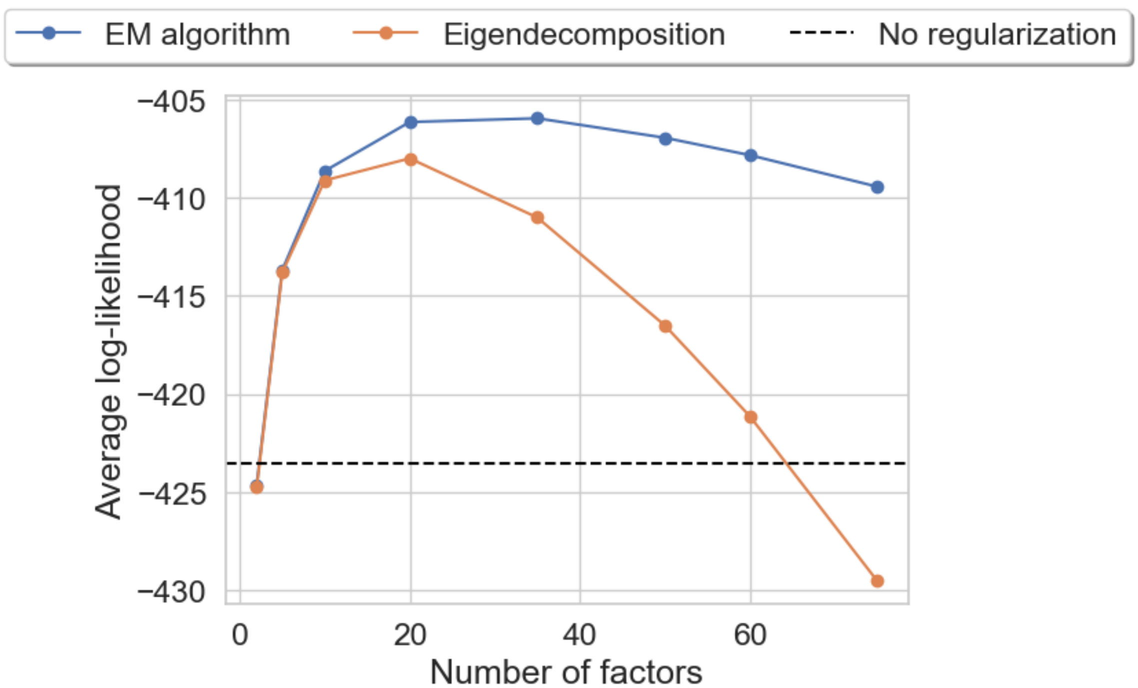
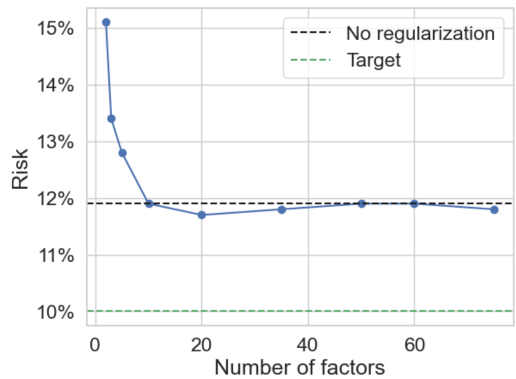
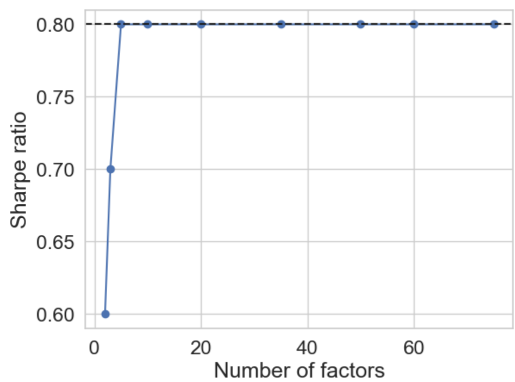


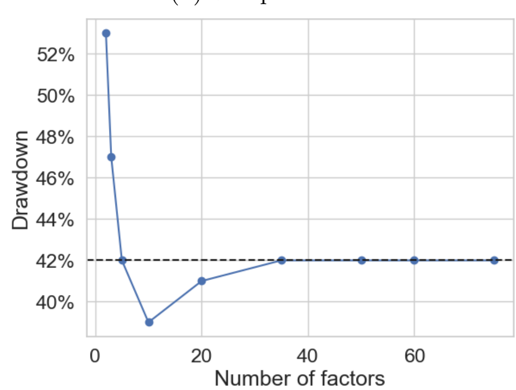
Figure 2.14: Log-likelihood versus the number of factors, obtained by fitting our covariance estimate with a factor model.



(a) Risk.



(b) Sharpe ratio.



(c) Drawdown.

Figure 2.15: Portfolio metrics for minimum variance portfolios constructed via factor models with various number of factors.

2.9 Smooth covariance predictions

We address here a secondary objective for a covariance prediction $\hat{\Sigma}_t$, which is that it vary smoothly across time. Perhaps the main reason for desiring smoothness of the covariance estimate is that it can lead to reduced trading in portfolio construction methods; it can also lead to improved portfolio performance, even without taking into account transaction costs.

To some extent smoothness happens naturally, since whatever method is used to form $\hat{\Sigma}_t$ from $r_1, \ldots, r_{t-1}$ is likely to yield a similar prediction $\hat{\Sigma}_{t+1}$ from $r_1, \ldots, r_t$. It is also possible to further smooth the predictions over time, perhaps trading off some performance, *e.g.*, in log-likelihood regret.

We have already mentioned that the weight optimization problem (2.6) can be modified to encourage smoothness of the weights over time. We can also directly smooth the prediction $\hat{\Sigma}_t$, to get a smooth version $\hat{\Sigma}_t^{\text{sm}}$. A very simple approach is to let $\hat{\Sigma}_t^{\text{sm}}$ be a EWMA of $\hat{\Sigma}_t$, with a half-life chosen as a trade-off between smoother predictions and performance. This EWMA post-processing is equivalent to choosing $\hat{\Sigma}_t^{\text{sm}}$ to minimize

$$\left\|\hat{\Sigma}_t^{\text{sm}} - \hat{\Sigma}_t\right\|_F^2 + \lambda \left\|\hat{\Sigma}_t^{\text{sm}} - \hat{\Sigma}_{t-1}^{\text{sm}}\right\|_F^2,$$

where λ is a positive regularization parameter used to control the trade-off between smoothness and performance, or equivalently, the half-life of the EWMA post-processing. Here the first term is a loss, and the second is a regularizer that encourages smoothly varying covariance predictions.

We can create more sophisticated smoothing methods by changing the loss or the regularizer in this optimization formulation of smoothing. For example, we can use the Kullback-Leibler (KL) divergence as a loss. With regularizer $\lambda\|\hat{\Sigma}_t^{\text{sm}} - \hat{\Sigma}_{t-1}^{\text{sm}}\|_F$ (no square in this case), we obtain a piecewise constant prediction, which roughly speaking only updates the prediction when needed. This is a convex optimization problem which can be solved quickly and reliably [48].

2.9.1 Data and experimental setup

We consider again the Fama-French factor returns from §2.5.1, over the same time horizon. We use the CM-IEWMA covariance predictor with the same parameters as in §2.5.2.

Smoothly varying covariance. In the first experiment we smooth the CM-IEWMA covariance estimates by applying a EWMA, which corresponds to the $\lambda \left\|\hat{\Sigma}_t^{\text{sm}} - \hat{\Sigma}_{t-1}^{\text{sm}}\right\|_F^2$ regularizer. For different EWMA half-lives we attain different levels of smoothness.

Piecewise constant covariance. In the second experiment we smooth the CM-IEWMA covariance estimates by applying the $\lambda \left\|\hat{\Sigma}_t^{\text{sm}} - \hat{\Sigma}_{t-1}^{\text{sm}}\right\|_F$ regularizer. For different values of λ we attain piecewise constant covariance predictors with different update frequencies.

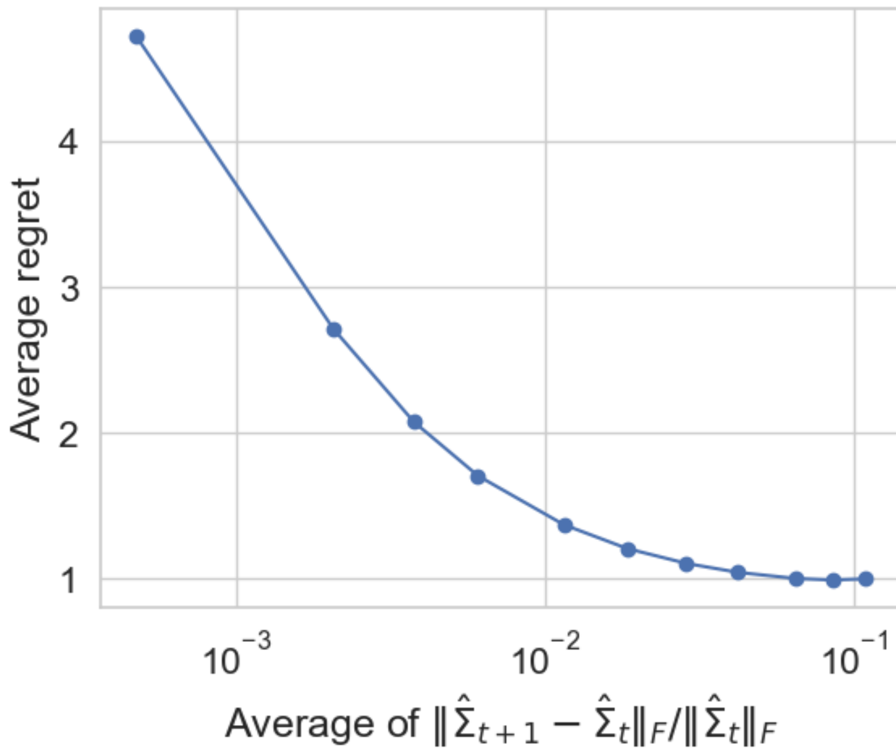


Figure 2.16: Average regret versus smoothness when using EWMA smoothing of the covariance predictor.

2.9.2 Empirical results

Smoothly varying covariance. Figure 2.16 shows the regret versus smoothness for various levels of smoothness. As seen, we can reduce the smoothness by a factor of (roughly) four without losing much performance in terms of regret. This can obviously be useful in practice since a smoother covariance estimate would, for example, reduce trading.

Table 2.15 shows the portfolio metrics for various values of λ for the minimum variance portfolio with the same parameters as in §2.6.4; here the turnover is defined as the average of $252 \times \|w_{t+1} - w_t\|_1 / \|w_t\|_1$ over all times t in the evaluation period. Interestingly, the right amount of smoothing not only reduces turnover, but also improves portfolio performance in terms of Sharpe ratio and drawdown, while keeping the desired volatility level. Too much smoothing, however, leads to reduced portfolio performance. Figure 2.17 shows the yearly annualized Sharpe ratios for $\lambda = 10^{-4}$, indicating a stable performance over time.

Figure 2.18 shows the portfolio weights for three different EWMA half-lives.

As seen, EWMA smoothing leads to smoothly varying portfolio weights, while the weights vary significantly when no smoothing is applied.

Table 2.15: Portfolio metrics for various EWMA half-lives used for smoothing the covariance. Half-life 0 means no smoothing.

Half-life/days	Return/%	Risk/%	Sharpe	Drawdown/%	Turnover/%
0	1.2	2.1	0.5	21	1855
10	1.4	2.1	0.7	16	310
100	1.8	2.1	0.9	15	56
250	2.1	2.1	1.0	**13**	30
5000	2.9	2.6	**1.1**	21	9

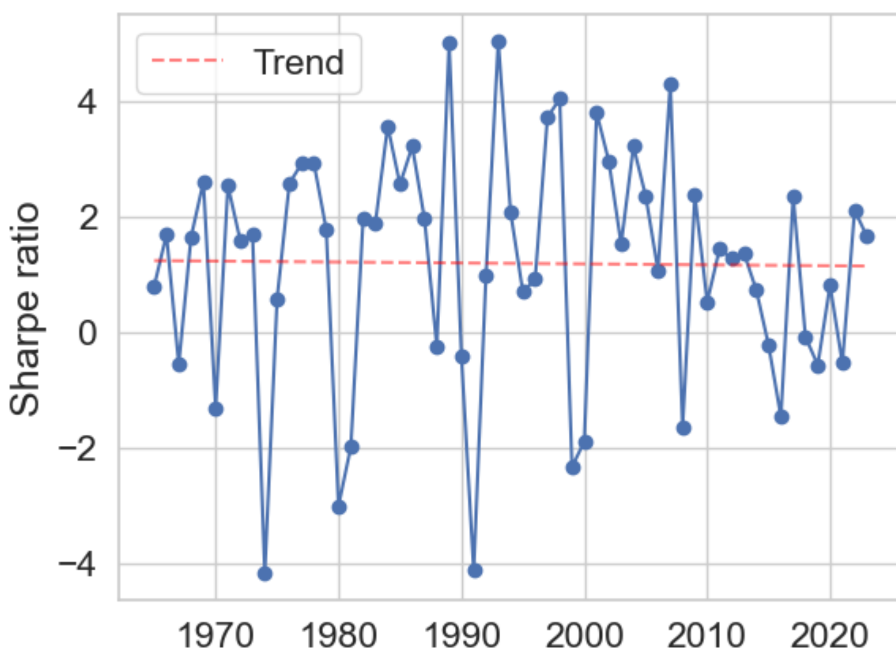
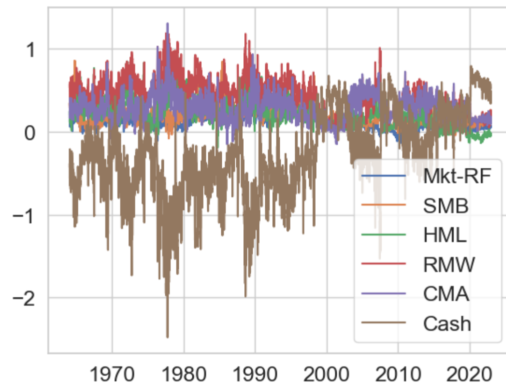
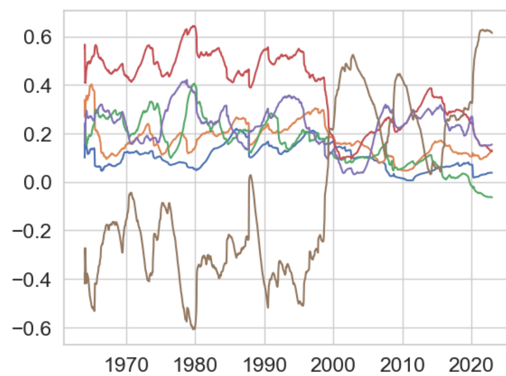


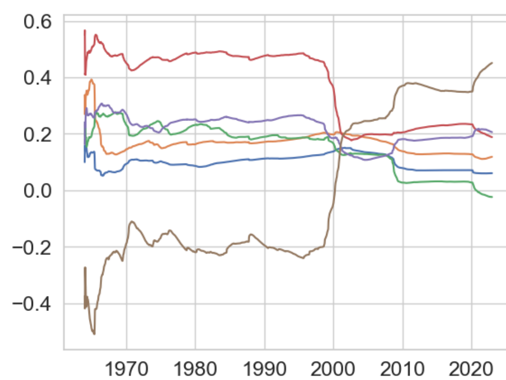
Figure 2.17: Yearly annualized Sharpe ratios for the minimum variance portfolio when smoothing the CM-IEMA covariance predictor with a 250-day half-life EWMA.



(a) No smoothing.



(b) Half-life of 250 days.



(c) Half-life of 5000 days.

Figure 2.18: Portfolio weights for three different regularization parameters λ.

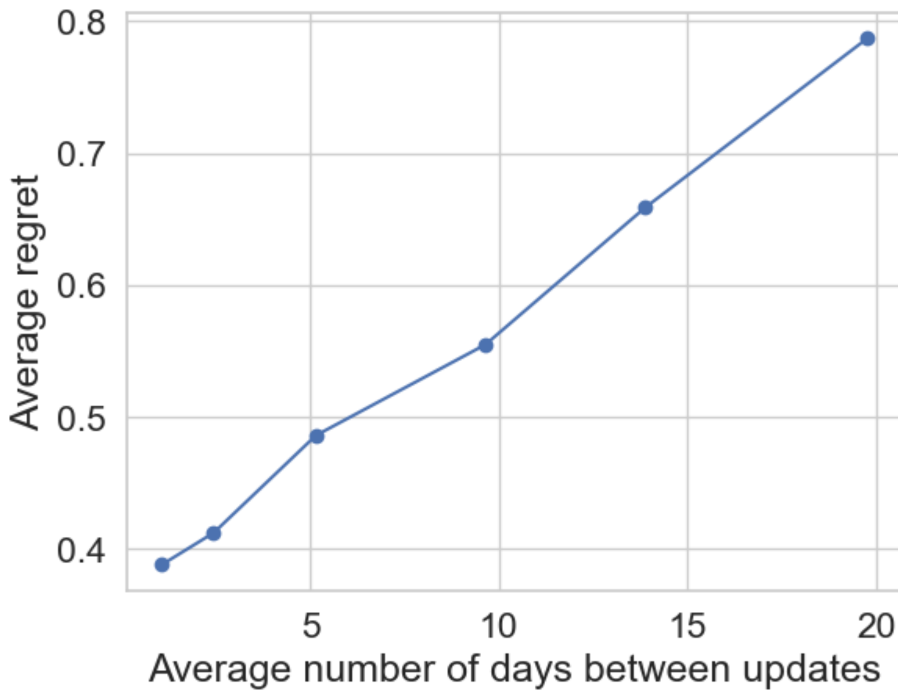


Figure 2.19: Average regret versus time between covariance updates.

Table 2.16: Portfolio metrics for various regularization parameters λ. $\lambda = 0$ means no smoothing.

λ	Return/%	Risk/%	Sharpe	Drawdown/%	Turnover/%
0	1.2	2.1	0.5	21	1855
5×10^{-5}	1.9	2.0	1.0	14	1190
10^{-4}	2.4	1.9	**1.3**	**9**	112
10^{-3}	2.6	2.1	1.2	17	7
7.5×10^{-3}	3.0	4.8	0.6	31	0

Piecewise constant covariance. Figure 2.19 shows the regret versus the update frequency of the covariance estimate. There is a clear trade-off between the regret and update frequency. Roughly speaking, we could update the covariance matrix weekly without losing much in terms of regret.

As mentioned, a piecewise constant predictor can be desirable in practice, since it encourages us not updating the portfolio weights, which in turn reduces trading costs. Table 2.16 shows the portfolio metrics for various values of λ for the minimum variance portfolio from §2.6.4. As seen, smoothing can significantly reduce turnover, and interestingly improve the Sharpe ratio and drawdown noticeably while maintaining the correct risk level. Figure 2.20 shows the yearly annualized Sharpe ratios for $\lambda = 10^{-4}$. The performance is relatively stable over time, with a small downward trend.

To illustrate the impact of smoothing we show the portfolio weights for three different values of

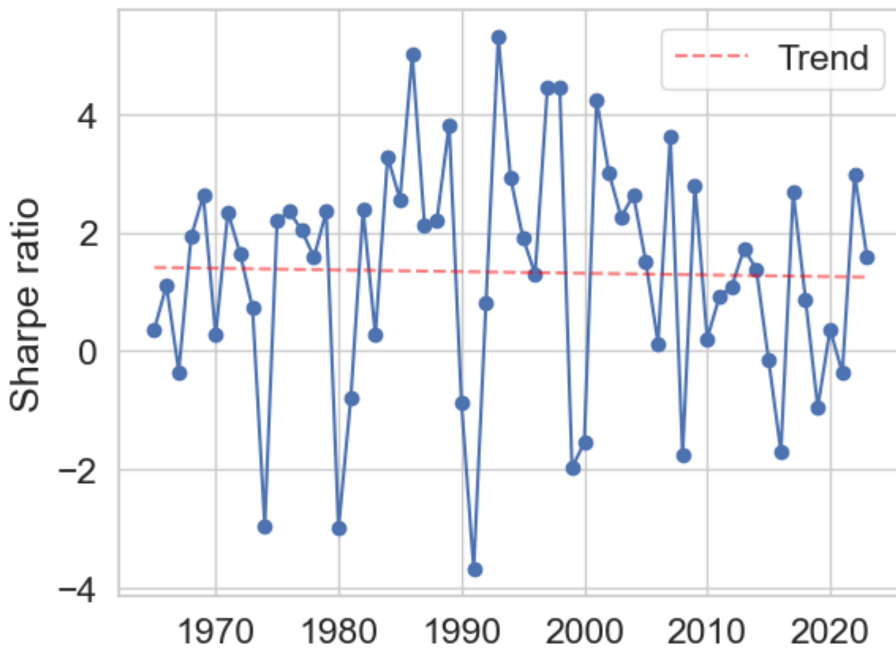


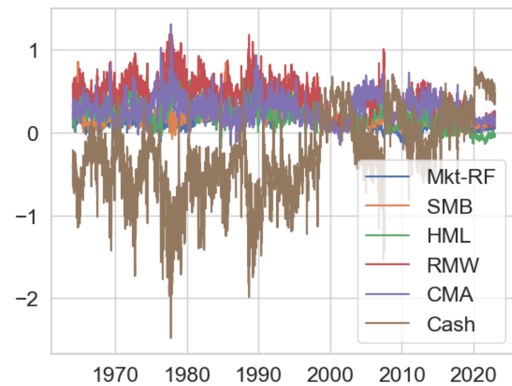
Figure 2.20: Yearly annualized Sharpe ratios for the minimum variance portfolio with a piecewise constant CM-IEMA covariance predictor using $\lambda = 10^{-4}$.

λ in figure 2.21. Without smoothing the portfolio weights are updated significantly every day. For $\lambda = 10^{-4}$ the weights are updated around once or twice a month. Finally, for $\lambda = 10^{-5}$ the weights are updated on average every half a year, with only four big weight updates over the whole trading period. Interestingly, the weight updates for $\lambda = 10^{-5}$ correspond precisely in time to the volatile regime around 1980, the 2000 dot-com bubble, the 2008 financial crisis, and the 2020 pandemic. In short, we can conclude from table 2.16 and figure 2.21 that smoothing can lead to less trading and improve the portfolio performance.

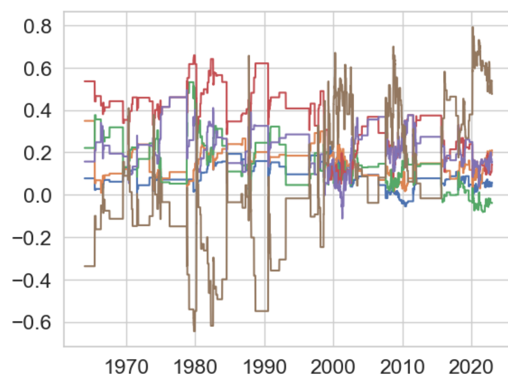
Finally, we note that there is some deviation between the regret metric and portfolio performance. As seen from figure 2.19 regret increases as we update the covariance matrix less than every other week. However, as seen from table 2.16 and figure 2.21, portfolio performance can improve notably when updating the covariance matrix only every few months, or even years.

2.10 Simulating returns

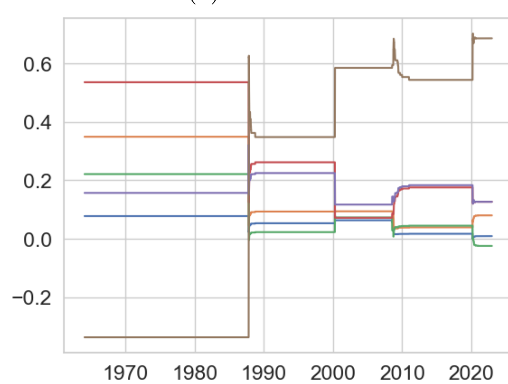
Our model can be used to simulate future returns, when seeded by past realized ones. To do this, we start with realized returns for periods $1, \dots, t-1$, and compute $\hat{\Sigma}_t$ using our method. Then we generate or sample r_t^{sim} from $\mathcal{N}(0, \hat{\Sigma}_t)$. We then find $\hat{\Sigma}_{t+1}$ using the returns $r_1, \dots, r_{t-1}, r_t^{\text{sim}}$. We generate r_{t+1}^{sim} by sampling from $\mathcal{N}(0, \hat{\Sigma}_{t+1})$. This continues.



(a) $\lambda = 0$.



(b) $\lambda = 10^{-3}$.



(c) $\lambda = 10^{-4}$.

Figure 2.21: Portfolio weights for three different regularization parameters λ.

This simple method generates realistic return data in the short term. Of course, it does not include shocks or rapid changes in the return statistics that we would see in real data, but the generative return method has several practical applications. To mention just one, we can simulate 100 (say) different realizations over the next quarter (say), and use these to compute 100 performance metrics for our portfolio. This gives us a distribution of the performance metric that we might see over the next quarter.

2.10.1 Data and experimental setup

To illustrate the generative return method, we consider the five Fama-French factor returns from §2.5.1. Using the same setup as in §2.5.2 we compute CM-IEMWA covariance estimates, using data from January 1st 2011 to December 31 2013, *i.e.*, over a three-year period. Returns are then generated for 100 days, using the generative mode described above.

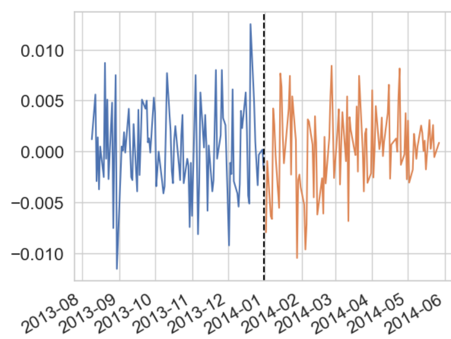
2.10.2 Empirical results

We illustrate the results by looking at the SMB factor, *i.e.*, we look at the marginal distribution of this factor. Figure 2.22 shows the true SMB factor returns and the simulated returns for two different random number generator seeds. As seen, we attain realistic returns that could be used to generate scenarios for downstream portfolio optimization tasks, for example.

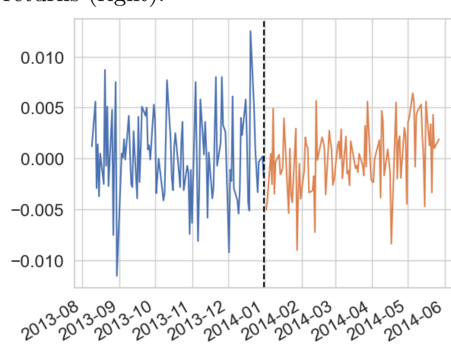
2.11 Conclusions

We have introduced a simple method for predicting covariance matrices of financial returns. Our method combines well known ideas such as EWMA, first estimating volatilities and then correlations, and dynamically combining multiple predictions. The method relies on solving a small convex optimization problem (to find the weights used in the combining), which is extremely fast and reliable. The proposed predictor requires little or no tuning or fitting, is interpretable, and produces results better than the popular EWMA estimate, and comparable to MGARCH. Given its interpretability, light weight, and good practical performance, we see it as a practical choice for many applications that require predictions of the covariance of financial returns.

(a) Observed returns.



(b) Observed returns (left) and simulated returns (right).



(c) Observed returns (left) and simulated returns (right).

Figure 2.22: Observed and simulated SMB factor returns for two different seeds. The vertical line separates the in-sample (observed returns) and out-of-sample (simulated returns) periods.

Chapter 3

Markowitz portfolio construction at seventy

3.1 Introduction

Harry Markowitz's 1952 paper *Portfolio Selection* [209] was a true breakthrough in our understanding of and approach to investing. Before Markowitz there was (almost) no mathematical approach to investing. As a 25-year-old graduate student, Markowitz founded modern portfolio theory, and methods inspired by him would become the most widely used portfolio construction practices over the next 70 years (and counting).

Before Markowitz, diversification and risk were fuzzy concepts. Investors loosely connected risk to the probability of loss, but with no analytical rigor around that connection. Ben Graham, who along with David Dodd wrote *Security Analysis* [126], once commented that investors should own "a minimum of ten different issues and a maximum of about thirty" [125].

There were a few precursors, such as an article by de Finetti, that contained some similar ideas before Markowitz; see [74, 269] for a discussion and more of the history of mathematical formulation of portfolio construction. Another notable precursor is John Burr Williams' 1938 *Theory of Investment Value* [306]. He argued that the value of a company was the present value of future dividends. His book is full of mathematics, and Williams predicted that "mathematical analysis is a new tool of great power, whose use promises to lead to notable advances in investment analysis". That prediction came true with Markowitz's work. Indeed, Markowitz considered Williams' book as part of his inspiration. According to Markowitz, "the basic concepts of portfolio theory came to me one afternoon in the library while reading John Burr Williams' *Theory of Investment Value*".

For many years, the lack of data and accessible computational power [213] rendered Markowitz's ideas impractical, despite his pragmatic approach. In 1963, William Sharpe published his market model [279], designed to speed up the Markowitz calculations. This model was a one-factor risk model (the factor was the market return), with the assumption that all residual returns are uncorrelated. His paper stated that solving a 100-asset problem on an IBM 7090 computer required 33 minutes, but his simplified risk model reduced it to 30 seconds. He also commented that computers could only handle 249 assets at most with a full covariance matrix, but 2000 assets with the simplified risk model. Today such a problem can be solved in microseconds; we can routinely solve problems with tens of thousands of assets and substantially more factors in well under one second.

Markowitz portfolio construction has thrived for many years in spite of claims of various alleged deficiencies. These have included the method's sensitivity to data errors and estimation uncertainty, its single-period nature to handle what is fundamentally a multi-period problem, its symmetric definition of risk, its neglect of higher moments like skewness and kurtosis, and its neglect of transaction costs and leverage risk. We will address these alleged criticisms and show that standard techniques in modern approaches to optimization effectively deal with them without altering Markowitz's vision for portfolio selection.

In 1990 Markowitz was awarded the Nobel Memorial Prize in Economics for his work on portfolio theory, shared with Merton Miller and William Sharpe. For more light on the fascinating historic details we recommend an interview with Markowitz [213], his acceptance speech for the Nobel Prize [214], and his remarks in the introduction to the *Handbook of Portfolio Construction* [133].

3.1.1 The original Markowitz idea

Markowitz identified two steps in the portfolio selection process. In a first step, the investor forms beliefs about the expected returns of the assets, expressed as a vector μ, and their covariances, expressed as a covariance matrix Σ, which gives the volatilities of asset returns and the correlations among them. These beliefs are the core inputs for the second step, which is the optimization of the portfolio based on these quantities.

He introduced the expected returns–variance of returns (E–V) rule, which states that an investor desires to achieve the maximum expected return for a portfolio while keeping its variance or risk below a given threshold. Convex programming was not a well developed field at that time, and Markowitz used a geometric interpretation in the space of portfolio weights [209] to solve the problem we would now express as

$$\begin{array}{ll} \text{maximize} & \mu^T w \\ \text{subject to} & w^T \Sigma w \leq (\sigma^{\text{tar}})^2, \\ & \mathbf{1}^T w = 1, \end{array} \tag{3.1}$$

with variable $w \in \mathbf{R}^n$, the set of portfolio weights, where $\mathbf{1}$ is the vector with all entries one. The data in the problem are $\mu \in \mathbf{R}^n$, the vector of expected asset returns, and Σ, the $n \times n$ covariance matrix of asset returns. The positive parameter σ^{tar} is the target portfolio return standard deviation or volatility. (We define the weights and describe the problem more carefully in §3.2.)

There are many other ways to formulate the trade-off of expected return and risk as an optimization problem [49, 10]. One very popular method maximizes the *risk-adjusted return*, which is the expected portfolio return minus its variance, scaled by a positive *risk-aversion parameter*. This leads to the optimization problem [130]

$$\begin{array}{ll} \text{maximize} & \mu^T w - \gamma w^T \Sigma w \\ \text{subject to} & \mathbf{1}^T w = 1, \end{array} \tag{3.2}$$

where γ is the risk-aversion parameter that controls the trade-off between risk and return. Both problems (3.1) and (3.2) give the full trade-off curve of Pareto optimal weights, as σ^{tar} or γ vary from 0 to ∞ (although (3.1) can be infeasible when σ^{tar} is too small). One advantage of the first formulation (3.1) is that the parameter σ^{tar} that controls the volatility is interpretable as, simply, the target risk level. The risk-aversion parameter γ appearing in (3.2) is less interpretable. We will have more to say about the parameters that control trade-offs in portfolio construction in §3.4.

Both problems (3.1) and (3.2) have analytical solutions. For example the solution of (3.2) is given by

$$w^\star = \frac{1}{2\gamma}\Sigma^{-1}(\mu + \nu^\star \mathbf{1}), \qquad \nu^\star = \frac{2\gamma - \mathbf{1}^T\Sigma^{-1}\mu}{\mathbf{1}^T\Sigma^{-1}\mathbf{1}}.$$

(The scalar $\nu^\star$ is the optimal dual variable [49, Chap. 5].) We note here the appearance of the inverse covariance matrix. To compute $w^\star$ we would not compute the inverse, but rather solve two sets of equations to find $\Sigma^{-1}\mu$ and $\Sigma^{-1}\mathbf{1}$ [51]. Still, the appearance of Σ^{-1} in the expressions for the solutions give us a hint that the method can be sensitive to the input data when the covariance matrix Σ is nearly singular. These analytical formulas can also be used to back out so-called implied returns, *i.e.*, the mean μ for which a given portfolio is optimal. For example the *market implied return* μ^{mkt} is the return for which the optimal weights are the market weights, *i.e.*, proportional to asset capitalization.

Both formulations (3.1) and (3.2) are referred to as the basic Markowitz problem, or *mean-variance optimization*, since they both trade off the mean and variance of the portfolio return. In his original paper Markowitz also noted that additional constraints can be added to the problem, specifically the constraint that $w \geq 0$ (elementwise), which means the portfolio is long-only, *i.e.*, it does not contain any short positions. With this added constraint, the two problems above do not have simple analytical solutions. But the formulation (3.2), with the additional constraint $w \geq 0$, is a quadratic program (QP), a type of convex optimization problem for which numerical solvers were developed already in the late 1950s [307]. In that early paper on QP, solving the Markowitz problem (3.2) with the long-only constraint $w \geq 0$ was listed as a prime application. Today we can solve either formulation reliably, with essentially any set of convex portfolio constraints.

Since the 1950s we have seen a truly stunning increase in computer power, as well as the development of convex optimization methods that are fast and reliable, and high-level languages that allow users to express complex convex optimization problems in a few lines of clear code.

These advances allow us to extend Markowitz's formulation to include a large number of practical constraints and additional terms, such as transaction cost or the cost incurred when holding short positions. In addition to directly handling a number of practical issues, these generalizations of the basic Markowitz method also address the issue of sensitivity to the input data μ and Σ. This chapter describes one such generalization of the basic Markowitz problem, that works well in practice.

Out of respect for Markowitz, and because the more generalized formulation we present here is nothing more than an extension of his original idea, we will refer to these more complex portfolio construction methods also as Markowitz methods. When we need to distinguish the extension of Markowitz's portfolio construction that we recommend from the basic Markowitz method, we refer to it as Markowitz++. (In computer science, the post-script ++ denotes the successor.)

3.1.2 Alleged deficiencies

The frequent criticism of Markowitz's work is a testament to its importance. These criticisms usually fall into one or more of the following (related) categories.

It's sensitive to data errors and estimation uncertainty. The sensitivity of Markowitz portfolio construction to input data is well documented [236, 229, 274, 55, 65], and already hinted at by the inverse covariance that appears in the analytical solutions of the basic Markowitz method. This sensitivity, coupled with the challenge of estimating the mean and covariance of the return, leads to portfolios that exacerbate errors or deficiencies in the input data to find unrealistic and poorly performing portfolios. Some authors argue that choosing a portfolio by optimization, as Markowitz's method does, is essentially an estimation-error maximization method. This is still a research topic that draws much attention. In the recent papers [122, 284] the authors quantify how the (basic) Markowitz portfolio is affected by estimation errors in the covariance matrix.

This criticism is justified, on the surface. Markowitz portfolio construction can perform poorly when it is naïvely implemented, for example by using empirical estimates of mean and covariance on a trailing window of past returns. The critical practical issues of taming sensitivity and gracefully handling estimation errors are readily addressed using techniques such as regularization and robust optimization, described in more detail in §3.1.3. A recent paper on this topic is [26].

It implicitly assumes risk symmetry. Markowitz portfolio construction uses variance of the portfolio return as its risk measure. With this risk measure a portfolio return well above the mean is just as bad as one that is well below the mean, whereas the former is clearly a good event, not a bad one. This observation should at least make one suspicious of the formulation, and has motivated a host of proposed alternatives, such as defining the risk taking into account only the downside [210, Chap. IX]. This criticism is also valid, on the surface. But when the parameters are chosen appropriately, and the data are reasonable, portfolios constructed from mean-variance optimization do not suffer from this alleged deficiency. Moreover, it has been shown that the mean-variance optimal portfolio is the same as the optimal portfolio attained from maximizing expected utility under a broad range of distributional assumptions on the asset returns, including asymmetric ones [26, 275, 227, 11, 64].

We should maximize expected utility. A more academic version of the previous criticism is that portfolios should be constructed by maximizing the expected value of a concave increasing utility function of the portfolio return [303]. The utility in mean-variance optimization (with risk-adjusted return objective) is $U(R) = R - \gamma R^2$, where R is the portfolio return. This utility function is concave, but only increasing for $R < 1/(2\gamma)$; above that value of return, it decreases, putting us in the awkward position of seeming to prefer smaller returns over larger ones.

This criticism is also valid, taken at face value; the quadratic utility above is indeed not increasing. Markowitz himself addressed the issue in a 1979 paper with H. Levy that argued that while mean-variance optimization does not appear to be the same as maximizing an expected utility, it is a very good approximation; see [192] and [215, Chap. 2]. But in fact it turns out that Markowitz portfolio construction *does* maximize the expected value of a concave increasing utility function. Specifically if we model the returns as Gaussian, and use the exponential utility $U(R) = 1 - \exp(-\gamma R)$, then the expected utility is the risk-adjusted return, up to an additive constant [204]. In other words, Markowitz portfolio construction *does* maximize expected utility of portfolio return, for a specific concave increasing utility function and a specific asset return distribution. Beyond practical considerations and the specific case of Gaussian distributions, [26] recently expanded the applicability of Markowitz portfolio construction by demonstrating its equivalence to maximizing expected utility under a wide array of distributional assumptions. This broader applicability encompasses not only elliptical distributions, such as the t-distribution, but also extends to distributions with fat tails and those that exhibit asymmetry. For more on the relationship between Markowitz portfolio construction and expected utility maximization, see [275, 227, 11, 64].

It considers only the first and second moments of the return. Mean-variance optimization naturally only considers the first two moments of the distribution. It would seem that taking higher moments like skewness and kurtosis into account might better describe investor preferences [61, 320]. This, coupled with the fact that the tails of asset returns are not well modeled by a Gaussian distribution [100], suggests that portfolio construction should consider higher moments than the first and second.

While it is possible to construct small academic examples where mean-variance optimization does poorly due to its neglect of higher moments, simple mean-variance optimization does very well on practical problems. In [204] the authors extend Markowitz by maximizing exponential utility, but with a more complex Gaussian mixture model of asset returns. Such a distribution is general, in that it can approximate any distribution. Their method evidently handles higher moments, but empirically gives no boost in performance on practical problems.

Markowitz himself addressed the common misconception that he labeled the "Great Confusion" [213, 215, 211, 212], stating that Gaussian returns are merely a sufficient but not a necessary condition on the return distribution for mean-variance optimization to work well and that mean and variance are good approximations for expected utility.

It's a greedy method. Portfolios are generally not just set up and then held for one investment period; they are rebalanced, and sometimes often. Problems in which a sequence of decisions are made, based on newly available information, are more accurately modeled not as simple optimization problems, but instead as stochastic control problems, also known as sequential decision making under uncertainty [174, 175, 23, 28]. In the context of stochastic control, methods that take into account only the current decision and not future ones are called *greedy*, and in some cases can perform very poorly. This criticism is also, on its face, valid. Using Markowitz portfolio construction repeatedly, as is always done in practice, is a greedy method.

We can readily counter this criticism. First, in the special case with risk-adjusted return and quadratic transaction costs, and no additional constraints, the stochastic optimal policy can be worked out, and coincides with a single-period Markowitz portfolio [131, 17]. This suggests that when other constraints are present, and the transaction cost is not quadratic, the (greedy) Markowitz method should not be too far from stochastic optimal.

Second, there are extensions of Markowitz portfolio construction, called *multi-period* methods, that plan a sequence of trades over a horizon, and then execute only the first trade; see, *e.g.*, [43, 193, 219]. These multi-period methods can work better than so-called single-period methods, for example when a portfolio is transitioning between two managers, or being set up or liquidated over multiple periods. But in almost all other cases, single-period methods work just as well as multi-period ones.

The third response to this criticism more directly addresses the question. In the paper *Performance Bounds and Suboptimal Policies for Multi-Period Investment* [46], the authors develop bounds on how well a full stochastic control trading policy can do, and show empirically that single-period Markowitz trading essentially does as well as a full stochastic control policy (which is impractical if there are more than a handful of assets). So while there are applications where greedy policies do much more poorly than a true stochastic control policy, it seems that multi-period trading is not one of them.

3.1.3 Robust optimization and regularization

Here we directly address the question of sensitivity of Markowitz portfolio construction to the input data μ and Σ. As mentioned above, the basic methods are indeed sensitive to these parameters. But this sensitivity can be mitigated and tamed using techniques that are widely used in other applications and fields, robust optimization and regularization.

Robust optimization. Modifying an optimization-based method to make it more robust to data uncertainty is done in many fields, using techniques that have differing names. When optimization is used in almost any application, some of the data are not known exactly, and solving the optimization problem without recognizing this uncertainty, for example by using some kind of mean or typical values of the parameters, can lead to very poor practical performance. *Robust optimization* is a subfield of optimization that develops methods to handle or mitigate the adverse effects of parameter uncertainty; see, *e.g.*, [24, 296, 113, 25, 29, 196]. These methods tend to fall in one of two approaches: statistical or worst-case deterministic. In a statistical model, the uncertain parameters are modeled as random variables and the goal is to optimize the expected value of the objective under this distribution, leading to a *stochastic optimization problem* [277], [49, Chap. 6.4.1]. A worst-case deterministic uncertainty model posits a set of possible values for parameters, and the goal is to optimize the worst-case value of the objective over the possible parameter values [30], [49, Chap. 6.4.2]. Another name for worst-case robust optimization is *adversarial optimization*, since we can model the problem as us choosing values for the variables to obtain the best objective, after which an adversary chooses the values of the parameters so as to achieve the worst possible objective. Worst-case robust optimization has many variations and goes by many names. For example when the set of possible parameter values is finite, they are called *scenarios* or *regimes*, and optimizing for the worst-case scenario is called *worst-case scenario optimization*. While these general approaches sound quite different, they often lead to very similar solutions, and both can work well in applications. Robust optimization methods work by modifying the objective or constraints to model the possible variation in the data.

One very successful application of robust optimization is in *robust control*, where a control system is designed so that the control performance is not too sensitive to changes in the system dynamics [319, 172]. So-called linear quadratic optimal control was developed around 1960, and used in many applications. Its occasional sensitivity to the data (in this case, the dynamic model of the system being controlled) was noted then; by the early 1990s robust control methods were developed, and are now very widely used.

Regularization. Regularization is another term for methods that modify an optimization problem to mitigate sensitivity to data. It is almost universally used in statistics and machine learning when fitting models to data. Here we fit the parameters of a model to some given training data, accounting for the fact that the training data set could have been different [293, 141]. This process of regularization can be done explicitly by adding a penalty term to the objective, and also implicitly by adding constraints to the problem that prevent extreme outcomes. Regularization can often be interpreted as a form of robust optimization; see, *e.g.*, [49, Chap. 6.3–6.4]. Markowitz explicitly advocates the use of constraints in portfolio optimization [155, Foreword].

The high level story. Robust optimization and regularization both follow the same high level story, and both can be applied to the Markowitz problem. The story starts with a basic optimization-based method that relies on data that are not known precisely. We then modify the optimization problem, often by adding additional objective terms or constraints. Doing this *worsens* the in-sample performance. But if done well, it *improves* out-of-sample performance. Roughly speaking, robustification and regularization tell the optimizer to not fully trust the data, and this serves it well out-of-sample.

In portfolio construction a long-only constraint can be interpreted as a form of regularization [158]. A less extreme version is to add a leverage aversion cost term [156] or impose a leverage limit [210], which can help avoid many of the data sensitivity issues. We will describe below some effective and simple robustification methods for portfolio construction.

Regularization can (and should) also be applied to the forecasting of the mean and covariance in Markowitz portfolio construction. The Black-Litterman approach to estimating the mean returns regularizes the estimate toward the market implied return [33]. A return covariance estimate can be regularized using *shrinkage*, another term for regularized estimation in statistics [186].

3.1.4 Convex optimization

Over the same 70-year period since Markowitz's original work, there has been a parallel advance in mathematical optimization, and especially convex optimization, not to mention stunning increases in available computer power. Roughly speaking, convex optimization problems are mathematical optimization problems that satisfy certain mathematical properties. They can be solved reliably and efficiently, even when they involve a very large number of variables and constraints, and involve nonlinear, even nondifferentiable, functions [49].

Shortly before Markowitz published his paper on portfolio selection, George Dantzig developed the simplex method [72], which allowed for the efficient solution of linear programs. In 1959, Wolfe [307] extended the simplex method to QP problems, citing Markowitz's work as a motivating application. This close connection between portfolio construction and optimization was no coincidence, since Dantzig and Markowitz were colleagues at RAND.

Since then, the field of convex optimization has grown tremendously. Today, convex optimization is a mature field with a large body of theory, algorithms, software, and applications [49]. Being able to solve optimization problems reliably and efficiently is crucial for portfolio construction, especially for back-testing or simulating a proposed method on historical or synthesized data, where portfolio construction has to be carried many times. Thus, any extension of the Markowitz objective or additional constraints should be convex to ensure tractability. As we will see, this is hardly a limitation in practice.

Solvers. The dominant convex optimization problem form is now the *cone program*, a generalization of linear programming that handles nonlinear objective terms and constraints [242, 49, 197, 300]. There are now a number of reliable and efficient solvers for such problems, including open-source ones like ECOS [85], Clarabel [124], and SCS [243], and commercial solvers such as MOSEK [9], GUROBI [136], and CPLEX [70]. A recent open-source solver for QPs is OSQP [286].

Domain-specific languages. Convex optimization is also now very accessible to practitioners, even those without a strong background in the mathematics or algorithms of convex optimization, thanks to high-level domain-specific languages (DSLs) for convex optimization, such as CVXPY [2, 82], CVX [128], Convex.jl [297], CVXR [112], and YALMIP [200]. These DSLs make it easy to specify complex, but convex, optimization problems in a natural, human readable way. The DSLs transform the problem from the human readable form to a lower level form (often a cone program) suitable for a solver. These DSLs make it easy to develop convex optimization based methods, as well as to modify, update, and maintain existing ones. As a result, CVXPY is used at many quantitative hedge funds today, as well as in many other applications and industries. The proposed extension of Markowitz's portfolio construction method that we describe below is a good example of the use of CVXPY. It is a complex problem involving nonlinear and nondifferentiable functions, but its specification in CVXPY takes only a few tens of lines of clear readable code, given in appendix A.2. The overhead of translating the human readable problem specification into a cone program is typically small. Additionally, in some DSLs, such as CVXPY, problems can be parametrized [1], such that they can be solved for a range of values of the parameters, making the translation overhead negligible. Related to DSLs are modeling layers provided by some solver, such as MOSEK's Fusion API [9], which provides a high-level interface to the solver. Less focused on convex optimization, there are other modeling languages such as JuMP [202] and Pyomo [140, 60] that do not verify convexity, but provide flexibility in modeling a wide range of optimization problems, including nonconvex ones.

Code generators. Code generators like CVXGEN [223] and CVXPYgen [273] are similar to DSLs. They support high level specification of a problem (family) but instead of directly solving the problem, they generate custom low level code (typically C) for the problem that is specified. This code can be compiled to a very fast and totally reliable solver, suitable for embedded real-time applications. For example, CVXGEN-generated code guides all of SpaceX's Falcon 9 and Falcon Heavy first stages to their landings [35].

3.1.5 Previous work

The literature on portfolio construction is vast, and focusing on the practical implementation of Markowitz's ideas, we do not attempt to survey it here in detail. Instead, we highlight only a few major developments that are relevant to our work. For a detailed overview see, *e.g.*, [130, Chap. 14], [239, Chap. 6], and [69, 176, 63].

Building on Markowitz's framework, the field of portfolio construction has undergone substantial evolution. Notable contributions include Sharpe's Capital Asset Pricing Model [279] and the Black-Litterman model [33]. A pivotal figure in bringing the field to the forefront of the industry was Barr Rosenberg, whose research evolved to become the Barra risk model [267, 281], first used for risk modeling and then in portfolio optimization. The introduction of risk parity models [206] brought a focus on risk distribution. Additionally, hierarchical risk parity, a recent advancement, offers a more intricate approach to risk allocation, considering the hierarchical structure of asset correlations [78]. These developments reflect the field's dynamic adaptation to evolving market conditions and analytical techniques.

Software. Dedicated software helped practitioners access the solvers and DSLs mentioned earlier, and has facilitated the wide acceptance of Markowitz portfolio construction. A wealth of software packages have been developed for portfolio optimization, many (if not most) with Python interfaces, both open-source and commercial. Examples range from simple web-based visualization tools to complex trading platforms. Here we mention only a few of these software implementations.

On the simpler end Portfolio Visualizer [121] is a web-based tool that allows users to back-test and visualize various portfolio strategies. PyPortfolioOpt [220] and Cvxportfolio [43] are Python packages offering various portfolio optimization techniques. PyPortfolioOpt includes mean-variance optimization, Black-Litterman allocation [33], and more recent alternatives like the Hierarchical Risk Parity algorithm [78], while Cvxportfolio [43] supports multi-period strategies. The skfolio [80] package offers similar functionality to PyPortfolioOpt, with a focus on interoperability with the scikit-learn [251] machine learning library. Another Python implementation is proposed in [271], where the authors introduce an approach to multicriteria portfolio optimization. Quantlib [291] is an alternate open-source software package for modeling, trading, and risk management. Implementations of the critical line algorithm [217, 216, 218], which finds the entire efficient frontier of the Markowitz problem, have also been developed [216, 15] and extended [157].

The list of commercial software is also extensive. MATLAB's Financial Toolbox [54, 222] includes functions for mathematical modeling and statistical analysis of financial data, including portfolio optimization. Another example is Axioma, which on top of its popular risk model offers a portfolio optimizer [264].

Other software packages include Portfolio123 [260], PortfoliosLab [261], and PortfolioLab by Hudson & Thames [290]. Additionally, many solvers, such as MOSEK [9, 10], provide extensive examples of portfolio optimization problems, making them easy to use for portfolio optimization.

3.1.6 This chapter

Our goal is to describe an extension of the basic Markowitz portfolio construction method that includes a number of additional objective terms and constraints that reflect practical issues and address the issue of sensitivity to inevitable forecasting errors. We give a minimal formulation that is both simple and practical; we make no attempt to list all possible extensions that a portfolio manager (PM) might wish to add.

While the resulting optimization problem might appear complex, containing nonlinear nondifferentiable functions, it is convex, which means it can be solved reliably and efficiently. It can also be specified in a DSL such as CVXPY in just a few tens of lines of clear simple code. We can solve even large instances of the optimization problem very quickly, making it practical to carry out extensive back-testing to predict performance or adjust parameter values. One additional advantage of our formulation is that parameters that need to be specified are generally more interpretable than those appearing in basic formulations. For example a PM specifies a target risk and a target turnover instead of some parameters that are less directly related to them.

Most of the material in this chapter is not new but scattered across many sources, in different formats, and indeed in different application fields. Some of our recommendations are widely accepted and industry standard, but others are rarely discussed in the literature and even less commonly used in practice.

The authors bring a diverse set of backgrounds to this chapter. Some of us have applied Markowitz portfolio construction day-to-day in research, writing, and real portfolios. Others approach Markowitz's method from the perspective of optimization and control in engineering. Control systems engineering has a long history and is widely applied in essentially all engineering applications. Most applications of control engineering use methods based on models that are either wrong or heavily simplified. While naïve implementations of these methods do not work well (or worse), simple sensible modifications, similar to the ones we describe later in this chapter, work very well in practice.

These different backgrounds together can provide a new perspective and bring modern tools to the endeavor Markowitz began. These techniques have made Markowitz's method even more applicable and useful to investors.

Software. We have created two companion software packages. One is designed for pedagogical purposes, uses limited parameter testing and checking, and very closely follows the terminology and notation of the paper. It is available at

`https://github.com/cvxgrp/markowitz-reference`.

The second package is a robust and flexible implementation, which is better suited for practical use. It is available at

`https://github.com/cvxgrp/cvxmarkowitz`.

Outline. In §3.2 we set up our notation, define weights and trades, and describe various objective terms and constraints. Return and risk forecasts are covered in §3.3. In §3.4 we pull together the material of the previous two sections to define the (generalized) Markowitz trading problem, which we refer to as Markowitz++. In §3.5 we present some simple numerical experiments that illustrate how the extra terms robustify the basic Markowitz trading policy, and how parameters are tuned via back-testing to improve good performance.

3.2 Portfolio holdings and trades

This section introduces the notation and terminology for portfolio holdings, weights, and trades, fundamental objects in portfolio construction independent of the trading strategy. We follow the notation of [43], with the exceptions of handling the cash weight separately and dropping the time period subscript.

3.2.1 Portfolio weights

Universe. We consider a portfolio consisting of investments (possibly short) in n assets, plus a cash account. We refer to the set of assets we might hold as the *universe* of assets, and n as the size of the universe. These assets are assumed to be reasonably liquid, and could include, for example, stocks, bonds, or currencies.

Asset and cash weights. To describe the portfolio investments, we work with the *weights* or fractions of the total portfolio value for each asset, with negative values indicating short positions. We denote the weights for the assets as w_i, $i = 1, \ldots, n$, and collect them into a portfolio weight vector $w = (w_1, \ldots, w_n) \in \mathbf{R}^n$. The weights are readily interpreted: $w_i = 0.05$ means that 5% of the total portfolio value is held in asset i, and $w_k = -0.01$ means that we hold a short position in asset k, with value 1% of the total portfolio value. The dollar value of asset i held is Vw_i, where V is the total portfolio value, assumed to be positive.

We denote the weight for the cash account, *i.e.*, our cash value divided by the portfolio value, as c. If c is negative, it represents a loan. When $c > 0$ we say the portfolio is *diluted* with cash; when $c < 0$, the portfolio is *margined*. The dollar value of the cash account is Vc.

By definition the weights sum to one, so we have

$$\mathbf{1}^T w + c = 1, \tag{3.3}$$

where $\mathbf{1}$ is the vector with all entries one. The first term, $\mathbf{1}^T w$, is the total weight on the non-cash assets, and we refer to it as the total asset weight. The cash weight is one minus the total asset weight, *i.e.*, $c = 1 - \mathbf{1}^T w$.

Several portfolio types can be expressed in terms of the holdings. A *long-only* portfolio is one with all asset weights nonnegative, *i.e.*, $w \geq 0$ (elementwise). A portfolio with $c = 0$, *i.e.*, no cash holdings, is called *fully invested*. In such a portfolio we have $\mathbf{1}^T w = 1$, *i.e.*, the total asset weight is one. As another example, a *cash-neutral* portfolio is one with $c = 1$. For a cash-neutral portfolio we have $\mathbf{1}^T w = 0$, *i.e.*, the total (net) asset weight is zero.

Leverage. The *leverage* of the portfolio, denoted L, is

$$L = \sum_{i=1}^{n} |w_i| = \|w\|_1.$$

(Several other closely related definitions are also used. Our definition is commonly referred to as the *gross leverage* [8].) The leverage does not include the cash account.

In a long-only portfolio, the leverage is equal to the total asset weight. The *130-30 portfolio* [189] refers to a fully invested portfolio with leverage $L = 1.6$. For such a portfolio, the total weight of the short positions (*i.e.*, negative w_i) is -0.3 and the total weight of the long positions (*i.e.*, positive w_i) is 1.3.

Benchmark and active weights. In some cases our focus is on portfolio performance relative to a *benchmark portfolio*. We let $w^{\mathrm{b}} \in \mathbf{R}^n$ denote the weights of the benchmark. Typically the benchmark does not include any cash weight, so $\mathbf{1}^T w^{\mathrm{b}} = 1$. We refer to $w - w^{\mathrm{b}}$ as the *active weights* of our portfolio. A positive active weight on asset i, *i.e.*, $w_i - w_i^{\mathrm{b}} > 0$, means our portfolio is *over-weight* (relative to the benchmark) on asset i; a negative active weight, $w_i - w_i^{\mathrm{b}} < 0$, means our portfolio is *under-weight* on asset i.

3.2.2 Holding constraints and costs

Several constraints and costs are associated with the portfolio holdings w and c.

Weight limits. Asset and cash *weight limits* have the form

$$w_i^{\min} \leq w_i \leq w_i^{\max}, \quad i = 1, \ldots, n, \quad c^{\min} \leq c \leq c^{\max},$$

where $w^{\min}$ and $w^{\max}$ are given vectors of lower and upper limits on asset weights, and $c^{\min}$ and $c^{\max}$ are given lower and upper limits on the cash weight. We write the asset weight inequalities in vector form as $w^{\min} \leq w \leq w^{\max}$. We have already encountered a simple example: a long-only portfolio has $w^{\min} = 0$.

Portfolio weight limits can reflect hard requirements, for example that a portfolio must (by legal or regulatory requirements) be long-only. Portfolio weight limits can also be used to avoid excessive concentration of a portfolio, or limit short positions. For example, $w^{\max} = 0.15$ means that our portfolio cannot hold more than 15% of the total portfolio value in any one asset. (Here we adopt the convention that in a vector-scalar inequality, the scalar is implicitly multiplied by $\mathbf{1}$.) As another example, $w^{\min} = -0.05$ means that the short position in any asset can never exceed 5% of the total portfolio value. For large portfolios it is reasonable to also limit holdings relative to the asset capitalization, *e.g.*, to require that our portfolio holdings of each asset are no more than 10% of the asset capitalization.

Weight limits can also be used to capture the portfolio manager's views on how the market will evolve. For example, she might insist on a long position for some assets, and a short position for some others.

When a benchmark is used, we can impose limits on active weights. For example $|w - w^{\mathrm{b}}| \leq 0.10$ means that no asset in the portfolio can be more than 10% over-weight or under-weight.

Leverage limit. In addition to weight limits, we can impose a leverage limit,

$$L \leq L^{\mathrm{tar}}, \tag{3.4}$$

where L^{tar} is a specified maximum or target leverage value. (Other authors have suggested including leverage as a penalty term in the objective, to model leverage aversion and identify the optimal amount of leverage in the presence of leverage aversion [156].)

Holding costs. In general a fee is paid to borrow an asset in order to enter a short position. Analogously we pay a borrow cost fee for a negative cash weight. We will assume these holding costs are a linear function of the negative weights, *i.e.*, of the form

$$\phi^{\mathrm{hold}}(w, c) = (\kappa^{\mathrm{short}})^T (-w)_+ + \kappa^{\mathrm{borrow}} (-c)_+, \tag{3.5}$$

where $(a)_+ = \max\{a, 0\}$ denotes the nonnegative part, applied elementwise and in its first use above. Here $\kappa^{\mathrm{short}} \geq 0$ is the vector of borrow cost (rates) for the assets, and $\kappa^{\mathrm{borrow}} \geq 0$ is the borrow cost for cash.

Other holding constraints. There are many other constraints on weights that might be imposed, some convex, and others not. A *concentration limit* is an example of a useful constraint that is convex. It states that the sum of the K largest absolute weights cannot exceed some limit. As a specific example, we can require that no collection of five assets can have a total absolute weight of more than 30% [257, 10]. A *minimum nonzero holding* constraint is an example of a commonly imposed nonconvex constraint. It states that any nonzero weight must have an absolute value exceeding some given minimum, such as 0.5%. (This one is easily handled using a heuristic based on convex optimization; see §3.4.3.)

3.2.3 Trades

Trade vector. We let w^{pre} and c^{pre} denote the pre-trade portfolio weights, *i.e.*, the portfolio weights before we carry out the trades to construct the portfolio given by w and c. We need the pre-trade weights to account for transaction costs. We refer to

$$z = w - w^{\text{pre}}, \tag{3.6}$$

the current weights minus the previous ones, as the (vector of) *trades* or the *trade list*. These trades have a simple interpretation: $z_i = 0.01$ means we buy an amount of asset i equal in value to 1% of our total portfolio value, and $z_i = -0.03$ means we sell an amount of asset i equal to 3% of the portfolio value.

Since $\mathbf{1}^T w^{\text{pre}} + c^{\text{pre}} = 1$, we have

$$c = c^{\text{pre}} - \mathbf{1}^T z, \tag{3.7}$$

i.e., the post-trade cash weight is the pre-trade cash weight minus the net weight of the trades. This does not include holding and transaction costs, discussed below.

Turnover. The quantity

$$T = \frac{1}{2}\sum_{i=1}^{n} |z_i| = \frac{1}{2}\|z\|_1$$

is the *turnover*. Here too, several other different but closely related definitions are also used, for example the minimum of the total weight bought and the total weight sold [130, Chap. 16]. A turnover $T = 0.01$ means that the average of total amount bought and total amount sold is 1% of the total portfolio value. The turnover is often annualized, by multiplying by the number of trading periods per year.

3.2.4 Trading constraints and costs

We typically have constraints on the trade vector z, as well as a trading cost that depends on z.

Trade limits. Trade limits impose lower and upper bounds on trades, as

$$z^{\min} \leq z \leq z^{\max},$$

where $z^{\min}$ and $z^{\max}$ are given limits. These trade limits can be used to limit market participation, defined as the ratio of the magnitude of each trade to the trading volume, using, *e.g.*,

$$|z| \leq 0.05v, \tag{3.8}$$

where $v \in \mathbf{R}^n$ is the trading volumes of the assets, expressed as multiples of the portfolio value. This constraint limits our participation for each asset to be less than 5%. (It corresponds to trade limits $z^{\max} = -z^{\min} = 0.05v$.) Since the trading volumes are not known when z is chosen, we use a forecast instead of the realized trading volumes.

Turnover limit. In addition to trade limits, we can limit the turnover as

$$T \leq T^{\text{tar}}, \tag{3.9}$$

where T^{tar} is a specified turnover limit.

Trading cost. Trading cost refers to the cost of carrying out a trade. For example, if we buy a small quantity of an asset, we pay the ask price, while if we sell an asset, we receive the bid price. Since the nominal price of an asset is the midpoint between the ask and bid prices, we can think of buying or selling the asset as doing so at the nominal price, plus an additional positive cost that is the trade amount times one-half the bid-ask spread. This *bid-ask spread transaction cost* has the form

$$\sum_{i=1}^{n} \kappa_i^{\text{spread}} |z_i| = (\kappa^{\text{spread}})^T |z|,$$

where $\kappa^{\text{spread}} \in \mathbf{R}^n$ is the vector of one-half the asset bid-ask spreads (which are all positive). This is the transaction cost expressed as a fraction of the portfolio value. For small trades this is a reasonable approximation of transaction cost.

For larger trades we 'eat through' the order book. To buy a quantity of an asset, we buy each ask lot, in order from lowest price, until we fill our order. An analogous situation occurs when selling. This means that we end up paying more per share than the ask price when buying, or receiving less than the bid price when selling. This phenomenon is called *market impact*.

A useful approximation of transaction cost that takes market impact into account is

$$\phi^{\text{trade}}(z) = (\kappa^{\text{spread}})^T |z| + (\kappa^{\text{impact}})^T |z|^{3/2}, \tag{3.10}$$

where the first term is the bid-ask spread component of transaction cost, and the second models the market impact, *i.e.*, the additional cost incurred as the trade eats through the order book. The vector κ^{impact} has positive entries and typically takes the form

$$\kappa_i^{\text{impact}} = a s_i v_i^{-1/2},$$

where s_i is the volatility of asset i over the trading period, v_i is the volume of market trading, expressed as a multiple of the portfolio value, and a is a constant on the order of one; see [130, 199, 295, 221] and [43, §2.3]. Evidently the transaction cost increases with volatility, and decreases with market volume. Several other approximations of transaction cost are used [6, 94].

Liquidation cost. Suppose we liquidate the portfolio, *i.e.*, close out all asset positions, which corresponds to the trade vector $z = -w$. The *liquidation cost* is

$$\phi^{\text{trade}}(-w) = (\kappa^{\text{spread}})^T|w| + (\kappa^{\text{impact}})^T|w|^{3/2}.$$

If the liquidation is carried out over multiple periods, the bid-ask term stays the same, but the market impact term decreases. For this reason a common approximation of the liquidation cost ignores the market impact term. A liquidation cost constraint has the form

$$(\kappa^{\text{spread}})^T|w| \leq \ell^{\max}, \tag{3.11}$$

where $\ell^{\max}$ is a maximum allowable liquidation cost, such as 1%. This is a weight constraint; it limits our holdings in less liquid assets, which have higher bid-ask spreads. It can be interpreted as a liquidity-weighted leverage (taking the bid-ask spread as a proxy for liquidity). When all assets have the same bid-ask spread, the liquidation constraint reduces to a leverage constraint. For example with all bid-ask spreads equal to 0.001 (*i.e.*, 10 basis points or bps) and a maximum liquidation cost $\ell^{\max} = 0.01$ (*i.e.*, 1% of the total portfolio value), the liquidation cost limit (3.11) reduces to a leverage limit (3.4) with $L^{\text{tar}} = 10$. The connection between liquidity and leverage is explored further in [88].

Transaction cost forecasts. When the trades z are chosen, we do not know the bid-ask spreads, the volatilities, or the volumes. Instead we use forecasts of these quantities in (3.8), (3.10), and (3.11). Simple forecasts, such as a trailing average or median of realized values, are typically used. More sophisticated forecasts take can into account calendar effects such as seasonality, or the typically low trading volume the day after Thanksgiving.

3.3 Return and risk forecasts

3.3.1 Return

Gross portfolio return. We let r_i denote the return, adjusted for dividends, splits, and other corporate actions, of asset i over the investment period. We collect these asset returns into a return vector $r = (r_1, \ldots, r_n) \in \mathbf{R}^n$. The portfolio return from asset i is $r_i w_i$. We let r^{rf} denote the risk-free interest rate, so the return in the cash account is $r^{\text{rf}} c$. The (gross) total portfolio return is then

$$R = r^T w + r^{\text{rf}} c.$$

This gross return does not include holding or trading costs. A closely related quantity is the *excess return*, the portfolio return minus the risk-free return, $R - r^{\rm rf} = r^T w + r^{\rm rf}(c-1)$.

Net portfolio return. The net portfolio return is the gross return minus the holding costs and transaction costs,

$$R^{\rm net} = R - \phi^{\rm hold}(w) - \phi^{\rm trade}(z). \tag{3.12}$$

Active return. The *active portfolio return* is the return relative to a benchmark portfolio,

$$r^T w + r^{\rm rf} c - r^T w^{\rm b} = r^T (w - w^{\rm b}) + r^{\rm rf} c.$$

If we subtract holding and trading costs we obtain the *net active portfolio return*.

Cash as slack. Since we do not know but only forecast the bid-ask spread, volatility, and volume, which appear in the transaction cost (3.10) (which is itself only an approximation) we should consider the post-trade cash c in (3.7) as an approximation that uses a forecast of holding and transaction costs, not the realized holding and transaction costs. We do not expect the realized post-trade cash weight to be exactly c.

3.3.2 Probabilistic asset return model

When we choose the trades z we do not know the asset returns r. Instead, we model r as a multivariate random variable with mean $\mu \in \mathbf{R}^n$ and covariance matrix $\Sigma \in \mathbf{S}^n_{++}$ (the set of symmetric positive definite $n \times n$ matrices),

$$\mathbf{E}\, r = \mu, \qquad \mathbf{E}(r-\mu)(r-\mu)^T = \Sigma.$$

The entries of the mean μ are often referred to as *trading signals* [154]. The asset return mean and covariance are forecasts, as described below. The asset return volatilities $s \in \mathbf{R}^n$ appearing in the transaction cost model (3.10) can be expressed as $s = \mathbf{diag}(\Sigma)^{1/2}$, where the squareroot is elementwise.

Expected return and risk. With this statistical model of r, the portfolio return R is a random variable with mean $\bar{R} = \mathbf{E}\, R$ and variance $\sigma^2 = \mathbf{var}\, R$ given by

$$\bar{R} = \mu^T w + r^{\rm rf} c, \qquad \sigma^2 = w^T \Sigma w.$$

The *risk* of the portfolio is defined as the standard deviation of the portfolio return, *i.e.*, σ.

Similarly, the active return $R^{\rm a}$ is a random variable with mean and variance

$$\bar{R}^{\rm a} = \mu^T (w - w^{\rm b}) + r^{\rm rf} c = \bar{R} - \mu^T w^{\rm b}, \qquad (\sigma^{\rm a})^2 = (w - w^{\rm b})^T \Sigma (w - w^{\rm b}),$$

and the *active risk* is σ^{a}. The risk and active risk are often given in annualized form, obtained by multiplying them by the squareroot of the number of periods per year.

The parameters μ and Σ are estimates or forecasts of the statistical model of asset returns, which is itself an approximation. For this reason the risk σ is called the *ex-ante* risk, to distinguish it from the standard deviation of the realized portfolio returns when trading, the *ex-post* risk. Similarly we refer to σ^{a} as the ex-ante active risk.

Optimizing expected return and risk. We have two objectives, high expected return and low risk. Perhaps the most common method for combining these objectives is to form a *risk-adjusted return*,

$$\bar{R} - \gamma\sigma^2,$$

where $\gamma > 0$ is the *risk aversion parameter*. Maximizing risk-adjusted return (possibly with other objective terms, and subject to constraints) gives the desired portfolio. Increasing γ gives us a portfolio with lower risk and also lower expected return. The risk aversion parameter allows us to explore the risk-return trade-off. This risk-adjusted return approach became popular in part because the resulting optimization problem is typically a quadratic program (QP), for which reliable solvers were developed even in the 1960s.

Another approach is to maximize expected return (possibly with other objective terms), subject to a *risk budget* or *risk target* constraint

$$\sigma \leq \sigma^{\mathrm{tar}}. \tag{3.13}$$

(The corresponding optimization problem is not a QP, but is readily handled by convex optimization solvers developed in the 1990s [198, 241, 288, 294].) This formulation seems more natural, since a portfolio manager will often have a target risk in her mind, *e.g.*, 8% annualized. This is the basic formulation that we recommend.

There are many other ways to combine expected return and risk. For example, we can maximize the return/risk ratio, called the *Sharpe ratio* (with no benchmark) or *information ratio* (with a benchmark). This problem too can be solved via convex optimization, at least when the constraints are simple [52].

3.3.3 Factor model

In practice, and especially for large universes, it is common to use a *factor model* for the returns. The factor return model, with k factors (typically with $k \ll n$), has the form

$$r = Ff + \epsilon, \tag{3.14}$$

where $F \in \mathbf{R}^{n \times k}$ is the *factor loading matrix*, $f \in \mathbf{R}^k$ is the vector of *factor returns*, and $\epsilon \in \mathbf{R}^n$ is the *idiosyncratic return*. The term Ff is interpreted as the component of asset returns explainable or predicted by the factor returns.

At portfolio construction time the factor loading matrix F is known, and the factor return f and idiosyncratic return ϵ are modeled as uncorrelated random variables with means and covariance matrices

$$\mathbf{E}\, f = \bar{f}, \qquad \mathbf{cov}\, f = \Sigma^{\mathrm{f}}, \qquad \mathbf{E}\, \epsilon = \bar{\epsilon}, \qquad \mathbf{cov}\, \epsilon = D,$$

where D is diagonal (with positive entries). The entries $\bar{\epsilon}$, the means of the idiosyncratic returns, are also referred to as the *alphas*, especially when there is only one factor which is the overall market return. They are the part of the asset returns not explained by the factor returns.

With the factor model (3.14) the asset return mean and covariance are

$$\mu = F\bar{f} + \bar{\epsilon}, \qquad \Sigma = F\Sigma^f F^T + D.$$

The return covariance matrix in a factor model has a special form, low rank plus diagonal. The portfolio return mean and variance are

$$\bar{R} = (F\bar{f})^T w + \bar{\epsilon}^T w + r^{\mathrm{rf}} c, \qquad \sigma^2 = (F^T w)^T \Sigma^{\mathrm{f}} (F^T w) + w^T D w.$$

The factor returns are constructed to have explanatory power for the returns of assets in our universe. For equities, they are typically the returns of other portfolios, such as the overall market (with weights proportional to capitalization), industries, and style portfolios like the celebrated Fama-French factors [101, 102]. For bonds, the factors are typically constructed from yield curves, interest rates, and spreads. These traditional factors are interpretable.

Factors can also can be constructed directly from previous realized asset returns using methods such as principal component analysis (PCA) [14, 13, 190, 191, 255, 254]. Aside from the first principal component, which typically is close to the market return, these factors are less interpretable.

Factor and idiosyncratic returns. A factor model gives an alternative method to specify the expected return as $\mu = F\bar{f} + \bar{\epsilon}$, where $\bar{f}$ is a forecast of the factor returns and $\bar{\epsilon}$ is a forecast of the idiosyncratic returns, *i.e.*, the asset alphas. One common method uses only a forecast of the factor returns, with $\bar{\epsilon} = 0$, so $\mu = F\bar{f}$. A complementary method assumes zero factor returns, so we have $\mu = \bar{\epsilon}$, *i.e.*, the mean asset returns are the same as the idiosyncratic asset mean returns.

Factor betas and neutrality. Under the factor model (3.14), the covariance of the portfolio return R with the factor returns f is the k-vector

$$\mathbf{cov}(R, f) = \Sigma^{\mathrm{f}} F^T w.$$

The *betas* of the portfolio with respect to the factors divide these covariances by the variance of the factors,

$$\beta = \mathbf{diag}(s^{\mathrm{f}})^{-2}\Sigma^{\mathrm{f}}F^T w,$$

where $s^{\mathrm{f}} = \mathbf{diag}(\Sigma^{\mathrm{f}})^{1/2}$ is the vector of factor return volatilities.

The constraint that our portfolio return is uncorrelated (or has zero beta) with the ith factor return f_i, under the factor model (3.14), is

$$\mathbf{cov}(R, f)_i = (\Sigma^f F^T w)_i = 0. \qquad (3.15)$$

This is referred to as *factor neutrality* (with respect to the ith factor). It is a simple linear equality constraint, which can be expressed as $a^T w = 0$, where a is the ith column of $F\Sigma^{\mathrm{f}}$. Factor neutrality constraints are typically used with active weights. In this case, factor neutrality means that the portfolio beta matches the benchmark beta for that factor. This also is a linear equality constraint that can be expressed as $\beta_i = \beta_i^{\mathrm{b}}$, with β^{b} the benchmark betas.

Advantages of a factor model. Especially with large universes, the factor model (specified by F, Σ^{f}, and D) can give a better estimate of the return covariance, compared to methods that directly estimate the $n \times n$ matrix Σ [163]. Another substantial advantage is computational. By exploiting the low-rank plus diagonal structure of the return covariance with a factor model, we can reduce the computational complexity of solving the Markowitz optimization problem from $O(n^3)$ flops (without exploiting the factor model) to $O(nk^2)$ flops (exploiting the factor form). These computational savings can be dramatic, *e.g.*, for a whole world portfolio with $n = 10000$ and $k = 100$, where we obtain a 10000 fold decrease in solve time; see §3.5.6.

3.3.4 Return and risk forecasts

Here we briefly discuss the forecasting of μ and Σ (or F, Σ^{f}, and D in a factor model). Markowitz himself did not address the question of estimating μ and Σ; when asked by practitioners how one should choose these forecasts, his reply was [276]

"*That's your job, not mine.*"

It is well documented that poor or naïve estimates of these, *e.g.*, the sample mean and covariance, can yield poor portfolio performance [228]. But even reasonable forecasts will have errors, which can degrade performance. We show some methods to mitigate this forecast uncertainty in §3.3.5.

Asset returns estimate. The expected returns vector μ is by far the most important parameter in the portfolio construction process, and methods for estimating it, or the factor and idiosyncratic return means are for obvious reasons in general proprietary. It is also the most challenging data to estimate. There is no consensus on how to estimate the mean returns, and the literature is vast.

Regularization methods can improve mean estimates. As an example, the Black-Litterman model [33] allows a portfolio manager to incorporate her views on how the expected returns differ from the market consensus, and in essence acts as a form of regularization of the portfolio toward the market portfolio. Another method that serves implicitly as regularization is winsorization, where the mean estimates are clipped when they go outside a specified range [305], [130, Chap. 14]. Yet another method is cross-sectionalization, where the preliminary estimate of returns μ is replaced with $\tilde{\mu}$, the same values monotonically mapped to (approximately) a Gaussian distribution [130, Chap. 14].

Return covariance estimate. There are many ways to estimate the covariance matrix, with or without a factor model. Approaches that work well in practice include the exponentially weighted moving average (EWMA) [245], dynamic conditional correlation (DCC) [92], and iterated EWMA [18]. For a detailed discussion on how to estimate a covariance matrix for financial return data, see [163] and the references therein.

3.3.5 Making return and risk forecasts robust

In this section we address methods to mitigate the impact of forecast errors in return and covariance estimation, which can lead to poor performance. This directly addresses one of the main criticisms of the Markowitz method, that it is too sensitive to estimation errors. Here, we briefly review how to address robust return mean and covariance estimation, and refer the reader to [43, §4.3] and [99, 296, 71] for more detailed discussions.

Robust return forecast. We model our uncertainty in the mean return vector by giving an interval of possible values for each return mean. We let $\mu \in \mathbf{R}^n$ denote our nominal estimate of the return means, and we take the nonnegative vector $\rho \in \mathbf{R}^n_+$ to describe the half-width or radius of the uncertainty intervals. Thus we imagine that the return can be any vector of the form $\mu + \delta$, where $|\delta_i| \leq \rho_i$. For example $\mu_i = -0.0010$ and $\rho_i = 0.0005$ means that the mean return for asset i lies in the range $[-15, -5]$ bps.

We define the *worst-case mean portfolio return* as the minimum possible mean portfolio return consistent with the given ranges of asset return means:

$$R^{\mathrm{wc}} = \min\{(\mu + \delta)^T w \mid |\delta| \leq \rho\}.$$

We can think of this as an adversarial game. The portfolio manager (PM) chooses the portfolio w, and an adversary then chooses the worst mean return consistent with the given uncertainty intervals. This second step has an obvious solution: We choose $\mu_i - \rho_i$ when $w_i \geq 0$, and we choose $\mu_i + \rho_i$ when $w_i < 0$. In words: For long positions the worst return is the minimum possible; for short positions the worst return is the maximum possible. With this observation, we obtain a simple formula for the worst-case portfolio mean return,

$$R^{\text{wc}} = \bar{R} - \rho^T |w|. \tag{3.16}$$

The first term is the nominal mean return; the second term, which is nonpositive, gives the degradation of return induced by the uncertainty. We refer to $\rho^T|w|$ as the portfolio *return forecast error penalty* in our return forecast. The return forecast error penalty has a nice interpretation as an uncertainty-weighted leverage.

When the portfolio is long-only, so $w \geq 0$, the worst-case asset returns are obvious: they simply take their minimum values, $\mu - \rho$. In this case the worst-case portfolio mean return (3.16) is the usual mean portfolio return, with each nominal asset return reduced by its uncertainty.

The return forecast uncertainties ρ can be chosen by several methods. One simple method is to set all entries the same, and equal to some quantile of the entries of $|\mu|$, such as the 20th percentile. A more sophisticated method relies on multiple forecasts of the returns, and sets μ as the mean or median forecast, and ρ as some measure of spread, such as standard deviation, of the forecasts.

Robust covariance forecast. We can also consider uncertainty in the covariance matrix. We let Σ denote our nominal estimate of the covariance matrix. We imagine that the covariance matrix has the form $\Sigma + \Delta$ where $\Delta \in \mathbf{S}^n$ (the set of symmetric $n \times n$ matrices) where the perturbation Δ satisfies

$$|\Delta_{ij}| \leq \varrho(\Sigma_{ii}\Sigma_{jj})^{1/2},$$

where $\varrho \in [0, 1)$ defines the level of uncertainty. For example, $\varrho = 0.04$ means that the diagonal elements of the covariance matrix can change by up to 4% (so the volatilites can change by around 2%), and the asset return correlations can change by up to around 4%. (You should not trust anyone who claims that his asset return covariance matrix estimate is more accurate than this.)

We define the *worst-case portfolio risk* as the maximum possible risk over covariance matrices consistent with our uncertainty set,

$$(\sigma^{\text{wc}})^2 = \max\{w^T(\Sigma + \Delta)w \mid |\Delta_{ij}| \leq \varrho(\Sigma_{ii}\Sigma_{jj})^{1/2}\}.$$

This can be expressed analytically as [43, §4.3]

$$(\sigma^{\text{wc}})^2 = \sigma^2 + \varrho\left(\sum_{i=1}^n \Sigma_{ii}^{1/2}|w_i|\right)^2. \tag{3.17}$$

The second term is the *covariance forecast error penalty*. It has a nice interpretation as an additive regularization term, the square of a volatility-weighted leverage. The worst-case risk can be expressed using Euclidean norms as

$$\sigma^{\mathrm{wc}} = \left\| \left(\sigma, \sqrt{\varrho}(\mathbf{diag}(\Sigma)^{1/2})^T |w_i| \right) \right\|_2. \tag{3.18}$$

When the portfolio is long-only, the worst-case risk (3.17) can be simplified. In this case, the worst-case risk is the risk using the covariance matrix $\Sigma + \varrho s s^T$, where $s = \mathbf{diag}(\Sigma)^{1/2}$ is vector of asset volatilities, under the nominal covariance.

3.4 Convex optimization formulation

3.4.1 Markowitz problem

In this section we assemble the objective terms and constraints described in §3.2 and §3.3 into one convex optimization problem. We obtain the Markowitz problem

$$\begin{array}{ll} \text{maximize} & R^{\mathrm{wc}} - \gamma^{\mathrm{hold}} \phi^{\mathrm{hold}}(w, c) - \gamma^{\mathrm{trade}} \phi^{\mathrm{trade}}(z) \\ \text{subject to} & \mathbf{1}^T w + c = 1, \quad z = w - w^{\mathrm{pre}}, \\ & w^{\min} \leq w \leq w^{\max}, \quad L \leq L^{\mathrm{tar}}, \quad c^{\min} \leq c \leq c^{\max}, \\ & z^{\min} \leq z \leq z^{\max}, \quad T \leq T^{\mathrm{tar}}, \\ & \sigma^{\mathrm{wc}} \leq \sigma^{\mathrm{tar}}, \end{array} \tag{3.19}$$

with variables $w \in \mathbf{R}^n$ and $c \in \mathbf{R}$, and positive parameters γ^{hold} and γ^{trade} that allow us to scale the holding and transaction costs, respectively. Despite the nonlinear and nondifferentiable functions appearing in the objective and constraints, this is a convex optimization problem, which can be very reliably and efficiently solved. We can add other convex objective terms to this problem, such as factor neutrality or liquidation cost limit, or work with active risk and return with a benchmark.

The objective is our forecast of the (robustified, worst-case) net portfolio return, with the holding and transaction costs scaled by the parameters γ^{hold} and γ^{trade}, respectively. The first line of constraints relate the pre-trade portfolio, which is given, and the post-trade portfolio, which is to be chosen. The second line of constraints are weight limits, and the third line contains the trading constraints. The last line of constraints is the (robustified, worst-case) risk limit.

Data. We divide the constants that need to be specified in the problem (3.19) into two groups, *data* and *parameters*, although the distinction is not sharp. Data are quantities we *observe* (such as the previous portfolio weights) or forecast (such as return means, market volumes, or bid-ask spreads):

- *Pre-trade portfolio weights* w^{pre} and c^{pre}.
- *Asset return forecast* μ.
- *Risk model* Σ, or for a factor model, F, Σ^{f}, and D.

- *Holding cost parameters* κ^{short} and κ^{borrow}.
- *Trading cost parameters* κ^{spread} and κ^{impact} (which in turn depend on the forecast bid-ask spreads, asset volatilities, and market volume).

Parameters. Parameters are quantities that we *choose* in order to obtain good investment performance, or to reflect portfolio manager preferences, or to comply with legal requirements or regulations. These are

- *Target risk* σ^{tar}.
- *Holding and trading scale factors* γ^{hold} and γ^{trade}.
- *Weight and leverage limits* $w^{\min}$, $w^{\max}$, $c^{\min}$, $c^{\max}$, and L^{tar}.
- *Trade and turnover limits* $z^{\min}$, $z^{\max}$, and T^{tar}.
- *Mean and covariance forecast uncertainties* ρ and ϱ.

We list the mean and covariance forecast uncertainties as parameters since they are closer to being chosen than measured or estimated. When the mean return uncertainties are chosen as described above from a collection of return forecasts, they would be closer to data.

Initial default choices for parameters. The target risk, and the weight, leverage, trade, and turnover limits are interpretable and can be assigned reasonable values by the PM. The return and risk uncertainty parameters ρ and ϱ can be chosen as described above. The hold and trade scale factors can be chosen to be around one.

To improve performance the PM will want to adjust or tune these parameter values around their natural or default values, as discussed in §3.5.4.

3.4.2 Softening constraints

The Markowitz problem (3.19) includes a number of constraints. This can present two challenges in practice. First, it can lead to substantial trading, for example to satisfy our leverage or ex-ante worst-case risk limits, even when they would have been violated only slightly, which can lead to poor performance due to excessive trading. Second, the problem can be infeasible, meaning there is no choice of the variables that satisfy all the constraints. This can complicate back-tests or simulations, as well as running the trading policy in production, where such infeasiblilities naturally occur most frequently during periods of market stress.

Soft constraints. Here we explain a standard method in optimization, in which some of the constraints can be softened, which means we allow them to be somewhat violated, if needed. In optimization, softness refers to how much we care about different values of an objective. We can think of the objective as infinitely soft: We will accept any objective value, but we prefer larger values (if we are maximizing). We can think of constraints as infinitely hard: We will not accept any violation of them, even if it is only by a small amount. *Soft constraints*, described below, are in between. They should normally act as constraints, but when needed, they can be violated. When a soft constraint is violated, and by how much, depends on our priorities, with high priority meaning that the constraint should be violated only when absolutely necessary.

Consider a (hard) constraint such as $f \leq f^{\max}$. This means that we will not accept any choice of the variables for which $f > f^{\max}$. To make it a *soft constraint*, we remove the constraint from the problem and form a penalty term

$$\gamma(f - f^{\max})_+$$

which we subtract from the objective, when we are maximizing. The number $(f - f^{\max})_+$ is the *violation* of the original constraint $f \leq f^{\max}$. The positive parameter γ is called the *priority parameter* associated with the softened constraint. In this context, we refer to the parameter $f^{\max}$ as a *target* for the value of f, not a limit. With the softened constraint, we can accept variable choices for which $f > f^{\max}$, but the optimizer tries to avoid this given the penalty paid (in the objective) when this occurs. Softening constraints preserves convexity of a problem.

Markowitz problem with soft constraints. A number of constraints in (3.19) should be left as (hard) constraints. These include the constraints relating the proposed and previous weights, *i.e.*, the first line of constraints in (3.19). When the portfolio is long-only, the constraint $w \geq 0$ should be left as a hard constraint, and similarly for a constraint such as $c \geq 0$, *i.e.*, that we do not borrow cash. When a leverage limit is strict or imposed by a mandate, it should be left as a hard constraint; when it is imposed by the portfolio manager to improve performance, or more likely, to help her avoid poor outcomes, it can be softened.

The other constraints in (3.19) are candidates for softening. Weight and trade limits, including leverage and turnover limits, should be softened (except in the cases described above). The worst-case risk limit $\sigma \leq \sigma^{\text{tar}}$ should be softened, with a risk penalty term

$$\gamma^{\text{risk}}(\sigma - \sigma^{\text{tar}})_+$$

subtracted from the objective. When the associated priority parameter γ^{risk} is chosen appropriately, this allows us to occasionally violate our risk limit a bit when the violation is small. We refer to the softened Markowitz problem as the Markowitz++ problem.

One nice attribute of the Markowitz++ problem is that it is always feasible; the choice $z = 0$, *i.e.*, no trading, is always feasible, even when it is a poor choice. This means that the softened Markowitz problem can be used to define a trading policy that runs with little or no human intervention (with, however, any soft constraints that exceed their targets reported to the portfolio manager).

Priority parameters. When we soften the worst-case risk, leverage, and turnover constraints, we gain several more parameters,

$$\gamma^{\text{risk}}, \quad \gamma^{\text{lev}}, \quad \gamma^{\text{turn}}.$$

Evidently the larger each of these priority parameters is, the more reluctant the optimizer is to violate it. (Here we anthropomorphize the optimization problem solver.) When the priority parameters are large, the associated soft constraints are effectively hard. Beyond these observations, however, it is hard to know what values should be used.

Choosing priority parameters. Here we describe a simple method to obtain reasonable useful initial values for the priority parameters associated with softened constraints. Our method is based on Lagrange multipliers or dual variables. Suppose we solve a problem with hard constraints, and obtain optimal Lagrange multipliers for each of the constraints. If we use these Lagrange multipliers as priorities in a softened version of the problem, all the original constraints will be satisfied. Roughly speaking, the Lagrange multipliers give us values of priorities for which the soft constraints are effectively hard. We would want to use priority values a bit smaller, so that the original constraints can occasionally be violated.

Now we describe the method in detail. We start by solving multiple instances of the problem with hard constraints, for example in a back-test, recording the values of the Lagrange multipliers for each problem instance (when the problem is feasible). We then set the priority parameters to some quantile, such as the 80th percentile, of the Lagrange multipliers. With this choice of priority parameters, we expect (very roughly) the original constraints to hold around 80% of the time. For hard constraints that are only occasionally tight, another method for choosing the priority parameters is as a fraction of the maximum Lagrange multiplier observed.

Using this method we can obtain reasonable starting values of the priority parameters. The final choice of priority parameters is done by back-testing and parameter tuning, starting from these reasonable values, as discussed in §3.4.4.

3.4.3 Nonconvex constraints and objectives

All objective terms and constraints discussed so far are convex, and the Markowitz problem (3.19), and its softened version, are convex optimization problems. They can be reliably and efficiently solved.

Some other constraints and objective terms are not convex. The most obvious one is that the trades must ultimately involve an integer number of shares. As a few other practical examples, we might limit the number of nonzero weights, or insist on a minimum nonzero weight absolute value. When these constraints are added to (3.19), the problem becomes nonconvex. Great advances have been made in solvers that handle so-called mixed-integer convex problems [180], and these can be used to solve these portfolio construction problems. The disadvantage is longer solve time, compared to a similar convex problem, and sometimes, dramatically longer solve time if we insist on solving the problem to global optimality. A convex portfolio construction problem that can be solved in a small fraction of a second can take many seconds, or even minutes or more, to solve when nonconvex constraints are added.

For production, where the problem is solved daily, or even hourly, this is fine. The slowdown incurred with nonconvex optimization is however very bad for back-testing and validation, where many thousands, or hundreds of thousands, of portfolio construction problems are to be solved. One sensible approach is to carry out back-testing using a convex formulation, so as to retain the speed and reliability of a convex optimization, and run a nonconvex version in production. As a variant on this, back-tests using convex optimization can be used for parameter search, and one final back-test with a nonconvex formulation can be used to be sure the results are close. Running backtests using only convex constraints works because the nonconvex constraints typically only have a small impact on the portfolio and its performance.

Heuristics based on convex optimization. Essentially all solvers for nonconvex problems that attempt to find a global solution rely on convex optimization under the hood [147]. The issue is that a very large number of convex optimization problems might need to be solved to find a global solution.

But many nonconvex constraints can be handled heuristically by solving just a few convex optimization problem. As a simple example we might simply round the numbers of shares in a trade list to an integer. This rounding should have little effect unless the portfolio value is very small.

Other nonconvex constraints are readily handled by heuristics that involve solving just a handful of convex problems. One general method is called relax-round-solve [83]. We illustrate this method to handle the constraint that the minimum nonzero weight absolute value is 0.001 (10 bps). First we solve the problem ignoring this constraint. If the weights satisfy the constraint, we are done (and the choice is optimal). If not, we set a threshold and divide the assets into those with absolute weight smaller than the threshold, those with weights larger than the threshold, and those which are less than minus the threshold. We then add constraints to the original problem, setting the weights to zero, more than 0.001, and less than -0.001, depending on the weights found in the first problem. These are convex constraints, and when we solve the second time we are guaranteed to satisfy the nonconvex constraint. We thus solve two convex problems. In the first one, we essentially decide which weights will be zero, which will be more than the minimum nonzero long weight, and which will be short more than the minimum. In the second one we adjust all the weights, ensuring that the minimum absolute nonzero weight constraint holds.

3.4.4 Back-testing and parameter tuning

Back-testing. Back-testing refers to simulating a trading strategy using historical data. To do this we provide the forecasts for all quantities needed, including the mean return and covariance, for Markowitz portfolio construction in each period. In each period these forecasts, together with the parameters, are sent in to the Markowitz portfolio construction method, which determines a set of trades. We then use the *realized* values of return, volatility, bid-ask spread, and market volume to compute the (simulated) realized net return R_t^{net}, where the subscript gives the time period. Note that while the Markowitz trading engine uses forecasts of various quantities, the simulation uses the realized historical values. This gives a reasonably realistic approximation of what the result would have been, had we actually carried out this trading. (It is still only an approximation, since it uses our particular trading cost model. Of course a more complex or realistic trading cost model could be used for simulation.) The back-test simulation can also include practical aspects like trading only an integer number of shares or blocks of shares. The simulation can also include external cash entering or leaving the portfolio, such as liabilities that must be paid each period.

In the simulation we log the trajectory of the portfolio. We can compute various quantities of interest such as the realized return, volatility, Sharpe or information ratio, turnover, and leverage, all potentially over multiple time periods such as quarters or years. We can determine the portfolio value versus periods, given by

$$V_t = V_1 \prod_{\tau=1}^{t-1} (1 + R_\tau^{\text{net}}),$$

where V_1 is the portfolio value at period $t = 1$. From this we can evaluate quantities like the average or maximum value of drawdown over quarters or years.

Variations. The idea of back-testing or simulating portfolio performance can be used for several other tasks. In one variation on a back-test called a *stress test*, we use historical data modified to be more challenging, *e.g.*, lower returns or higher costs than actually occurred.

Another variation called *performance forecasting* uses data that are simulated or generated, starting from the current portfolio out to some horizon like one year in the future, or the end of current fiscal year. We generate some number of possible future values of quantities such as returns, along with the corresponding forecasts of them, and simulate the performance for each of these. This gives us an idea of what we can expect our future performance to be, for example as a range of values or quantiles.

Yet another variation is a *retrospective what-if* simulation. Here we take a live portfolio and go back, say, three months. Starting from the portfolio holdings at that time, we simulate forward to the present, after making some changes to our trading method, *e.g.*, modifying some parameters. The fact that the current portfolio value would be higher (according to our simulation) if the PM had reduced the target risk three months ago is of course not directly actionable. But it still very useful information for the PM.

Parameter tuning. Perhaps the most important use of back-testing is to help the PM choose parameter values in the Markowitz portfolio construction problem. While some parameters, like the target risk, are given, others are less obvious. For example, how should we choose γ^{trade}? The default value of one is our best guess of what the single period transaction cost will be. But perhaps we get better performance with $\gamma^{\text{trade}} = 2$, which means, roughly speaking, that we are exaggerating trading cost by a factor of two. The result, of course, is a reduction in trading compared to the default value one. This will result in smaller realized transaction costs, but also, possibly, higher return, or smaller drawdown. The back-test will reveal what would happen in this case (to the limits of the back-test accuracy).

To choose among a set of choices for parameters, we carry out a back-test with each set, and evaluate multiple metrics, such as realized returns, volatility, and turnover. Our optimization problem contains target values for these, based on our forecasts and models; in a back-test we obtain the ex-post or realized values of these metrics.

To make a final choice of parameters, we must scalarize our metrics, *i.e.*, create one scalar metric from them, so we can choose among different sets of parameter values. For example we might choose to maximize Sharpe ratio, subject to other metrics being within specified bounds. Or we could form some kind of weighted combination of the individual metrics.

At the very minimum, a PM should always carry out back-tests in which all of her chosen parameters are, one by one, increased or decreased by, say, 20%. Even with 10 parameters, this requires only 20 back-tests. If any of these back-tests results in substantially improved performance, she will need to explain or defend her choices.

This simple method of changing one parameter at a time can be extended to carry out a crude but often effective parameter search. We cycle over the parameters, increasing or decreasing each and carrying out a back-test. When we find a new set of parameter values that has better performance than the current set of values, we take it as our new values. This continues until increasing or decreasing each parameter value does not improve performance.

Another traditional method of parameter tuning is *gridding*. We choose a small number of candidate values for each parameter, and then carry out a back-test for each combination, evaluating multiple performance metrics. Of course this is practical only when we are choosing just a few parameters, and we consider only a few candidate values for each one. Gridding is often carried out with a first crude parameter gridding, with the candidate values spaced by a factor of ten or so; then, when good values of these parameters are found, a more refined grid search is used to focus in on parameters near the good ones found in the first crude search. In any case there is no reason to find or specify parameter values very accurately; specifying them to even 10% is not needed. For one thing, the back-test itself is only an approximation. If a back-test reveals that $\gamma^{\text{trade}} = 2.1$ works well, but that $\gamma^{\text{trade}} = 1.9$ and $\gamma^{\text{trade}} = 2.3$ work poorly, it is very unlikely that our trading method will work well in practice. Similar to the way we want our trading policy to be robust to variations in the input data, we also want it to be robust to variation in the parameters.

More sophisticated parameter search methods can also be used. Many such methods build a statistical model of the good parameter values found so far, and obtain new values to try by sampling from the distribution; see, *e.g.*, [205] for more discussion. Another option is to obtain not just the value of some composite metric, but also its gradient with respect to the parameters. This very daunting computation can be carried out by automatic differentiation systems that can differentiate through the solution of a convex optimization problem, such as CVXPYlayers [1, 42].

3.5 Numerical experiments

In this section we present numerical experiments that illustrate the ideas and methods discussed above. In the first set of experiments, described in §3.5.2, we show the effect of several constraints and objective terms that serve as effective regularizers and improve performance. In §3.5.4 we illustrate how parameter tuning via back-tests can improve performance, and in §3.5.6 we show how the methods we describe scale with problem size.

3.5.1 Data and back-tests

Data. Throughout the experiments we use the same data set, which is based on the stocks in the S&P 100 index. We use daily adjusted close price data from 2000-01-04 to 2023-09-22. We exclude stocks without data for the entire period, and acknowledge that this inherent survivorship bias in the data set would make it unsuitable for a real portfolio construction method, but it is sufficient for our experiment, which is only concerned with the relative performance of the different methods. We end up with a universe of $n = 74$ assets. In addition to the price data, we use bid-ask spread data to estimate the trading costs, as well as the effective federal funds rate [104] for short term borrowing and lending. We make the data set available with the code for reproducibility and experimentation at

https://github.com/cvxgrp/markowitz-reference.

Mean prediction. Simple estimates of the means work poorly, so in the spirit of [43], we use synthetic return predictions to simulate a proprietary mean prediction method. For each asset, the synthetic returns for each day are given by

$$\hat{r}_t = \alpha(r_t + \epsilon_t),$$

where ϵ_t is a zero-mean Gaussian noise term with variance chosen to obtain a specified information coefficient and r_t is the mean return of the asset in the week starting on day t. We take the noise variance to be $\sigma^2(1/\alpha - 1)$, where α is the square of the information coefficient, and σ^2 is the variance of the return. (These mean predictions are done for each asset separately.) We choose an information coefficient of $\sqrt{\alpha} = 0.15$. Using this parameterization, the sign of the return is predicted correctly in 52.1% of all observations, with this number ranging from 50.3% to 54.1% for the individual assets.

Covariance prediction. For the covariance prediction, we use a simple EWMA estimator, *i.e.*, the covariance matrix at time t is estimated as

$$\hat{\Sigma}_t = \alpha_t \sum_{\tau=1}^{t} \beta^{t-\tau} r_\tau r_\tau^T,$$

where

$$\alpha_t = \left(\sum_{\tau=1}^{t} \beta^{t-\tau} \right)^{-1} = \frac{1-\beta}{1-\beta^t}$$

is the normalization constant, and $\beta \in (0,1)$ is the decay factor. (We use the second moment as the covariance, since the contribution from the mean term is negligible.) We use a half-life of 125 trading days, which corresponds to a decay factor of $\beta \approx 0.994$. We note that the specific choice of the half-life does not change the results of the experiments qualitatively.

Spread. Our simulations include the transaction cost associated with bid-ask spread. In simulation we use the realized bid-ask spread; for the Markowitz problems we use a simple forecast of spread, the average realized bid-ask spread over the previous five trading days.

Shorting and leverage costs. We use the effective federal funds rate as a proxy for interest on cash for both borrowing and lending. When shorting an asset we add a 5% annualized spread over the effective federal funds rate to approximate the shorting cost in our simulation. For forecasting, we set κ^{short} to 7.5% annualized, and κ^{borrow} to the effective federal funds rate.

Back-tests. We use a simple back-test to evaluate the performance of the different methods. We start with a warm-up period of 500 trading days for our estimators leaving us with 5,686 trading days, or approximately 22 years of data. The first 1,250 trading days (five years) are used to initialize the priority parameters. This leaves us with 4,436 out-of-sample trading days, approximately 17 years. Starting with an initial cash allocation of $1,000,000, we call the portfolio construction method each day to obtain the target weights. We then execute the trades at the closing price, rebalancing the portfolio to the new target weights, taking into account the weight changes due to the returns from the previous day. Buy and sell orders are executed at the ask and bid prices, respectively, and interest is paid on borrowed cash and short stocks, and received on cash holdings.

	Return	Volatility	Sharpe	Turnover	Leverage	Drawdown
Equal weight	14.1%	20.1%	0.66	1.2	1.0	50.5%
Basic Markowitz	3.7%	14.5%	0.19	1145.2	9.3	78.9%
Weight-limited	20.2%	11.5%	1.69	638.4	5.1	30.0%
Leverage-limited	22.9%	11.9%	1.86	383.6	1.6	14.9%
Turnover-limited	19.0%	11.8%	1.54	26.1	6.5	25.0%
Robust	15.7%	9.0%	1.64	458.8	3.2	24.7%
Markowitz++	38.6%	8.7%	4.32	28.0	1.8	7.0%
Tuned Markowitz++	41.8%	8.8%	4.65	38.6	1.6	6.4%

Table 3.1: Back-test results for different trading policies.

3.5.2 Taming Markowitz

In this first experiment we show how a basic Markowitz portfolio construction method can lead to the undesirable behavior that would prompt the alleged deficiencies described in §3.1.2. We then show how adding just one more reasonable constraint or objective term improves the performance, taming the basic Markowitz method.

Basic Markowitz. We start by solving the basic Markowitz problem (3.1) for each day in the data set, with the risk target set to 10% annualized volatility. Unsurprisingly the basic Markowitz problem results in poor performance, as seen in the second line of table 3.1. It has low mean return, high volatility (well above the target 10%), a low Sharpe ratio, high leverage and turnover, and a maximum drawdown of almost 80%. This basic Markowitz portfolio performs considerably worse than an equal-weighted portfolio, which we give as a baseline on the top line of table 3.1.

Markowitz with regularization. In a series of four experiments we show how adding just one more reasonable constraint or objective term to the basic Markowitz method can greatly improve the performance.

In the first experiment we add portfolio weight limits of 10% for long positions and -5% for short positions. We limit the cash weight to lie between -5% and 100% (which guarantees feasibility). Adding these asset and cash weight limits leads to a significant improvement in the performance of the portfolio shown in the third row of table 3.1, with the Sharpe ratio increasing to 1.69 (from 0.19), and the maximum drawdown decreasing to 30%. In addition the realized volatility, 11.5%, is closer to the target value 10% than the basic Markowitz trading policy. The turnover is still very high, however, and the maximum leverage is still large at above 5.

In the second experiment we add a leverage limit to the basic Markowitz problem, with $L^{\text{tar}} = 1.6$. This one additional constraint also greatly improves performance, as seen in the fourth row of table 3.1, but with a lower turnover and (not surprisingly) a lower maximum leverage, which is at our target value 1.6.

Our third experiment adds a turnover limit of $T^{\text{tar}} = 25$ to the basic Markowitz problem. This additional constraint drops the turnover considerably, to a value near the target, but still achieves high return, Sharpe ratio, and even lower maximum drawdown.

Our fourth experiment adds robustness to the return and risk forecasts. As simple choices we set all entries of ρ to the 20th percentile of the absolute value of the return forecast at each time step, and use $\varrho = 0.02$. This robustification also improves performance. Not surprisingly the realized risk comes in under our target, since we use the robust risk ex-ante; we could achieve realized risk closer to our desired target 10% by increasing the target to something like 11.5% (which we didn't do).

3.5.3 Markowitz++

In the four experiments described above, we see that adding just one reasonable additional constraint or objective term to the basic Markowitz problem greatly improves the performance. In our last experiment, we include all of these constraints and terms, with parameters

$$\begin{aligned}
&\gamma^{\text{hold}} = 1, \quad \gamma^{\text{trade}} = 1, \quad \sigma^{\text{tar}} = 0.10 \\
&c^{\min} = -0.05, \quad c^{\max} = 1.00, \quad w^{\min} = -0.05, \quad w^{\max} = 0.10, \quad L^{\text{tar}} = 1.6 \\
&z^{\min} = -0.10, \quad z^{\max} = 0.10, \quad T^{\text{tar}} = 25.
\end{aligned}$$

The mean uncertainty parameter ρ is chosen as the 20th percentile of the absolute value of the return forecast, and $\varrho = 0.02$. We soften the risk target, leverage limit, and turnover limit, using the priority parameters

$$\gamma^{\text{risk}} = 5 \times 10^{-2}, \quad \gamma^{\text{lev}} = 5 \times 10^{-4}, \quad \gamma^{\text{turn}} = 2.5 \times 10^{-3}.$$

These were chosen as the 70th percentiles for the corresponding Lagrange multipliers of the hard constraints in the basic Markowitz problem for the risk and turnover limits, and as 25% of the maximum Lagrange multiplier for the leverage limit, over the five years leading up to the out-of-sample study. (We selected γ^{lev} this way since the corresponding constraint was active very rarely in the basic Markowitz problem.)

With this Markowitz++ method, we obtain the performance listed in the second from bottom row of table 3.1. It is considerably better than the performance achieved by adding just one additional constraint, as in the four previous experiments, and very much better than the basic Markowitz method. Not surprisingly it achieves good performance on all metrics, with a high Sharpe ratio, reasonable tracking of our volatility target, modest turnover and leverage, and very small maximum drawdown. When the parameters are tuned annually, as detailed in the next section, we see even more improvement, as shown in the bottom row of table 3.1.

The Sharpe ratios on the bottom two rows are high. We remind the reader that our data has survivorship bias and uses synthetic (but realistic) mean return forecasts, so the performance should not be thought of as implementable. But the differences in performance of the different trading methods is significant.

3.5.4 Parameter tuning

In this section we show how parameter tuning can be used to improve the performance of the portfolio construction method. We will tune the parameters γ^{hold}, γ^{trade}, γ^{lev}, γ^{risk}, and γ^{turn}, keeping the other parameters fixed. We start from the values used in Markowitz++.

Experimental setup. We tune the parameters at the start of every year, on the previous two years of data, and then fix the tuned parameters for the following year. To tune the parameters we use the simple cyclic tuning method described in §3.4.4. We cycle through the parameters one by one. Each time a parameter is encountered in the loop, we increase it by 25%; if this yields an improvement in the performance (defined below), we keep the new value and continue with the next parameter; if not, we decrease the parameter by 20% and check if this yields an improvement. We continue this process until a full loop through all parameters does not yield any improvement. By improvement in performance we mean that all the following are satisfied:

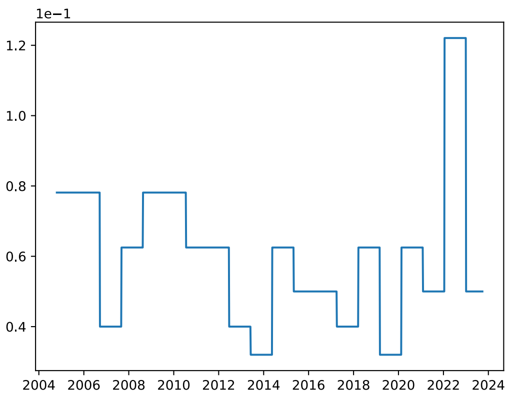
- The in-sample Sharpe ratio increases.
- The in-sample annualized turnover is no more than 50.
- The in-sample maximum leverage is no more than 2.
- The in-sample annualized volatility is no more than 15%.

Results. Tuning the parameters every year yields the performance given in the last row of table 3.1. We see a modest but significant boost in performance over untuned Markowitz++.

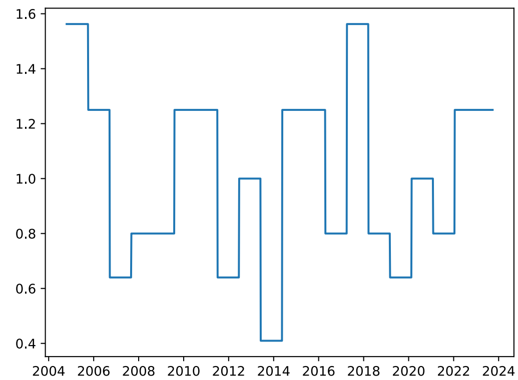
The tuned parameters over time are shown in figure 3.1. We can note several intuitive patterns in the parameter values. For example, γ^{risk} increases during 2008 to account for the high uncertainty in the market during this period. Similarly, γ^{turn} decreases during the same period, likely to allow us to trade more freely to satisfy the other constraints; interestingly γ^{trade} increases during the same period, likely to push us toward more liquid stocks when trading increases. During the same period γ^{lev} increases to reduce leverage. Similar patterns can be observed in 2020.

Tuning evolution. Here we show an example of the evolution of tuning, showing both in- and out-of-sample values of Sharpe ratio, turnover, leverage, and volatility. The in-sample period is April 19, 2016 to March 19, 2018, and the out-of-sample period March 20, 2018 to March 4, 2019. These are shown in figure 3.2. This tuning process converged after 45 back-tests to the parameter values

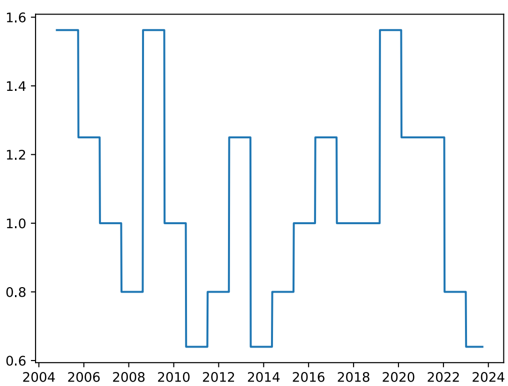
$$\gamma^{\text{risk}} = 4 \times 10^{-2}, \quad \gamma^{\text{hold}} = 0.64, \quad \gamma^{\text{trade}} = 0.64, \quad \gamma^{\text{lev}} = 5 \times 10^{-4}, \quad \gamma^{\text{turn}} = 1.6 \times 10^{-3}.$$



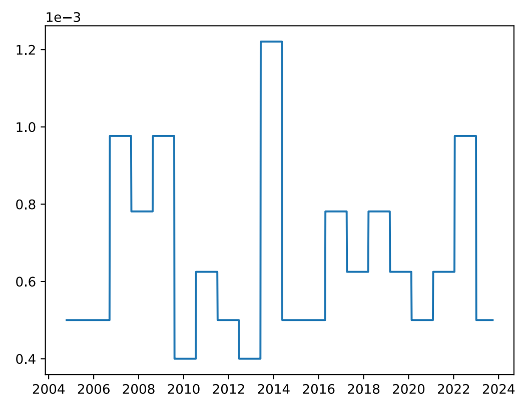
(a) γ^{risk}



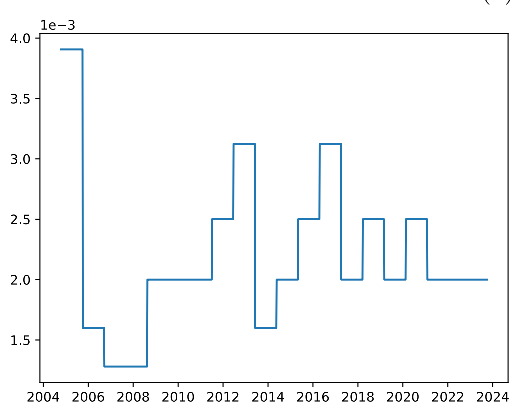
(b) γ^{hold}



(c) γ^{trade}

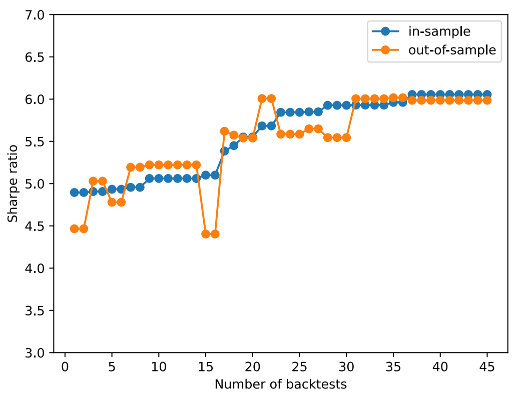


(d) γ^{lev}

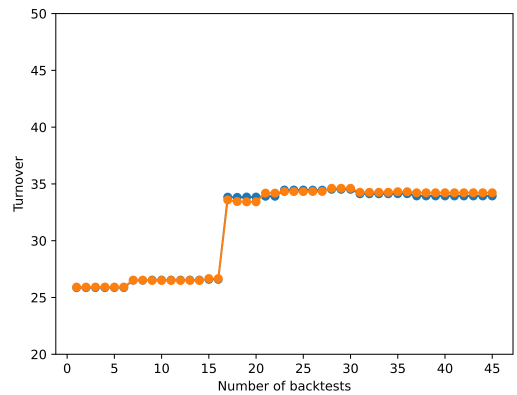


(e) γ^{turn}

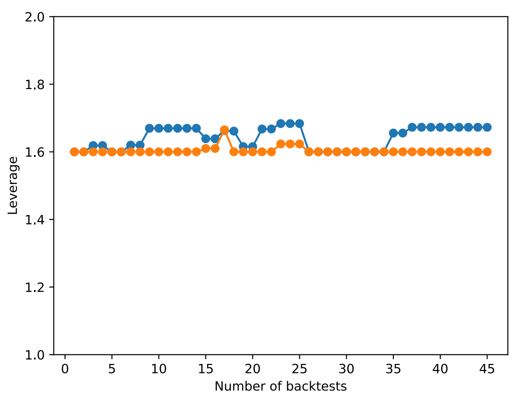
Figure 3.1: Tuned parameters over time.



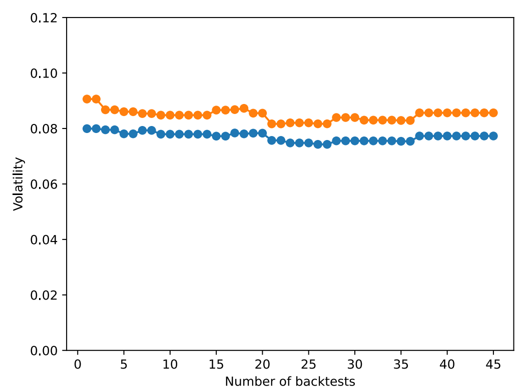
(a) Sharpe ratio.



(b) Turnover.



(c) Leverage.



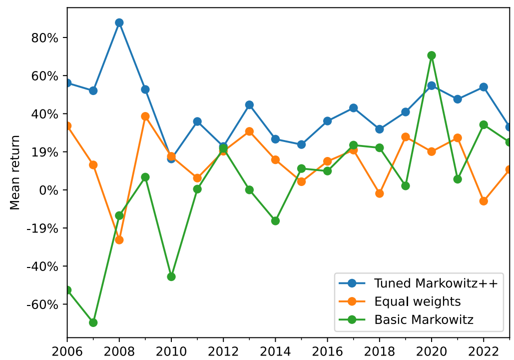
(d) Volatility.

Figure 3.2: Parameter tuning results.

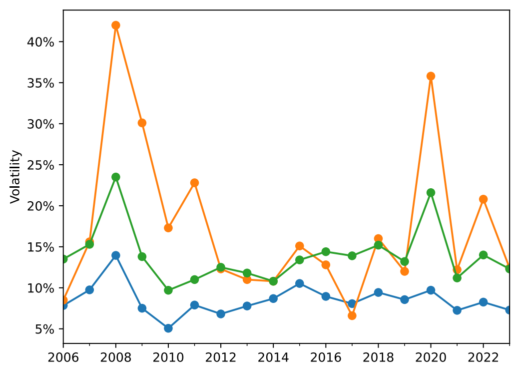
We can see that tuning increases the Sharpe ratio both in- and out-of-sample, while keeping the leverage, turnover, and volatility reasonable. In this example we end up changing 4 of our 5 adjustable parameters, although not by much, which shows that our initial default parameter values were already quite good. Still, we obtain a significant boost in performance.

3.5.5 Annual performance

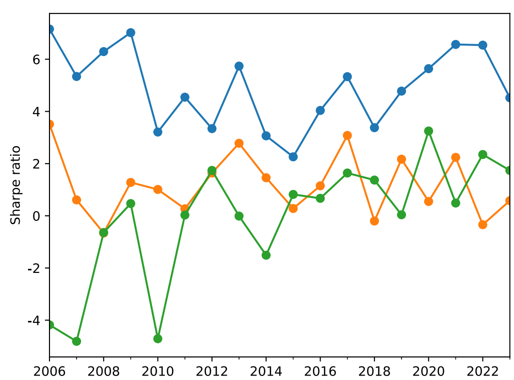
The performance analyses described above and summarized in table 3.1 give aggregate metrics over a 17 year out-of-sample period, long enough to include multiple distinct market regimes as a well as a few market crashes. For such a long back-test, it is interesting to see how the performance in individual years varies with different market regimes. The realized annual return, volatility, and Sharpe ratio are shown in figure 3.3, for basic Markowitz, equal weights, and tuned Markowitz++. Here we see that Markowitz++ not only gives the performance improvements seen in table 3.1, but in addition has less variability in performance across different market regimes.



(a) Yearly annualized returns.



(b) Yearly annualized volatilities.



(c) Yearly annualized mean Sharpe ratios.

Figure 3.3: Yearly annualized metrics for the equal weight portfolio, basic Markowitz, and tuned Markowitz++.

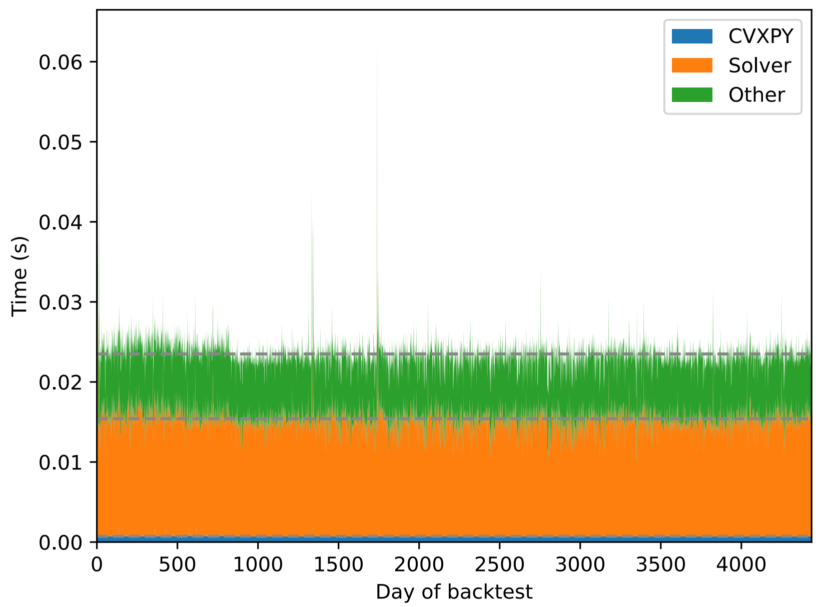


Figure 3.4: Timing results for the Markowitz problem with a single backtest setting.

3.5.6 Scaling

We now turn from the performance of the portfolios to the algorithmic performance of the portfolio construction method itself.

Small problems. We start with the small problem used in the previous section, with $n = 74$ assets, and without a factor risk model. Figure 3.4 shows the time required for each of the 4,436 days in the back-test, broken down into updating and logging (shown in green), CVXPY overhead (shown in blue, negligible), and solver time, the time required to solve the resulting cone program. (We do not count factorizing the covariance matrix, or computing the mean forecasts, since these are done ahead of time, and the time is amortized across all back-tests.)

The 17 year back-test, which involve solving 4,436 problems, takes around 104 seconds on a MacBook Pro with an M1 Pro processor, or about 23ms per day on average. About 63% of the time is spent in the solver, which in this case is MOSEK [9], with other solvers giving similar results, including open-source solvers such as ECOS [85], Clarabel [124], and SCS [243]. Only 3% of the time is spent in the compilation step using CVXPY. The averages for each component of the timings are indicated by the horizontal lines in the figure.

For a small problem like this one, we can carry out a one-year back-test (around 250 trading days) in around six seconds, on a single thread. A single processor with 32 threads can carry out

Assets n	Factors k	Solve time (s)
100	10	0.01
500	20	0.07
500	50	0.10
2,000	50	0.23
2,000	100	0.22
10,000	50	0.65
10,000	100	0.89
50,000	200	9.00
50,000	500	17.77

Table 3.2: Average solve times for Markowitz++ problem for MOSEK, for different problem sizes.

around 20,000 one-year back-tests in an hour. There is little excuse for a PM who does not carry out many back-tests, even if only to vary the parameters around their chosen values.

Large problems. We now investigate the scaling of the method with problem size. As outlined in §3.3.3, a factor model improves the scaling from $O(n^3)$ to $O(nk^2)$. To illustrate this, we solve the Markowitz problem for different values of n and k using randomly generated but realistic data. Table 3.2 shows the average solve time for each problem size across 30 instances using MOSEK. (Solve times with open-source solvers such as Clarabel were a bit longer.) We can see that even very large problems can be solved with stunning speed.

We solved many more problems than those shown in table 3.2, and used the solve times to fit a log-log model, approximating the solve time as an^bk^c, with parameters a, b, c. We obtained coefficients $b = 0.79$ and $c = 1.72$, consistent with the theoretical scaling of $O(nk^2)$.

When the problems are even larger, generic software reaches its limits. In such cases, users may consider switching to first-order methods like the Alternating Direction Method of Multipliers (ADMM) [232, 108, 109, 248, 47], which can offer better scalability and efficiency for very large problems.

3.6 Conclusions

It was Markowitz's great insight to formulate the choice of an investment portfolio as an optimization problem that trades off multiple objectives, originally just expected return and risk, taken to be the standard deviation of the portfolio return. His original proposal yielded an optimization problem with an analytical solution for the long-short case, and a QP for the long-only case, both of which were tractable to solve (for very small problems) even in the 1950s. Since then, stunning advances in computer power, together with advances in optimization, now allow us to formulate and solve much more complex optimization problems, that directly handle various practical constraints and mitigate the effects of forecasting errors. We can solve these problems fast enough that very large numbers of back-tests can be carried out, to give us a good idea of the performance we can expect, and to help choose good values of the parameters. It is hardly surprising that these methods are widely used in quantitative trading today.

While we have vastly more powerful computers, far better software, and easier access to data, than Markowitz did in 1952, we feel that the more complex Markowitz++ optimization problem simply realizes his original idea of an optimization-based portfolio construction method that takes multiple objectives into account.

Chapter 4

Finding moving-band stat-arbs via convex-concave optimization

4.1 Introduction

We consider the problem of finding a *statistical arbitrage* (stat-arb), *i.e.*, a portfolio with mean-reverting price. Roughly speaking, this means that the price of the portfolio stays in a band, and varies over it. Such a stat-arb is traded in the obvious way, buying when the price is in the low part of the band and selling when it is in the high part of the band. In financial markets, price movements are often perceived as largely unpredictable. However, quantitative strategies, such as stat-arb trading, challenge this view by exploiting subtle and often transient relationships between asset prices. Traditional stat-arbs focus on portfolios consisting of two or possibly three underlying assets. When the portfolio contains two assets, trading the stat-arb is called *pairs trading*. The assets in a pair are called *co-moving assets*. Pairs trading involves identifying pairs of assets that have historically exhibited co-integrated price movements. When the prices of these assets deviate from their typical relation, traders can position themselves to profit from the expectation that the prices will revert to their historical relationship. This underlying principle of mean reversion assumes that asset prices will fluctuate within a defined range over time, eventually returning to equilibrium after periods of divergence. Pairs trading and similar strategies provide market participants with a systematic and data-driven approach to profiting from inefficiencies, thereby playing a crucial role in contemporary quantitative trading frameworks. Pairs can be found by exhaustive search over all $n(n-1)/2$ pairs of assets in a universe of n assets. When the weights of the two assets in pairs trading are $+1$ and -1, the portfolio value is the *spread* (between the two prices).

In this paper we propose a new method for finding stat-arbs that can contain multiple (more than two) assets, with general weights. The problem is formulated as a nonconvex optimization problem in which we maximize the portfolio price variation subject to the price staying within a fixed band, along with a leverage limit, over some training period. Although this approach requires maximizing a convex function, we show how to approximately solve it using the convex-concave procedure [282, 194].

Our second contribution is to introduce the concept of a *moving-band stat-arb*. In this approach, the price of the portfolio varies in a band that changes over time, centered at the recent average portfolio price. (We refer to a traditional stat-arb as a *fixed-band stat-arb*.) We show that the same method we use to find fixed-band stat-arbs can be used to find moving-band stat-arbs, despite the apparent complexity of the average price also depending on the portfolio. Moving-band stat-arbs are traded in the same obvious way as fix-band stat-arbs, buying when the price is in the low part of the band and selling when it is in the high part of the band; but with moving-band stat-arbs, the center of the band changes over time. Moving-band stat-arb trading resembles trading using *Bollinger bands* [41, 40], except that the bands are associated with the price of a carefully constructed portfolio, and not a single asset. Our empirical studies show that moving-band stat-arbs out-perform fixed-band stat-arbs in terms of profit, and remain profitable for longer out-of-sample periods.

4.1.1 Related work

Stat-arbs. Stat-arb trading strategies date back to the 1980s when a group of Morgan Stanley traders, led by Nunzio Tartaglia, developed pairs trading [259, 114]. This strategy involves identifying pairs of assets whose price tends to move together, hence referred to as pairs trading. In pairs trading, the spread, *i.e.*, price difference between two assets, is tracked and positions entered when this difference deviates from its mean. This trading strategy has enjoyed widespread popularity, with its success substantiated by numerous empirical studies in various markets like equities [12], commodities [237, 298], and currencies [107]; see, *e.g.*, [114, 12, 256, 145, 178, 62, 152, 87].

In the general setting, a stat-arb consists of multiple assets in a portfolio that exhibits a mean-reverting behavior [106, §10.5]. Stat-arb trading is a widely used strategy in quantitative finance. The literature on stat-arbs is extensive and generally splits into several categories: finding stat-arbs, modeling the (mean-reverting) portfolio price, and trading stat-arbs. We give a brief review of these here and refer the reader to [177] for a comprehensive overview of the literature.

Finding stat-arbs. Probably the simplest approach to finding pairs of co-moving securities is the distance approach. The distance pairs trading strategy finds assets whose (normalized) prices have moved closely historically, in an exhaustive search through pairs of assets [114]. Assets whose prices have a low sum of squared deviation from each other are considered for trading. The distance approach is simple and intuitive, although it does not necessarily find good pairs [177]. The objective itself is to minimize the distance between two asset prices, which does not directly relate to the desired properties of a stat-arb, which crucially should also have a high variance. This paper addresses this issue by directly optimizing for large fluctuations around the mean.

The co-integration approach is another popular method for finding co-moving securities. Co-integration is an important concept in the econometrics literature [162, 5], and dates back to Engle and Granger's works in the 1980s (for which Granger was awarded the 2003 Nobel Memorial Prize in Economic Sciences) [127, 95]. The idea is that the absence of stationarity in a multivariate time series may be explained by common trends, which would make it possible to find linear combinations of assets that are stationary and hence mean-reverting. Thus, the co-integration approach is based on identifying linear combinations of assets that result in a stationary time series [177]. In [302], the most cited work on co-integration based pairs trading, potential asset pairs are found based on statistical measures, which are then tested for co-integration using an adapted version of the Engle-Granger test. Several co-integration based methods have been proposed to extend the pairs trading strategy to more than two assets. For example, in [316, 318, 317] the authors consider a (non-convex) optimization problem for finding high variance, mean-reverting portfolios. Their strategy is based on finding a portfolio of spreads, defined by a co-integration subspace, and implemented using sequential convex optimization. Their proposed optimization problem, *i.e.*, maximizing variance subject to a mean-reversion criterion is similar to our problem formulation. However, our problem differs significantly in that we do not rely on any co-integration analysis or statistical testing. Rather, we directly optimize for a high variance portfolio that is mean-reverting.

Asset pairs can also be found using machine learning methods. In [272], the authors use unsupervised learning and propose a density-based clustering algorithms to cluster assets. Then, within asset clusters, pairs of assets are chosen for trading depending on co-integration, as well as mean-reversion tendency and frequency. Modern machine learning methods are also explored in [178], where the authors propose the use of deep neural networks, gradient-boosted trees, and random forests for finding stat-arb portfolios. Another recent study of deep-learning stat-arb finding is [134]. Earlier work on using machine learning for finding stat-arbs includes, *e.g.*, [84, 234, 289, 151, 150].

Modeling the stat-arb spread. When a co-moving set of assets has been identified, the next step is to model the portfolio price (or spread between the assets for a pair). Perhaps the most popular approach is to model the spread using stochastic control theory. It is common to consider investments in a mean-reverting spread and a risk-free asset and to model the spread as an Ornstein-Uhlenbeck process [235, 169]. Other methods include those borrowing tools from time series analysis [177]. In [89] the authors propose a mean-reverting Gaussian Markov chain model for modeling the spread between two assets. Copulas have also been proposed to model the joint distribution of the spread, both for pairs and for larger sets of assets [75, 287, 179]. In [75] the authors suggest modeling the spread using linear state space models.

Trading stat-arbs. We mention here a number of stat-arb trading methods, ranging from simple ones based on the intuitive idea of buying when the price is low and selling when it is high, to more complex ones based on learning the price statistics and using stochastic control. One simple method is based on *hysteresis*, as used in a conventional thermostat. In this approach we buy (enter into a long position) when the price drops below a threshold, and sell (switch to a short position) when the price goes above another threshold. The thresholds are typically based on price bands, as discussed below. A variation on this method sets the thresholds based on the standard deviation of the price, as proposed in [114]. Another simple method is *linear trading*, where we take a position proportional to the difference between the band midpoint price and the current price. This method can also be modified to use volatility-based bands instead of fixed bands. (Such a trading policy is a simple Markowitz policy, with the mean return given by the difference between the midpoint price and the current price.)

Other methods for trading stat-arbs follow directly from the spread models described above. When the spread is modeled as an Ornstein-Uhlenbeck process, the optimal trading strategy is found by solving a stochastic control problem [235, 169, 27]. In [310, 262, 312] the authors model the spread as an autoregressive process, a discretization of the Ornstein-Uhlenbeck process, and show how to trade a portfolio of spreads under proportional transaction costs and gross exposure constraints using model predictive control. With the copula approaches of [75, 287, 179], the trading strategy is based on deviations from confidence intervals. The machine learning methods of [272, 178, 134, 84, 234, 289, 151, 150] also include trading strategies.

Exiting a stat-arb. A stat-arb will not keep its mean-reverting behavior forever. Hence, a strategy for exiting a stat-arb, *i.e.*, closing the position, is needed. One approach is to exit when the spread reaches a certain threshold in terms of its standard deviation or in terms of a price band, as described below. Another simple method is to exit after a fixed time-period, as was done in [114].

A simple variation on any of these exit methods does not exit the position immediately when the exit condition is first satisfied. Instead it reduces the position to zero slowly (*e.g.*, linearly) over some fixed number of periods.

Price-bands. Price-bands are popular in technical analysis, and are used to identify trading opportunities based on the price of an asset relative to its recent price history [188]. The most popular is the Bollinger band [40]. It is constructed by computing the M-day moving average (with common choice $M = 21$) of the asset price, denoted μ_t at time t, and the corresponding price standard deviation σ_t. The Bollinger band is then defined as the interval $[\mu_t - k\sigma_t, \mu_t + k\sigma_t]$, where $k > 0$ is a parameter, typically taken to be $k = 2$. A price signal is extracted based on if the price is near the top or bottom of the band. For a detailed description of the Bollinger band and other trading-bands, we refer the reader to [41, 40].

4.1.2 This paper

This paper proposes a new method for finding stat-arbs, with two main contributions. The first contribution of our method is to formulate the search for stat-arbs as an optimization problem that intuitively and directly relates to the desired properties of a stat-arb: its price should remain in a band (*i.e.*, be mean-reverting) and also should have a high variance. Since our method is based on convex optimization, it readily scales to large universes of assets. It can and does find stat-arbs with ten or more assets, well beyond our ability to carry out an exhaustive search.

Our second contribution is to introduce the concept of a moving-band stat-arb. While the idea of a price band is widely known and used in technical analysis trading, the main difference is that we apply it to carefully constructed portfolios (*i.e.*, stat-arbs), instead of single assets.

Our focus is on finding both fixed-band and moving-band stat-arbs, and not on trading them. Our numerical experiments use a simple linear trading policy, and a simple time-based exit condition. (We have also verified that similar results are obtained using hysteris-based trading policies.) We also do not address the question of how one might trade a portfolio of stat-arbs, the focus of a forth-coming paper.

4.1.3 Outline

The rest of the paper is organized as follows. In §4.2 we propose a new method for finding traditional stat-arbs, *i.e.*, with a fixed band. We extend this method to moving bands in §4.3. In §4.4 we present experiments on real data, and in §4.5 we conclude the paper.

4.2 Finding fixed-band stat-arbs

We consider a vector time series of prices of a universe of n assets, denoted $P_t \in \mathbf{R}^n$, $t = 1, 2, \dots, T$, denoted in USD per share. (We presume these are adjusted for dividends and splits.) We consider a portfolio of these assets given by $s \in \mathbf{R}^n$, denoted in shares, with $s_i < 0$ denoting short positions. The price or net value of the portfolio is the scalar time series $p_t = s^T P_t$, $t = 1, \dots, T$. The asset prices P_t are positive, but the portfolio price p_t need not be, since the entries of s can be negative. For future use, we define the average price of the n assets as the vector $\bar{P} = (1/T)\sum_{t=1}^{T} P_t$.

We seek a portfolio for which p_t consistently varies over a band (interval of prices) with two goals. It should stay in the price band, and also, vary over the band consistently; that is, it should frequently vary between the high end of the band and the low end of the band. We refer to such a portfolio s as a stat-arb, so we use the term to refer to the general concept as well as a specific portfolio. Our method differs from the traditional statistical framework, where one would seek a portfolio s for which p_t is co-integrated.

4.2.1 Formulation as convex-concave problem

We formulate the search for a stat-arb s as an optimization problem. The condition that p_t varies over a band is formulated as $-1 \leq p_t - \mu \leq 1$, where μ is the midpoint of the band. Here we fix the width of the price band as 2; s and μ can always be scaled so this holds. We express the desire that p_t vary frequently over the band by maximizing its volatility. Since s, p, and μ can all be multiplied by -1 without any effect on the constraints or objective, we can assume that $\mu \geq 0$ without loss of generality.

We arrive at the problem

$$\begin{array}{ll} \text{maximize} & \sum_{t=2}^{T}(p_t - p_{t-1})^2 \\ \text{subject to} & -1 \leq p_t - \mu \leq 1, \quad p_t = s^T P_t, \quad t = 1, \ldots, T \\ & |s|^T \bar{P} \leq L, \quad \mu \geq 0, \end{array} \tag{4.1}$$

with variables $s \in \mathbf{R}^n$, $p \in \mathbf{R}^T$, and $\mu \in \mathbf{R}$, where the absolute value in the last constraint is elementwise. The problem data are the vector price time series P_t, $t = 1, \ldots, T$, and the positive parameter L.

Note that $|s|^T \bar{P}$ is the average total position of the portfolio, sometimes called its leverage, so the constraint $|s|^T \bar{P} \leq L$ is a leverage constraint; it limits the total position of the portfolio. The leverage is a weighted ℓ_1 norm of s, and so tends to lead to sparse s, *i.e.*, a portfolio that concentrates in a few assets, a typical desired quality of a stat-arb. The problem (4.1) is a nonconvex optimization problem, since the objective is a convex function, and we wish to maximize it; we will explain below how we can approximately solve it.

4.2.2 Interpretation via a simple trading policy

While our formulation of the stat-arb optimization problem (4.1) makes sense on its own, we can further motivate it by looking at the profit obtained using a simple trading policy. Suppose we hold quantity $q_t \in \mathbf{R}$ of the portfolio, *i.e.*, we hold the portfolio $q_t s \in \mathbf{R}^n$ (in shares). We assume that $q_0 = q_T = 0$, *i.e.*, we start and end with no holdings. In period t we buy $q_t - q_{t-1}$ and pay $p_t(q_t - q_{t-1})$. The total profit is then

$$\sum_{t=1}^{T-1} q_t(p_{t+1} - p_t). \tag{4.2}$$

We will relate this to our objective in (4.1) above, with the simple trading policy

$$q_t = \mu - p_t, \quad t = 1, \ldots, T-1, \tag{4.3}$$

which we refer to as a linear trading policy since the holdings are a linear function of the difference between the band midpoint and the current price. This trading policy holds nothing when $p_t = \mu$, *i.e.*, the price is in the middle of the band. When the price is low, $p_t = \mu - 1$, we hold $q_t = +1$, and when it is high, $p_t = \mu + 1$ we hold $q_t = -1$.

With the simple linear policy (4.3) and the boundary conditions $q_0 = q_T = 0$, the profit (4.2) is, after some algebra,

$$\frac{1}{2}\sum_{t=2}^{T}(p_t - p_{t-1})^2 + \frac{(p_1-\mu)^2 - (p_T-\mu)^2}{2}.$$

The first term is one half our objective. The second term is between $-1/2$ and $1/2$, since $(p_T-\mu)^2$ and $(p_1-\mu)^2$ are both between 0 and 1. Thus the profit is at least

$$\frac{1}{2}\left(\sum_{t=2}^{T}(p_t - p_{t-1})^2 - 1\right). \tag{4.4}$$

This shows that our objective is the profit of the simple linear policy (4.3), scaled by one-half, plus a constant. In particular, if the objective of the problem (4.1) exceeds one, the simple linear policy makes a profit.

4.2.3 Solution method

Convex-concave procedure. We solve the problem (4.1) approximately using sequential convex programming, specifically the convex-concave procedure [282, 194]. Let k denote the iteration, with s^k the portfolio and p_t^k the portfolio price in the kth iteration. In each iteration of the convex-concave procedure, we linearize the objective, replacing the quadratic function $f(p) = \sum_{t=2}^{T}(p_t - p_{t-1})^2$ with the affine approximation

$$\hat{f}(p; p^k) = f(p^k) + \nabla f(p^k)^T(p - p^k) = \nabla f(p^k)^T p + c,$$

where c is a constant (*i.e.*, does not depend on p). This linearization is a lower bound on the true objective, *i.e.*, we have $f(p) \geq \hat{f}(p; p^k)$ for all p. For completeness we note that

$$(\nabla f(p))_t = \begin{cases} 2(p_1 - p_2) & t = 1 \\ 2(2p_t - p_{t-1} - p_{t+1}) & t = 2, \dots, T-1 \\ 2(p_T - p_{T-1}) & t = T. \end{cases}$$

We now solve the linearized problem

$$\begin{array}{ll} \text{maximize} & \hat{f}(p; p^k) \\ \text{subject to} & -1 \leq p_t - \mu \leq 1, \quad p_t = s^T P_t, \quad t = 1, \dots, T \\ & |s|^T \bar{P} \leq L, \quad \mu \geq 0, \end{array} \tag{4.5}$$

with variables p_t, s and μ. This is a convex problem, in fact a linear program (LP), and readily solved [50]. We take the solution of this problem as the next iterate p^{k+1}, s^{k+1}, μ^{k+1}. This simple algorithm converges to a local solution of (4.1), typically in at most a few tens of iterations.

Cleanup phase. The leverage constraint $|s|^T\bar{P} \leq L$ encourages sparse solutions, but in some cases the convex-concave procedure converges to a portfolio with a few small holdings. To achieve even sparser portfolios, we can carry out a clean-up step once the convex-concave procedure has converged. We first determine the subset of assets for which s_i is zero or small, as measured by its relative weight in the portfolio, *i.e.*, i for which

$$|s_i|\bar{P}_i \leq \eta|s|^T\bar{P},$$

where η is a small positive constant such as 0.05. We then solve the problem again, this time with the constraint that all such s_i are zero. This takes just a few convex-concave iterations, and can be repeated, which results in sparse portfolios in which every asset has weight at least η.

Implementation. To make the optimization problem better conditioned, we scale the prices P_t so that (after scaling) $\bar{P} = \mathbf{1}$, the vector with all entries one. Thus after scaling, the leverage $|s|^T\bar{P}$ becomes the ℓ_1 norm of s. We also scale the gradient (or objective) to be on the order of magnitude one. These scalings do not affect the solution, but make the method less vulnerable to floating point rounding errors.

Initialization. The final portfolio found by the convex-concave procedure, plus the cleanup phase, depends on the initial portfolio s^1. It can and does converge to different final portfolios for different starting portfolios. With a random initialization we can find multiple stat-arbs for the same universe. (We also get some duplicates when the method converges to the same final portfolio from different initial portfolios.) We have found that uniform initialization of the entries of s^1 in the interval $[0,1]$ works well in practice. Thus, from one universe and data set, we can obtain multiple stat-arbs.

4.3 Finding moving-band stat-arbs

4.3.1 Moving-band stat-arbs

In the fixed-band stat-arb problem (4.1) μ is constant, so the midpoint of the trading band does not vary with time. In this section we describe a simple but powerful extension in which the stat-arb band midpoint changes over time. One simple (and traditional) choice is to define μ_t as the mean of the trailing prices p_t, for example the mean over the last M periods,

$$\mu_t = \frac{1}{M}\sum_{\tau=t-M+1}^{t} p_\tau.$$

(This requires knowledge of the prices $P_0, P_{-1}, \ldots, P_{-M+1}$.) In this formulation, μ_t is also a function of s, the portfolio, but it is a known linear function of it. Any other linear expression for the average recent price could be used, *e.g.*, exponentially weighted moving average (EWMA). A moving-band stat-arb is a portfolio s in which the price p_t stays in a moving band with width two and midpoint μ_t, and also has high variance.

Trading policy. The simple linear trading policy (4.3) can be modified in the obvious way, as

$$q_t = \mu_t - p_t, \quad t = 1, \ldots, T-1. \tag{4.6}$$

4.3.2 Finding moving-band stat-arbs

We arrive at the optimization problem

$$\begin{array}{ll} \text{maximize} & \sum_{t=2}^{T} (p_t - p_{t-1})^2 \\ \text{subject to} & -1 \leq p_t - \mu_t \leq 1, \quad p_t = s^T P_t, \quad t = 1, \ldots, T \\ & |s|^T \bar{P} \leq L, \quad \mu_t = (1/M) \sum_{\tau=t-M+1}^{t} p_\tau, \quad t = 1, \ldots, T, \end{array} \tag{4.7}$$

with variables $s \in \mathbf{R}^n$, $p_1, \ldots, p_T$, and $\mu_1, \ldots, \mu_T$. (The latter two sets of variables are simple linear functions of s.) In this problem we have an additional parameter M, the memory for the band midpoint.

Note that in the fixed-band stat-arb problem (4.1), μ is a scalar variable that we freely choose; in the moving-band stat-arb problem (4.7), μ_t varies over time, and is itself a function of s. Despite this complication, the moving-band stat-arb problem (4.7) can be approximately solved using exactly the same convex-concave method as the fixed-band stat-arb problem (4.1); the only difference is in the convex constraints.

4.4 Numerical experiments

We illustrate our method with an empirical study on historical asset prices. Everything needed to reproduce the results is available online at

`https://github.com/cvxgrp/cvxstatarb`.

4.4.1 Experimental setup

Data set. We use daily data of the CRSP US Stock Databases from the Wharton Research Data Services (WRDS) portal [308]. The data set consists of adjusted prices of 15405 assets from January 4th, 2010, to December 30th, 2023, for a total of 3282 trading days.

Monthly search for stat-arbs. Starting December 23, 2011, and every 21 trading days thereafter, until July 6th, 2022, we use the convex-concave method with 10 different random initial portfolios. From these 10 stat-arbs we add the unique ones, defined by the set of assets in the stat-arb, to our current set of stat-arbs. All together we solve fixed-band and moving-band problems 1270 times.

Parameters. We set $L = \$50$ for the fixed-band stat-arb and $L = \$100$ for the moving-band stat-arb. For the moving-band stat-arb, we take $M = 21$ days, *i.e.*, we use the trailing month average price as the midpoint, a value commonly used for Bollinger bands [40, 41]. We note that our results are not very sensitive to the choices of L and M, and that choosing a larger L for the moving-band than for the fixed-band stat-arb is reasonable since we expect the price of a portfolio to vary less around its short-term moving midpoint than around a fixed midpoint.

Trading policy. We use the simple linear trading policies defined in (4.3) and (4.6) for the fixed-band and moving-band stat-arbs, respectively. We use a simple time-based exit condition, where we trade a stat-arb for $T^{\max}$ trading days, and then exit the position uniformly (linearly) over the next T^{exit} trading days. This means we take

$$q_t = (1 - \alpha_t)(\mu - p_t), \quad t = T^{\max}, \ldots, T^{\max} + T^{\text{exit}} - 1,$$

where $\alpha_t = (t + 1 - T^{\max})/T^{\text{exit}}$. We use parameter values $T^{\max} = 63$ for the fixed-band stat-arb and $T^{\max} = 125$ for the moving-band stat-arb, and $T^{\text{exit}} = 21$ for both. We also exit a stat-arb if the value of the stat-arb plus a cash account drops below a given level, as described below.

4.4.2 Simulation and metrics

We simulate and evaluate a stat-arb as follows.

Cash account. Each stat-arb is initialized with a cash account value

$$C_0 = \nu |s|^T P_0,$$

where ν is a positive parameter and P_0 is the price vector the day before we start trading the stat-arb. We take $\nu = 0.5$ in our experiments. The cash account is updated as

$$C_{t+1} = C_t - (q_{t+1} - q_t) p_{t+1} - \phi_t, \quad t = 1, \ldots, T,$$

where ϕ_t is the transaction and holding cost, consisting of the trading cost at time $t + 1$ and the holding cost over period t, described below. The cash account plus the long position is meant to be the collateral for the short positions.

Portfolio net asset value. The net asset value (NAV) of the portfolio, including the cash account, at time t is then

$$V_t = C_t + q_t p_t.$$

(Note that $V_0 = C_0$.) The profit at time t is

$$V_t - V_{t-1} = q_t p_t + C_t - q_{t-1} p_{t-1} - C_{t-1} = q_t(p_t - p_{t-1}) - \phi_t,$$

which agrees with the profit formula (4.2), after accounting for transaction and holding costs.

NAV based termination. If the NAV goes below 25% of C_0, we liquidate the stat-arb portfolio. We do this for two reasons. First, we do not want the portfolio to have negative value, *i.e.*, to go bust. Second, this constraint ensures that the short positions are always fully collateralized by the long positions plus the cash account, with a margin of at least half of our initial investment. All of our metrics include early termination stat-arbs.

Trading and shorting costs. Our numerical experiments take into account transaction costs, *i.e.*, we buy assets at the ask price, which is the (midpoint) price plus one-half the bid-ask spread, and we sell assets at the bid price, which is the price minus one-half the bid-ask spread. Note that while we do not take into account transaction cost in our simple trading policy, we do in simulation and accounting. We use 0.5% as a proxy for the annual shorting cost of assets, which is well above what is typically observed in practice for liquid assets [73, 117, 173, 86]. (We also note that the method is robust against shorting costs; the results are consistent even when shorting costs exceed 10% per annum.)

Metrics. The profit of a stat-arb is

$$\sum_{t=1}^{T} (V_t - V_{t-1}),$$

where $T = T^{\max} + T^{\text{exit}}$ is the number of trading days in the evaluation period. The return at time t is

$$r_t = \frac{V_t - V_{t-1}}{V_{t-1}}, \quad t = 1, \ldots, T.$$

We report several standard metrics based on the daily returns r_t. The average return is

$$\bar{r} = \frac{1}{T} \sum_{t=1}^{T} r_t,$$

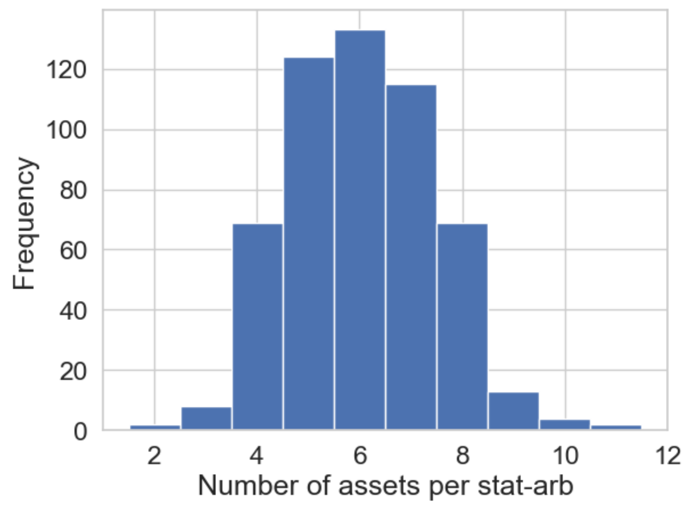


Figure 4.1: Distribution of the number of assets per fixed-band stat-arb.

which we multiply by 250 to annualize. The risk (return volatility) is

$$\left(\frac{1}{T} \sum_{t=1}^{T} (r_t - \overline{r})^2 \right)^{1/2},$$

which we multiply by $\sqrt{250}$ to annualize. The annualized Sharpe ratio is the ratio of the annualized average return to the annualized risk. Finally, the maximum drawdown is

$$\max_{1 \leq t_1 < t_2 \leq T} \left(1 - \frac{V_{t_2}}{V_{t_1}} \right),$$

the maximum drop in value form a previous high.

4.4.3 Results for fixed-band stat-arbs

Stat-arb statistics. After solving the fixed-band stat-arb problem 1270 times, we found 539 unique stat-arbs. These stat-arbs contained between 2 and 11 assets, with a median value of 6, as shown in figure 4.1. Over time the number of active stat-arbs ranges up to 30, with a median value of 16, as shown in figure 4.2. Out of the 539 stat-arbs, 17 (around 3%) were terminated before the end of the evaluation period, due to the NAV falling below 25% of the initial investment.

Metrics. Table 4.1 summarizes metrics related to the profitability of the fixed-band stat-arbs.

Of the 539 stat-arbs, 68% were profitable. The average annualized return is 2%, with an average annualized risk of 35%, and an average annualized Sharpe ratio of 0.81. The maximum drawdown

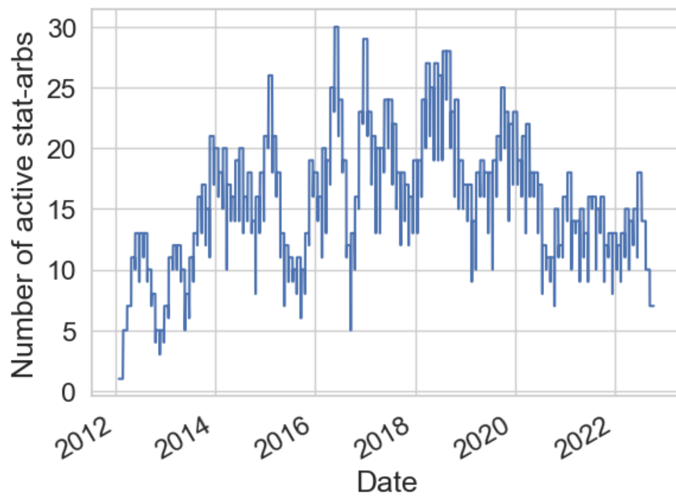


Figure 4.2: Number of active fixed-band stat-arbs over time.

was on average 17% over the four-month trading period for each stat-arb.

Profitability	
Fraction of profitable stat-arbs	68%
Annualized return	
Average	2%
Median	18%
75th percentile	34%
25th percentile	-5%
Annualized risk	
Average	35%
Median	21%
75th percentile	38%
25th percentile	13%
Annualized Sharpe ratio	
Average	0.81
Median	1.01
75th percentile	1.87
25th percentile	-0.21
Maximum drawdown	
Average	17%
Median	10%
75th percentile	19%
25th percentile	5%

Table 4.1: Metric summary for 539 fixed-band stat-arbs.

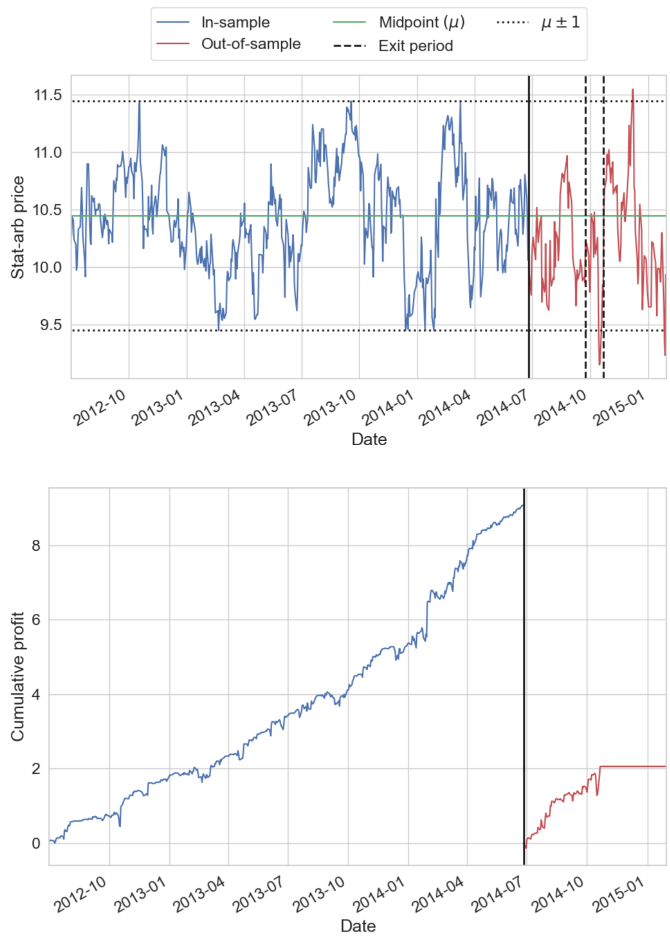


Figure 4.3: A fixed-band stat-arb strategy that made money. *Top.* Price. *Bottom.* Cumulative profit.

Example stat-arbs. We show the detailed evolution of two stat-arbs, one that made money and one that lost money, in figures 4.3 and 4.4, respectively. These were chosen to have average returns around the 70th and 25th percentiles of the return distribution across our 539 stat-arbs. As expected both stat-arbs are very profitable in-sample. The first one continues to be profitable out-of-sample. The first one, which made money, contained the assets

Amgen
Walgreens
Anadarko Petroleum
Yum! Brands
Energy Transfer LP
Alexion Pharmaceuticals.

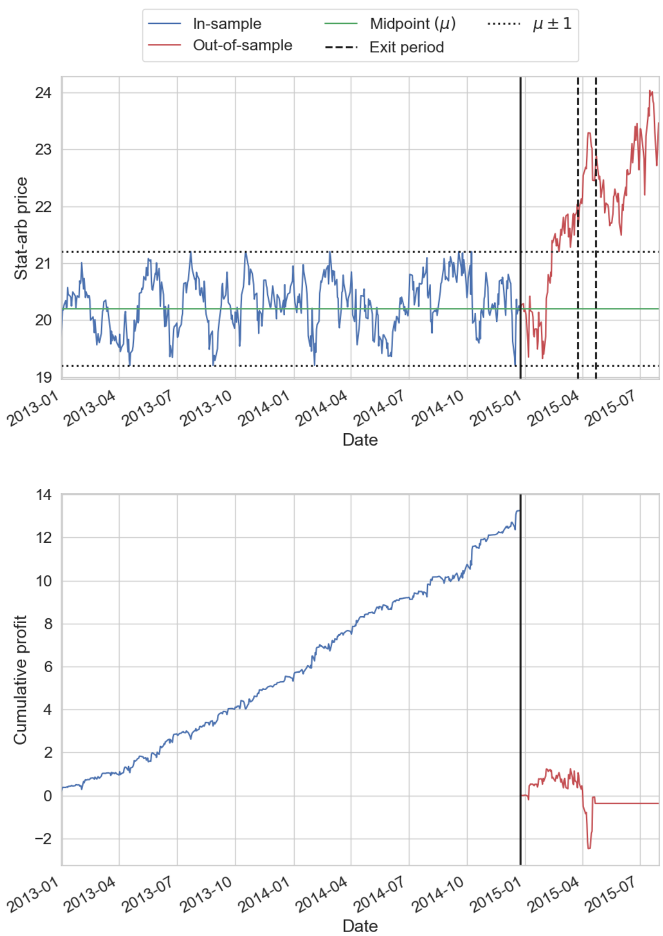


Figure 4.4: A fixed-band stat-arb strategy that lost money. *Top.* Price. *Bottom.* Cumulative profit.

The second one, which lost money, contained the assets

Target
Vanguard
Deutsche Bank
The Charles Schwab Corporation
Delta Air Lines
Alexion Pharmaceuticals
The Archer-Daniels-Midland Company

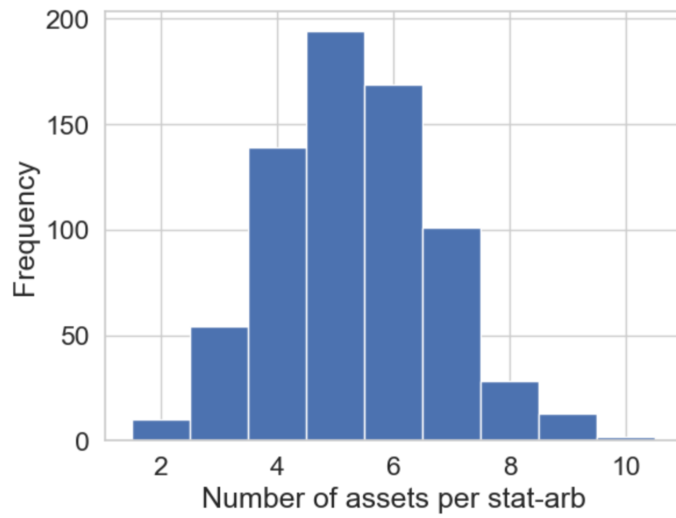


Figure 4.5: Distribution of the number of assets per moving-band stat-arb.

4.4.4 Results for moving-band stat-arbs

Stat-arb statistics. We found 711 unique moving-band stat-arbs (compared with 539 fixed-band stat-arbs). These stat-arbs contained between 1 and 11 assets, with a median value of 5. The full distribution is shown in figure 4.5. The number of active moving-band stat-arbs over time is shown in figure 4.6. The median number of active stat-arbs is 40. This is considerably larger than the number of active fixed-band stat-arbs since we find more of them, and they are active (by our choice) almost twice as long. None out of the 711 moving-band stat-arbs were terminated before the end of the evaluation period, due to the NAV falling below 25% of the initial investment.

Metrics. Table 4.2 summarizes the profitability of the moving-band stat-arbs. A large majority (79%) of the stat-arbs are profitable. The average annualized return was 16%, with an average annualized risk of 20%, and an average annualized Sharpe ratio of 0.84. The maximum drawdown was on average 12% over the seven-month trading period for each stat-arb.

Comparison with fixed-band stat-arbs. Our first observation is that far fewer of the moving-band stat-arbs were terminated early due to low NAV than the fixed-band stat-arbs, despite their running for a period almost twice as long. Comparing tables 4.1 and 4.2 we see that the metrics for fixed-band stat-arbs are more variable, with a larger range in each of the metrics. The moving-band stat-arbs are more profitable than the fixed-band stat-arbs, but the difference is not large.

Example stat-arbs. Two stat-arbs, picked to represent roughly the 70th and 15th percentiles of the return distribution across the 711 stat-arbs, are illustrated in figures 4.7 and 4.8, respectively. Again,

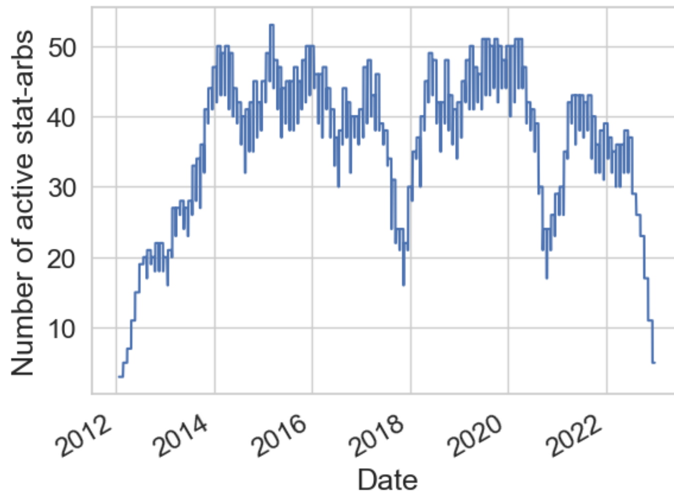


Figure 4.6: Number of active moving-band stat-arbs over time.

Profitability	
Fraction of profitable stat-arbs	79%
Annualized return	
Average	16%
Median	12%
75th percentile	24%
25th percentile	3%
Annualized risk	
Average	20%
Median	15%
75th percentile	25%
25th percentile	9%
Annualized Sharpe	
Average	0.84
Median	0.87
75th percentile	1.51
25th percentile	0.21
Maximum drawdown	
Average	12%
Median	9%
75th percentile	15%
25th percentile	5%

Table 4.2: Metric summary for 711 moving-band stat-arbs.

both stat-arbs are profitable in-sample, and the first one continues to be profitable out-of-sample. The first one, which made money, contained the assets

Morgan Stanley
Monsanto
Walgreens
Accenture Plc
ArcelorMittal
Pioneer Natural Resources

The second one, which lost money, contained the assets

Lockheed Martin
ServiceNow
Gilead Sciences
NXP Semiconductors

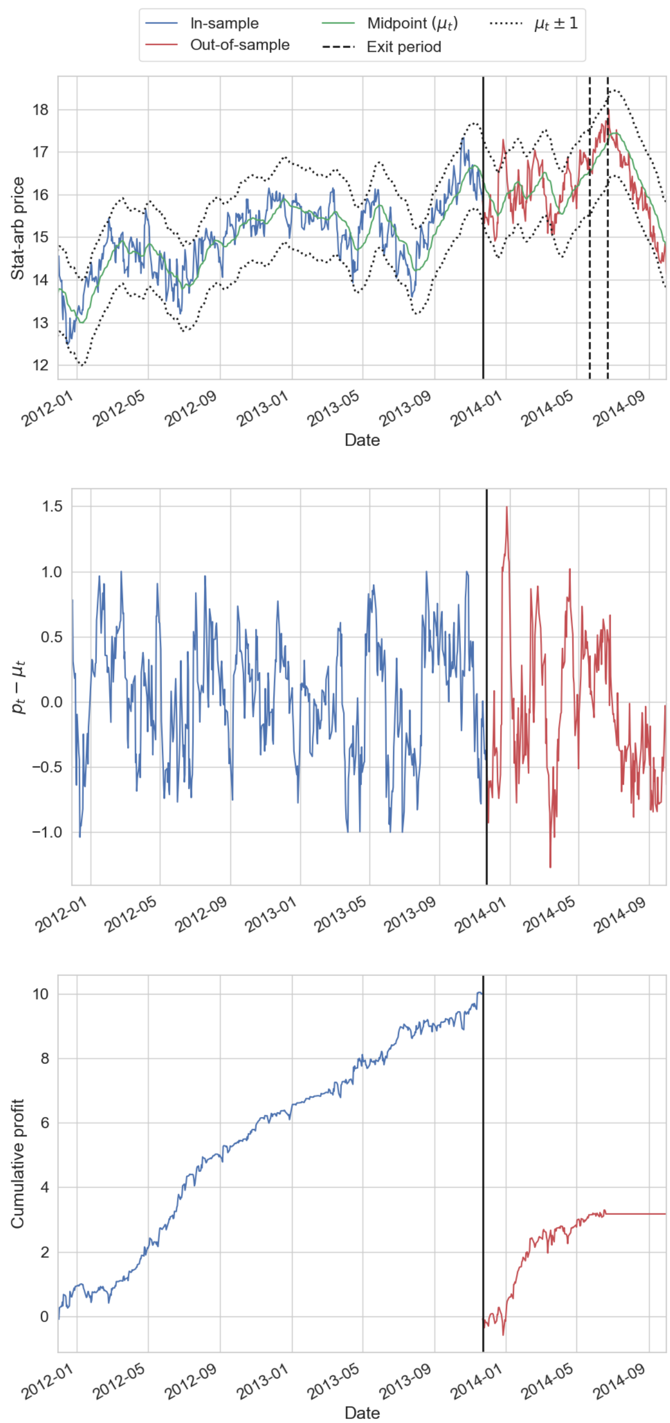


Figure 4.7: A moving-band stat-arb strategy that made money. *Top.* Price. *Middle.* Price relative to the trailing mean. *Bottom.* Cumulative profit.

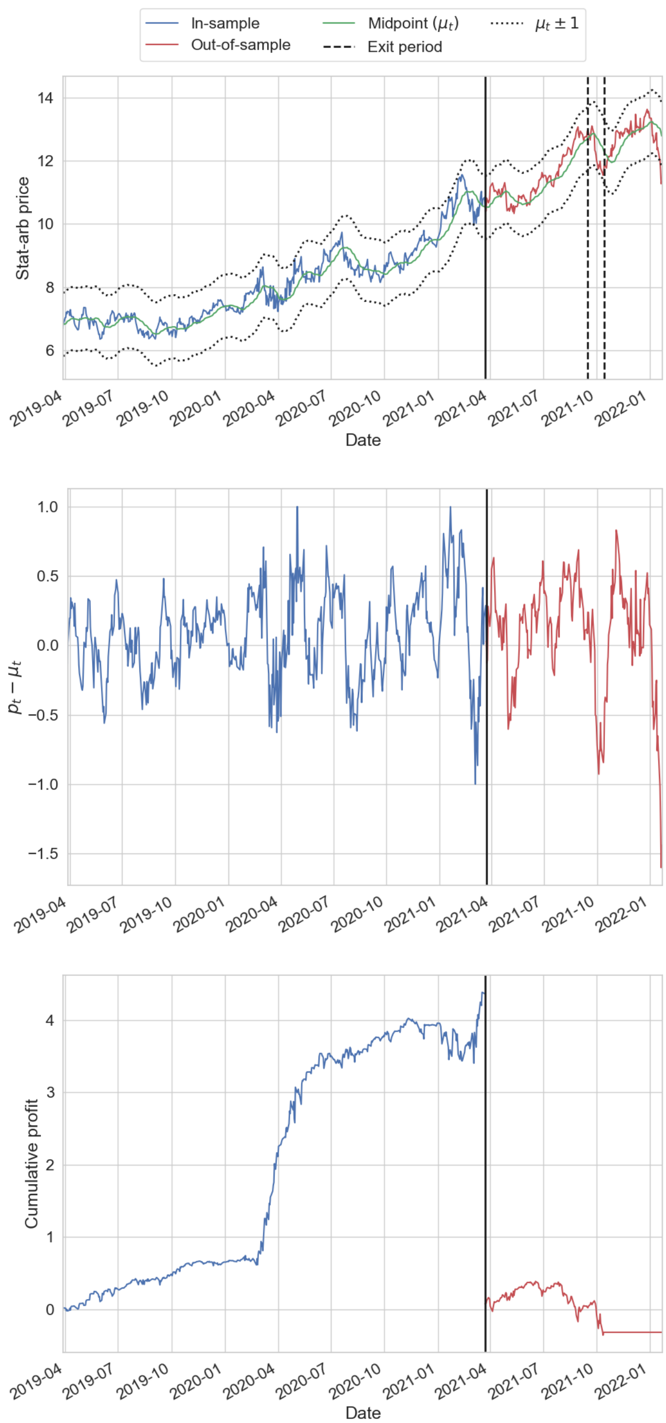


Figure 4.8: A moving-band stat-arb strategy that lost money. *Top.* Price. *Middle.* Price relative to the trailing mean. *Bottom.* Cumulative profit.

4.5 Conclusions and comments

We have formulated the problem of finding stat-arbs as a nonconvex optimization problem which can be approximately solved using the convex-concave procedure. We have introduced moving-band stat-arbs, which combine ideas from statistical arbitrage and price band trading.

Our empirical study on historical data shows that moving-band stat-arbs perform better than fixed-band stat-arbs, and remain profitable for longer out-of-sample periods. Our empirical study uses very simple trading and exit policies; we imagine that with more sophisticated ones such as those cited above, the results would be even better. Our focus in this paper is on finding stat-arbs, and not on trading them.

Variations and extensions. We mention here several ideas that we tried out, but were surprised to find did not improve the empirical results.

- *Asset screening.* We construct stat-arbs using assets only within an industry or sector.
- *Validation.* We split past asset prices into a training and a test set. We find candidate stat-arbs using the training data and then test them on the test data. We then only trade those with good test performance.
- *Incorporating transaction costs in the trading policy.* We modify the linear policy to take into account transaction costs. (Our simulations take trading cost into account, but our simple linear trading policy does not.)
- *Hysteresis-based trading.* We use a hysteresis-based trading policy, which can help reduce transaction costs compared to the linear policy.

Trading a portfolio of stat-arbs. We have focussed on finding individual stat-arbs. A next obvious topic is how to trade a portfolio of stat-arbs. This will be addressed in an upcoming paper by the authors. We simply note here that the results presented in this paper are all for single stat-arbs, and so fall somewhere in between individual assets and a full portfolio.

Chapter 5

Managing a dynamic basket of moving-band stat-arbs

5.1 Introduction

We consider the problem of managing a portfolio of statistical arbitrages (stat-arbs), where each stat-arb is a mean-reverting portfolio of a subset of an n-asset universe. Trading strategies based on stat-arbs have been around since the 1980s and are popular due to their documented success in practice. The strategy is based on the idea that if the price of a portfolio of assets is mean-reverting, we can profit by buying the portfolio when it is cheap and selling it when it is expensive. The literature on stat-arbs is vast, and typically focuses on one of two aspects: (i) finding a linear combination of asset prices that is mean-reverting, or (ii) finding a trading policy that profits by exploiting the mean-reversion. To the best of our knowledge, there exists no satisfying solution to the problem of managing a portfolio of multiple stat-arbs, that is independent of the process of finding them. This paper aims to fill this gap and proposes a solution based on a Markowitz optimization approach. In particular, we focus on the case of moving-band stat-arbs (MBSAs), recently introduced in [164], where the midpoint of the price band is allowed to move over time.

5.1.1 Related work

Stat-arbs. Stat-arb strategies have been popular ever since their introduction by Nunzio Tartaglia in the 1980s [259, 114]. The first strategy was based on pairs trading, where the price difference between two assets is tracked and positions entered when this difference deviates from its mean. Its success has been demonstrated in numerous empirical studies in various markets like equities [12], commodities [237, 298], and currencies [107]; see, *e.g.*, [114, 12, 256, 145, 178, 62, 152, 87]. The general setting of a stat-arb is a portfolio of (possibly more than two) assets that exhibits a mean-reverting behavior [106, §10.5]. The literature on stat-arbs tends to split into several categories: finding stat-arbs, modeling the (mean-reverting) portfolio price, and trading stat-arbs. For a detailed overview of the literature, we refer the reader to [177].

MBSA. MBSAs were recently introduced as an extension of the tradition stat-arb, in which the midpoint of the price band is allowed to move over time [164]. Although the method is new, it relies on principles related to Bollinger bands that have been known for decades [40, 41]. The MBSA relies on solving a small convex optimization problem to find a portfolio of (possibly more than two) assets that vary within a moving band. The method has been shown to work well in practice, and is used in the empirical study in this paper.

Trading stat-arbs. With some exceptions, the literature on trading stat-arbs is mostly limited to pairs trading or trading individual stat-arbs, rather than managing a portfolio of several stat-arbs. Several methods are based on co-integration [162, 5, 127, 95, 302]. For example, in [316, 318, 317] the authors consider a (non-convex) optimization problem for finding high variance, mean-reverting portfolios, in a co-integration space of several stat-arb spreads. Their strategy is based on finding a portfolio of spreads, defined by a co-integration subspace, and implemented using sequential convex optimization. In [310, 262, 312] the authors model the spread of a pair of assets as an autoregressive process, a discretization of the Ornstein-Uhlenbeck process, and show how to trade a portfolio of spreads under proportional transaction costs and gross exposure constraints using model predictive control. The authors of [311] use co-integration techniques to find pairs of assets that are mean-reverting, and show how to construct optimal mean-variance portfolios of these pairs.

Our approach differs from the above methods in that we do not assume any particular model of the price process, and we do not use statistical analysis like co-integration tests to find stat-arbs. Instead, we assume several stat-arbs (defined below as a portfolio and a price signal) are given, and the problem is to find the optimal allocation to these stat-arbs. In other words, we decouple the problem of finding stat-arbs from the problem of portfolio construction.

5.1.2 Outline

The rest of this paper is structured as follows. In §5.2 we review MBSAs and set our notation. In §5.3 we show how to manage and evaluate a dynamic basket of MBSAs. We present an empirical study of the method on recent historical data in §5.4, and give conclusions in §5.5.

5.2 Moving-band stat-arbs

MBSAs were recently introduced in [164]. We start with a universe of n assets, with $P_t \in \mathbf{R}^n_{++}$ denoting the price (suitably adjusted) in USD, in period $t = 1, 2, \ldots$.

5.2.1 Midpoint price and alpha

An MBSA is defined by a vector $s \in \mathbf{R}^n$ of asset holdings (in shares), with negative entries denoting short positions. The vector s is typically sparse, with only a modest number of nonzero entries, but that will not affect our method. The MBSA price in period t is $p_t = s^T P_t$. The MBSA *midpoint price* is given by the trailing M-period average price,

$$\mu_t = \frac{1}{M} \sum_{\tau=t-M+1}^{t} p_\tau, \tag{5.1}$$

where M is the rolling window memory. For a good MBSA, the difference of the price and midpoint price, $p_t - \mu_t$, will oscillate in a band; we obtain a profit by buying when the price difference is low and selling when it is high. We associate with the MBSA the 'alpha' value

$$\alpha_t = \mu_t - p_t. \tag{5.2}$$

If α_t is positive, we expect the price of the MBSA to increase, and if it is negative, we expect the price to decrease.

5.2.2 MBSA lifetime

In [164] we show how to discover MBSAs (*i.e.*, the vector s) by approximately solving a constrained variance maximization problem, using the convex-concave procedure [282, 194].

Each MBSA has a lifetime, starting at creation or discovery time $t = d$ and ending at time $t = e > d$, when we choose to decommission it. We say it is active over the periods $t = d, d+1, \ldots, e$. (See [164, §2.3] for details.)

5.2.3 Multiple MBSAs

We consider a collection of multiple active MBSAs that changes over time as new ones are discovered and some existing ones are decommissioned. We denote K_t as the number of MBSAs active during period t, and index the MBSAs as $k = 1, \ldots, K_t$. The kth MBSA is defined by the holdings $s^{(k)} \in \mathbf{R}^n$, which gives rise to the price $p_t^{(k)} = (s^{(k)})^T P_t$, and alpha

$$\alpha_t^{(k)} = \mu_t^{(k)} - p_t^{(k)},$$

where $\mu_t^{(k)}$ is the moving midpoint of the kth MBSA. We collect the holdings of the K_t MBSAs in a matrix

$$S_t = [s^{(1)} \ \cdots \ s^{(K_t)}] \in \mathbf{R}^{n \times K_t},$$

and introduce the K_t-vectors

$$p_t = (p_t^{(1)}, \dots, p_t^{(K_t)}), \qquad \mu_t = (\mu_t^{(1)}, \dots, \mu_t^{(K_t)}), \qquad \alpha_t = (\alpha_t^{(1)}, \dots, \alpha_t^{(K_t)}),$$

i.e., we extend the notation introduced in the previous paragraph to the setting of multiple MBSAs.

5.3 Managing a dynamic basket of MBSAs

5.3.1 Portfolio holdings and trades

Arb-level and asset-level portfolio holdings. The (MBSA) holdings during period t are denoted by $q_t \in \mathbf{R}^{K_t}$. (The units of q_t are 'shares' of the MBSAs.) The corresponding asset-level holdings are $h_t = S_t q_t \in \mathbf{R}^n$ (in shares), or, in USD, $P_t \circ (S_t q_t)$, where $\circ$ denotes the elementwise (Hadamard) product.

The cash account is denoted by $c_t \in \mathbf{R}$ (in USD). Since we will use the cash account as part of our collateral for short positions, we will assume $c_t > 0$ for all t, *i.e.*, we do not borrow cash. The total portfolio value at time t is then $p_t^T q_t + c_t$ in USD.

Trading cost. We denote the (asset level) trades vector at time t as $z_t = h_t - h_{t-1}$, *i.e.*, the change in asset-level holdings from time $t-1$ to time t, in shares. The trading cost at time t is given by

$$(\kappa_t^{\text{trade}})^T |z_t|,$$

where $\kappa_t^{\text{trade}} \in \mathbf{R}_+^n$ is the vector of one half the bid-ask spread for each asset at time t, in USD per share, and the absolute value is elementwise.

Holding cost. The holding cost is given by

$$(\kappa_t^{\text{short}})^T (-h)_+,$$

where $\kappa_t^{\text{short}} \in \mathbf{R}^n$ is the vector of shorting rates for each asset over period t, in USD per share per trading period. Here $(u)_+ = \max\{u, 0\}$ is the nonnegative part of u, and in the expression above it is applied elementwise.

5.3.2 Markowitz objective and constraints

We manage a dynamic basket of MBSAs using a Markowitz optimization approach [208, 210, 44], maximizing the expected return adjusted for transaction and holding costs, subject to constraints on the portfolio holdings which include a risk limit. We assume that we know the current asset prices P_t, the arb-to-asset transformation matrix S_t, and the previous portfolio holdings h_{t-1} (and q_{t-1}) and cash value c_{t-1}. Our goal is to find the new portfolio holdings h_t (and q_t and c_t). We denote the candidate values, which are optimization variables, as h, q, and c, respectively.

Objective. The objective is to maximize the alpha exposure, adjusted for transaction and shorting costs,

$$\alpha_t^T q - \gamma^{\text{trade}}(\kappa_t^{\text{trade}})^T(h - h_{t-1}) - \gamma^{\text{short}}(\kappa_t^{\text{short}})^T(-h)_+,$$

where γ^{trade} and γ^{short} are positive parameters trading off the alpha exposure, transaction cost, and holding cost. The true values of κ_t^{trade} and κ_t^{short} are not known at time t, but are estimated from historical data. This objective is a concave function of the variables h and q.

Cash-neutrality. We assume that the portfolio is cash-neutral [44, 130], *i.e.*,

$$p_t^T q = 0. \tag{5.3}$$

In other words, the total portfolio value is equal to the value of the cash account. (We can also add a limit on the market exposure (or beta), but we found empirically that this makes a small difference on top of the cash neutrality constraint.) This constraint is often used in long-short trading strategies [130, Chap. 15]. The cash neutral constraint is a linear equality constraint on the variable q.

Collateral constraint. We require that

$$P_t^T(h)_+ + (c)_+ \geq \eta(P_t^T(h)_- + (c)_-),$$

where $\eta \geq 1$ is a parameter. Here $(u)_- = (-u)_+ = \max\{-u, 0\}$ is the nonpositive part of u, applied elementwise to h above. This constraint ensures that the long position is at least η times the short position. In the form above, it is not a convex constraint, but subtracting $(P_t^T(h_t)_- + (c_t)_-)$ from both sides yields the equivalent convex constraint

$$P_t^T h + c \geq (\eta - 1)(P_t^T(h)_- + (c)_-).$$

The cash-neutrality constraint implies $P_t^T h = 0$, so this simplifies to

$$c \geq (\eta - 1)P_t^T (h)_-,$$

i.e., the cash account is at least $(\eta - 1)$ times the short position. For example, we can set $\eta = 2.02$ to keep a collateral of 102% of the short position, as has typically been required by regulation [73, 117].

MBSA size limit constraints. We add the constraints

$$|q|_k (p_t)_k \leq \xi_t^{(k)} c, \quad k = 1, \ldots, K_t,$$

i.e., the absolute value of the position in the kth MBSA is at most $\xi_t^{(k)}$ times the total portfolio value (which is the same as the cash account), where $\xi_t^{(k)}$ is a parameter. These constraints serve two purposes. First, they act as a form of regularization in case there are more MBSAs than assets, in which case the covariance matrix in the risk constraint (defined below) will be ill-conditioned or singular. Second, they limit extreme positions in any single MBSA. During the active period $t = d, d+1, \ldots, e$ of an MBSA, we set $\xi_t^{(k)} = \xi$ for $t = d, d+1, \ldots, e-l$, and then linearly decrease $\xi_t^{(k)}$ to zero over the next l periods, where l is a parameter, say $l = 21$ periods, and ξ is a parameter common to all MBSAs. In other words we use the size limit constraints to gradually decommission MBSAs. These constraints are linear inequality constraints on the variables q and c, and so are convex.

Risk constraint. The traditional definition of risk of a portfolio is the variance of the portfolio return, expressed as a quadratic form of the asset weights with an estimated asset return covariance matrix. Taking the squareroot we obtain the standard deviation of portfolio return, *i.e.*, its volatility, which we can convert to USD by multiplying by the portfolio value.

Here we propose a different risk measure, which we have found empirically to work better, and is more in line with the spirit of MBSAs. Our risk is based on fluctuation of the portfolio value around its expected midpoint, and is directly expressed in USD. Recall that the portfolio value is $p_t^T q_t$, which we expect to fluctuate around the midpoint value $\mu_t^T q_t$, both of these in USD. We take the risk to be an estimate of the short term mean square value of the difference of the portfolio value and the midpoint value, $(p_t - \mu_t)^T q_t$. We express this mean-square value (now using q, not q_t) as

$$q^T \Sigma_t q,$$

where Σ_t is an average of the recent values of $(p_\tau - \mu_\tau)(p_\tau - \mu_\tau)^T$. We limit our risk using the constraint

$$\|\Sigma_t^{1/2} q\|_2 \leq \sigma c,$$

where $\sigma > 0$ is a parameter setting the target risk (which is unitless) and c is the portfolio value. This risk limit is a convex constraint in the variables q and c, specifically a second-order cone (SOC) constraint [50, §4.4.2].

The covariance matrix Σ_t can be estimated in many ways; see, *e.g.*, [163]. We express it in terms of asset prices as

$$\Sigma_t = S_t^T \Sigma_t^P S_t,$$

where $\Sigma_t^P \in \mathbf{R}^{n\times n}$ is an estimate of the short-term covariance matrix of the asset prices. We estimate Σ_t^P as follows. First, define the centered price vectors

$$\tilde{P}_t = P_t - \bar{P}_t, \quad t = 1, \ldots, T,$$

where $\bar{P}_t$ is the M-period rolling window mean of the asset prices. (This is the same M used to define the MBSA midpoints in (5.1).) Then, Σ_t^P is an average of the recent values of $\tilde{P}_\tau \tilde{P}_\tau^T$. If a linear estimator, like a rolling window or exponentially weighted moving average (EWMA), is used to estimate this expectation, then this is equivalent to estimating Σ_t directly using the same method, *i.e.*, to center the MBSA prices and then compute an average of the outer product of the centered prices. We will use the iterated EWMA predictor [163, 18], to estimate the covariance matrix of the centered prices $\tilde{P}_t$. (This is not equivalent to estimating Σ_t directly with an iterated EWMA, but in practice very similar.)

5.3.3 Markowitz formulation

We assemble the objective and constraints described above. We take h_t, q_t, and c_t, as optimal values of h, q, and c in the optimization problem

$$\begin{array}{ll} \text{maximize} & \alpha_t^T q - \gamma^{\text{trade}} (\kappa_t^{\text{trade}})^T (h - h_{t-1}) - \gamma^{\text{short}} (\kappa_t^{\text{short}})^T (-h)_+ \\ \text{subject to} & h = S_t q, \quad c \geq (\eta - 1) P_t^T h_-, \quad c = c_{t-1} + p_t^T q_{t-1}, \\ & p_t^T q = 0, \quad |q|_k (p_t)_k \leq \xi_t^{(k)} c, \quad k = 1, \ldots, K_t, \\ & \|\Sigma_t^{1/2} q\|_2 \leq \sigma c, \end{array} \tag{5.4}$$

with variables $h \in \mathbf{R}^n$, $q \in \mathbf{R}^{K_t}$, and c. The problem data are:

- h_{t-1}, q_{t-1}, c_{t-1}, the previous portfolio holdings at the asset-level and arb-level, and cash;
- κ_t^{trade} and κ_t^{short}, (predictions of) the trading and holding costs;
- S_t, the arb-to-asset transformation matrix;
- η, the collateral parameter;
- P_t and p_t, the asset and MBSA prices;

- $\xi_t^{(k)}$, the MBSA size limits;
- Σ_t, a prediction of the short-term covariance matrix of the MBSA prices;
- σ, a target risk, expressed as a fraction of the portfolio value.

The problem (5.4) is a convex optimization problem, more specifically one that can be transformed to a second-order cone program (SOCP). The solution to this optimization problem gives the new MBSA portfolio q_t, the asset-level portfolio holdings h_t, and the cash account c_t at time t.

Extensions and variations. We can add additional constraints, such as a leverage constraint, asset position limits, maximum market exposure, *etc.* [44]. We can also soften some constraints, if it is not critical that they are satisfied exactly and softening improves performance [44]. For example, to soften the arb-to-asset constraint, we remove the constraint $h = S_t q$ and add a penalty term $\gamma^{\text{hold}} \|P_t \circ (h - S_t q)\|_1$ to the objective. (Note that softening the arb-to-asset constraint implicitly also softens the risk and cash-neutrality constraints if these are expressed in terms of the arb-level portfolio q.) Softening some constraints allows the optimizer to choose values that violate the original hard constraints when necessary, which can reduce unnecessary trading, and therefore transaction cost.

5.4 Numerical experiments

We illustrate the MBSA portfolio management strategy on recent historical data. Code to replicate the experiments is available at

`https://github.com/cvxgrp/cvxstatarb`.

5.4.1 Data and parameters

Data set. We gather daily price data from the CRSP US Stock Databases using the Wharton Research Data Services (WRDS) portal [308]. The data set consists of adjusted asset prices of 15405 stocks from January 4, 2010, to December 30, 2023, for a total of 3282 trading days.

Monthly search for MBSAs. We search for MBSAs every 21 trading days, with the same setup and parameters as described in detail in [164, §4.1].

Dynamic management of the MBSAs. Every day we solve the Markowitz optimization problem (5.4) to rebalance our portfolio. Each MBSA is kept in the portfolio for 500 trading days. After that ξ is reduced linearly to zero over the next 21 trading days.

Risk model. As described in §5.3.2, we decompose the covariance matrix as

$$\Sigma_t = S_t^T \Sigma_t^P S_t,$$

where Σ_t^P is the short-term covariance matrix of the asset prices. The covariance matrix Σ_t^P is estimated using the iterated EWMA (IEWMA) predictor (see [92] and [163, Ch. 2.5]) on the centered prices $\tilde{P}_t = P_t - \bar{P}_t$, where $\bar{P}_t$ is the 21-day rolling window mean of the asset prices. For the IEWMA predictor, we use a 125-day half-life for volatility estimation, and a 250-day half-life for correlation estimation. To reduce trading induced by the risk-model, we smooth the covariances with a 250-day half-life EWMA [163, Chap. 9.2].

Parameters. For the MBSA problem we use the same parameters as in [164, §4.1]. For the Markowitz optimization problem we use $\gamma^{\text{trade}} = 1$, $\eta = 1$, $\xi = 1$, and $\sigma^{\text{tar}} = 10\%$ annualized.

Trading and shorting costs. Our numerical experiments take into account transaction costs, *i.e.*, we buy assets at the ask price, which is the (midpoint) price plus one-half the bid-ask spread, and we sell assets at the bid price, which is the price minus one-half the bid-ask spread. We use 0.5% as a proxy for the annual shorting cost of stocks, which is well above what is typically observed in practice for liquid stocks [73, 117, 173, 86]. We also note that we have tested the method with shorting costs upwards of 10% annually, and it remains profitable.

5.4.2 Simulation and metrics

This section explains how we simulate the portfolio, and the metrics we use to evaluate the performance.

Cash account. We initialize the portfolio with a cash account at time $t = 0$, $c_0 = 1$. The cash account evolves as

$$c_{t+1} = c_t - (q_{t+1} - q_t)^T p_{t+1} - \phi_t = c_t + q_t^T p_{t+1} - \phi_t, \quad t = 0, \ldots, T,$$

where ϕ_t is the transaction and holding cost, consisting of the trading cost at time $t+1$ and the holding cost over period t. (The last equality follows from the cash-neutrality constraint.)

Portfolio NAV. The net asset value (NAV) of the portfolio, including the cash account, at time t is

$$V_t = c_t + q_t^T p_t = c_t.$$

Due to the market neutrality constraint in (5.4) we have $V_t = c_t$, *i.e.*, the NAV is equal to the cash account.

Return. The return at time t is

$$r_t = \frac{V_t - V_{t-1}}{V_{t-1}}, \quad t = 1, \ldots, T.$$

We report several standard metrics based on the returns r_t. The average return is

$$\overline{r} = \frac{1}{T}\sum_{t=1}^{T} r_t,$$

which we multiply by 250 to annualize. The return volatility is

$$\left(\frac{1}{T}\sum_{t=1}^{T}(r_t - \overline{r})^2\right)^{1/2},$$

which we multiply by $\sqrt{250}$ to annualize. The annualized Sharpe ratio is the ratio of the annualized average return to the annualized volatility. Finally, the maximum drawdown is

$$\max_{1 \le t_1 < t_2 \le T} \left(1 - \frac{V_{t_2}}{V_{t_1}}\right),$$

the maximum fractional drop in value form a previous high.

Turnover. The turnover at time t is

$$T = \frac{1}{2}\sum_{i=1}^{n} |((P_t \circ h_t)_i - (P_t \circ h_{t-1})_i)/V_t| = \frac{1}{2}\|P_t \circ (h_t - h_{t-1})\|_1/V_t,$$

which we multiply by 250 to annualize. It measures the amount of trading in the portfolio [130, Chap. 16]. For example, a turnover of 0.01 means that the average of total amount bought and total amount sold is 1% of the total portfolio value.

Active return and risk. We define the active return as the return of the portfolio minus the return of the market, which we take to be the S&P 500. The active risk is the standard deviation of the active return.

Residual return and risk. Given the portfolio returns $r_1, r_2, \ldots, r_T$, and the corresponding market returns $r_1^m, r_2^m, \ldots, r_T^m$, we construct the linear model

$$r_t = \beta r_t^m + \theta_t, \quad t = 1, \ldots, T,$$

Return	Volatility	Sharpe ratio	Turnover	Drawdown
19%	12%	1.61	136	15%

Table 5.1: Metrics for portfolio of MBSAs.

Active return	Active risk	Residual return	Residual risk	Beta	Information ratio
8%	20%	18%	11%	11%	1.53

Table 5.2: Comparison of portfolio of MBSAs to the market.

where βr_t^m is the return explained by the market (with $\beta \in \mathbf{R}^n$), and $\theta_t \in \mathbf{R}^n$ is the residual return at time t. The residual risk is the standard deviation of the residual return. The mean residual return is often referred to as the alpha of the portfolio [130] (not to be confused with the alpha of an MBSA). The information ratio is the ratio of the portfolio alpha to the residual risk.

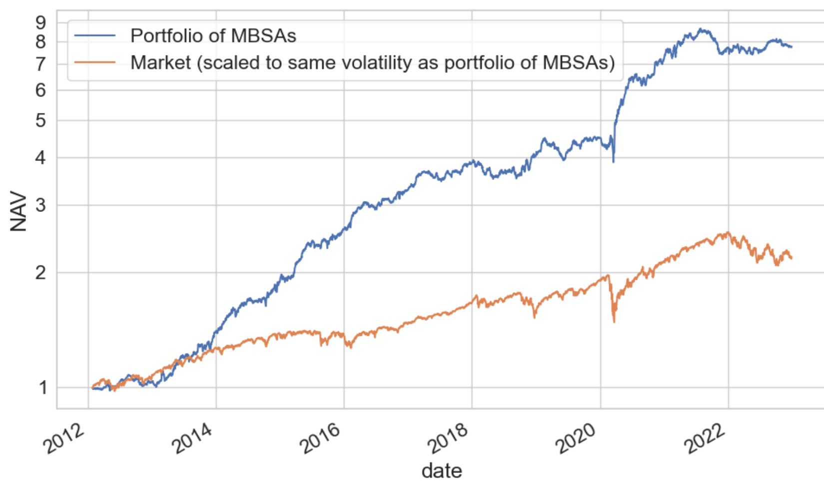
5.4.3 Results

Portfolio performance. The portfolio performance is summarized in table 5.1. We attain an average annual return of 19% at an annual volatility of 12%, corresponding to a Sharpe ratio of 1.61, with a maximum drawdown of 15% over the roughly 10-year period. The average turnover is 136, which corresponds to a daily turnover of roughly 50%. In comparison to other stat-arb strategies in the literature, this seems like a reasonable level of turnover [134].

Active and residual return and risk. Here we compare the MBSA strategy to the market, represented by the S&P 500 with an initial investment of $1 and diluted with cash to attain the same annualized risk as the MBSA strategy. Table 5.2 gives a numerical summary. The MBSA strategy attains an annualized Sharpe ratio of 1.61 (compared to 0.66 for the market) with a residual return (alpha) of 18% and a market beta of 11%. The information ratio is 1.53.

NAV evolution. Figure 5.1 shows the NAV of the MBSA strategy and the market, and the 250-day half-life EWMA correlation between the two.

As seen, the MBSA strategy outperforms the market, with very low correlation, 15% over the whole period. This suggests mixing the MBSA strategy with the market. For example, mixing 90% of the MBSA strategy and 10% of the market yields an annualized Sharpe ratio of 1.66, slightly better than the MBSA strategy alone.



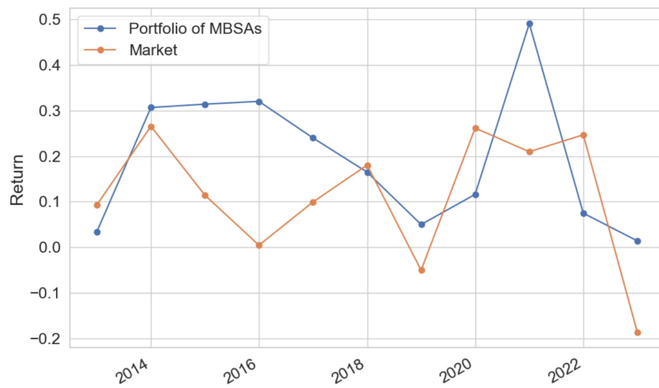
(a) NAV evolution.

(b) 250-day half-life EWMA correlation to market.

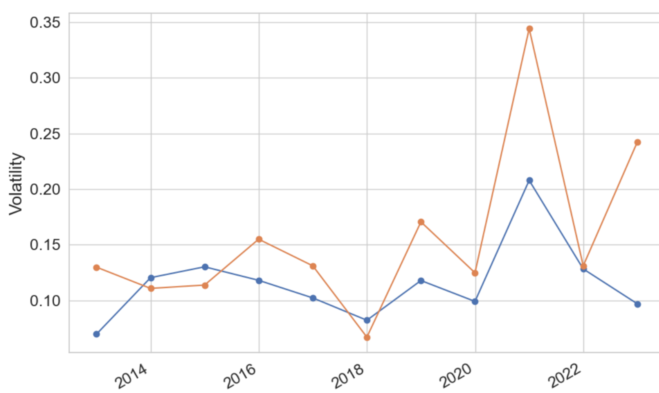
Figure 5.1: A comparison summary of the dynamically managed portfolio of MBSAs to the market.

Annual performance. Here we break down the metrics reported above over the 11-year period into the performance metrics for each year. Figure 5.2 shows the annual performance of the MBSA strategy and the market. The MBSA strategy has positive return in each of the 11 years, whereas the market has negative return in 2 of the 11 years. The MBSA strategy outperforms the market in 8 of the 11 years, and has more stable performance.

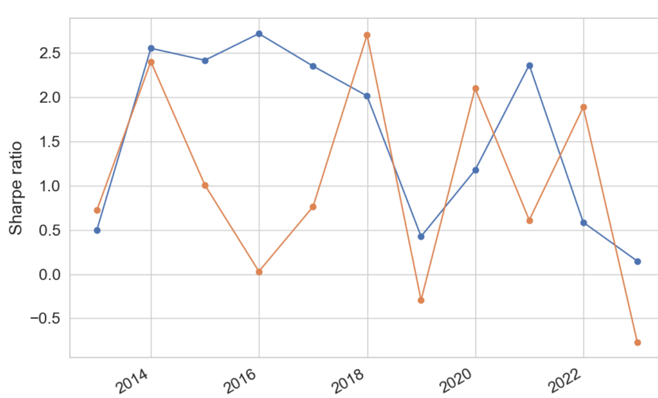
Finally, figure 5.3 shows the annual residual return and risk, and market beta. As seen, most of the MBSA success is unexplained by the market.



(a) Annual returns.

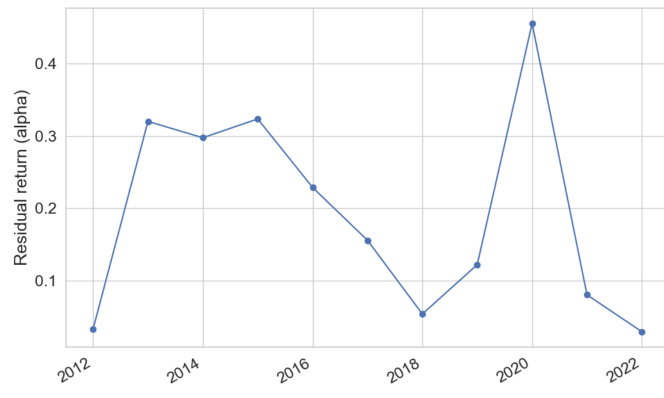


(b) Annual volatilities.

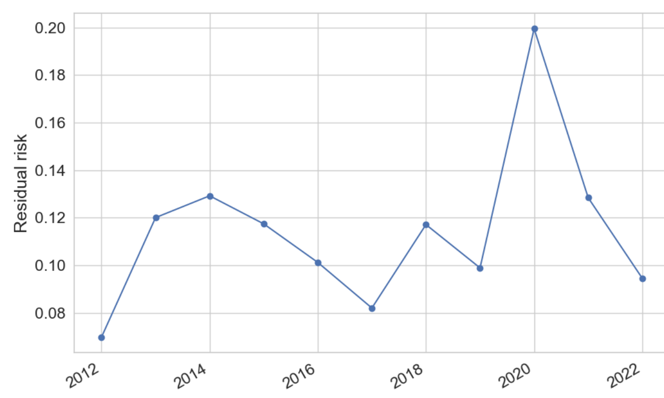


(c) Annual Sharpe ratios.

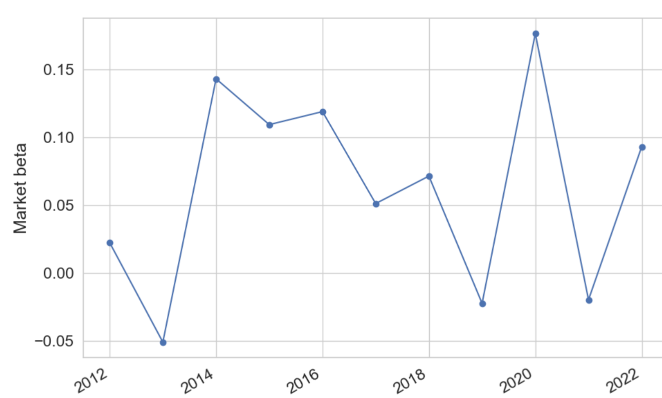
Figure 5.2: Annual portfolio performance.



(a) Annual residual returns (alphas).



(b) Annual residual risks.



(c) Annual market betas.

Figure 5.3: Annual residual return and risk, and market beta.

5.5 Conclusion

We have shown how to manage a dynamic basket of moving-band stat-arbs, based on a long-short Markowitz optimization strategy. We presented an empirical study of the method on recent historical data, showing that it can to outperform the market with low correlation.

Chapter 6

Simple and effective portfolio construction with crypto assets

6.1 Introduction

Since the introduction of cryptocurrencies in 2009 [238], the field of crypto trading has rapidly grown. In this note we consider the problem of constructing a portfolio of assets, including a combination of traditional financial assets such as stocks and bonds with crypto assets such as Bitcoin and Ethereum. Our conclusion is that despite the well known extreme swings of crypto currency values, simple standard portfolio construction methods suffice to realize the benefits of including crypto assets in a portfolio.

Figure 6.1 shows the normalized prices of Bitcoin (BTC), Ethereum (ETH), and the S&P 500 index (SP500) over the past six years. To the eye it reasonably seems that crypto returns are quite different from traditional returns. Table 6.1 lists some metrics for these asset returns over the same period, which also suggest that crypto asset returns fundamentally differ from traditional asset returns. For example, ETH at one point dropped in value by a factor of almost $20\times$, while the maximum drop in value of SP500 is only one third.

Stylized facts of crypto returns, such as extreme volatility, heavy tails, excess kurtosis, and

Table 6.1: Performance metrics for BTC, ETH, and SP500.

Metric	BTC	ETH	SP500
Return (%)	43.3	47.0	13.7
Volatility (%)	58.1	71.6	19.5
Sharpe	0.75	0.66	0.70
Drawdown (%)	83.3	93.9	33.9

Figure 6.1: Normalized prices of BTC, ETH, and SP500.

skewness, have been well documented [195, 148, 170]. Figure 6.2 shows quantile-quantile (QQ) plots of the log returns of BTC, ETH, and SP500 over the last six years. All three return distributions have tails that deviate significantly from the normal distribution, with the crypto asset returns exhibiting more extreme tail behavior than the market index. The 1st and 99th return percentiles are shown in the plots as dashed green lines; here too we see that crypto asset returns have considerably bigger tails than the market index, even when normalized to have the same volatility.

The documented characteristics of crypto asset returns have made conventional investors hesitant to include them in their portfolios, due to perceptions of risk and unpredictability. While asset managers may have additional concerns, such as legislative risk and other fundamental factors like the ongoing debate about the legitimacy of crypto as a real asset [313, 182, 181, 67], we will focus here on the concerns related to the statistical properties of crypto asset returns.

These characteristics of crypto asset returns have also led to a wealth of research on how to extend traditional portfolio construction methods to include crypto assets, including methods similar to those used for traditional assets [146, 56, 59, 149], as well as more complex machine learning driven methods [270, 160, 203, 161, 265, 270].

Despite the documented differences between crypto and conventional asset returns, some authors have argued that the two asset classes are fundamentally similar [247, Chap. 2], even though the crypto asset returns and volatilities are much higher. We agree with this perspective. In this note we show a simple method for constructing a portfolio of traditional and crypto assets using a risk allocation framework, (hopefully) debunking the idea that novel and complex machine learning

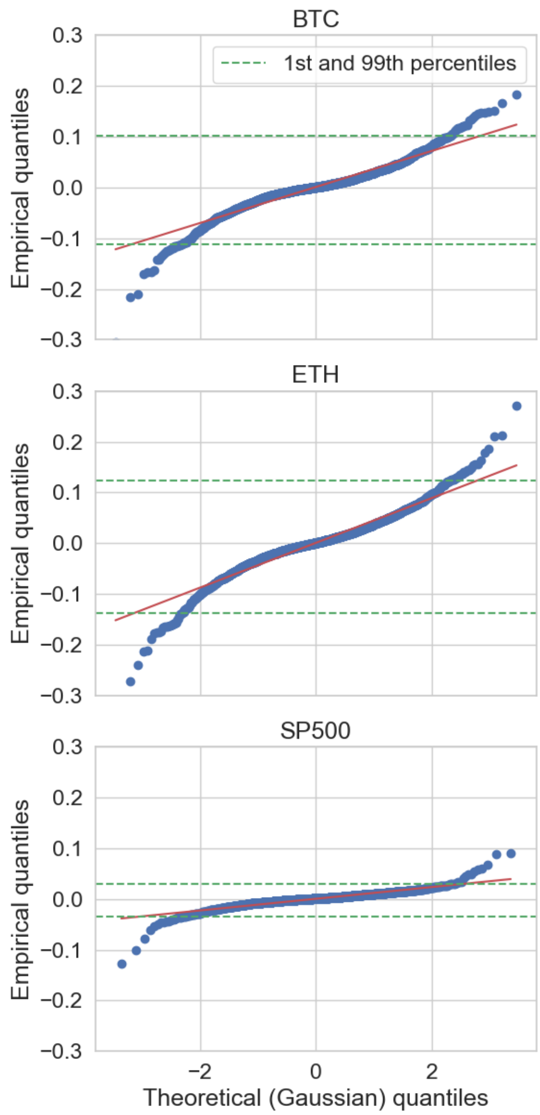


Figure 6.2: Quantile-quantile plot of log returns of BTC, ETH, and SP500.

approaches are necessary to manage a portfolio that includes crypto assets. Based on a back-test of the risk allocation method, we propose an even simpler portfolio construction method, reminiscent of the traditional 60/40 stocks/bonds split, which consists of a 90/10 split of traditional and crypto assets, followed by dynamic (time-varying) dilution with cash, to achieve a given ex-ante risk. We refer to this simple portfolio as DD90/10.

Outline. In §6.2 we review previous and related work. We introduce in §6.3 an approach for constructing a portfolio that combines traditional and crypto assets within a risk allocation framework. We illustrate the performance of this method on historical data in §6.4. Finally, in §6.5 we propose the DD90/10 portfolio allocation strategy, and show that it has performance similar to the risk allocation strategy.

6.2 Related work

6.2.1 Stylized facts of financial return data

Stylized facts of financial return series refer to a set of empirical observations that are consistently observed across different financial markets and asset classes. We review some of the most prominent stylized facts for financial assets, and refer the reader to [247, Chap. 2] for a more comprehensive overview.

Equities. Stylized facts for equity returns include the non-normal distribution of returns, characterized by heavy tails and excess kurtosis [207, 58]. Another fact is volatility clustering, where large price movements tend to be followed by additional large movements, regardless of direction. While returns themselves are generally uncorrelated over time, the absolute or squared returns often display strong autocorrelation, highlighting a pattern in the magnitude of fluctuations. Furthermore, the leverage effect is a notable feature, where negative returns increase future volatility to a greater extent than positive returns of the same size. Equity return distributions have also been shown to exhibit asymmetry between positive and negative returns [159].

Cryptocurrencies. Cryptocurrencies exhibit several patterns similar to traditional financial assets, but with more extreme behavior [148, 119]. They are highly volatile, with large price fluctuations and heavy-tailed return distributions. Volatility tends to cluster, with periods of high volatility followed by more volatility. While returns show no autocorrelation in the short term, return magnitudes do exhibit autocorrelation. Cryptocurrencies also exhibit asymmetry in returns [170].

6.2.2 Portfolio construction

Portfolio construction involves selecting a combination of assets by balancing the trade-off between expected return and portfolio risk. Here we discuss key historical contributions to the field, along with some recent advancements. For more extensive overviews, refer to texts such as [130, Chap. 14], [240, Chap. 6], and the studies by [69, 176, 63, 135, 45].

Markowitz portfolio construction. Before Markowitz's seminal work in 1952, portfolio construction was largely based on heuristics and rules of thumb. Markowitz introduced a quantitative framework for portfolio construction, where the return of a portfolio is modeled as a random variable, and the expected return is maximized for a desired level of risk [209]. Despite its simplicity, and having over 70 years of history, the Markowitz model remains the foundation of quantitative investing to this day [45].

Extensions of the Markowitz model include the Black-Litterman model [33, 34, 32], fully flexible views [226], and conditional value-at-risk (CVaR) optimization [266], to name a few. The Black-Litterman model is a Bayesian approach to portfolio construction, where the prior distribution is based on the equilibrium market portfolio, and the posterior distribution is updated with user-specific views on the expected returns of the assets. Fully flexible views is a generalization of Black-Litterman, allowing for nonlinear views on the returns. CVaR optimization replaces the variance in the Markowitz model with the conditional value-at-risk of the portfolio, penalizing the tail risk of the portfolio, and directly addresses the issue of a non-normal return distribution.

Machine learning based portfolio construction. Typically, portfolio construction is split into two parts: data modelling and portfolio optimization [247, Chap. 1]. Data modelling is concerned with predicting the expected returns and covariances of the assets, and portfolio optimization concerns selecting a portfolio of assets that trades off expected return and risk. However, with the growing popularity of machine learning, and documented criticism of the Markowitz model [228], in recent years several studies have proposed machine learning based portfolio construction methods as alternatives to this traditional framework [137, 19]. In particular, it has become popular to combine the two parts of portfolio construction into a single end-to-end machine learning model, where market features are fed into the model, and the model outputs a trade list. The argument for this has been that splitting portfolio construction into two parts is suboptimal, as there are uncertainties in the return forecasts that are not accounted for in the portfolio optimization component of portfolio construction [171, §5]. Although theoretically appealing, we have yet to see a wide-spread adoption of these methods in practice, and the Markowitz model remains the dominant framework for portfolio construction [45]. We refer the reader to [171, §5] for a detailed review of end-to-end machine learning models for portfolio construction.

Risk-based portfolio construction. Risk-based portfolio construction methods rely on estimates of the asset return covariances, but do not require an estimate of the expected returns of the assets, making them attractive for practitioners who do not have access to data sources for estimating expected returns reliably. A trivial example of a risk-based portfolio construction method is the equally weighted portfolio. Another popular portfolio is the minimum variance portfolio, which is also the mean-variance efficient portfolio when expected returns are equal. Other risk-based portfolio construction methods include risk parity [263], where the risk contribution of each asset is equal, and maximum diversification portfolios [66]. These portfolios can all be computed via convex optimization [163, §4.4], which makes them reliable, fast, and practical [49]. These portfolio construction methods can be implemented in just a few lines of domain specific languages for convex optimization such as CVXPY [82].

Risk and covariance estimation. Risk-based portfolio construction methods rely on estimates of the portfolio risk. There are in general two ways to estimate portfolio risk. The first is to use a realized measure of the variance of the portfolio. There are many such methods, including the exponentially weighted moving average (EWMA), methods based on mean absolute deviation or the rolling median [115, 116], as well as autoregressive conditional heteroskedasticity (ARCH) and generalized ARCH (GARCH) models [91, 36, 93]. The second way to estimate portfolio risk is to leverage a covariance matrix of the asset returns. The literature on covariance estimating in finance is vast, but probably the most popular method is to use an iterated covariance matrix [92, 18]. This method decomposes the covariance Σ as $\Sigma = VRV$, where V is a diagonal matrix with the asset standard deviations on the diagonal and R is the correlation matrix. Typically, V and R are estimated separately, using EWMAs with different half-lives. We will refer to this method as the iterated EWMA (IEWMA) [163, §2.5]. It is also possible to dynamically adjust the half-lives of the EWMAs, to account for time-varying market conditions [163].

6.2.3 Crypto trading

Several methods to managing portfolios of cryptocurrencies have been proposed, and these tend to be separated from investment strategies proposed for traditional assets.

Diversification. Many studies have noted that the returns of cryptocurrencies are uncorrelated with traditional assets [16, 314, 146]. This means that they can be leveraged to diversify a portfolio of traditional assets, and thus increase the risk-adjusted return of the portfolio [146, 16].

Portfolio construction. Much of the literature on cryptocurrency trading is focused on machine learning or deep learning [270]. In [160] an end-to-end convolutional neural network, taking in raw price data and outputting a trade list, is proposed. A deep Q-learning portfolio management framework is proposed in [203]. The authors of [161] introduce a financial-model-free reinforcement learning framework, incorporating convolutional neural networks, recurrent neural networks, and long short-term memory models. In [265] ARIMA models, convolutional neural networks, and long short-term memory methods are used for cryptocurrency price forecasting and multiple portfolio construction strategies are evaluated with the forecasted prices as input. For a more detailed review of machine learning in cryptocurrency portfolio management, see *e.g.*, [270, §4.2].

Some crypto studies use traditional portfolio construction methods, such as the Markowitz framework. In [146], the authors show that crypto assets can improve the performance of a mean-variance optimized portfolio. The authors of [56] show how to implement a mean-variance optimized portfolio of cryptocurrencies and that it outperforms an equally weighted portfolio as well as single cryptocurrencies. The paper [120] proposes an extension of the Markowitz model, combining random matrix theory and network measures to manage a portfolio of crypto assets. In [258] the authors design crypto portfolios using variance-based constraints in the Black-Litterman model, to account for estimation uncertainties.

Other studies focus on risk-based portfolio construction methods, *i.e.*, those that do not rely on an estimate of the expected returns of the assets. In [59] multiple risk-based portfolio construction methods are evaluated, and the authors find that most of them outperform single cryptocurrencies and the equally weighted portfolio. The authors of [149] find that minimizing the variance and conditional value-at-risk of a portfolio of cryptocurrencies, yields a portfolio that outperforms the market.

6.3 Constrained risk allocation

The most common approach to portfolio construction is to formulate the problem as a trade-off between expected return and risk, as suggested by Markowitz in 1952 [209]. The main practical challenge with this framework is that it requires estimating expected returns of the assets. These are very difficult to estimate and, for obvious reasons, successful estimation techniques are proprietary; large hedge funds and asset managers have entire teams dedicated to estimating expected returns, using data sources for which they pay large premiums [247]. Here we describe a risk allocation approach, which does not require an estimate of the expected returns of the assets. It is based on the idea of risk parity [263], with additional constraints on the portfolio weights and a risk limit. We refer to this method as constrained risk allocation (CRA).

6.3.1 Constrained risk allocation problem

We consider a portfolio of n non-cash assets, plus cash. We denote the asset weights as $w \in \mathbf{R}^n$, with $w \geq 0$, where w_i is the fraction of the total portfolio value held in asset i. We let $c \geq 0$ denote the fraction of the portfolio value held in cash, so we have $\mathbf{1}^T w + c = 1$, where $\mathbf{1}$ is the vector with all entries one. We refer to $\mathbf{1}^T w$ as our asset exposure.

Let Σ denote the $n \times n$ (estimate of the) covariance matrix of the returns. Assuming cash is risk-free, the portfolio risk is $w^T \Sigma w$. (The volatility is the squareroot of this.) Inspired by the identity

$$w^T \Sigma w = \sum_{i=1}^{n} w_i (\Sigma w)_i,$$

we define the risk contribution of asset i as $w_i(\Sigma w)_i$. In risk allocation we specify the fraction of risk to be held in each asset, as the vector $\rho \in \mathbf{R}^n$, with $\rho > 0$ and $\mathbf{1}^T \rho = 1$. We interpret ρ_i as the fraction of total portfolio risk contributed by asset i. (We assume that all entries of ρ are positive; if any were zero, we simply would not include that asset in the portfolio.) Thus we have

$$w_i(\Sigma w)_i = \rho_i w^T \Sigma w, \quad i = 1, \ldots, n. \tag{6.1}$$

The special case $\rho = (1/n)\mathbf{1}$ corresponds to risk parity, where all assets contribute an equal fraction of the risk.

The CRA problem is defined as

$$\begin{array}{ll} \text{minimize} & c \\ \text{subject to} & w \geq 0, \quad c \geq 0, \quad \mathbf{1}^T w + c = 1 \\ & w_i(\Sigma w)_i = \rho_i w^T \Sigma w, \quad i = 1, \ldots, n \\ & w^T \Sigma w \leq \sigma^2, \quad F w \leq g, \end{array} \tag{6.2}$$

where $w \in \mathbf{R}^n$ and $c \in \mathbf{R}$ are the variables, σ^2 is the maximum allowed risk, and $F \in \mathbf{R}^{m \times n}$ and $g \in \mathbf{R}^m$ describe constraints on the portfolio. In words: We choose the portfolio to minimize cash holdings (or equivalently, maximize asset exposure), subject to a given risk allocation, a total risk limit, and some additional constraints on the weights.

We will assume that $g > 0$ and $F \geq 0$ (elementwise), with each row of F nonzero. We will soon see that this implies the CRA problem (6.2) always has a unique solution. The weight constraints can be used to enforce a maximum weight on each asset, or a maximum weight on a subset of assets, *e.g.*, crypto assets.

6.3.2 Solution via convex optimization

The CRA problem (6.2) is not itself a convex optimization problem, but it can be solved efficiently via convex optimization. We first consider the risk allocation constraints (6.1) alone, together with $w \geq 0$. It can be shown that w satisfies these constraints if and only if it has the form

$$w = \alpha x^\star,$$

where $\alpha \geq 0$ and $x^\star \in \mathbf{R}^n$ is the unique solution of the convex optimization problem

$$\text{minimize} \quad (1/2)x^T \Sigma x - \sum_{i=1}^{n} \rho_i \log x_i, \tag{6.3}$$

with variable $x \in \mathbf{R}^n$ (and implicit constraint $x > 0$). See, *e.g.*, [52, §17]. Thus the set of weights that satisfy the risk allocation constraints is a ray, with a direction that can be found by solving a convex optimization problem.

Now we take this very specific form for w and substitute it back into the original CRA problem (6.2), dropping the risk allocation constraints and $w \geq 0$ since they are automatically satisfied. This gives us the problem

$$\begin{array}{ll} \text{minimize} & c \\ \text{subject to} & \alpha \geq 0, \quad c \geq 0, \quad \alpha \mathbf{1}^T x^\star + c = 1 \\ & \alpha^2 (x^\star)^T \Sigma x^\star \leq \sigma^2, \quad \alpha F x^\star \leq g, \end{array}$$

with scalar variables α and c. Note that the quantities $\mathbf{1}^T x^\star$, $(x^\star)^T \Sigma x^\star$, and $F x^\star$ are constants in this problem.

We can solve this simple problem analytically. Minimizing c is the same as maximizing α. Along with $\alpha \geq 0$, all constraints on α are (positive) upper limits:

$$\alpha \leq \frac{1}{\mathbf{1}^T x^\star}, \quad \alpha \leq \frac{\sigma}{((x^\star)^T \Sigma x^\star)^{1/2}}, \quad \alpha \leq \frac{g_i}{(F x^\star)_i}, \quad i = 1, \ldots, m.$$

(Each of the denominators is positive due to our assumptions and $x^\star > 0$.) It follows that the solution is

$$\alpha^\star = \min \left\{ \frac{1}{\mathbf{1}^T x^\star}, \ \frac{\sigma}{((x^\star)^T \Sigma x^\star)^{1/2}}, \ \frac{g_i}{(F x^\star)_i}, \quad i = 1, \ldots, m \right\}. \tag{6.4}$$

Roughly speaking: Scale the unconstrained risk allocation weights as large as possible with all constraints holding.

Summary. The two step solution procedure is summarized as follows:

1. Solve the optimization problem (6.3) to obtain $x^\star$.

2. The unique solution of the CRA problem is then given by $w^\star = \alpha^\star x^\star$ where $\alpha^\star$ is given by (6.4).

We note that Feng and Palomar have suggested a more sophisticated formulation of the CRA problem, in which the risk allocations need only hold approximately [105, 309], [247, Chap. 11]. This formulation can be approximately solved by solving a sequence of convex problems. Our simple formulation is, however, good enough for us to make our main point.

Table 6.2: Performance metrics for the six assets.

Metric	BTC	ETH	Cnsmr	Manuf	HiTec	Hlth
Return (%)	43.5	47.1	14.1	11.4	20.7	10.8
Volatility (%)	58.1	71.6	19.3	20.5	23.9	18.0
Sharpe	0.73	0.60	0.76	0.58	0.90	0.62
Drawdown (%)	83.3	93.9	28.5	42.7	35.4	26.8

Variation. We can modify the way $\alpha^\star$ is computed in (6.4). Instead of estimating the standard deviation of the unconstrained risk allocation portfolio $x^\star$ as $\left((x^\star)^T \Sigma x^\star\right)^{1/2}$, we compute the realized return trajectory of the portfolio $x^\star$, *i.e.*, $(x^\star)^T r_t$, $t = 1, 2, \ldots$, where r_t is the vector of asset returns at time t. We then compute an estimate of the standard deviation of the portfolio return trajectory, using, *e.g.*, a EWMA. Thus to compute the scaling (which sets the cash dilution) we directly estimate the standard deviation of the return trajectory with the unconstrained risk allocation weights, rather than find it from our estimated covariance matrix (which we use to compute the risk allocation weights $x^\star$ in step 1.) We have found that leads to a modest but significant improvement in portfolio performance.

6.4 CRA results

6.4.1 Data and experimental setup

Data. We consider daily close prices of two crypto assets, BTC and ETH, with data from LSEG Data and Analytics. We also consider four daily traded industry portfolios: consumer goods and services, manufacturing and utilities, technology and communications, as well as healthcare, medical equipment, and drugs; these were obtained from Kenneth French's data library [110]. The data spans from September 8th, 2017, to September 22nd, 2024, for a total of 2565 days, or 1729 trading days. (Although crypto assets are traded every day, we rebalance our portfolios only on market trading days; we do, however, realize gains and losses on crypto assets on weekends and holidays.) Figure 6.3 shows the normalized price evolution of the six assets, and table 6.2 lists some metrics for them. The data and code to reproduce the results are available at

`https://github.com/cvxgrp/crypto_portfolio`.

Risk model. We estimate the covariance matrix of the assets using an iterated EWMA, described in detail in [163, §2.5]. We use a 63-day half-life for the volatility estimate and a 125-day half-life for the correlation estimate. The estimated volatilities of the six assets are shown in figure 6.4. Two examples (from different time-periods) of the estimated correlation matrix are shown in figure 6.5.

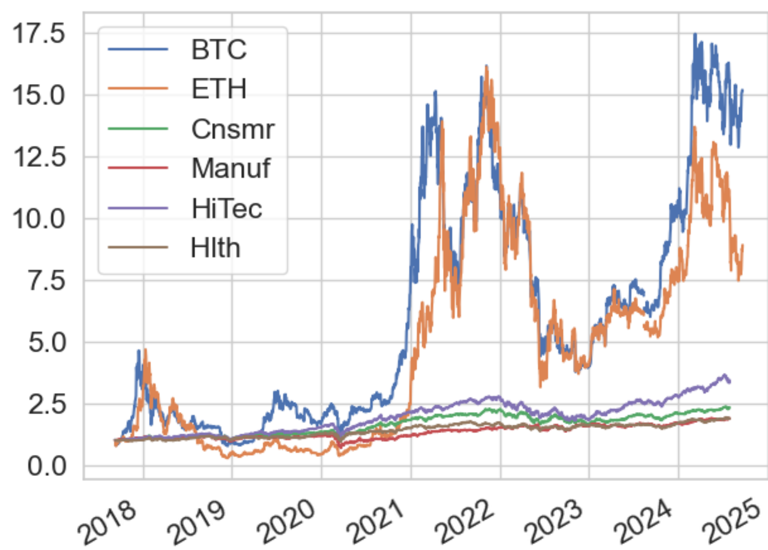


Figure 6.3: Normalized prices of BTC, ETH, and four industry portfolios.

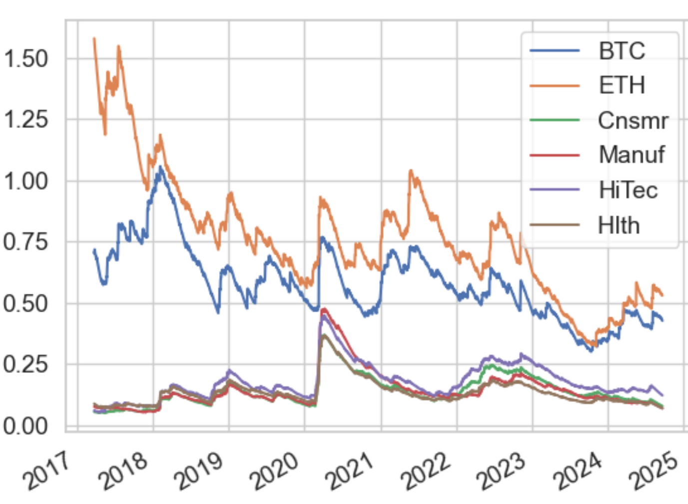
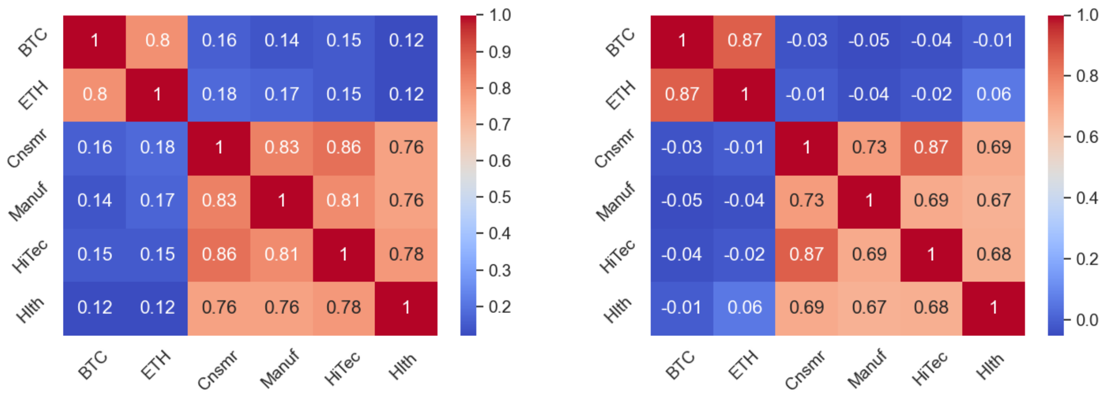


Figure 6.4: Estimated annualized volatilities of the six assets.



(a) March 1st, 2020.

(b) May 1st, 2023.

Figure 6.5: Estimated correlation matrices on two different dates.

Simulation and parameters. We simulate three portfolios.

- *Industries* contains only the four industry portfolios and cash.
- *Crypto* contains only the two crypto assets and cash.
- *Combined* contains all six assets and cash.

We rebalance the portfolios every trading day, using risk parity. We impose a 10% annualized risk limit on the portfolio, *i.e.*, $\sigma = 0.1\sqrt{D}$, where $D = 250$ is the number of trading days in a year. To estimate the risk of the unconstrained risk parity portfolio $x^\star$, we use a 10-day half-life EWMA of the portfolio return. We also impose a 10% maximum weight constraint on crypto assets, *i.e.*, for BTC and ETH combined. These limits were chosen as reasonable values that one might use in practice; the results are not sensitive to these choices.

6.4.2 Metrics

We describe the metrics used to evaluate the performance of the portfolios over the time interval $t = 1, \ldots, T$.

Return. The (realized) return of the portfolio at time t is given by

$$w_t^T r_t,$$

Table 6.3: Portfolio performance metrics.

Metric	Industries	Crypto	Combined
Return (%)	6.0	4.5	8.2
Volatility (%)	8.2	6.0	8.2
Sharpe	0.73	0.75	1.00
Drawdown (%)	12.5	15.9	19.6

where r_t and w_t are the vector of (realized) asset returns and the portfolio weights at time t, respectively. The annualized (realized) return is given by

$$\bar{r} = \frac{D}{T}\sum_{t=1}^{T} w_t^T r_t.$$

Volatility. The (realized) annualized volatility of the portfolio is given by

$$\left(\frac{D}{T}\sum_{t=1}^{T}(r_t - \bar{r})^2\right)^{1/2}.$$

Sharpe ratio. The Sharpe ratio is the ratio of the annualized return to the annualized volatility.

Drawdown. Let V_t denote the portfolio value in time period t, starting from $V_1 = 1$, with returns compounded or re-invested. These are found from the recursion $V_{t+1} = (1 + w_t^T r_t)V_t$, $t = 1, \ldots, T-1$. The (maximum) drawdown of the portfolio is defined as

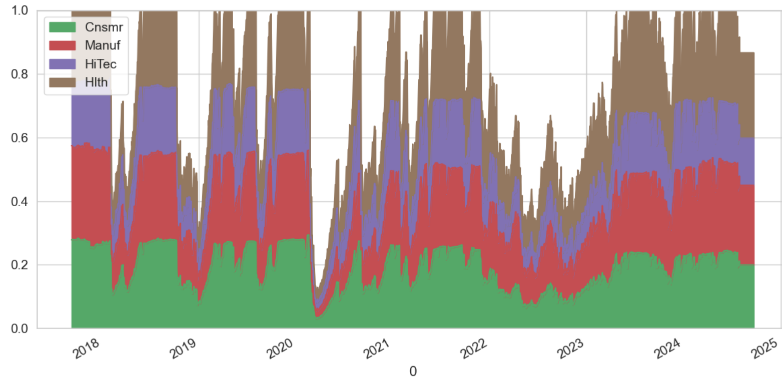
$$\max_{1 \le t_1 < t_2 \le T}\left(1 - \frac{V_{t_2}}{V_{t_1}}\right),$$

i.e., the maximum fractional drop in value from a previous high.

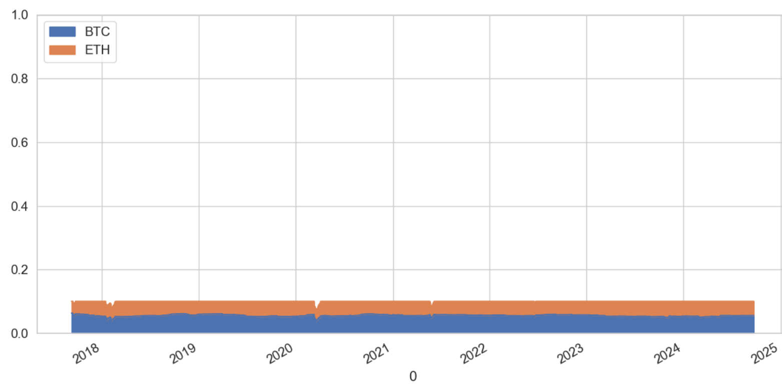
6.4.3 Results

Portfolio weights. The portfolio weights of the three portfolios are shown in figure 6.6. As expected, the crypto portfolio holds mostly cash, due to the 10% crypto limit. The industry and combined portfolios have more diversified weights, varying over time. As expected these portfolios hold a lot of cash during the turbulent 2020 period. On average, the industry portfolio holds 25% cash, the crypto portfolio 90% cash, and the combined portfolio 33% cash.

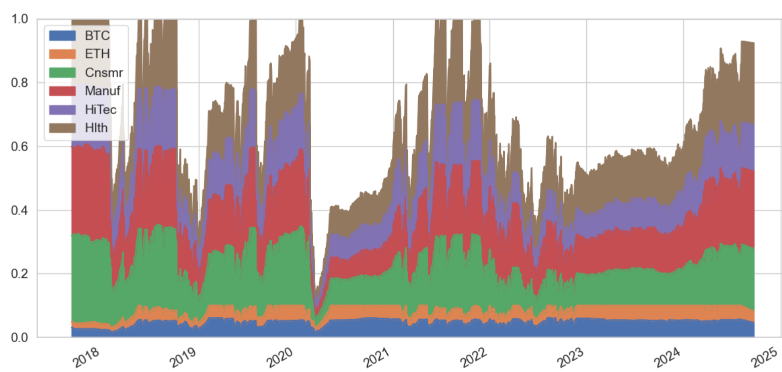
Performance. Figure 6.7 shows the value of the three portfolios over time, normalized to one at the start of the simulation. (SR denotes Sharpe ratio.) The aggregate performance of the three portfolios is shown in table 6.3. Including crypto assets in the portfolio clearly gives a noticeable



(a) Industry portfolio.



(b) Crypto portfolio.



(c) Combined portfolio.

Figure 6.6: Portfolio weights of the three portfolios. The cash weight is shown as uncolored.

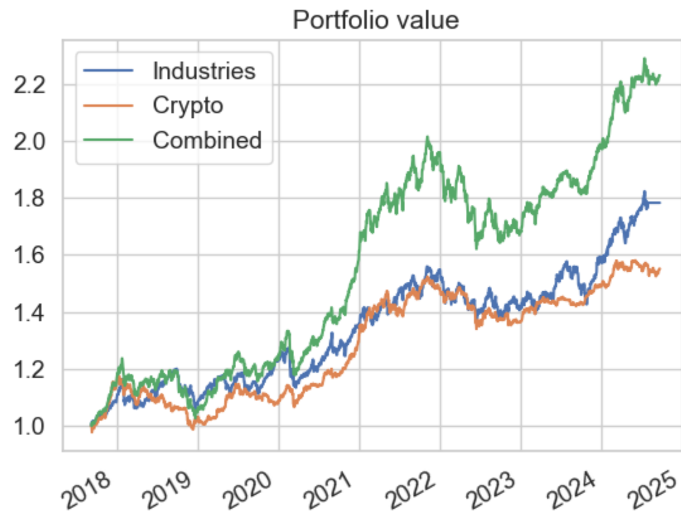


Figure 6.7: Portfolio values of the three portfolios.

boost, despite allocating less than 10% of the portfolio to them. The combined portfolio has a higher return and Sharpe ratio than the industry portfolio, while having similar volatility and drawdown. To get a better understanding of the performance differences, figure 6.8 shows the return, volatility, and Sharpe ratio of the three portfolios over each year of the simulation.

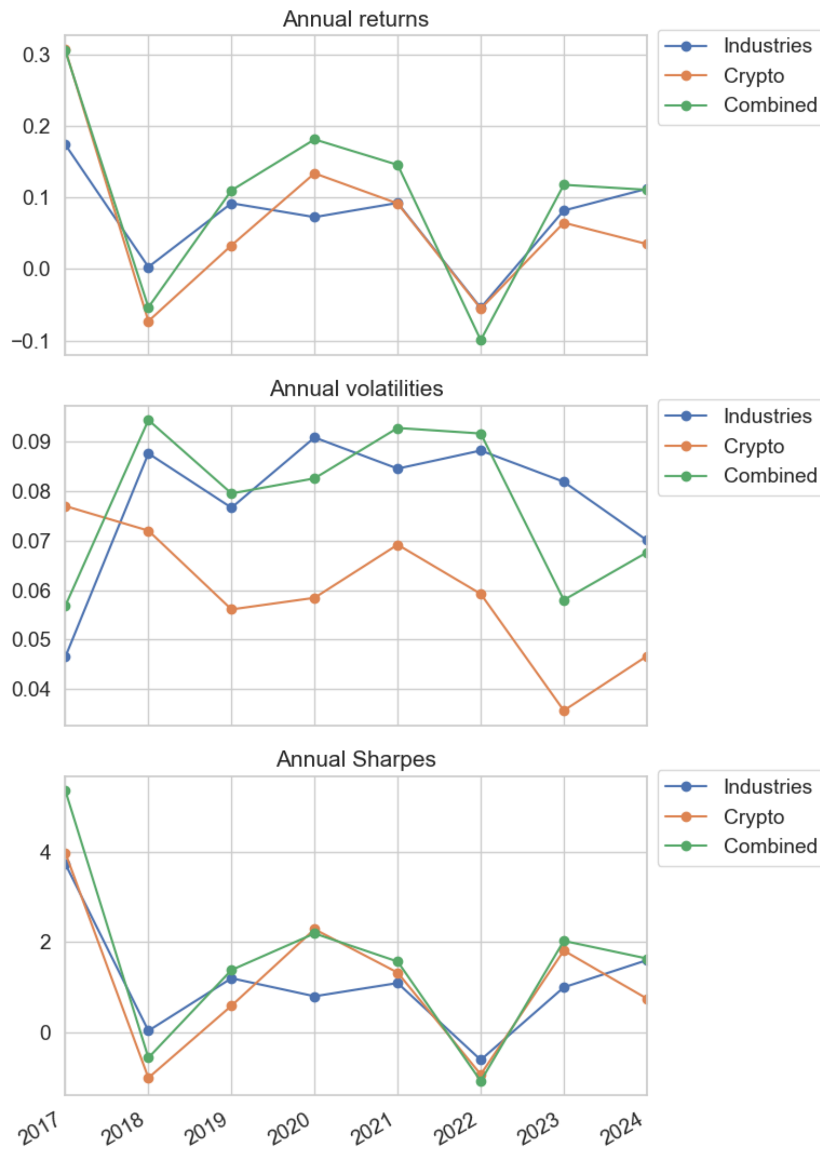


Figure 6.8: Annual performance metrics of the three portfolios.

Table 6.4: Shapley attributions by asset category.

Metric	Cnsmr	Manuf	HiTec	Hlth	Crypto	Total
Return (%)	1.8	0.7	2.5	0.8	2.5	8.2
Volatility (%)	1.8	1.8	2.0	1.9	0.7	8.2
Sharpe	0.20	0.06	0.26	0.08	0.40	1.0
Drawdown (%)	3.3	2.4	3.2	3.5	7.3	19.6

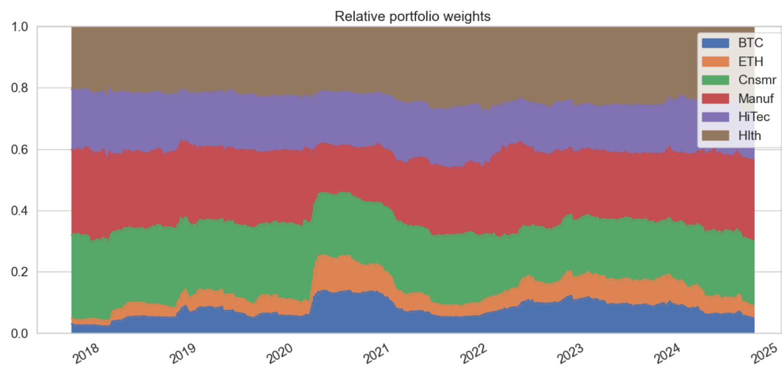


Figure 6.9: Relative weights of the combined portfolio.

6.4.4 Shapley attributions

We would like to attribute the performance of the portfolio to the different asset classes. Shapley values account for each assets' contribution to the portfolio, ensuring a fair allocation. They are uniquely characterized by satisfying a collection of desirable properties, including fairness, monotonicity, and full attribution [278, 153, 315, 111, 246, 231]. We will now look at the Shapley attributions to each industry and to crypto as a whole, for the combined portfolio. The Shapley attributions of the different asset classes for the combined portfolio are shown in table 6.4. Crypto assets have the highest attribution to return and Sharpe. All assets other than crypto have around a 2% contribution to volatility; crypto has a noticeably lower volatility contribution. Crypto assets have the highest contribution to drawdown.

6.5 Dynamically diluted 90/10 portfolio

Figure 6.9 shows the relative non-cash weights, *i.e.*, $w/\mathbf{1}^T w$ for the combined portfolio over time. We see that, apart from 2020, the relative weights are relatively stable and evenly distributed with about 10% in crypto assets (equally split between BTC and ETH), and 90% roughly equally split between the four industry portfolios. This motivates an even simpler portfolio construction method, akin to the popular 60/40 stocks/bonds allocation.

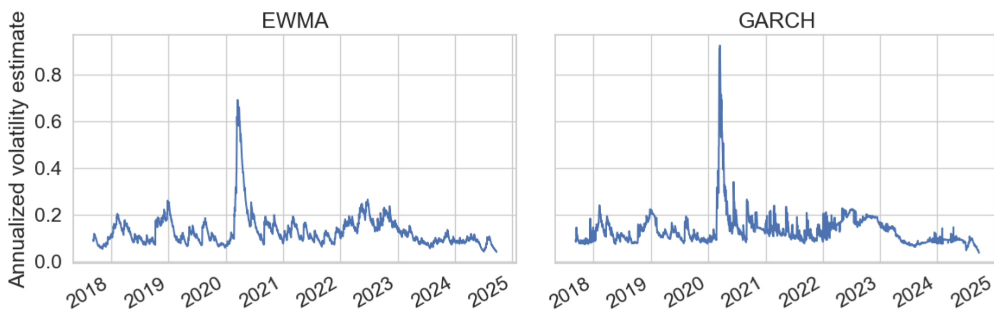


Figure 6.10: Annualized volatility estimates of the 90/10 portfolio.

Table 6.5: Performance metrics DD90/10 and CRA.

Metric	DD90/10 (EWMA)	DD90/10 (GARCH)	CRA
Return (%)	10.4	10.1	8.2
Volatility (%)	9.8	9.7	8.2
Sharpe	1.06	1.04	1.00
Drawdown (%)	19.9	19.7	19.6

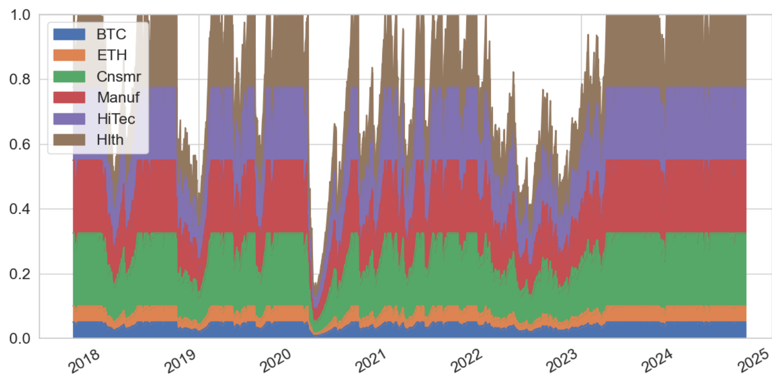
- *90/10 portfolio.* Construct a portfolio consisting of 90% equities (*e.g.*, the four industries with equal weights) and 10% crypto (*e.g.*, equally split between BTC and ETH).
- *Dynamic cash dilution.* Based on an estimate of the recent volatility of the 90/10 portfolio, dilute the 90/10 portfolio with cash to achieve the target risk σ and respect weight limits. We refer to this portfolio as the dynamically diluted 90/10 (DD90/10) portfolio.

6.5.1 DD90/10 results

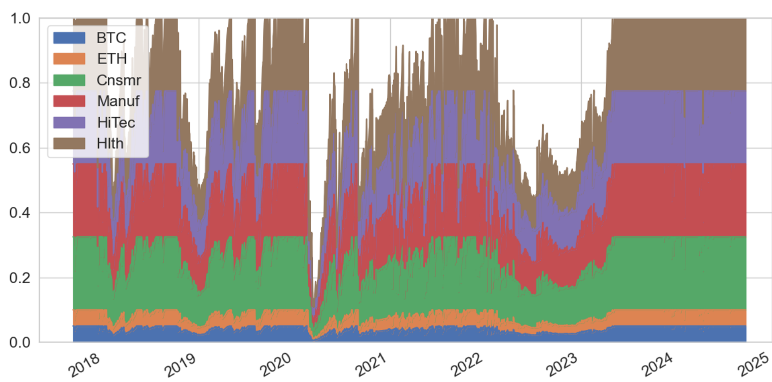
Volatility estimators. We evaluate the performance of the DD90/10 portfolio using two different volatility estimators: a 10-day half-life EWMA, and a GARCH(1,1) model refitted every day on the last 250 days of data, using the arch package in Python [283]. (We tried several other volatility estimators; all gave similar results.) The volatility estimates of the 90/10 portfolio are shown in figure 6.10.

Weights. The weights of the DD90/10 portfolios are shown in figure 6.11. The weights are quite similar, with the GARCH estimator being noticeably more reactive during some periods.

Performance. The performance of the DD90/10, compared to the (combined) CRA portfolio, is shown in table 6.5. Figure 6.12 shows the value of the three portfolios over time.



(a) EWMA volatility estimator.



(b) GARCH volatility estimator.

Figure 6.11: Portfolio weights of the DD90/10 portfolios with EWMA and GARCH volatility estimators.

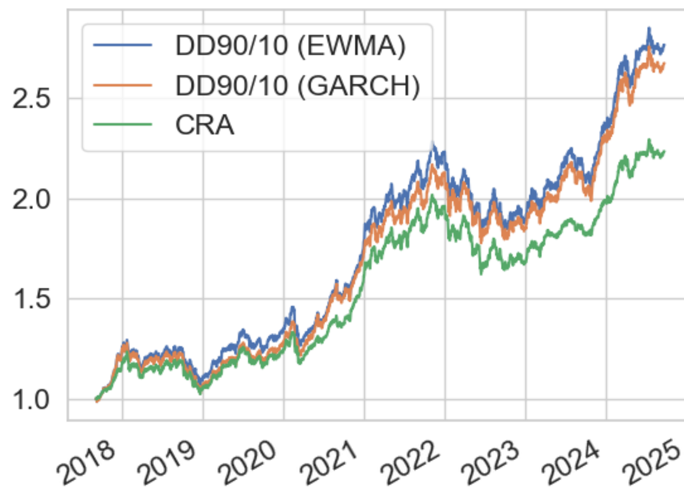


Figure 6.12: Portfolio values of the DD90/10 and CRA portfolios.

6.6 Conclusions

We have illustrated that, despite the documented extreme behaviors of crypto assets, simple traditional portfolio construction techniques can be used to include them in a diversified portfolio. We show this using two standard portfolio construction methods, one based on risk parity, and the other a fixed set of relative weights, with each one dynamically diluted with cash to achieve a target ex-ante risk. The addition of even a modest crypto weight of 10% increases the return and Sharpe ratio of the portfolio significantly, without significantly increasing volatility or drawdown.

Appendix A

Supplementary material for chapter 3

A.1 Coding tricks

The problem described in §3.4.1 can essentially be typed directly into a DSL such as CVXPY, with very few changes. In this section we mention a few simple tricks in formulating the problem (for a DSL) that lead to better performance.

Quadratic forms versus Euclidean norms. Traditional portfolio construction optimization formulations use quadratic forms such as $w^T \Sigma w$. Modern convex optimization solvers can directly handle the Euclidean norm without squaring to obtain a quadratic form. Using norm expressions instead of quadratic forms is often more natural, and has better numerical properties. For example a risk limit, traditionally expressed using a quadratic form as

$$w^T \Sigma w \leq (\sigma^{\text{tar}})^2,$$

is better expressed using a Euclidean norm as

$$\|L^T w\|_2 \leq \sigma^{\text{tar}},$$

where L is the Cholesky factor of Σ, *i.e.*, $LL^T = \Sigma$, with L lower triangular with positive diagonal entries.

Exploiting the factor model. To exploit the factor model, it is critical to *never* form the covariance matrix $\Sigma = F \Sigma^{\text{f}} F^T + D$. The first disadvantage of doing this is that we have to (needlessly) store an $n \times n$ matrix, which can be a challenge when n is on the order of tens of thousands. In addition, the solver will be slowed by a dramatic factor as mentioned in §3.3.3.

To exploit the factor model, we introduce the data matrix $\tilde{F} = FL$, where L is the Cholesky factor of Σ^{f}, so $\tilde{F}\tilde{F}^T = F\Sigma^{\mathrm{f}}F^T$. The portfolio variance is

$$\sigma^2 = w^T \tilde{F}\tilde{F}^T w + w^T D w = \|\tilde{F}^T w\|_2^2 + \|D^{1/2} w\|_2^2,$$

so the risk can be expressed using Euclidean norms as

$$\sigma = \left\| \left(\|\tilde{F}^T w\|_2, \|D^{1/2} w\|_2 \right) \right\|_2.$$

In this expression, the outer norm is of a 2-vector; the inner lefthand norm is of a k-vector, and the inner righthand norm is of an n-vector. Here we should be careful to express D as a diagonal matrix, or to express $D^{1/2}w$ as the elementwise (Hadamard) product of two vectors.

A.2 CVXPY code listing

We provide a reference implementation for the problem described in §3.4. This implementation is not optimized for performance, contains no error checking, and is provided for illustrative purposes only. For a more performant and robust implementation, we refer the reader to the cvxmarkowitz package [132]. Below, we assume that the data and parameters are already defined in corresponding data structures. The complete code for the reference implementation is available at

https://github.com/cvxgrp/markowitz-reference.

```
import cvxpy as cp

w, c = cp.Variable(data.n_assets), cp.Variable()

z = w - data.w_prev
T = cp.norm1(z) / 2
L = cp.norm1(w)

# worst-case (robust) return
factor_return = (data.F @ data.factor_mean).T @ w
idio_return = data.idio_mean @ w
mean_return = factor_return + idio_return + data.risk_free * c
return_uncertainty = param.rho_mean @ cp.abs(w)
return_wc = mean_return - return_uncertainty

# worst-case (robust) risk
factor_risk = cp.norm2((data.F @ data.factor_covariance_chol).T @ w)
idio_risk = cp.norm2(cp.multiply(data.idio_volas, w))
risk = cp.norm2(cp.hstack([factor_risk, idio_risk]))
risk_uncertainty = param.rho_covariance**0.5 * data.volas @ cp.abs(w)
risk_wc = cp.norm2(cp.hstack([risk, risk_uncertainty]))

asset_holding_cost = data.kappa_short @ cp.pos(-w)
cash_holding_cost = data.kappa_borrow * cp.pos(-c)
holding_cost = asset_holding_cost + cash_holding_cost

spread_cost = data.kappa_spread @ cp.abs(z)
impact_cost = data.kappa_impact @ cp.power(cp.abs(z), 3 / 2)
trading_cost = spread_cost + impact_cost
```

```
30
31 objective = (
32     return_wc
33     - param.gamma_hold * holding_cost
34     - param.gamma_trade * trading_cost
35 )
36
37 constraints = [
38     cp.sum(w) + c == 1,
39     param.w_min <= w, w <= param.w_max,
40     L <= param.L_tar,
41     param.c_min <= c, c <= param.c_max,
42     param.z_min <= z, z <= param.z_max,
43     T <= param.T_tar,
44     risk_wc <= param.risk_target,
45 ]
46
47 problem = cp.Problem(cp.Maximize(objective), constraints)
48 problem.solve()
```

We start by importing the CVXPY package in line 1 and define the variables of the problem in line 3. The variable `w` is the vector of asset weights, and `c` is the cash weight. We then define the trade vector `z`, turnover `T`, and leverage `L` in lines 5–7 to simplify the notation in the remainder of the code.

In the next block we first define the mean return in lines 10–12, taking into account the factor and idiosyncratic returns, as well as the risk-free rate. We then define the uncertainty in the mean return in line 13, which then reduces the mean return to the worst-case return in line 14.

Similarly, the robust risk is obtained in lines 17–21 by first defining the factor and idiosyncratic risk components, which are combined to the portfolio risk. The uncertainty in the risk, which depends on the asset volatilities, is combined with the portfolio risk to obtain the worst-case risk in line 21. The holding cost is defined in lines 23–25, followed by the trading cost in lines 27–29.

We form the objective function in lines 31–35 by combining the worst-case return with the holding and trading costs, weighted by the corresponding parameters. The constraints are collected in lines 37–45, starting with the budget constraint, followed by the holding and trading constraints, and ending with the risk constraint.

Finally, the problem is defined in line 47, combining the objective and constraints. It is solved in line 48 by simply calling the `.solve()` method on the problem instance, with a suitable solver being chosen automatically.

In only 48 lines of code we have defined and solved the Markowitz problem with all the constraints and objectives described in §3.4. This underlines the power of using a DSL such as CVXPY to specify convex optimization problems in a way that closely follows the mathematical formulation.

Parameters. Using parameters can provide both a convenient way to specify the problem, as well as a way to reduce the overhead of CVXPY when solving multiple instances. To obtain this speedup requires some restrictions on the problem formulation. For a precise definition we refer the reader to [1]. Here we only mention that we require expressions to additionally be linear, or affine, in the parameters. For example, we can use CVXPY parameters to easily and quickly change the mean return by writing to the the `.value` attribute of the `mean` and `risk_free` parameters.

```
mean = cp.Parameter(n_assets)
risk_free = cp.Parameter()

mean_return = w @ mean + risk_free * c
```

In some cases, it is necessary to reformulate the problem to satisfy the additional restrictions required to obtain the speedup, *e.g.*, by introducing auxiliary variables. For convenience, we provide a parametrized implementation of the Markowitz problem in the code repository, where these reformulations have already been carried out.

Bibliography

[1] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, 2019.

[2] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[3] Y. Ait-Sahalia and D. Xiu. Using principal component analysis to estimate a high dimensional factor model with high-frequency data. *Journal of Econometrics*, 201(2):384–399, 2017.

[4] C. Alexander and A. Chibumba. Multivariate orthogonal factor GARCH. *University of Sussex, Mimeo*, 1997.

[5] C. Alexander and A. Dimitriu. Indexing and statistical arbitrage. *The Journal of Portfolio Management*, 31(2):50–63, 2005.

[6] R. Almgren and N. Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, pages 5–39, 2000.

[7] T. Andersen, T. Bollerslev, P. Christoffersen, and F. Diebold. Volatility and correlation forecasting. volume 1 of *Handbook of Economic Forecasting*, pages 777–878. Elsevier, 2006.

[8] A. Ang, S. Gorovyy, and G. van Inwegen. Hedge fund leverage. *Journal of Financial Economics*, 102(1):102–126, 2011.

[9] MOSEK ApS. MOSEK modeling cookbook, 2020.

[10] MOSEK ApS. MOSEK portfolio optimization cookbook, 2023.

[11] B. Auer, F. Schuhmacher, and H. Kohrs. Rehabilitating mean–variance portfolio selection: Theory and evidence. *The Journal of Portfolio Management*, 2023.

[12] M. Avellaneda and J. Lee. Statistical arbitrage in the US equities market. *Quantitative Finance*, 10(7):761–782, 2010.

[13] J. Bai. Inferential theory for factor models of large dimensions. *Econometrica*, 71(1):135–171, 2003.

[14] J. Bai and S. Ng. Large dimensional factor analysis. *Foundations and Trends in Econometrics*, 3(2):89–163, 2008.

[15] D. Bailey and M. de Prado. An open-source implementation of the critical-line algorithm for portfolio optimization. *Algorithms*, 6(1):169–196, 2013.

[16] W. Bakry, A. Rashid, S. Al-Mohamad, and N. El-Kanj. Bitcoin and portfolio diversification: A portfolio optimization approach. *Journal of Risk and Financial Management*, 14(7):282, 2021.

[17] S. Barratt and S. Boyd. Stochastic control with affine dynamics and extended quadratic costs. *IEEE Transactions on Automatic Control*, 67(1):320–335, 2021.

[18] S. Barratt and S. Boyd. Covariance prediction via convex optimization. *Optimization and Engineering*, 2022.

[19] S. Bartram, J. Branke, G. De Rossi, and M. Motahari. Machine learning for active portfolio management. *Journal of Financial Data Science*, 3(3):9–30, 2021.

[20] L. Bauwens, S. Laurent, and J. Rombouts. Multivariate GARCH models: a survey. *Journal of Applied Econometrics*, 21(1):79–109, 2006.

[21] L. Bauwens and E. Otranto. Modeling realized covariance matrices: a class of hadamard exponential models. *Journal of Financial Econometrics*, 21(4):1376–1401, 2023.

[22] L. Bauwens, G. Storti, and F. Violante. Dynamic conditional correlation models for realized covariance matrices. *CORE DP*, 60:104–108, 2012.

[23] R. Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[24] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

[25] A. Ben-Tal and A. Nemirovski. Robust optimization—methodology and applications. *Mathematical Programming*, 92:453–480, 2002.

[26] E. Benveniste, P. Kolm, and G. Ritter. Justifying mean-variance portfolio selection when asset returns are skewed. *Journal of Portfolio Management, Harry Markowitz Special Issue*, 50(4), 2024.

[27] W. Bertram. Analytic solutions for optimal statistical arbitrage trading. *Physica A: Statistical mechanics and its applications*, 389(11):2234–2243, 2010.

[28] D. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific, 2012.

[29] D. Bertsimas, D. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[30] H. Beyer and B. Sendhoff. Robust optimization—a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, 2007.

[31] C. Bishop. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[32] F. Black. Universal hedging: Optimizing currency risk and reward in international equity portfolios. *Financial Analysts Journal*, 45(4):16–22, 1989.

[33] F. Black and R. Litterman. Asset allocation: Combining investor views with market equilibrium. *Goldman Sachs Fixed Income Research*, 115(1):7–18, 1990.

[34] F. Black and R. Litterman. Global portfolio optimization. *Financial analysts journal*, 48(5):28–43, 1992.

[35] L. Blackmore. Autonomous precision landing of space rockets. In *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*, volume 46, pages 15–20. The Bridge, 2016.

[36] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

[37] T. Bollerslev. Modelling the coherence in short-run nominal exchange rates: A multivariate generalized ARCH model. *The Review of Economics and Statistics*, 72(3):498–505, 1990.

[38] T. Bollerslev, R. Engle, and J. Wooldridge. A capital asset pricing model with time-varying covariances. *Journal of Political Economy*, 96(1):116–131, 1988.

[39] T. Bollerslev, A. Patton, and R. Quaedvlieg. Multivariate leverage effects and realized semicovariance garch models. *Journal of Econometrics*, 217(2):411–430, 2020.

[40] J. Bollinger. Using Bollinger bands. *Stocks & Commodities*, 10(2):47–51, 1992.

[41] J. Bollinger. *Bollinger on Bollinger bands*. McGraw-Hill New York, 2002.

[42] S. Boyd, A. Agrawal, and S. Barratt. Embedded convex optimization for control. In *Proceedings of the 59th IEEE Conference on Decision and Control*, 2020.

[43] S. Boyd, E. Busseti, S. Diamond, R. Kahn, K. Koh, P. Nystrup, and J. Speth. Multi-period trading via convex optimization. *Foundations and Trends in Optimization*, 3(1):1–76, 2017.

[44] S. Boyd, K. Johansson, R. Kahn, P. Schiele, and T. Schmelzer. Markowitz portfolio construction at seventy, 2023. Working Paper, Stanford University.

[45] S. Boyd, K. Johansson, R. Kahn, P. Schiele, and T. Schmelzer. Markowitz portfolio construction at seventy. *Journal of Portfolio Management*, 50(8):117–160, 2024.

[46] S. Boyd, M. Mueller, B. O'Donoghue, and Y. Wang. Performance bounds and suboptimal policies for multi-period investment. *Foundations and Trends in Optimization*, 1(1):1–72, 2013.

[47] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[48] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[49] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[50] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[51] S. Boyd and L. Vandenberghe. *Introduction to applied linear algebra: Vectors, matrices, and least squares*. Cambridge University Press, 2018.

[52] S. Boyd and L. Vandenberghe. Convex optimization additional exercises. https://github.com/cvxgrp/cvxbook_additional_exercises, 2024.

[53] M. Braga. *Risk-based approaches to asset allocation: concepts and practical applications*. Springer, 2015.

[54] P. Brandimarte. *Numerical methods in finance and economics: a MATLAB-based introduction*. John Wiley & Sons, 2013.

[55] M. Brandt. Portfolio choice problems. In *Handbook of financial econometrics: Tools and techniques*, pages 269–336. Elsevier, 2010.

[56] A. Brauneis and R. Mestel. Cryptocurrency-portfolios in a mean-variance framework. *Finance Research Letters*, 28:259–264, 2019.

[57] C. Brooks, S. Burke, and G. Persand. Multivariate GARCH models: software choice and estimation issues. *Journal of Applied Econometrics*, 18(6):725–734, 2003.

[58] J. Bulla and I. Bulla. Stylized facts of financial time series and hidden semi-Markov models. *Computational statistics & data analysis*, 51(4):2192–2209, 2006.

[59] T. Burggraf. Risk-based portfolio optimization in the cryptocurrency world. *Available at SSRN 3454764*, 2019.

[60] M. Bynum, G. Hackebeil, W. Hart, C. Laird, B. Nicholson, J. Siirola, J. Watson, and D. Woodruff. *Pyomo–Optimization modeling in Pthon*, volume 67. Springer Science & Business Media, third edition, 2021.

[61] D. Cajas. Convex optimization of portfolio kurtosis. *SSRN Electronic Journal*, 2022.

[62] J. Caldeira and G. Moura. Selection of a portfolio of pairs based on co-integration: A statistical arbitrage strategy. *Available at SSRN 2196391*, 2013.

[63] S. Chakrabarty and A. Kanaujiya. *Mathematical Portfolio Theory and Analysis*. Springer Nature, 2023.

[64] G. Chamberlain. A characterization of the distributions that imply mean—variance utility functions. *Journal of Economic Theory*, 29(1):185–201, 1983.

[65] J. Chen and M. Yuan. Efficient portfolio selection in a large market. *Journal of Financial Econometrics*, 14(3):496–524, 2016.

[66] Y. Choueifaty and Y. Coignard. Toward maximum diversification. *The Journal of Portfolio Management*, 35(1):40–51, 2008.

[67] J. Ciolli and J. Kara. Paul Krugman: Bitcoin is a more obvious bubble than housing was. *Business Insider*, 2013.

[68] J. Cochrane. *Asset pricing: Revised edition*. Princeton university press, 2009.

[69] G. Cornuejols and R. Tütüncü. *Optimization methods in finance*, volume 5. Cambridge University Press, 2006.

[70] IBM ILOG Cplex. V12. 1: User's manual for CPLEX. *International Business Machines Corporation*, 46(53):157, 2009.

[71] Z. Dai and F. Wang. Sparse and robust mean–variance portfolio optimization problems. *Physica A: Statistical Mechanics and its Applications*, 523:1371–1378, 2019.

[72] G. Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity Analysis of Production and Allocation*, 13:339–347, 1951.

[73] G. D'avolio. The market for borrowing stock. *Journal of financial economics*, 66(2-3):271–306, 2002.

[74] B. de Finetti. Il problema dei "pieni". *Giornale dell'Istituto Italiano degli Attuari*, 11:1–88, 1940.

[75] C. Eduardo De Moura, A. Pizzinga, and J. Zubelli. A pairs trading strategy based on linear state space models and the Kalman filter. *Quantitative Finance*, 16(10):1559–1573, 2016.

[76] G. De Nard, O. Engle, R.and Ledoit, and M. Wolf. Large dynamic covariance matrices: Enhancements based on intraday data. *Journal of Banking & Finance*, 138:106426, 2022.

[77] G. De Nard, O. Ledoit, and M. Wolf. Factor models for portfolio selection in large dimensions: The good, the better and the ugly. *Journal of Financial Econometrics*, 19(2):236–257, 2021.

[78] M. De Prado. Building diversified portfolios that outperform out of sample. *Journal of Portfolio Management*, 42(4):59–69, 2016.

[79] D. deBrito, M. Medeiros, and R. Ribeiro. Forecasting large realized covariance matrices: The benefits of factor models and shrinkage. *Available at SSRN 3163668*, 2018.

[80] H. Delatte. skfolio. https://github.com/skfolio/skfolio, 2023.

[81] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

[82] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[83] S. Diamond, R. Takapoui, and S. Boyd. A general system for heuristic minimization of convex functions over non-convex sets. *Optimization Methods and Software*, 33(1):165–193, 2018.

[84] M. Dixon, D. Klabjan, and J. Bang. Implementing deep neural networks for financial market prediction on the Intel Xeon Phi. In *Proceedings of the 8th Workshop on High Performance Computational Finance*, New York, NY, USA, 2015. Association for Computing Machinery.

[85] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *2013 European control conference*, pages 3071–3076. IEEE, 2013.

[86] I. Drechsler and Q. Drechsler. The shorting premium and asset pricing anomalies. Technical report, National Bureau of Economic Research, 2014.

[87] C. Dunis, G. Giorgioni, J. Laws, and J. Rudy. Statistical arbitrage and high-frequency data with an application to Eurostoxx 50 equities. *Liverpool Business School*, 2010.

[88] C. Edirisinghe, J. Chen, and J. Jeong. Optimal leveraged portfolio selection under quasi-elastic market impact. *Operations Research*, 71(5):1558–1576, 2023.

[89] R. Elliott, J. Van Der Hoek, and W. Malcolm. Pairs trading. *Quantitative Finance*, 5(3):271–276, 2005.

[90] R. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

[91] R. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica: Journal of the econometric society*, pages 987–1007, 1982.

[92] R. Engle. Dynamic conditional correlation. *Journal of Business & Economic Statistics*, 20(3):339–350, 2002.

[93] R. Engle and T. Bollerslev. Modelling the persistence of conditional variances. *Econometric reviews*, 5(1):1–50, 1986.

[94] R. Engle, R. Ferstenberg, and J. Russell. Measuring and modeling execution cost and risk. *Journal of Portfolio Management*, 38(2):14–28, 2012.

[95] R. Engle and C. Granger. Co-integration and error correction: Representation, estimation, and testing. *Econometrica*, 55(2):251–276, 1987.

[96] R. Engle and K. Kroner. Multivariate simultaneous generalized ARCH. *Econometric Theory*, 11(1):122–150, 1995.

[97] R. Engle, O. Ledoit, and M. Wolf. Large dynamic covariance matrices. *Journal of Business & Economic Statistics*, 37(2):363–375, 2019.

[98] R. Engle and K. Sheppard. Theoretical and empirical properties of dynamic conditional correlation multivariate garch, 2001.

[99] F. Fabozzi, P. Kolm, D. Pachamanova, and S. Focardi. *Robust portfolio optimization and management*. John Wiley & Sons, 2007.

[100] E. Fama. The behavior of stock-market prices. *Journal of Business*, 38(1):34–105, 1965.

[101] E. Fama and K. French. The cross-section of expected stock returns. *Journal of Finance*, 47(2):427–465, 1992.

[102] E. Fama and K. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, 1993.

[103] J. Fan, A. Furger, and D. Xiu. Incorporating global industrial classification standard into portfolio allocation: A simple factor-based large covariance matrix estimator with high-frequency data. *Journal of Business & Economic Statistics*, 34(4):489–503, 2016.

[104] Federal Reserve Bank of St. Louis. Federal funds effective rate (DFF). `https://fred.stlouisfed.org/series/DFF`, 2023. Accessed on 2023-09-22.

[105] Y. Feng and D. Palomar. Scrip: Successive convex optimization methods for risk parity portfolio design. *IEEE Transactions on Signal Processing*, 63(19):5285–5300, 2015.

[106] Y. Feng and D. Palomar. A signal processing perspective on financial engineering. *Foundations and Trends® in Signal Processing*, 9(1–2):1–231, 2016.

[107] T. Fischer, C. Krauss, and A. Deinert. Statistical arbitrage in cryptocurrency markets. *Journal of Risk and Financial Management*, 12(1):31, 2019.

[108] C. Fougner. POGS - proximal operator graph solver. https://foges.github.io/pogs/, 2023. Accessed on 2023-09-22.

[109] C. Fougner and S. Boyd. *Parameter Selection and Preconditioning for a Graph Form Solver*, pages 41–61. Springer International Publishing, 2018.

[110] K. French. Kenneth french data library. https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html#Research, 2024.

[111] D. Fryer, I. Strümke, and H. Nguyen. Shapley values for feature selection: The good, the bad, and the axioms. *IEEE Access*, 9:144352–144360, 2021.

[112] A. Fu, B. Narasimhan, and S. Boyd. CVXR: An R package for disciplined convex optimization. *arXiv preprint arXiv:1711.07582*, 2017.

[113] V. Gabrel, C. Murat, and A. Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.

[114] E. Gatev, W. Goetzmann, and G. Rouwenhorst. Pairs trading: Performance of a relative-value arbitrage rule. *The Review of Financial Studies*, 19(3):797–827, 2006.

[115] R. Geary. Moments of the ratio of the mean deviation to the standard deviation for normal samples. *Biometrika*, pages 295–307, 1936.

[116] R. Geary. Testing for normality. *Biometrika*, 34(3/4):209–242, 1947.

[117] C. Geczy, D. Musto, and A. Reed. Stocks are special too: An analysis of the equity lending market. *Journal of Financial Economics*, 66(2-3):241–269, 2002.

[118] A. Ghalanos. rmgarch: Multivariate garch models. *R package version*, pages 1–3, 2019.

[119] B. Ghosh, E. Bouri, J. Bum Wee, and N. Zulfiqar. Return and volatility properties: Stylized facts from the universe of cryptocurrencies and NFTs. *Research in International Business and Finance*, 65:101945, 2023.

[120] P. Giudici, P. Pagnottoni, and G. Polinesi. Network models to enhance automated cryptocurrency portfolio management. *Frontiers in artificial intelligence*, 3:22, 2020.

[121] SRL Global. Portfolio visualizer. https://www.portfoliovisualizer.com/, 2023.

[122] L. Goldberg, A. Papanicolaou, and A. Shkolnik. The dispersion bias. *SIAM Journal on Financial Mathematics*, 13(2):521–550, 2022.

[123] G. Golub and C. Van Loan. *Matrix computations*. JHU press, 2013.

[124] P. Goulart and Y. Chen. Clarabel documentation, 2024. Accessed on 2023-09-07.

[125] B. Graham. *The intelligent investor*. New York: Harper, 4th edition, 1973.

[126] B. Graham and D. Dodd. *Security analysis*. McGraw-Hill, 6th edition, 2009.

[127] C. Granger. *Co-integrated variables and error-correcting models*. PhD thesis, Discussion Paper 83-13. Department of Economics, University of California, 1983.

[128] M. Grant and S. Boyd. CVX: MATLAB software for disciplined convex programming, version 2.1, 2014.

[129] R. Grinold and R. Kahn. Active portfolio management. 2000.

[130] R. Grinold and R. Kahn. *Active portfolio management*. McGraw-Hill, 2000.

[131] R. Grinold and R. Kahn. *Advances in Active Portfolio Management: New Developments in Quantitative Investing*. McGraw-Hill, 2020.

[132] Stanford University Convex Optimization Group. CVXMARKOWITZ. `https://github.com/cvxgrp/cvxmarkowitz`, 2023.

[133] J. Guerard, editor. *Handbook of Portfolio Construction*. Springer, 2010.

[134] J. Guijarro-Ordonez, M. Pelger, and G. Zanotti. Deep learning statistical arbitrage. *arXiv preprint arXiv:2106.04028*, 2021.

[135] A. Gunjan and S. Bhattacharyya. A brief review of portfolio optimization techniques. *Artificial Intelligence Review*, 56(5):3847–3886, 2023.

[136] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[137] I. Gurrib. Machine learning and portfolio management: A review. *Annals of Mathematics and Computer Science*, 5:31–43, 2022.

[138] P. Hansen, Z. Huang, and H. Shek. Realized garch: a joint model for returns and realized measures of volatility. *Journal of Applied Econometrics*, 27(6):877–906, 2012.

[139] P. Hansen, A. Lunde, and J. Nason. The model confidence set. *Econometrica*, 79(2):453–497, 2011.

[140] W. Hart, J. Watson, and D. Woodruff. Pyomo: Modeling and solving mathematical programs in Python. *Mathematical Programming Computation*, 3(3):219–260, 2011.

[141] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*, volume 2. Springer, 2009.

[142] N. Hautsch, L. Kyj, and P. Malec. Do high-frequency data improve high-dimensional portfolio allocations? *Journal of Applied Econometrics*, 30(2):263–290, 2015.

[143] E. Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[144] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[145] S. Hogan, R. Jarrow, M. Teo, and M. Warachka. Testing market efficiency using statistical arbitrage with applications to momentum and value strategies. *Journal of Financial economics*, 73(3):525–565, 2004.

[146] O. Holovatiuk. Cryptocurrencies as an asset class in portfolio optimisation. *Central European Economic Journal*, 7(54):33–55, 2020.

[147] R. Horst and H. Tuy. *Global optimization: Deterministic approaches*. Springer Science & Business Media, 2013.

[148] A. Hu, C. Parlour, and U. Rajan. Cryptocurrencies: Stylized facts on a new investible instrument. *Financial Management*, 48(4):1049–1068, 2019.

[149] Y. Hu, S. Rachev, and F. Fabozzi. Modelling crypto asset price dynamics, optimal crypto portfolio, and crypto option valuation. *arXiv preprint arXiv:1908.05419*, 2019.

[150] N. Huck. Pairs selection and outranking: An application to the S&P 100 index. *European Journal of Operational Research*, 196(2):819–825, 2009.

[151] N. Huck. Pairs trading and outranking: The multi-step-ahead forecasting case. *European Journal of Operational Research*, 207(3):1702–1716, 2010.

[152] N. Huck. Large data sets and machine learning: Applications to statistical arbitrage. *European Journal of Operational Research*, 278(1):330–342, 2019.

[153] F. Huettner and M. Sunder. Axiomatic arguments for decomposing goodness of fit according to Shapley and Owen values. 2012.

[154] M. Isichenko. *Quantitative portfolio management: The art and science of statistical arbitrage*. John Wiley & Sons, 2021.

[155] B. Jacobs and K. Levy. *Equity management: Quantitative analysis for stock selection*. McGraw-Hill, 2000.

[156] B. Jacobs and K. Levy. Leverage aversion, efficient frontiers, and the efficient region. *Journal of Portfolio Management*, 39(3):54–64, 2013.

[157] B. Jacobs, K. Levy, and H. Markowitz. Portfolio optimization with factors, scenarios, and realistic short positions. *Operations Research*, 53(4):586–599, 2005.

[158] R. Jagannathan and T. Ma. Risk reduction in large portfolios: Why imposing the wrong constraints helps. *Journal of Finance*, 58(4):1651–1683, 2003.

[159] L. Jiang, K. Wu, G. Zhou, and Y. Zhu. Stock return asymmetry: Beyond skewness. *Journal of Financial and Quantitative Analysis*, 55(2):357–386, 2020.

[160] Z. Jiang and J. Liang. Cryptocurrency portfolio management with deep reinforcement learning. In *2017 Intelligent systems conference (IntelliSys)*, pages 905–913. IEEE, 2017.

[161] Z. Jiang, D. Xu, and J. Liang. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*, 2017.

[162] S. Johansen. Modelling of co-integration in the vector autoregressive model. *Economic Modelling*, 17(3):359–373, 2000.

[163] K. Johansson, G. Ogut, M. Pelger, T. Schmelzer, and S. Boyd. A simple method for predicting covariance matrices of financial returns. *Foundations and Trends in Econometrics*, 12(4):324–407, 2023.

[164] K. Johansson, T. Schmelzer, and S. Boyd. Finding moving-band statistical arbitrages via convex-concave optimization, 2023. Working Paper, Stanford University.

[165] K. Johansson, T. Schmelzer, and S. Boyd. Finding moving-band statistical arbitrages via convex–concave optimization. *Optimization and Engineering*, 2024.

[166] K. Johansson, T. Schmelzer, and S. Boyd. A markowitz approach to managing a dynamic basket of moving-band statistical arbitrages. *arXiv preprint arXiv:2412.02660*, 2024.

[167] Kasper Johansson and Stephen Boyd. Simple and effective portfolio construction with crypto assets. *Crypto Insights and Trends, Competition Policy International*, pages 1–16, January 2025.

[168] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[169] J. Jurek and H. Yang. Dynamic portfolio selection in arbitrage. In *EFA 2006 Meetings Paper*, 2007.

[170] M. Karim, Mohamed E. Rasid, A. Noman, and L. Yarovaya. Exploring asymmetries in cryptocurrency intraday returns and implied volatility: New evidence for high-frequency traders. *International Review of Financial Analysis*, page 103617, 2024.

[171] B. Kelly and D. Xiu. Financial machine learning. *Foundations and Trends® in Finance*, 13(3-4):205–363, 2023.

[172] I. Khalil, J. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall, 1996.

[173] D. Kim and B. Lee. Shorting costs and profitability of long–short strategies. *Accounting & Finance*, 63(1):277–316, 2023.

[174] M. Kochenderfer. *Decision making under uncertainty: theory and application*. MIT Press, 2015.

[175] M. Kochenderfer, T. Wheeler, and K. Wray. *Algorithms for decision making*. MIT Press, 2022.

[176] P. Kolm, R. Tütüncü, and F. Fabozzi. 60 years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, 234(2):356–371, 2014.

[177] C. Krauss. Statistical arbitrage pairs trading strategies: Review and outlook. *Journal of Economic Surveys*, 31(2):513–545, 2017.

[178] C. Krauss, X. Do, and N. Huck. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 259(2):689–702, 2017.

[179] C. Krauss and J. Stübinger. Non-linear dependence modelling with bivariate copulas: Statistical arbitrage pairs trading on the S&P 100. *Applied Economics*, 49(52):5352–5369, 2017.

[180] J. Kronqvist, D. Bernal, A. Lundell, and I. Grossmann. A review and comparison of solvers for convex MINLP. *Optimization and Engineering*, 20(2):397–455, 2018.

[181] P. Krugman. Golden cyberfetters. *The New York Times*, 2011.

[182] P. Krugman. Adam Smith hates bitcoin. *Business Insider*, 2013.

[183] S. Laurent, J. Rombouts, and F. Violante. On loss functions and ranking forecasting performances of multivariate volatility models. *Journal of Econometrics*, 173(1):1–10, 2013.

[184] O. Ledoit and M. Wolf. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of empirical finance*, 10(5):603–621, 2003.

[185] O. Ledoit and M. Wolf. Honey, I shrunk the sample covariance matrix. *The Journal of Portfolio Management*, 30(4):110–119, 2004.

[186] O. Ledoit and M. Wolf. Honey, I shrunk the sample covariance matrix. *Journal of Portfolio Management*, 30(4):110–119, 2004.

[187] O. Ledoit and M. Wolf. Robust performance hypothesis testing with the sharpe ratio. *Journal of Empirical Finance*, 15(5):850–859, 2008.

[188] J. Lee, J. Lee, and A. Prékopa. Price-bands: A technical tool for stock trading. *Fintech with Artificial Intelligence, Big Data, and Blockchain*, pages 221–246, 2021.

[189] M. Leibowitz, S. Emrich, and A. Bova. *Modern Portfolio Management: Active Long/Short 130/30 Equity Strategies*. Wiley, 2009.

[190] M. Lettau and M. Pelger. Estimating latent asset-pricing factors. *Journal of Econometrics*, 218(1):1–31, 2020.

[191] M. Lettau and M. Pelger. Factors that fit the time series and cross-section of stock returns. *Review of Financial Studies*, 33(5):2274–2325, 2020.

[192] H. Levy and H. Markowitz. Approximating expected utility by a function of mean and variance. *American Economic Review*, 69(3):308–317, 1979.

[193] X. Li, A. Uysal, and J. Mulvey. Multi-period portfolio optimization using model predictive control with mean-variance and risk parity frameworks. *European Journal of Operational Research*, 299(3):1158–1176, 2022.

[194] T. Lipp and S. Boyd. Variations and extension of the convex–concave procedure. *Optimization and Engineering*, 17:263–287, 2016.

[195] Y. Liu and A. Tsyvinski. Risks and returns of cryptocurrency. *The Review of Financial Studies*, 34(6):2689–2727, 2021.

[196] M. Lobo. *Robust and convex optimization with applications in finance*. PhD thesis, Stanford university, 2000.

[197] M. Lobo, L. Vandenberghe, S. Boyd, and H Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1):193–228, 1998.

[198] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and Its Applications*, 284(1-3):193–228, 1998.

[199] T. Loeb. Trading cost: The critical link between investment information and results. *Financial Analysts Journal*, 39(3):39–44, 1983.

[200] J. Lofberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, pages 284–289. IEEE, 2004.

[201] J. Longerstaey and M. Spencer. *Riskmetrics: Technical Document*. JP Morgan and Reuters, 1996.

[202] M. Lubin, O. Dowson, J. Dias Garcia, J. Huchette, B. Legat, and J. Vielma. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, 2023.

[203] G. Lucarelli and M. Borrotti. A deep Q-learning portfolio management framework for the cryptocurrency market. *Neural Computing and Applications*, 32:17229–17244, 2020.

[204] E. Luxenberg and S. Boyd. Portfolio construction with Gaussian mixture returns and exponential utility via convex optimization. *Optimization and Engineering*, pages 1–20, 2023.

[205] G. Maher, S. Boyd, M. Kochenderfer, C. Matache, D. Reuter, A. Ulitsky, S. Yukhymuk, and L. Kopman. A light-weight multi-objective asynchronous hyper-parameter optimizer. *arXiv preprint arXiv:2202.07735*, 2022.

[206] S. Maillard, T. Roncalli, and J. Teïletche. The properties of equally weighted risk contribution portfolios. *Journal of Portfolio Management*, 36(4):60–70, 2010.

[207] H. Malmsten and T. Teräsvirta. Stylized facts of financial time series and three popular models of volatility. *European Journal of pure and applied mathematics*, 3(3):443–477, 2010.

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

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

[210] H. Markowitz. *Portfolio selection: Efficient diversification of investments*. John Wiley, 1959.

[211] H. Markowitz. The early history of portfolio theory: 1600–1960. *Financial Analysts Journal*, 55:5–16, 7 1999.

[212] H. Markowitz. *Harry Markowitz: Selected works*, volume 1. World Scientific, 2009.

[213] H. Markowitz. *From Portfolio Theory to Practice: An AQR interview with Harry Markowitz*. AQR Capital Management, 2016.

[214] H. Markowitz. Nobel Prize lecture, 2023. Accessed on 2023-12-13.

[215] H. Markowitz and K. Blay. *Risk-return analysis: The theory and practice of rational investing*, volume 1. McGraw-Hill, 2014.

[216] H. Markowitz, D. Starer, H. Fram, and S. Gerber. *Avoiding the Downside: A Practical Review of the Critical Line Algorithm for Mean-Semivariance Portfolio Optimization*, chapter 16, pages 369–415. 2020.

[217] H. Markowitz and P. Todd. *Mean-variance analysis in portfolio choice and capital markets*, volume 66. John Wiley & Sons, 2000.

[218] H. Markowitz, P. Todd, G. Xu, and Y. Yamane. Computation of mean-semivariance efficient sets by the critical line algorithm. *Annals of Operations Research*, 45:307–317, 1993.

[219] H. Markowitz and E. Van Dijk. Single-period mean-variance analysis in a changing world. *Financial Analysts Journal*, 59(2):30–44, 2003.

[220] R. Martin. Pyportfolioopt: portfolio optimization in Python. *Journal of Open Source Software*, 6(61):3066, 2021.

[221] I. Mastromatteo, B. Tóth, and J. Bouchaud. Agent-based models for latent liquidity and concave price impact. *Physical Review E*, 89:042805, Apr 2014.

[222] MathWorks. MATLAB financial toolbox documentation, 2023.

[223] J. Mattingley and S. Boyd. CVXGEN: A code generator for embedded convex optimization. *Optimization and Engineering*, 13:1–27, 2012.

[224] A. McNeil, R. Frey, and P. Embrechts. *Quantitative risk management: concepts, techniques and tools-revised edition*. Princeton university press, 2015.

[225] J. Menchero, D. Orr, and J. Wang. *The Barra US equity model (USE4), methodology notes*. MSCI Barra, 2011.

[226] A. Meucci. Fully flexible views: Theory and practice. *arXiv preprint arXiv:1012.2848*, 2010.

[227] J. Meyer and R. Rasche. Sufficient conditions for expected utility to imply mean-standard deviation rankings: empirical evidence concerning the location and scale condition. *The Economic Journal*, 102(410):91–106, 1992.

[228] R. Michaud. The Markowitz optimization enigma: Is 'optimized' optimal? *Financial analysts journal*, 45(1):31–42, 1989.

[229] R. Michaud and R. Michaud. *Efficient asset management: a practical guide to stock portfolio optimization and asset allocation*. Oxford University Press, 2008.

[230] J. Mincer and V. Zarnowitz. The evaluation of economic forecasts. In *Economic forecasts and expectations: Analysis of forecasting behavior and performance*, pages 3–46. NBER, 1969.

[231] N. Moehle, S. Boyd, and A. Ang. Portfolio performance attribution via shapley value. *arXiv preprint arXiv:2102.05799*, 2021.

[232] N. Moehle, J. Gindi, S. Boyd, and M. Kochenderfer. Portfolio construction as linearly constrained separable optimization. *Optimization and Engineering*, 24(3):1667–1687, 2022.

[233] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7195–7201, 2016.

[234] B. Moritz and T. Zimmermann. Deep conditional portfolio sorts: The relation between past and future stock returns. *LMU Munich and Harvard University Working paper*, 2014.

[235] S. Mudchanatongsuk, J. Primbs, and W. Wong. Optimal pairs trading: A stochastic control approach. In *2008 American control conference*, pages 1035–1039. IEEE, 2008.

[236] P. Muller. *Empirical tests of biases in equity portfolio optimization: What do ex-ante mean-variance optimal portfolios look like ex-post?*, pages 80–98. Cambridge University Press, 1993.

[237] T. Nakajima. Expectations for statistical arbitrage in energy futures markets. *Journal of Risk and Financial Management*, 12(1):14, 2019.

[238] S. Nakamoto. A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, 4(2):15, 2008.

[239] R. Narang. *Inside the Black Box: The Simple Truth about Quantitative Trading*. J. Wiley & Sons, 2009.

[240] R. Narang. *Inside the Black Box: A Simple Guide to Systematic Investing*. J. Wiley & Sons, 2024.

[241] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, PA, 1994.

[242] Y. Nesterov and A. Nemirovsky. Conic formulation of a convex programming problem and duality. *Optimization Methods and Software*, 1(2):95–115, 1992.

[243] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169:1042–1068, 2016.

[244] D. Oh and A. Patton. High-dimensional copula-based distributions with mixed frequency data. *Journal of Econometrics*, 193(2):349–366, 2016.

[245] J. Ongerstaey and M. Spencer. *RiskMetrics: Technical Document*. JP Morgan and Reuters, 1996.

[246] A. Owen and C. Prieur. On Shapley value for measuring importance of dependent inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):986–1002, 2017.

[247] D. Palomar. *Portfolio Optimization*. Cambridge University Press, 2024.

[248] N. Parikh and S. Boyd. Block splitting for distributed optimization. *Mathematical Programming Computation*, 6(1):77–102, 2013.

[249] A. Patton. Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, 160(1):246–256, 2011.

[250] A. Patton and K. Sheppard. Evaluating volatility and correlation forecasts. In *Handbook of financial time series*, pages 801–838. Springer, 2009.

[251] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[252] M. Pelger. Understanding systematic risk: A high-frequency approach. *The Journal of Finance*, 75(4):2179–2220, 2020.

[253] M. Pelger. Markus pelger's data and code, 2023.

[254] M. Pelger and R. Xiong. Interpretable sparse proximate factors for large dimensions. *Journal of Business & Economic Statistics*, 40(4):1642–1664, 2022.

[255] M. Pelger and R. Xiong. State-varying factor models of large dimensions. *Journal of Business & Economic Statistics*, 40(3):1315–1333, 2022.

[256] M. Perlin. Evaluation of pairs-trading strategy at the Brazilian financial market. *Journal of Derivatives & Hedge Funds*, 15:122–136, 2009.

[257] S. Perrin and T. Roncalli. *Machine Learning Optimization Algorithms & Portfolio Allocation*. John Wiley & Sons, 2020.

[258] E. Platanakis and A. Urquhart. Portfolio management with cryptocurrencies: The role of estimation risk. *Economics Letters*, 177:76–80, 2019.

[259] A. Pole. *Statistical arbitrage: algorithmic trading insights and techniques*. John Wiley & Sons, 2011.

[260] Portfolio123. Portfolio123 website. https://www.portfolio123.com/, 2023.

[261] PortfoliosLab. PortfoliosLab website. https://portfolioslab.com/, 2023.

[262] J. Primbs and Y. Yamada. Pairs trading under transaction costs using model predictive control. *Quantitative Finance*, 18(6):885–895, 2018.

[263] E. Qian. Risk parity and diversification. *The Journal of Investing*, 20(1):119–127, 2011.

[264] Qontigo. Axioma by Qontigo. https://qontigo.com/products/axioma-risk, 2023.

[265] G. Ramkumar. Cryptocurrency portfolio construction using machine learning models. In *Contemporary Trends and Challenges in Finance: Proceedings from the 6th Wroclaw International Conference in Finance*, pages 103–122. Springer, 2021.

[266] T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

[267] B. Rosenberg. Prediction of common stock investment risk. *Journal of Portfolio Management*, pages 44–53, 1984.

[268] D. Rubin and D. Thayer. Em algorithms for ml factor analysis. *Psychometrika*, 47:69–76, 1982.

[269] M. Rubinstein. Bruno de Finetti and mean-variance portfolio selection. *Journal of Investment Management*, 4(3), 2006.

[270] I. Ruiz Roque da Silva, E. Junior, and P. Balbi. Cryptocurrencies trading algorithms: A review. *Journal of Forecasting*, 41(8):1661–1668, 2022.

[271] E. Sarmas, P. Xidonas, and H. Doukas. *Multicriteria portfolio construction with python*. Springer, 2020.

[272] S. Sarmento and N. Horta. *A Machine Learning Based Pairs Trading Investment Strategy*. Springer, 2020.

[273] M. Schaller, G. Banjac, S. Diamond, A. Agrawal, B. Stellato, and S. Boyd. Embedded code generation with CVXPY. *IEEE Control Systems Letters*, 6:2653–2658, 2022.

[274] T. Schmelzer and R. Hauser. Seven sins in portfolio optimization. *arXiv preprint arXiv:1310.3396*, 2013.

[275] F. Schuhmacher, H. Kohrs, and B. Auer. Justifying mean-variance portfolio selection when asset returns are skewed. *Management Science*, 67(12):7812–7824, 2021.

[276] S. Sexauer and L. Siegel. Harry Markowitz and the philosopher's stone. *Financial Analysts Journal*, 80(1):1–11, 2024.

[277] A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on stochastic programming: Modeling and theory*. SIAM, 2021.

[278] L. Shapley. A value for n-person games. *Contribution to the Theory of Games*, 2, 1953.

[279] W. Sharpe. A simplified model for portfolio analysis. *Management Science*, 9(2):277–293, 1963.

[280] W. Sharpe. Capital asset prices: A theory of market equillibrium under conditions of risk. *The Journal of Finance*, 19(3):425–442, 1964.

[281] A. Sheikh. BARRA's risk models. *Barra Research Insights*, pages 1–24, 1996.

[282] X. Shen, S. Diamond, Y. Gu, and S. Boyd. Disciplined convex-concave programming. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1009–1014. IEEE, 2016.

[283] K. Sheppard. Arch documentation. 2024.

[284] A. Shkolnik. On the markowitz enigma for minimum variance. In *Seminar of the Advanced Financial Technologies Laboratory*, December 7 2023. Talk presented at Stanford University.

[285] A. Silvennoinen and T. Teräsvirta. Multivariate GARCH models. In *Handbook of financial time series*, pages 201–229. Springer, 2009.

[286] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: An operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.

[287] J. Stübinger, B. Mangold, and C. Krauss. Statistical arbitrage with vine copulas. *Quantitative Finance*, 18(11):1831–1849, 2018.

[288] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625–653, 1999.

[289] L. Takeuchi and Y. Lee. Applying deep learning to enhance momentum trading strategies in stocks. In *Technical Report*. Stanford University Stanford, CA, USA, 2013.

[290] Hudson & Thames. Hudson & Thames website. https://hudsonthames.org/, 2023.

[291] The QuantLib contributors. QuantLib: a free/open-source library for quantitative finance. https://github.com/lballabio/QuantLib, 2023.

[292] Henri Theil. *Economic forecasts and policy*. North-Holland Rotterdam, 1961.

[293] A. Tikhonov and V. Arsenin. *Solutions of ill-posed problems*. V. H. Winston & Sons, 1977. Translated from Russian.

[294] K. Toh, M. Todd, and R. Tütüncü. SDPT3—a MATLAB software package for semidefinite programming, version 1.3. *Optimization Methods and Software*, 11(1-4):545–581, 1999.

[295] B. Tóth, Y. Lemperiere, C. Deremble, J. De Lataillade, J. Kockelkoren, and J. Bouchaud. Anomalous price impact and the critical nature of liquidity in financial markets. *Physical Review X*, 1(2):021006, 2011.

[296] R. Tütüncü and M. Koenig. Robust asset allocation. *Annals of Operations Research*, 132(1):157–187, 2004.

[297] M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd. Convex optimization in Julia. In *2014 First Workshop for High Performance Technical Computing in Dynamic Languages*, pages 18–28. IEEE, 2014.

[298] M. Vaitonis and S. Masteika. Statistical arbitrage trading strategy in commodity futures market with the use of nanoseconds historical data. In *Information and Software Technologies*, pages 303–313, Cham, 2017. Springer International Publishing.

[299] R. van der Weide. GO-GARCH: A multivariate generalized orthogonal GARCH model. *Journal of Applied Econometrics*, 17(5):549–564, 2002.

[300] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[301] D. Vassallo, G. Buccheri, and F. Corsi. A dcc-type approach for realized covariance modeling with score-driven dynamics. *International Journal of Forecasting*, 37(2):569–586, 2021.

[302] G. Vidyamurthy. *Pairs Trading: quantitative methods and analysis*, volume 217. John Wiley & Sons, 2004.

[303] J. Von Neumann and O. Morgenstern. *Theory of games and economic behavior, 2nd rev.* Princeton University Press, 1947.

[304] I. Vrontos, P. Dellaportas, and D. Politis. A full-factor multivariate GARCH model. *The Econometrics Journal*, 6(2):312–334, 2003.

[305] R. Welsch and X. Zhou. Application of robust statistics to asset allocation models. *REVSTAT–Statistical Journal*, pages 97–114, 2007.

[306] J. Williams. *The theory of investment value.* Cambridge, MA: Harvard University Press, 1938.

[307] P. Wolfe. The simplex method for quadratic programming. *Econometrica*, 27(3):382–398, 1959.

[308] WRDS. Wharton research data services. https://wrds-web.wharton.upenn.edu/wrds/, 2023.

[309] L. Wu, Y. Feng, and D. Palomar. General sparse risk parity portfolio design via successive convex optimization. *Signal Processing*, 170:107433, 2020.

[310] Y. Yamada and J. Primbs. Model predictive control for optimal portfolios with co-integrated pairs of stocks. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 5705–5710. IEEE, 2012.

[311] Y. Yamada and J. Primbs. Optimal trading with cointegrated pairs of stocks. In *Recent Advances In Financial Engineering 2011*, pages 183–202. World Scientific, 2012.

[312] Y. Yamada and J. Primbs. Model predictive control for optimal pairs trading portfolio with gross exposure and transaction cost constraints. *Asia-Pacific Financial Markets*, 25:1–21, 2018.

[313] J. Yarow. Paul Krugman responds to all the people throwing around his old internet quote. *Business Insider*, 2013.

[314] D. Yermack. Is bitcoin a real currency? An economic appraisal. In *Handbook of digital currency*, pages 29–40. Elsevier, 2024.

[315] H. Zhang, H. Singh, M. Ghassemi, and S. Joshi. Why did the model fail? Attributing model performance changes to distribution shifts. 2023.

[316] Z. Zhao and D. Palomar. Mean-reverting portfolio with budget constraint. *IEEE Transactions on Signal Processing*, 66(9):2342–2357, 2018.

[317] Z. Zhao, R. Zhou, and D. Palomar. Optimal mean-reverting portfolio with leverage constraint for statistical arbitrage in finance. *IEEE Transactions on Signal Processing*, 67(7):1681–1695, 2019.

[318] Z. Zhao, R. Zhou, Z. Wang, and D. Palomar. Optimal portfolio design for statistical arbitrage in finance. In *2018 IEEE Statistical Signal Processing Workshop (SSP)*, pages 801–805. IEEE, 2018.

[319] K. Zhou and J. Doyle. *Essentials of robust control*, volume 104. Prentice Hall, 1998.

[320] R. Zhou and D. Palomar. Solving high-order portfolios via successive convex approximation algorithms. *IEEE Transactions on Signal Processing*, 69:892–904, 2021.

[321] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. 2, 2003.