Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Félix Roudier

Portfolio Optimization and Genetic Algorithms

Master's Thesis
Department of Management, Technology and Economics - DMTEC
Chair of Entrepreneurial Risks - ER

Swiss Federal Institute of Technology (ETH) Zurich
Ecole Nationale des Ponts et Chaussées (ENPC) Paris

Supervisors:
Prof. Dr. Didier Sornette
Prof. Dr. Bernard Lapeyre

Zurich, May 17, 2007

To my office mates, who kindly let me make the coffee every morning...

Over the past few months, I've learned how to write decent code, changed my OS, read a bunch of theory and attended my first Nobel Lecture. Had I been a bit quicker on my feet and a lot better dressed, I might have squeezed a second one into the same day. With a little luck, I'll catch Joseph Stiglitz some other time.

Complex systems, finance, economics, physics and even some math. The research areas approached in the team are pretty broad. I've spread my attention and time a bit wide, but that was the whole point. A large part of what I've learned doesn't fit into this report.

The following pages are in no way an exhaustive review of portfolio theory. There are no proofs or theorems, no great results. This is just a summary of most of what I've done and what I would do if I had more time.

Abstract

Modern Portfolio Theory is based on Harry Markowitz's 1952 work on mean-variance portfolios. He stated that a rational investor should either maximize his expected return for a given level of risk, or minimize his risk for a given expected return. These two principles lead to an efficient frontier of portfolios, among which the investor is free to choose.

Fifty years on, there are no widely accepted practical implementations of mean-variance portfolio theory. The mean-variance approach puts excessive weights on assets with large excess returns, regardless of possible estimation errors. It yields unstable portfolios and extra gains don't make up for the excess transaction costs.

The goal of this Master's Thesis is to develop robust portfolio optimization methods. We design a multi-factor objective function reflecting our investment preferences and solve the subsequent optimization problem using a genetic algorithm.

Résumé

La théorie du portefeuille est fondée sur la description de portefeuilles moyenne-variance, donnée par Harry Markowitz en 1952. Il propose qu'un investisseur rationnel doit maximiser son rendement espéré pour un niveau de risque donné, ou minimiser son niveau de risque pour un rendement espéré donné. Ces deux principes permettent de générer une frontière de portefeuilles efficients, parmi lesquels l'investisseur peut choisir.

Cinquante ans plus tard, la théorie n'a toujours pas été mise en pratique de manière fiable et reconnue. L'approche moyenne-variance génère des portefeuilles déséquilibrés, surinvestis dans les titres à forte espérance en rendement. On ne prend pas en compte l'incertitude des données. Les portefeuilles construits sont instables et les frais de transaction supplémentaires ne sont pas compensés par des gains.

Le but de ce projet est de développer des méthodes d'optimisation robuste de portefeuilles. Nous définissons une fonction objective de plusieurs facteurs, que nous optimisons à l'aide d'un algorithme génétique.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> A man walks into a grocery store and buys a dozen eggs. On the way home he trips, drops his bag and breaks all the eggs. He goes back to the store and explains what happened.
>
> The shop owner answers: "Never put all your eggs in the same basket. Why don't you buy some of these extra-strong bags. They're really cheap."
>
> The man buys a couple of bags and a dozen more eggs. He splits them between the two bags and walks home, where he finds the eggs are bad.
>
> You can try getting rich by splitting up your eggs. You can also try selling bags.

It's 1976 and your neighbour Steve asks you if you want to help him build "computers" in his garage. Would you *willingly* have done it? Probably not. Of course, few people knew what a "computer" was at the time. Then again, most people who bought WorldCom stock didn't know much about telecommunications either... We choose our investments in mysterious ways. It all depends on what you expect from them. Are you ready to risk losing your money? Some of it? All of it? How long do you plan to keep it invested? A few weeks? A few years? How steady are those plans? Is there a chance you might have to use that money beforehand? How big of a chance?

Asset allocation is a decision problem. We must choose among different investment opportunities. This implies we should be able to compare investments, rank them according to preference. Portfolio theory describes how investors should allocate their wealth and manage their liabilites.

The standard approach to this problem was developped by Harry Markowitz in 1952 [1] [2]. He derived *mean-variance portfolio theory*, stating that a rational investor should either maximize his expected return for a given level of risk, or minimize his risk for a given expected return. These two principles

lead to an efficient frontier of portfolios, among which the investor is free to choose according to his risk preference. In this framework, variance serves as a measure of risk.

Fifty years on, mean-variance is still the standard approach to portfolio theory. Nevertheless, mean-variance optimized portfolios have been shown not to perform well. The approach puts excessive weights on assets with large excess returns, regardless of possible estimation errors in the input data [3] [4]. The portfolios it yields are very unstable and extra gains don't make up for the excess transaction costs due to excessive reallocation.

We consider a 2-step approach to portfolio management. We start by choosing a broad spectrum of stocks. This is our underlying portfolio. We then overweight or underweight certain stocks according to different criteria (tips, hunches and other good ideas). An example of this type of portfolio management would be to buy the index, then overweight certain stocks or subindices according to preferences and beliefs about the future.

The goal of this thesis is to develop robust portfolio optimization methods to choose the underlying portfolio. By robust, we mean that portfolios should be stable over time and have low sensitivity to estimation errors in the inputs. We design an objective function f taking different factors into account: performance, risk, diversification, turnover costs and eventually more. f reflects our investment preferences.

Our underlying portfolio is the solution to the following optimization problem:

$$\max f(x),\ x \in \Omega$$

where Ω is the space of possible portfolios.

Typical optimizers rely on quadratic programming and deterministic algorithms to find "optimal" portfolios. We will be using a *genetic algorithm*, which allows for much more freedom in the functional form we wish to solve. Genetic algorithms are heuristic and stochastic search methods. We don't determine in advance how the algorithm should act at each step and part of the search is random. Genetic algorithms are often well suited to find good solutions to optimization problems where the search space has many local minima and/or there are no known well-performing deterministic search methods.

In the mean-variance framework, the quality of input data is very important. As one author puts it, "mean-variance optimization is too powerful a tool for the quality of our data" [5]. We optimize over noise rather than information. To lessen the effect of noise, we apply re-sampling methods. We consider stock prices to be one realized outcome among many different possible ones. Using this historical data, we generate new data sets from which we hope to gather better information about asset returns and correlations.

Chapter 2

Foundations of Portfolio Theory

2.1 Asset prices and returns

There are many ways to make money and even more ways to lose it. You can bet on dog races and play roulette, you can crash expensive cars... We'll be concentrating on making money in financial markets, which is kind of like betting, except nobody will break your legs if you bring the house down, quite the contrary. Chances are that if you win big, the house will probably give you more money to play with.

"Financial markets" is a very broad term. There are markets for stocks, for bonds, for money, oil, pork bellies, wheat, insurance, the weather, and just about anything legal. If you have enough money, you will always find a financial institution willing to take the other side of your bet. We'll be dealing with portfolios of stocks, so we should quickly look at what a stock is.

2.1.1 Stocks

When you buy a stock, you buy a small part of the company which emitted the stock. If the small burger restaurant next door emitted 1000 shares and you bought 10 of those for 1000 dollars, you would be the rightful owner of 1% of Joe's Burgers. Depending on the type of stock, you now have a 1% stake in the voting rights and are entitled to 1% of future profits. *Congratulations!*

Was it a good investment? Joe's Burgers might expand wildly, change it's name, get a golden arches logo and make you rich. Or maybe people will stop eating burgers and take up tofu instead (a crazy idea, but who knows), in which case you might as well have bought 1000 burgers. More likely, the outcome will be somewhere in between. If the restaurant runs

well, and your share of the profits for the year is 100 dollars, you made a 10% return on your investment. Was it worth the trouble? You took the *risk* of seeing Joe's Burgers go bankrupt and lose all your money. You could also have become a burger king... And both scenarios might still come true in future years. So the value of your investment depends on your perception of the future, the burger market and your take on tofu.

Now if you want to sell your stock, you have to find a buyer who agrees to your price. This might not always be easy. Others might have a different take on burgers and tofu. You might not find a buyer, or you might find a buyer who agrees to a lower price. If you really have to sell and there's only one buyer, you'll have to agree and will probably lose money. This is called *liquidity risk*. Financial markets decrease this risk, by bringing buyers and sellers together, with hopes of increasing the amount of trading done and making it cheaper to do so.

2.1.2 Prices and returns

While Joe's Burgers isn't quoted on any major exchanges that we know of, *Starbucks* is. Let's use Starbucks (SBUX) stock to define the terms we will be using throughout this paper.

Prices We refer to the price of an asset at time t as P_t. We always take the closing price, either at the end of the day, the end of the week or the end of the month. We will sometimes refer to logprices, which we define as:

$$log\ p_t = log(P_t).$$

Figure 2.1 shows the price of SBUX at different time scales. Asset prices tend to vary a lot on a daily basis, less on a weekly basis. The less often you sample prices, the smoother your price curve gets. One way of saying this is that there is a lot of noise in the market and that you can smooth this out by taking wider time frames.

Returns We define the return on a stock from time $t-1$ to time t as:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

log-Returns For small variations of P_t, we have $P_t \simeq P_{t-1}$ and the return at time t is very small:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{P_t}{P_{t-1}} - 1 \simeq 0$$

For a small x, we can write the following first order approximation:

$$log(1+x) \simeq x$$

In the same way, for small price variations, we can also write:

$$log(\frac{P_t}{P_{t-1}}) = log(1 + \frac{P_t}{P_{t-1}} - 1) \simeq \frac{P_t}{P_{t-1}} - 1 = R_t$$

So for small price variations, log-returns are first-order equivalents of returns.

2.2 Mean-Variance Optimization

Asset prices vary over time. Some move up when others go down, some move together. Some have very small price variations.

Suppose you can invest your money in two different businesses: an ice cream factory and an umbrella factory. If you invest it all in the ice cream factory, you will earn big if it's a sunny year, and lose big if it's a rainy year. If you invest everything in the umbrella factory it's the other way around. If you invest half in both, the fluctuations in your income will be somewhat dampened. You won't win big, but you won't lose big either.

This investment strategy makes sense. It is a way of diversifying your investments. At the same time, it's a subjective process and can't be applied to a large universe of stocks. How do you guess the relative behaviours of General Electric (GE) and Starbucks (SBUX)? Portfolio Theory tries to define a systematic approach to choosing assets.

The standard approach to Modern Portfolio Theory was developped by Harry Markowitz [1] [2]. He derived *mean-variance portfolio theory*, stating that a rational investor should either maximize his expected return for a given level of risk, or minimize his risk for a given expected return. This framework allows us to compare different portfolios on a risk/return basis. It gives us the riskwise cost of a given level of expected return and tells us how to choose the fractions $\omega_1, ..., \omega_n$ of our portfolio we invest in each asset to best suit our risk/return expectations.

In the rest of this paper, we assume our investment universe is made up of n assets with returns $r_1, ..., r_n$, where $\forall i$, r_i is a random variable.

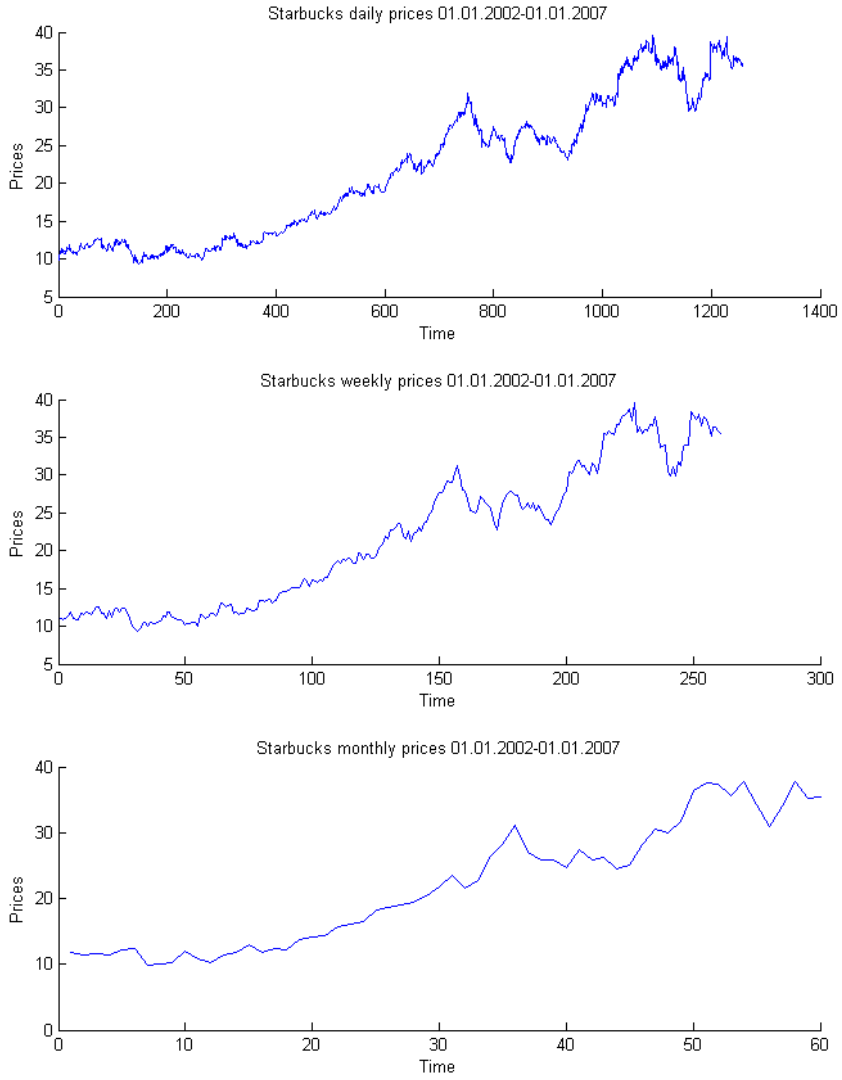

Fig. 2.1: From top to bottom: daily, weekly and monthly prices of Starbucks stock (SBUX) from 1st of January 2002 to 1st of January 2007.

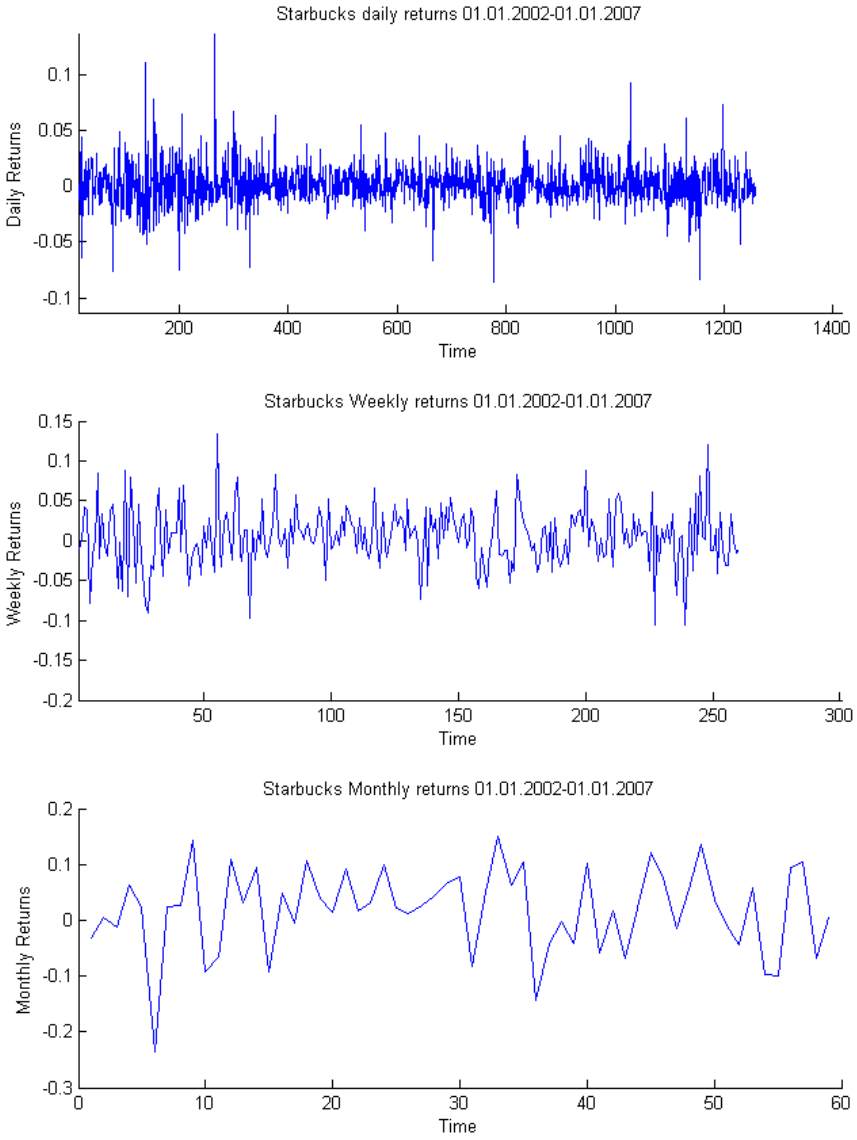

Fig. 2.2: From top to bottom: daily, weekly and monthly returns of Starbucks stock (SBUX) from 1st of January 2002 to 1st of January 2007.

2.2.1 Risk and Return

Return

We need an estimation of the expected returns for each asset. To start out, we simply take the historical mean. The expected returns of the individual assets are:

$$E[r_i] = \mu_i. \tag{2.1}$$

Thus the total expected return of portfolio P can be written as:

$$\mu_P = E[r_P] = \sum_{i=1}^{n} \omega_i E[r_i] = \sum_{i=1}^{n} \omega_i \mu_i \tag{2.2}$$

Risk

In this framework, the measure of risk is the variance of returns. For a given asset i:

$$\sigma_i^2 = Var(r_i) = E[(r_i - E[r_i])^2] = E[(r_i - \mu_i)^2] \tag{2.3}$$

The covariance of assets i and j is defined as:

$$\sigma_{ij} = Covar(r_i, r_j) = E[(r_i - E[r_i])(r_j - E[r_j])] = E[(r_i - \mu_i)(r_j - \mu_j)] \tag{2.4}$$

The correlation of assets i and j is defined as:

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \tag{2.5}$$

So we can write the variance of portfolio P as:

$$\sigma_P^2 = E[(r_P - \mu_P)^2] = E[(\sum_{i=1}^{n} \omega_i r_i - \sum_{i=1}^{n} \omega_i \mu_i)^2] = E[(\sum_{i=1}^{n} \omega_i (r_i - \mu_i))^2] \tag{2.6}$$

$$\sigma_P^2 = E[\sum_{i=1}^{n} \omega_i^2 (r_i - \mu_i)^2 + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_i \omega_j (r_i - \mu_i)(r_j - \mu_j)] \tag{2.7}$$

$$\sigma_P^2 = \sum_{i=1}^{n} (\sigma_i \omega_i)^2 + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_i \omega_j \sigma_{ij} \tag{2.8}$$

2.2.2 The virtues of diversification

Equation 2.8 shows us the effect of diversification. The variance of the portfolio can be lower than the weighted sums of individual variances. If we have an equally weighted portfolio of n assets, we get:

$$\sigma_P^2 = \frac{1}{n}\overline{\sigma_i}^2 + \frac{n-1}{n}\overline{\sigma_{ij}} \tag{2.9}$$

The first term of this equation is the risk associated with indiviual assets. If n is large, we can diversify this risk away and only the market risk remains:

$$\sigma_P^2 \longrightarrow \overline{\sigma_{ij}} \tag{2.10}$$

If we want to lower the risk of our portfolio even more, we have to invest in anti-correlated assets or assets with low correlations. In this manner, the average correlation factor σ_{ij} will go down.

2.2.3 Efficient frontiers and portfolio selection

For simplicity reasons, we consider a portfolio of 2 assets. We write the expected return μ_P and variance σ_P^2 of our portfolio as a function of our choice of weights (ω_1, ω_2)

$$\sigma_P^2 = (\sigma_1\omega_1)^2 + (\sigma_2\omega_2)^2 + 2\omega_1\omega_2\rho_{12}\sigma_1\sigma_2 \tag{2.11}$$

$$\mu_P = \omega_1\mu_1 + \omega_2\mu_2 \tag{2.12}$$

(ω_1, ω_2) have to verify some constraints:

- $\omega_1 + \omega_2 = 1$
- $\forall i,\ \omega_i \geq 0$

We replace ω_2 by $(1 - \omega_1)$, giving us the following expression of σ_P:

$$\sigma_P^2 = (\sigma_1\omega_1)^2 + (\sigma_2(1-\omega_1))^2 + 2\omega_1(1-\omega_1)\rho_{12}\sigma_1\sigma_2 \tag{2.13}$$

We then plot possible portfolios in a (σ, μ) plane for different values of $\rho_{12}, as shown in figure 2.3$.

Perfect correlation between assets $(\rho_{12} = 1)$

$$\sigma_P^2 = (\sigma_1\omega_1)^2 + (\sigma_2(1-\omega_1))^2 + 2\omega_1(1-\omega_1)\sigma_1\sigma_2 \tag{2.14}$$

Which factorizes into:

$$\sigma_P^2 = (\sigma_1\omega_1 + \sigma_2(1-\omega_1))^2 \tag{2.15}$$

And:

$$\mu_P = \omega_1\mu_1 + \omega_2\mu_2 \tag{2.16}$$

So:

$$\mu_P = \mu_1 + \frac{\mu_2 - \mu_1}{\sigma_2 - \sigma_1}(\sigma_P - \sigma_1) \tag{2.17}$$

The different portfolios available to an investor changing his asset allocation are on a straight line between a 100% investment in asset 1 and a 100% investment in asset 2, as shown in figure 2.3.

No correlation between assets ($\rho_{12} = 0$)

$$\sigma_P^2 = (\sigma_1\omega_1)^2 + (\sigma_2(1-\omega_1))^2 \tag{2.18}$$

We find the minimum-variance portfolio by solving $\frac{\partial\sigma_P}{\partial\omega_1} = 0$, which gives:

$$\omega_{1*} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{2.19}$$

This gives us a return of:

$$\mu_{P*} = \frac{\sigma_2^2\mu_1 + \sigma_1^2\mu_2}{\sigma_1^2 + \sigma_2^2} \tag{2.20}$$

The possible portfolios are on the curve described by the following equations:

$$\omega_1 = \frac{\sigma_2^2 - \sqrt{\sigma_2^4 - (\sigma_1^2 + \sigma_2^2)(\sigma_2^2 - \sigma_P^2)}}{\sigma_1^2 + \sigma_2^2} \tag{2.21}$$

And:

$$\mu_P = \omega_1\mu_1 + (1-\omega_1)\mu_2 \tag{2.22}$$

Perfect anti-correlation between assets ($\rho_{12} = -1$)

$$\sigma_P^2 = (\sigma_1\omega_1)^2 + (\sigma_2(1-\omega_1))^2 - 2\omega_1(1-\omega_1)\sigma_1\sigma_2 \tag{2.23}$$

Which factorizes into:

$$\sigma_P^2 = (\sigma_1\omega_1 - \sigma_2(1-\omega_1))^2 \tag{2.24}$$

And we get 2 distinct expressions for σ_P:

$$\sigma_P = \sigma_1\omega_1 - \sigma_2(1-\omega_1) \text{ or: } \sigma_P = \sigma_2(1-\omega_1) - \sigma_1\omega_1$$

We can choose ω_1 such that $\sigma_P = 0$:

$$\omega_{1*} = \frac{\sigma_2}{\sigma_2 + \sigma_1} \tag{2.25}$$

This gives us a return of:

$$\mu_{P*} = \frac{\sigma_2\mu_1 + \sigma_1\mu_2}{\sigma_1 + \sigma_2} \tag{2.26}$$

As well as 2 distinct expressions for μ_P:

$$\mu_P = \mu_{P*} + \frac{\sigma_P}{\sigma_2}(\mu_2 - \mu_{P*}) \text{ or: } \mu_P = \mu_{P*} + \frac{\sigma_P}{\sigma_1}(\mu_1 - \mu_{P*})$$

The possible portfolios are on 2 segments, as shown below.

Because of correlations between assets, it is possible to build a portfolio less risky than all individual assets, but with a higer expected return than the lowest expected return among these assets. For a given level of expected return ($\mu = r_0$), we can find the portfolio with the minimum expected variance. By changing r_0, we find a $frontier$ of so-called $efficient\ portfolios$, which all minimize variance for a level of expected return.

2.2.4 Closed-form solution of the Mean Variance Portfolio Problem

We write the problem described above in the following way:

$$\text{minimize } \vec{\omega}^T V \vec{\omega}$$

st:

$$\vec{\omega}^T \vec{\mu} = r_0 \tag{2.27}$$

$$\forall i, \omega_i \geq 0 \tag{2.28}$$

$$\vec{\omega}^T \vec{e} = 1 \tag{2.29}$$

Where $\vec{e} = (1, 1, .., 1)^T$, V is the covariance matrix of the assets, $\vec{\mu}$ is the vector of expected returns and r_0 the desired level of expected return of the portfolio. The solution for $\vec{\omega}$ is:

$$\vec{\omega} = \vec{V}^{-1} \begin{pmatrix} \vec{\mu} & \vec{1} \end{pmatrix} \vec{A}^{-1} \begin{pmatrix} r_o \\ 1 \end{pmatrix} \tag{2.30}$$

Where **A** is defined by:

$$\mathbf{A} = \begin{pmatrix} a' & b' \\ b' & c' \end{pmatrix} = \begin{pmatrix} \vec{\mu}^T \mathbf{V}^{-1} \vec{\mu} & \vec{\mu}^T \mathbf{V}^{-1} \vec{1} \\ \vec{\mu}^T \mathbf{V}^{-1} \vec{1} & \vec{1}^T \mathbf{V}^{-1} \vec{1} \end{pmatrix} \tag{2.31}$$

A complete derivation of the closed form solution to the mean-variance problem can be found in Elton and Gruber (1995) [6] or Merton (1972) [7].

2.2.5 Tracing the mean-variance efficient frontier

We compute the efficient frontier for a set of 15 assets. The portfolios on the frontier have the lowest possible variance for their expected of return, or highest expected return for their variance. Individual assets are plotted in the same mean/variance plane. All except the asset with highest expected return are inside the frontier plotted in figure 2.4.

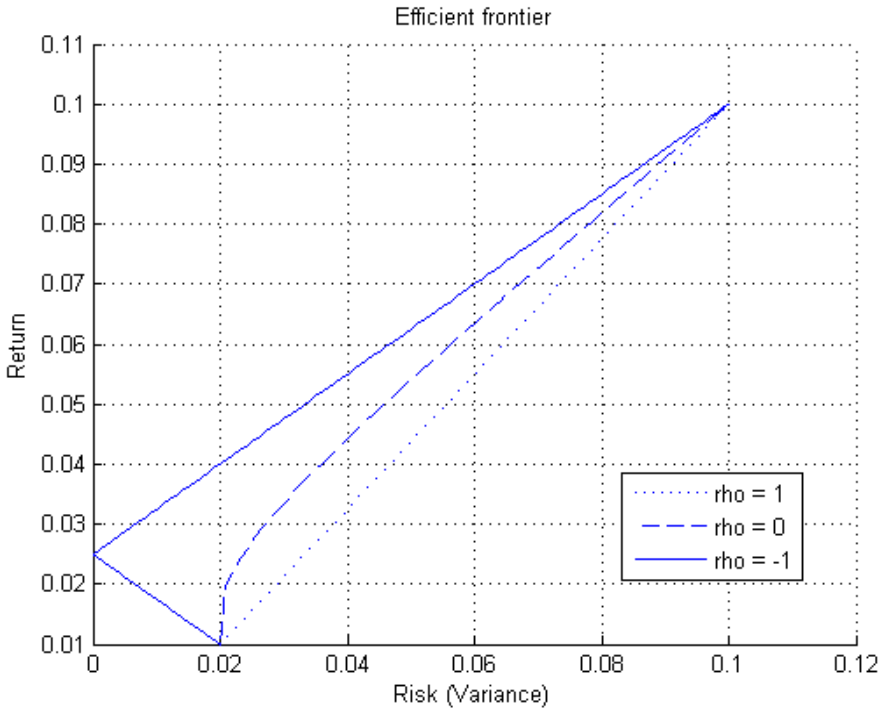

Fig. 2.3: Efficient frontier for different values of ρ

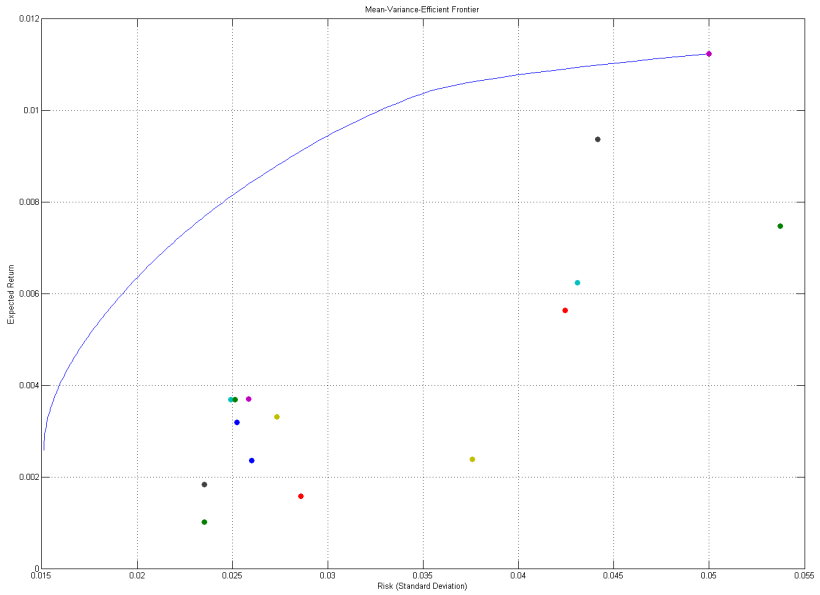

Fig. 2.4: Efficient frontier for a 15-asset universe. Each dot represents an asset in the mean/standard deviation plane. The resulting frontier stretches from a diversified low risk/low return portfolio to the maximum return portfolio (one asset portfolio)

2.2.6 Limits and problems

The mean-variance approach has been widely accepted by professionals. It is easy to understand and presents the concept of risk and return in a staightforward manner. However, this approach has yet to be proved efficient out of sample. Mean-variance optimization has limits, a few of which we want to show here.

Concentration MV-optimized portfolios are highly concentrated on a few assets with the "best" characteristics. Assets with high expected returns will be overweighted and we lose the benefit of diversification which optimization is supposed to provide.

Instability Because we don't take into account estimation inaccuracy and because MV-optimization concentrates on assets with "good" input characteristics, optimized portfolios are prone to instability. If asset characteristics change slightly, the process will re-allocate ressources on the "new" best assets, regardless of transaction costs and data inaccuracy. We can measure the concentration of a given portfolio by using the diversification index defined in the appendix. Figure 2.5 shows asset allocation and concentration over time, as given by an MV-optimization strategy. Concentration is represented by the Hehrfindal index. The curves exhibit jumps and we see that the process only invests in a small portion of the 15 possible assets. The optimal portfolio is highly concentrated on a few stocks. At the beginning of the period, 6 stocks account for portfolio variance. At the end of the period, 3 stocks suffice. One stock alone has a weight of 0.4. This goes against the whole idea of diversification, and points to one of the flaws of MV-optimization.

Sensitivity to input errors Because the MV-approach gives excessive weight to assets with larger expected returns, the resulting portfolios are highly sensitive to errors in the input data. In the optimization process, we consider the input factors $\vec{\mu}$ and $\mathbf{V}$ to be exact. No uncertainty in the data is allowed for, which goes against what we know about the accuracy of estimating asset returns and variance.

Examples We consider an investment universe of 15 stocks from the S&P500 and consider three problems. We plot the results in figures 2.5 and 2.6.

1. We solve for the minimum variance portfolio over time. We plot asset weights and the volatility adjusted number of assets in the minimum variance portfolio. This shows how diversified the portfolio is. As we can see, the optimal portfolio is concentrated on a few assets, with one asset weighing as much as 40% of the portfolio at one point. The

advantages of diversification are cancelled out by the optimization procedure.

2. An efficient frontier is a static representation of possible portfolios at a given moment. We plot the efficient frontier over time, using a one year rolling window, which we shift forward every day for one year. As we can see, the frontier changes over time. Portfolios have to be re-adjusted to remain efficient.

3. We choose a portfolio on the frontier and see how it evolves in the mean/variance plane over time: we don't readjust it, we just plot the current efficient frontier and see how the original portfolio compares.

2.2.7 Conclusion

Mean-Variance portfolio theory is easy to understand and easy to implement. Practitioners and academics alike use it as a reference point. However, it still hasn't been proved to have good investment value. It yields unstable portfolios, sensitive to slight changes in estimated returns and highly concentrated on a few assets. This goes against the stated goals of diversification and robustness.

The wild behaviour of optimal weights makes for excess transaction costs and offset any potential gains. The framework allows no room for uncertainty, even though the data we provide is very noisy and our estimations far from accurate. We can't set up the procedure so that it fits our perception of the market.

Most attempts to make MV optimization work consist in structuring constraints, setting bracket limits to asset weights. This doesn't address the fundamental flaws of the method.

We would like to test two different approaches to improve portfolio selection:

1. We use alternative risk/return frameworks.

2. We use data resampling to reduce the influence of noise on the optimization process.

Hopefully, the combination of these methods will yield better results.

2.3 Risk measures for an efficient frontier

Mean-variance portfolio theory defines a risk-return framework for comparing portfolios. In this framework, risk is defined as the variance of returns.

We quickly go over different risk measures, starting with variance, and explain what each has to offer and why it should or shouldn't be used. These

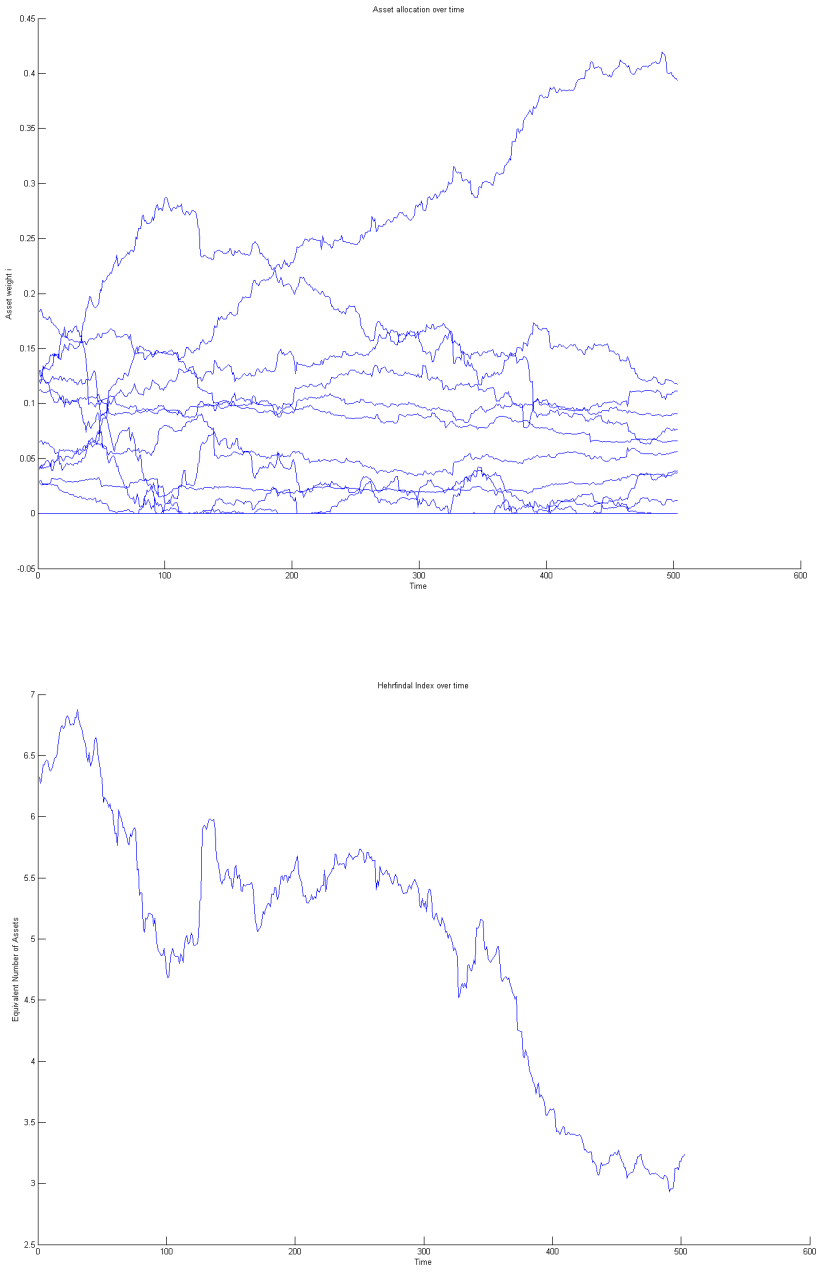

Fig. 2.5: *Top*: asset weights of the minimum variance portfolio over time in a 15-asset universe. *Bottom*: variance-adjusted equivalent number of assets in the minimum variance portfolio over time in the same 15-asset universe.

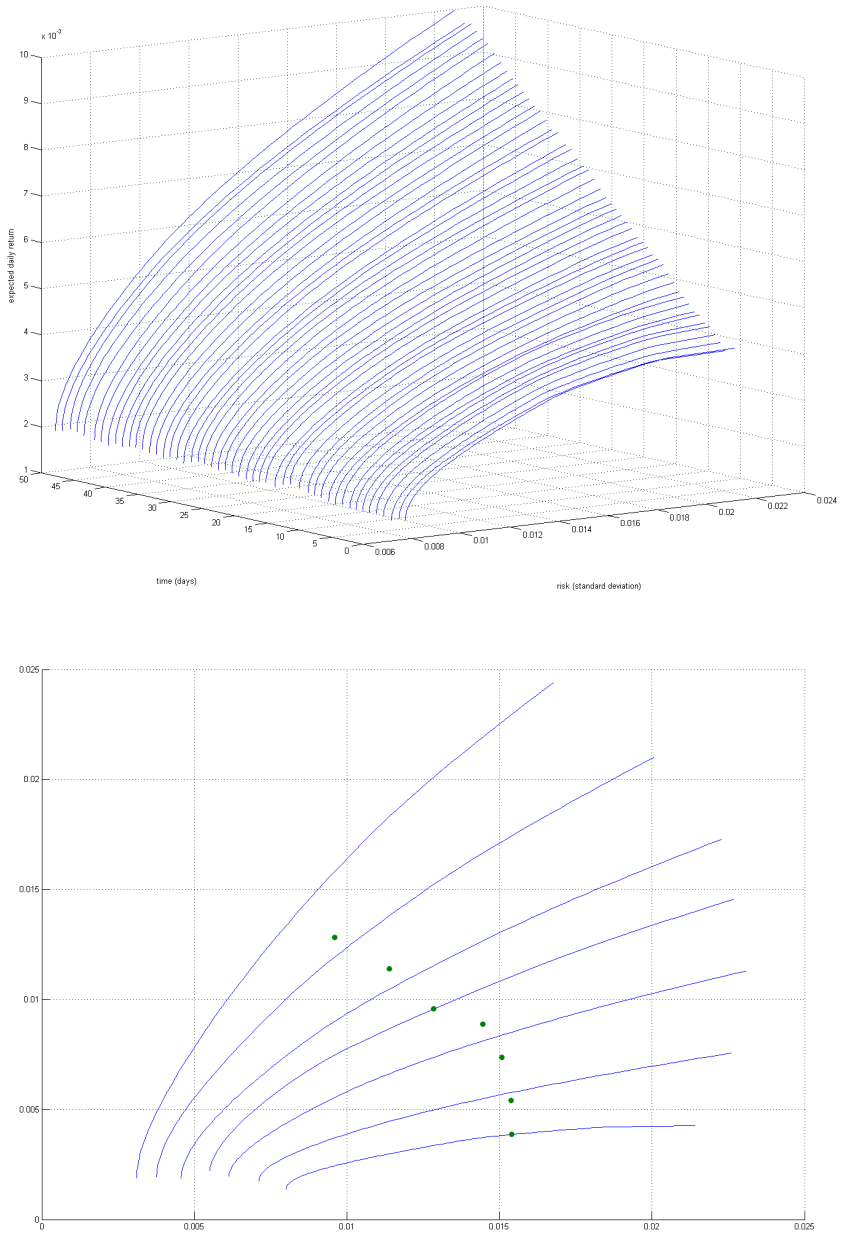

Fig. 2.6: *Top*: evolution of the efficient frontier for a 15-asset universe over 50 days, based on a 1 year rolling window. The shape of the frontier changes over time. *Bottom*: we choose a portfolio on the frontier and let it evolve over time in the mean/variance plane. Without readjustment, it drifts away from the frontier and is no longer optimal. The frontiers are drawn at one-month intervals, starting with the bottom curve at time 1.

are in no way rigorous mathematical definitions. The idea is just to get a feel for what these risk measures actually take into account, and what they don't. For a rigorous and complete overview of risk measures, we recommend Embrechts and McNeil (2005) [8].

2.3.1 Variance

Variance is defined as the second centered moment of returns around the mean. Simply put, it gives a measure of "how fat" a distribution is. In the case of normally distributed returns, it does a good job of describing the distribution.

Let r_i be a random variable, we define the variance σ_i^2 and standard deviation σ_i of r_i as:

$$Var(r_i) = \sigma_i^2 = E[(r_i - E[r_i])^2] = E[(r_i - \mu_i)^2] \tag{2.32}$$

If asset returns were normally distributed, they could then be fully described by their mean and variance. However, asset returns are most likely not normally distributed. Figure 2.7 shows the distribution of daily returns for starbucks stock (SBUX) over a period of five years, from january 2002 to january 2007, and a distribution of returns drawn from a normal distribution with mean and variance equal to the sample mean and variance of our starbucks data.

We see that the distributions don't have the same characteristics. Whereas the realized return distribution has a number of absolute returns over 0.08, the resampled normal distribution has no returns over 0.07, despite a greater amount of draws. Simply put: the realized return distribution exhibits heavy tails and a slim waist, while the resampled distribution has no tails and a heavy middle. A rigorous demonstration would have us give kurtosis for both distributions and compare the tails in a log-lin plot.

Variance doesn't put much more emphasis on extreme returns than on small returns. This doesn't describe most investors' risk-aversion: people don't like suffering large losses.

Another characteristic of variance is its symetry. It penalizes negative returns as much as positive ones. This once again doesn't reflect investor preference: people most likely don't mind positive returns, but want to avoid losses.

2.3.2 Semi-Variance

Semi-variance is defined in the same way as variance, but only takes into account returns under the average return. This is an improvement on variance, because we don't consider positive returns as risks, quite the contrary. While we don't want to be exposed to negative returns, we happily accept the *risk* of large positive returns. If the distribution of returns is gaussian, then

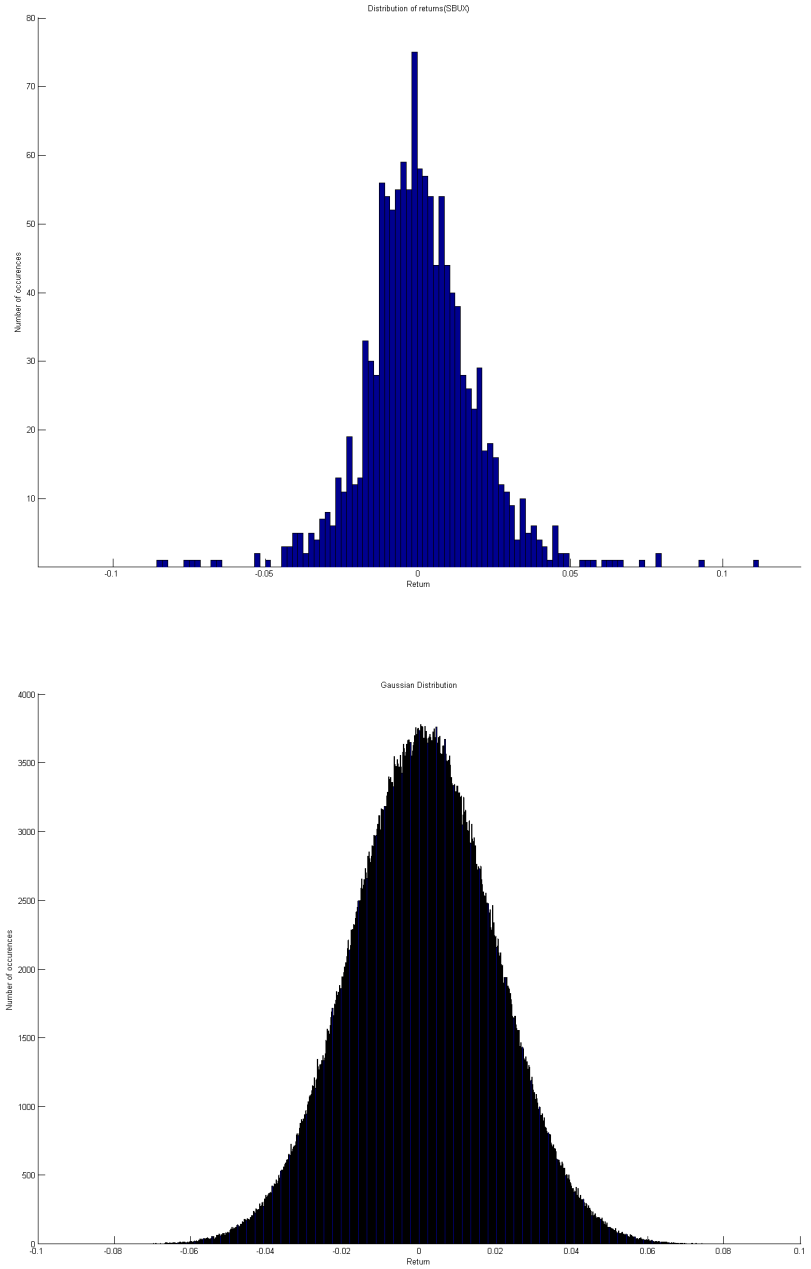

Fig. 2.7: *Top*: distribution of daily returns for Starbucks stock (SBUX) from January 1st 2002 to January 1st 2007. *Bottom*: Resampled normal distribution of returns drawn from a gaussian of same mean and variance. We see that the assumption of normality doesn't fit the data we have: the realized return distribution exhibits heavy tails and a slim waist, while the resampled distribution has no tails and a heavy middle.

variance and semi-variance are equivalent measures of risk. We can calculate and plot mean/semi-variance efficient frontiers as defined by Markowitz. Mean semi-variance portfolios are presented in Konno and Yamazaki (1991) [9].

2.3.3 Higher-order moments

The concept of variance is easy to grasp, but is ill-adapted for the real world. It doesn't put sufficient emphasis on the risk of large returns. In a gaussian world, there are no extreme events, so it doesn't matter that they aren't taken into account. The real world, however, isn't gaussian. In order to capture the risk of extreme returns, we can extend the concept of variance to the nth-order centered moment around the mean. This puts more weight on large returns. This measure "squashes" small fluctuations and emphasizes large returns.

$$M_n = E[(r_i - E[r_i])^n] = E[(r_i - \mu_i)^n] \tag{2.33}$$

Using the same framework defined by Markowitz, we can now trace generalized efficient frontiers for given levels of returns, as proposed in Sornette and Malevergne (2005) [10].

2.3.4 Value at Risk

An alternative risk measure to variance is Value at Risk (VaR), which is quite popular in the financial community. VaR is the maximum amount you stand to lose over a given period of time and for a given confidence level. For example, we can compute the value at risk for a given portfolio with a level of certainty of 95%. We get an amount x, of which we can say: "there's a 95% chance of not losing more than x over the next period". Typical confidence levels are 95% or 99%. We thus know that 95% of potential losses over the next period will be under x. But what can we say about the remaining 5%?

2.3.5 Expected Shortfall

Expected Shortfall is a measure of large risks. Given a 95% confidence interval for VaR, expected shortfall is what we stand to lose if we happen to fall into the unlucky 5%.

2.3.6 Drawdown measures

For some investors, fluctuations in portfolio value aren't very important, just as long as negative returns are followed by gains. Drawdown measures consider the risk of sustaining sequential losses. What is the worst that could happen to an investor buying at a peak and selling at a subsequent low over

a certain period? We consider maximum and average drawdown. Maximum drawdown over a given period is the maximum amount an investor can lose over a period in the worst case scenario: buying at peak value and selling at the lowest subsequent value. Average drawdown is the mean drawdown over the considered period.

In the same way we defined Value at Risk, we can define Drawdown at Risk. We can compute drawdown at risk x for a given level of confidence, say 95%. With 95% probability, there will be no drawdowns over x.

2.4 Historical data and resampling methods

We work with historical data. For every asset, we have the time series of prices and returns. Chances are, if we could go back in time and let prices evolve freely a second time from some given point, the outcome would not be exactly the same. One way of seeing this is that the prices we observe are one realized path of a stochastic process. This process is somewhat random and noisy, so using this single path to characterize an asset might yield bad results.

We would like to use this realized outcome to gather information about the underlying process: how are the returns distributed, how are assets correlated?

We consider three different methods for using data to estimate parameters. In our framework, we need estimates of the expected returns, variances and drawdowns for each asset in our investment universe. To illustrate how each method works, we once again consider (SBUX) stock, for which we get data over five years.

2.4.1 Historical approach

We can simply use historical data as estimates of the expected values of the parameters we want to characterize. We take the average return over our calibration window as an estimate of the expected return, and the covariance matrix of assets over our calibration window as an estimate of the expected covariances between assets. This is a naive approach, which we can improve by applying some kind of weighting to the data. We might want past data to play a smaller part in the estimation, and weight old data less than recent data.

2.4.2 Fitting distributions

Another approach is to fit a distribution to the data we observe, just as we did for starbucks stock (SBUX). Suppose we consider that returns are normally distributed. We want an estimate of the expected return $E(r_i)$. We observe the mean (μ) and standard deviation of historical returns (σ).

We then generate p alternative price paths for the asset by drawing returns from a normal distribution $N(\mu, \sigma^2)$. We do this p times and average the mean return over the n realizations.

2.4.3 Resampling methods

We can also do the same thing without making any assumptions about the form of the distribution function. We generate paths by drawing returns from the historical distribution. This has the advantage of not having to make any assumptions about the form of the distribution.

Correlation in time It's been shown that returns aren't always independant from one another. Large (absolute) returns tend to follow large (absolute) returns. There is a certain amount of persistance in the market.

If we draw returns from the whole historical distribution, we lose the correlation structure between returns. To preserve this structure, we cut our return series into smaller subsets, shuffle and re-assemble them. Think of it as a shotgun approach to data generation.

If we want to preserve n-day correlation, we consider strings of n-day returns. We place these return segments in a virtual bag from which we draw. By randomly cutting our return series, we generate return segments of different lengths. This might be the best way to replicate data. We will be keeping structure on different time scales.

2.5 Conclusion

In this chapter, we have explained the foundations of mean-variance portfolio theory. Its simplicity appeals to academics and practitioners alike. We have also seen that it still isn't possible to use these "optimal" portfolios directly.

To make theory more robust and draw investment-grade information from the mean-variance framework, we would like to add factors to our objective function and go beyond the simple expected mean and historical variance.

To solve these new problems, we will use a genetic algorithm. We describe genetic algorithms and there applications in the following chapter.

Chapter 3

Genetic Algorithms

3.1 General information

> Fishes are generally streamlined with a pointed snout and pointed posterior and a broad propulsive tail. Unlike the shape of a human body, a fish's body shape is ideal for speeding through the water without creating excess resistance. *MSN Encarta*

Nature often finds very good solutions to complicated and diverse problems. Humans aren't bad at designing aerodynamic or hydrodynamic shapes, especially when nature gives us examples of what is efficient. But whereas it takes a team of engineers to determine what shape a torpedo should have, fish have evolved from scratch.

Genetic algorithms are heuristic search methods inspired by nature. They can be used to find solutions to optimization problems where there are no "good" deterministic search methods. Principles of evolutionary biology, such as natural selection and reproduction serve as guidelines to evolve a population of solutions to a given problem.

3.1.1 Principles of GAs

The big advantage of genetic algorithms is that you don't have to specify all the details of a problem in advance. In fact, you don't even have to *know* all of the details. Potential solutions are evaluated by a fitness function representing the problem we want to solve. We then define an evolution procedure to produce new candidate solutions. The idea is that combining good solutions (solutions that score high on the fitness scale) should lead to better solutions. By adding some noise (mutating the candidate), we hope to find better solutions. Part of the evolution process consists in choosing the members which will form the next generation of solutions. There are many different methods.

Hopefully, this "oriented" random search will lead us to a "good" solution. It isn't necessarily the best solution, but since GAs are usually employed in cases where there are no easily found optimal solutions, a good solution is still better than nothing.

There are two main steps in genetic algorithms.

1. Write your problem in a way that solutions can easily be coded and changed automatically. This is the "mutate and reproduce" part of evolution.
2. Specify a fitness function f. We will use this function to compare and rank solutions. This is the "survival of the fittest" part of evoution.

3.1.2 Pseudo-code

1. Draw a random population of n candidates. This is the old population.
2. Generate an intermediate population of n candidates.
3. Compare and rank the $2n$ candidates
4. Choose the n highest scoring candidates as the new population.
5. Go to step 2.

We stop when certain criteria are met: number of cycles, time passed, good score...

3.1.3 Constraints

Optimization problems often have constraints, which we have to include in our algorithm. We describe two methods for doing this:

First method Add a penalty to the fitness function, so that candidates violating the constraint score less. Suppose we would like to solve the following problem:

$$\text{maximize } f(x, y)$$

$$\text{st } x + y = 1$$

We can include the constraints by building another fitness function g, such that:

$$g(x, y) = f(x, y) - p(x, y)$$

Where $p(x, y)$ is a penalty function which increases when the constraints are violated. For example, if K is some positive constant.

$$p(x, y) = K * |x + y - 1|$$

The bigger K is, the stronger the constraint. This keeps vectors which violate constraints on the sidelines of the selection process. At the same time, if the constraints are too harshly enforced, especially in the beginning, we might narrow down the population too much. The GA might converge to a sub-optimal solution or run very slowly. It might not even converge.

We might just be solving the constraint conditions. The reason for this is that our constraints define a subset of the whole search space. If this subset is too small and if we can't violate conditions temporarily, we might not find a solution.

The following problem is defined over the whole plane of pairs of real numbers.

$$\text{maximize } f(x, y)$$

$$\text{st } (x, y) \in R^2$$

The problem below is defined over a straight line.

$$\text{maximize } f(x, y)$$

$$\text{st } x + y = 1$$

Suppose our genetic algorithm randomly draws pairs of real numbers, then ranks these solutions, generates new ones and so on... The probability of randomly drawing points on a straight line in a plane universe is quite small (smaller than that actually...). Then, when you randomly mutate the points you have got, you will probably not move in the direction of the line, and keep "falling off" the line. So this method takes computer power, time, and lots of hope...

Second method Try to write the constraints into the algorithm. For example, if we want the coordinates of each candidate to sum up to one, we should normalize the intermediate population before ranking the different solutions.

$$x \longleftarrow \frac{x}{sum(x)}$$

Another way of thinking about these two methods is to say that the second method randomly chooses points, projects them ALL into the admissible search space and then optimizes over them, whereas the first method randomly draws points, forgets the constraints, optimizes all the same, then throws out all the points which weren't in the admissible search space to begin with.

3.1.4 Stringing beads

We are given a bag of black and white beads from which we make 20-bead bracelets. A friend randomly makes a bracelet and we want to make the exact same one. We can't look at the bracelet, but our friend can tell us how many beads match. If 10 beads match, we get 10 points, for 5 beads we get 5 points... We want to find the bracelet in as few trys as possible. If we were to make all the possible bracelets successively and compare each of them, we'd have to compare $2^{20} = 1048576$ bracelets. That's a lot of bracelets.

Set up

1. Each time we compare bracelets we increment our count.
2. Two symetrical bracelets are considered to be two different bracelets.
3. We lose no generalization by supposing that the random bracelet is all black beads.

GA representation First we have to choose how to represent our solution bracelet. We set black beads to be ones and whites beads to be zeroes. Each bracelet can be represented as a string of ones and zeroes.

10101010101000110101

We consider candidate vector x, where $\forall i, x_i$ is 0 or 1:

$$x = x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}x_{11}x_{12}x_{13}x_{14}x_{15}x_{16}x_{17}x_{18}x_{19}x_{20}$$

GA scoring Now we need to define a fitness function. We just sum up the string.

$$f(x) = \Sigma x_i, for\ i = 1..20$$

We draw an initial random population of 10 black and white bead bracelets.

10101010100101001101	10101010100110100101
00101110101011010010	10001010101101011001
10010000101011011110	10100000110101000001
00110100000010101010	10001010101101011001
10111100100101011000	00000000101101000101

We define the way our population evolves. At each generation, 2 things happen to all the vectors/bracelets:

1. Each bead will mutate with some small probability p_m, in which case we get:

$$10101010100\boxed{1}01001101 \longrightarrow 10101010100\boxed{0}01001101$$

2. With some small probability p_c, some bracelets will exchange whole parts of their sequence randomly. This is called a crossover.

$$10101010100\boxed{101001101} \longrightarrow 10101010100\mathbf{010101010}$$
$$00110100000\mathbf{010101010} \longrightarrow 00110100000\boxed{101001101}$$

Running the GA We run through this sequence of mutations and crossovers, getting a new population at each loop. Each time we compare a bracelet to the target one, we increment our counter.

Results On average, it takes the algorithm 70 comparisons with the target bracelet to find the exact solution. For one of these trys, we plot the average and best scores inside the population for each generation. The results are plotted in figure 3.1.

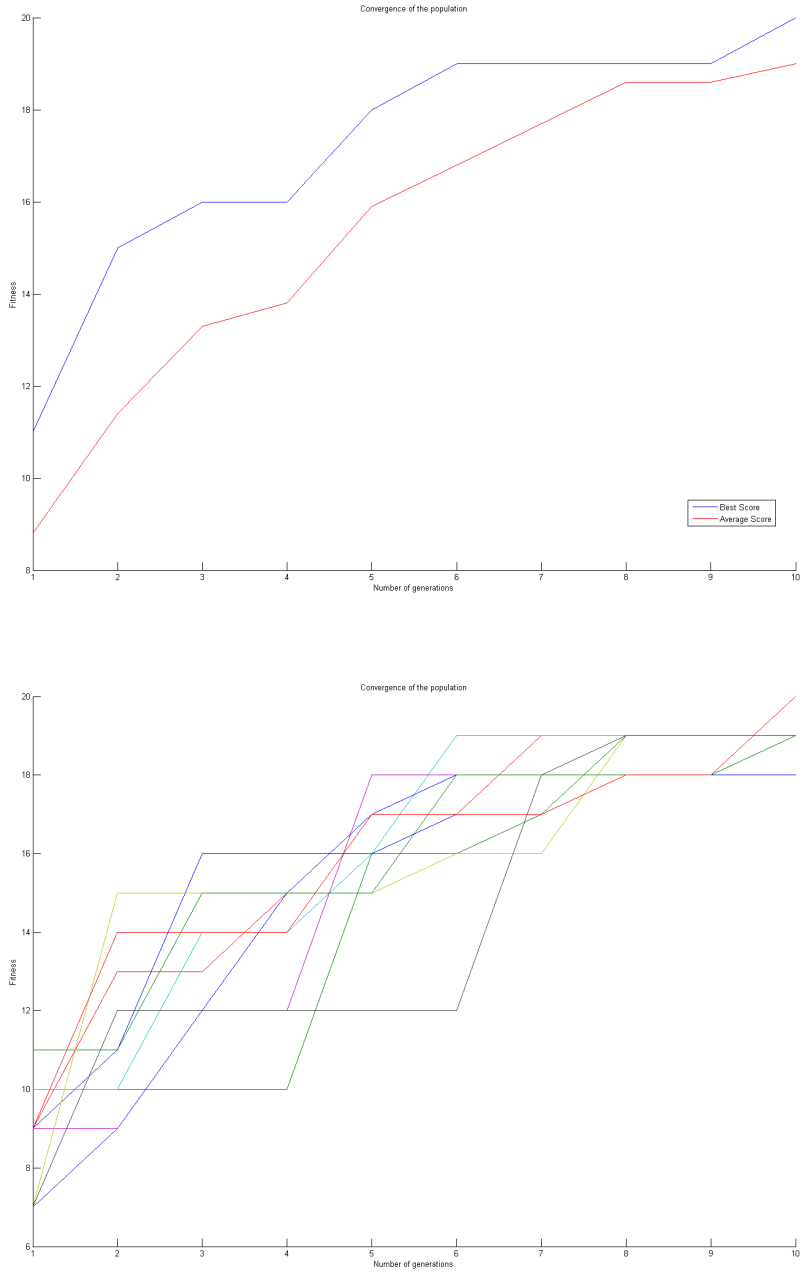

Fig. 3.1: *Top*: We plot the average (red line) and best (blue line) scores inside the population for every generation. *Bottom*: We plot the individual score of all the population members over generations. Each color is the evolution of one vector in the population. Score is plotted along the y-axis, generation number is on the x-axis.

3.1.5 Another example

We would like to solve the following problem:

$$\text{minimize } f(x_1, x_2) = |x_1 - 0.5| + |x_2 - 0.5|$$

$$\text{st } (x_1, x_2) \in R^2$$

Results We know the exact solution is $(x_1, x_2) = (0.5, 0.5)$. We plot the convergence of the algorithm towards the optimal solution in graphical form in figure 3.2.

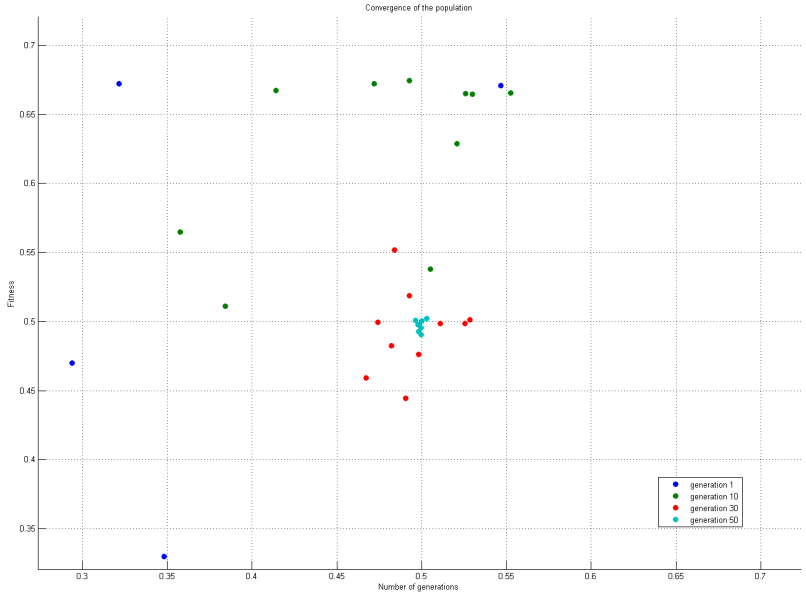

Fig. 3.2: We plot the pairs of solutions (x_1, x_2) for different generations. We see that each generation comes closer to the exact solution. We start out with a widespread cloud of points. Over time, this cloud gets more and more concentrated. At generation 50, all the pairs are very near the exact solution.

3.2 Defining our problem

We first have to write our problem in a GA-friendly way. We want to find *optimal portfolios* of assets, meaning portfolios which score high on our fitness scale.

We also have to define a set of constraints which each portfolio must verify.

Representing solutions We we can choose among n assets to build our portfolio. Asset i has weight ω_i in the whole portfolio, so we will represent a candidate portfolio x by the vector of it's asset weights:

$$x = (\omega_1, \omega_2, \ldots, \omega_n).$$

Constraints We will consider 2 constraints:

1. No short-selling. So we must have $\forall i, \omega_i \geq 0$
2. We invest all our money, but no more. So we must have $\Sigma\omega_i = 1$

3.2.1 Algorithm design and principles

The genetic algorithm we are using is based on the *differential evolution* code developped by Rainer Storn and Ken Price [11].

Pseudo-code

1. Draw a random population of n candidates. This is the *old population*.
2. While $iter < itermax$

   Evaluate each member's fitness. Find the best member.
3. Generate an *intermediate population* of n candidates:

   $iter = 1$

   for $i = 1$ to NP

   for each candidate i, randomly choose 2 different members from the population. Substract one from the other and add F times the difference to candidate i.

   for $j = 1$ to D

   randomly choose ω_j from either the best member (with probability CR) or the modified version of candidate i (with probability $1 - CR$). This is intermediate candidate i
4. for $i = 1$ to n

   compare candidate i to intermediate candidate i. The fittest member goes into the next generation and we now have the *new population*.
5. $iter = iter + 1$

   Go to step 2.

Parameters

- n number of individuals in the population
- D number of assets
- CR crossover ratio
- F mutation factor
- $itermax$ maximum number of generations in a run

Designing constraints

As we saw before, we can enforce the constraints either by including a penalty in the objective function, or by writing the algorithm so that the constraints aren't violated. We try both methods for our problem:

$$\text{maximize } f(x)$$

$$\begin{gathered}\text{st } x = (\omega_1, \omega_2, \ldots, \omega_n). \\ \forall i, \omega_i \geq 0 \\ \Sigma\omega_i = 1\end{gathered}$$

First method We include a penalty function in the objective function so that out-of-bounds candidates will score less, and get the following objective function:

$$g(x) = f(x) - \alpha\left(|min_i\left(\omega_i\right)| - min_i\left(\omega_i\right)\right) - \beta(|\Sigma\omega_i - 1|)$$

where α is a penalty factor for the first constraint and β for the second constraint.

Second method For both constraints, we choose to normalize and rectify the population at each generation, just before ranking.

As it turns out, the second method is much faster, so we stick to it.

Definitions

- *generation*: All members of an initial population are evaluated and the best member found. From this, we generate a new intermediate population. Each intermediate member i is compared to to initial member i. The best of the two is kept in the population.
- *run*: Starting with an initial random draw over the search space, we let the population evolve over generations, until a certain criterium is met. This series of generations is a *run*.

3.3 Calibrating the algorithm

The goal of this work is to develop a multi-objective fitness function and have the genetic algorithm find stable solutions to the problem thus defined.

3.3.1 Objective function

Our function is the sum of different functions:

$$f = f_1 + f_2 + f_3 + f_4$$

Where:

f_1 is a performance factor in terms of annualized returns.

f_2 is a risk cost.

f_3 is a transaction cost due to potential reallocation of the initial portfolio.

f_4 is a concentration cost.

3.3.2 Minimum-Variance portfolio

GAs are useful in cases where there are no analytical solutions or no good deterministic search methods. However, we would like to set the search parameters to make our GA efficient. We want to compare GA results to the analytical solution of a given problem.

In this case, we choose to find the minimum variance portfolio, for which we have an exact solution. It is actually a subfunction of our objective function, with f_1, f_3 and f_4 set to 0, and using variance as a risk measure.

Set up To see if our settings lead to good solutions, we measure the error between our search method and the exact result. So the problem we want to solve now is:

$$\text{minimize } f(x) = xVx'$$

$$\begin{gathered}\text{st } x = (\omega_1, \omega_2, \ldots, \omega_n). \\ \forall i, \omega_i \geq 0 \\ \Sigma\omega_i = 1\end{gathered}$$

Where V is the covariance matrix of asset returns

3.3.3 Computing-time and convergence

Convergence

There is no proof of convergence for the DE algorithm, so we can't be sure that every run will converge. And when a run converges, we have no idea how good the output portfolio is.

We would like to know how fast the population converges towards the best solution during one run. This way we should know how many generations we should let the GA go through at each run. Figure 3.3 shows the global convergence of the population.

Computing-time and the dimension of the problem

We would like to know how computing time increases with the number of assets. We define performance criteria for the algorithm to reach and see how time to completion increases with the number of assets.

We count the number of times the objective function has to be evaluated for the algorithm to find 5 portfolios within 1% variance of the minimum variance portfolio, for which we know the exact solution.

The idea is to calibrate the algorithm on a model for which we know the solution and use the settings to solve bigger problems.

We set the algorithm so that one *run* ends:

- after the convergence of the population, which we define by checking when the average population score reaches 90% of the best scoring individual.
- after finding a vector within the target range.

We then plot computing time in a $(D, nfeval(D))$ plane, as showed in figure 3.3.3, to get a rough idea of how the algorithm scales with respect to the dimension D of the problem.

Tracing the mean-variance efficient frontier

We now want to compute and plot the MV-efficient frontier using the GA algorithm. Since we have an exact solution, we can compare the results to see how good the algorithm is. We trace the exact efficient frontier and plot 10 points on the frontier using the GA.

We compute the frontier by maximizing a set of objective functions f_λ over our search space.

$$f_\lambda(x) = \lambda r'x - (1-\lambda)xVx'$$

Parameters

- λ amount of risk, $0 \leqslant \lambda \leqslant 1$
- x candidate portfolio
- r vector of historical mean returns

By varying λ from 0 to 1, we move along the frontier, from the minimum-variance portfolio to the maximum-risk/maximum-return portfolio, as shown in figure 3.5.

results We successfully compute and plot the frontier using the GA. In some cases, the GA actually performs better than the built-in closed form solution by matlab.

Even with small sample sizes, approximation errors keep matlab from inverting the covariance matrix correctly and the frontier can't be computed. The GA has no such problems, because we don't invert any matrices...

3.4 Practical measures for implementing the GA in our framework

3.4.1 Generations and runs

As we've explained before, we need to define stop criteria for the algorithm. If we don't it will continue searching for improvements in the population. However, when the population has converged, any improvements will be the result of a random search, and very slow.

We define different criteria:

1. When the average score in the population is less than 1% from the best score, we consider that the populatio has converged and end our search.

2. After a number x of runs without improvement in the best member, we stop our search.

3.4.2 Convergence problems

One of our main priorities is to find robust results. If we run the GA several times on the same problem, we want the find the same solution each time. GAs are good at finding "pockets" of extreme points in a search space, meaning they won't get stuck in a local extreme point. However, once they're in a convex subspace of the search space, they perform rather poorly compared to deterministic search methods. Another way of saying this is that GAs don't crawl over flat surfaces very well. There are 2 ways to solve this problem.

1. Run a deterministic solver after a number of runs to find the exact optimal portfolio. This would be time consuming, both in running time and programming time. At the same time, it goes against one of the stated goals of this paper. We aren't interested in finding the perfect portfolio at a given time. We want a portfolio which does well most of the time. Considering the imprecisions in data, there is no point in optimizing a portfolio to the extreme over one period.

2. Force convergence by defining a minimal allocation step. This goes well with practical applications: a fund manager will buy 1% or 5% of an asset, not 0.01%. In our case, we set a minimal investment levl of 1%.

3.4.3 Dynamic optimization

MV optimization was developped as a static view of possible investments. Our goal here is to develop a dynamic method. The framework we use calls the genetic algorithm repeatedly to find optimal portfolios. At each reallocation time, we need to re-run the algorithm. As stated before, we hope to find stable portfolios with small reallocation costs.

In keeping with this idea, we seed our population of random portfolios at time t with the best portfolio at time $t-1$. This way, we have at least one good starting point for our new function.

3.5 Conclusions

We chose to use a genetic algorithm for different reasons. They perform well in many cases where deterministic algorithms don't, such as non-convex problems. This allows us the possibility of using objective functions with non convex parts. The GA will simply crawl over the space and find good solutions. Local convergence problems were then solved by setting minimum increments in reallocation possibilites.

As it turns out, our objective functions were convex, and could have been solved using deterministic algorithms. However, this would have required a certain amount of programming to set up the algorithm for each function.

The GA approach simply required a few lines of simple code, defining the objective function. The basic framework didn't depend on the function we used and didn't have to be changed at each new trial.

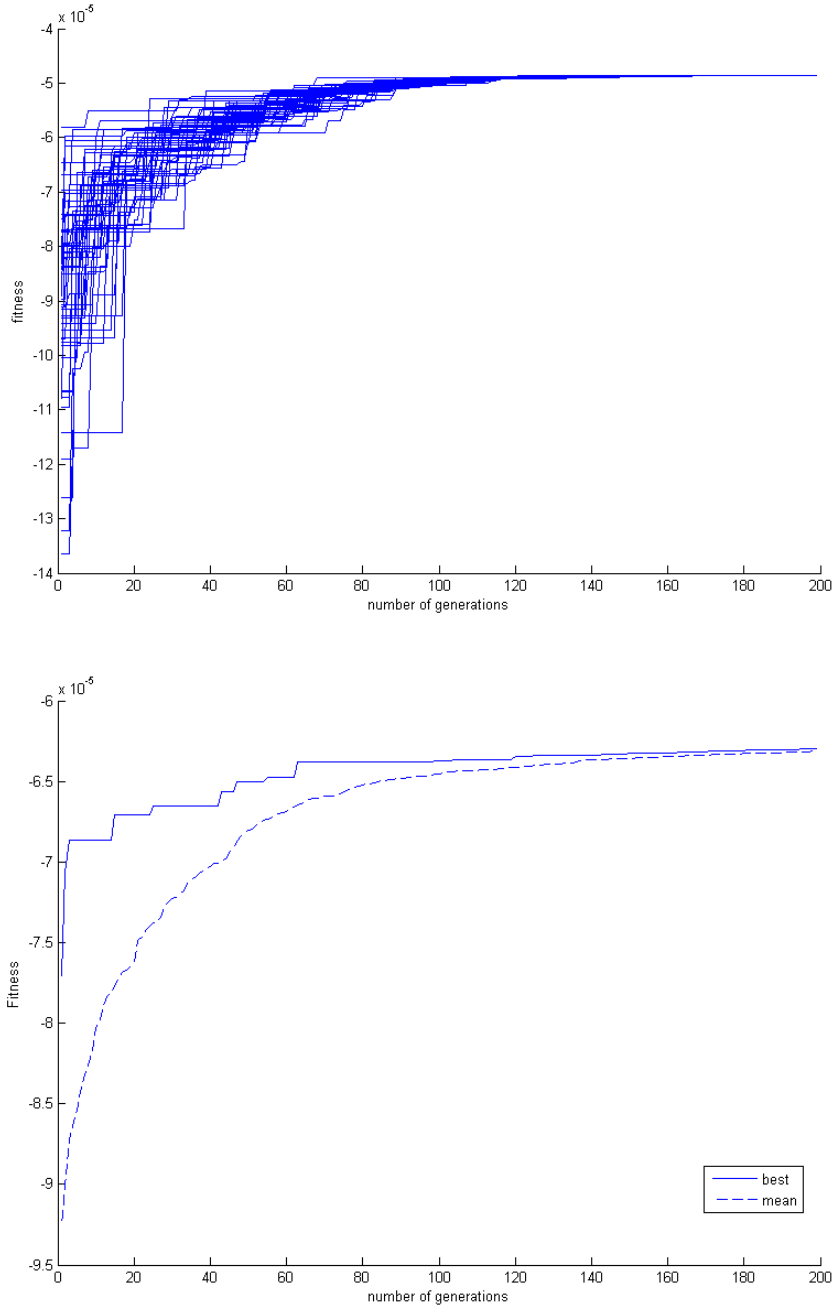

Fig. 3.3: *Top*: Fitness of every member of the population over time. *Bottom*: Best and average scores in the population over time. At one point in time, the population has completely converged and we can stop the algorithm. Beyond this time step, we are just randomly searching the space around one point. Fitness is plotted along the y-axis, generation along the x-axis.

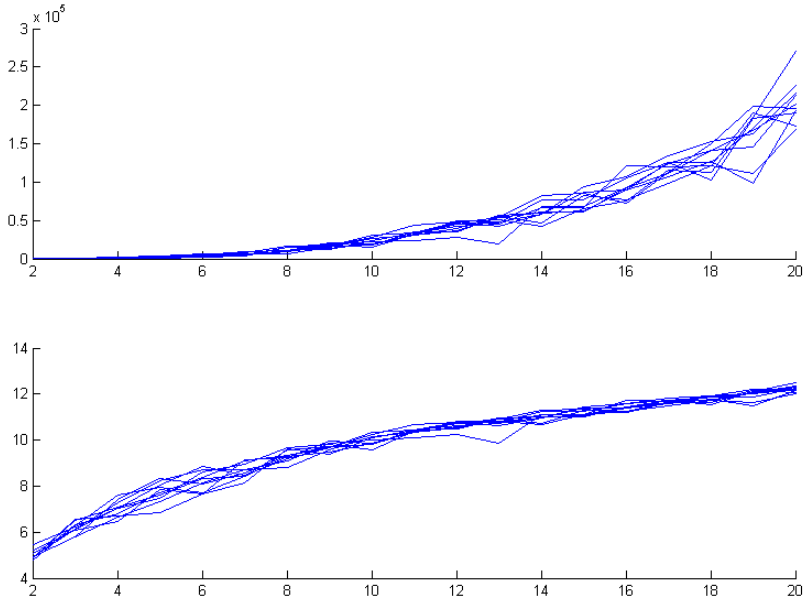

Fig. 3.4: Computing-time (y axis) as a function of the number of assets D (x axis), in a lin-lin (top plot) and log-lin (lower plot).

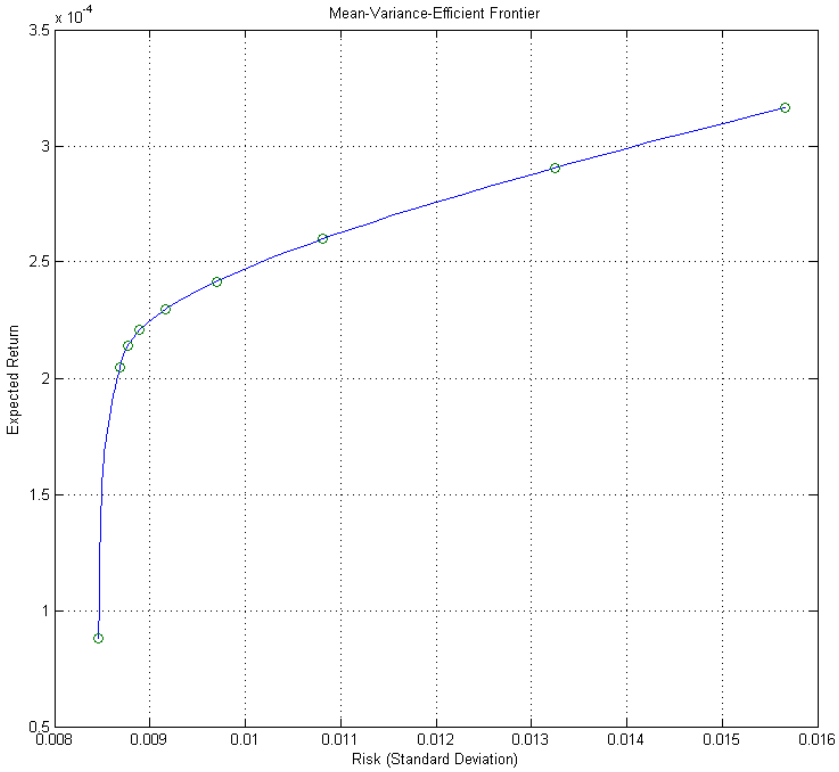

Fig. 3.5: Efficient frontier as computed with the closed form solution (continuous line) and with the GA (green points)

Chapter 4

Tests, framework and results

Mean-Variance as we have described it is a static process: at one point in time, we look at past data and decide which portfolio is optimal for our mean/variance preferences. The goal of this project is to develop dynamic portfolio optimization methods. We want to choose a portfolio and then reallocate assets so that the current portfolio is always optimal by our fitness standards.

Nothing keeps us from simply using MV-optimization over time. However, this method has been proved to be ineffective, due to excessive reallocation costs. We want to develop and then compare different strategies against a number of standard techniques, some of which are very simple and surprisingly effective.

4.1 Framework

We define the framework in which we compare existing methods as well as the ones we have developped.

4.1.1 Data

We consider three different times periods, corresponfing to different regimes. From January 1st 1995 to January 1st 1999, January 1st 1999 to January 1st 2003, and January 1st 2003 to January 1st 2007. This way we make sure that the results we get are robust over time.

4.1.2 Investment universe

We consider assets from the S&P500. We usually take subsets of 8 to 15 assets, sometimes more. Our algorithm is very computationally intensive and we have to limit ourselves to small data sets.

4.1.3 Backtesting procedure

At time t we choose a window of past returns of length T weeks. At a given time t, we consider a window stretching from $t - T$ to t. We denote this window W_t. It is a T x m matrix of weekly returns for the m assets in our investment universe.

We consider weekly re-allocation of assets. Daily re-allocation would result in excessive transaction costs. Many funds reallocate on a monthly basis. Ultimately we would like the algorithm to choose to reallocate when it seems right and not have to discretize the procedure.

Procedure

1. Get W_t.
2. Run the GA over W_t. Choose the optimal portfolio for time t, which we note P_t.
3. Reallocate assets from P_{t-1} to P_t.
4. Compute the realized return on our portfolio at time t using the realized returns at time t of all the assets in our universe and transaction costs.
5. Shift window W_t by removing the oldest return and adding asset returns at time t.
6. start over

Outputs

At the end of one period, we compute and plot networth over time. We compute different performance measures to compare portfolios.

4.2 Performance measures

How should we rank different strategies? One technique might yield a large return one year, and large losses the next. Another might have smaller returns, but less variations. In order to compare different strategies, we define performance metrics usually used in the financial community.

4.2.1 Sharpe Ratio

The Sharpe Ratio is the ratio of excess annualized returns over annualized standard deviation of returns σ. By excess returns, we mean returns over the riskless rate r_0. The Sharpe Ratio gives a measure of how much portfolio

returns fluctuate over a given period.

$$SR = \frac{r - r_0}{\sigma}$$

If a portfolio has a $r = 9\%$ yearly return and $\sigma^2 = 16\%$ variance over one year, and the riskless rate r_0 was 5%, then our portfolio has a Sharpe Ratio SR:

$$SR = 1$$

4.2.2 Calmar Ratio

For investors less interested in small fluctuations of their portfolio than in large risks, the Calmar Ratio is more appropriate. It is defined as the ratio of excess annualized returns over the maximum drawdown in a given period.

$$CR = \frac{r - r_0}{d}$$

If a portfolio has a $r = 10\%$ yearly return and a maximum drawdown $d = 10\%$ over one year, and the riskless rate $r_0$was 5%, then our portfolio has a Calmar Ratio CR:

$$CR = 0.5$$

4.3 Objective functions

We develop different objective functions, which we use as inputs to the optimization problem we solve at each time step:

$$\text{maximize } f(x)$$

$$\begin{gathered}\text{st } x = (\omega_1, \omega_2, \ldots, \omega_n). \\ \forall i, \omega_i \geq 0 \\ \Sigma\omega_i = 1\end{gathered}$$

4.3.1 Benchmarks

In order to compare strategies, we need to set a banchmark performance. Surprisingly, very simple techniques perform very well. We choose variations on the market portfolio as a benchmark to beat. This strategy is actually very good. In practice, it's very hard to "beat the market". An in-depth comparison of different strategies can be found in DeMiguel et al. (2005) [12].

Equally weighted approach

We simply buy equal shares in every asset available to us. This is diversification pushed to its extreme. As we will see, this technique is very efficient

and very hard to beat. Consequently, we use it as a reference portfolio. In this case, we readjust portfolio weights so that we stay on a $\frac{1}{n}$ weighting over time. This has consequences if transaction costs are high.

Equally weighted approach, then free

We buy into the market with equal shares in every asset and then let the weights drift away depending on each asset's returns. This is also a zero turnover strategy.

Value-weighted portfolio

In this case, we simply buy into the market proportionally to each asset's market weight. In a CAPM world, this is the optimal strategy. This is also a zero turnover strategy.

4.3.2 Standard functions

Minimum variance portfolio

We simply choose the portfolio with minimum variance over our calibrating window.

Minimum semi-variance portfolio

As explained before, variance is a symetric risk measure. We are only loss-averse, so choosing the portfolio with minimum historical semi-variance suits our risk aversion better.

Minimizing higher order moments

Variance is a bad measure of the risk of large returns. Using higher order moments, we emphasize large returns over small ones. It means that we are willing to stand small fluctuations in our networth in order to miss large drops. We test the fourth and eigth order centered moments around the mean.

4.3.3 Multi-factor objective functions

We design an objective function f as a sum of different functions:

$$f = f_1 + f_2 + f_3 + f_4$$

Where:
f_1 is a performance factor in terms of annualized returns.
f_2 is a risk cost.

f_3 is a transaction cost due to potential reallocation of the initial portfolio.

f_4 is a concentration cost.

We are developping a one-period model. At each time-step, we consider the optimal portfolio over the next period.

Performance f_1 We compute the mean historical return vector over our calibrating window and use it as an estimate of the expected return over the next period. This is a naïve approach.

Risk f_2 We start out using variance as a risk measure. For a given test portfolio, we compute the variance of returns over the calibrating window, and use it as an estimate of the variance of returns over the next period

Taking costs into account f_3 Reallocating assets has a cost. Each time we buy/sell assets, we pay a (not so small) transaction fee. This fee depends on the type of asset we want to buy (stocks, bonds, funds), how much of it we want to buy, who we are (an individual, a hedge-fund, a very big client...) and where we want to buy it (banks charge more for stocks in emerging markets for instance).

For our tests, we consider that we pay 0.1% on assets we buy and sell.

Imposing diversification f_4 Mean-variance portfolio theory is supposed to yield diversified portfolios. We've seen this isn't the case. We consider a function which penalizes highly concentrated portfolios. Asset managers sometimes recommend investing so that you risk the same amount of money on each asset. This is a risk-adjusted equally-weighted portfolio similar to the index function, but takes into account the fact that different assets have different risks. For example, if asset A is twice as risky as asset B, we invest twice as much in B as in A.

We use the diversification index H defined in the appendix. For an n asset investment universe, perfect diversification would occur if $\frac{1}{H} = n$. If $\frac{1}{H}$ is smaller, then it mean that some assets are overweighted.

We define f_4 as:

- $f_4 = -C_{concentration}(N_{concentration} - \frac{1}{H})$, if $\frac{1}{H} < N_{concentration}$
- $f_4 = 0$ otherwise
- $C_{concentration}$ is an adjustable parameter.
- H is calculated with equation A.1 of the diversification index.
- $N_{concentration}$ is an adjustable parameter.

The portfolio manager chooses the minimum number $N_{concentration}$ of volatility-adjusted diversified funds that he requires within the portfolio and the cost $C_{concentration}$ that he attributes to the fitness function for each incremental concentration point.

Combining factors

We want the sum of factors to make some kind of investment sense, so that we optimize some tangible function. We choose all our factors so that their dimension is a cost in terms of networth. The trick will be to find weights on each factor.

$f_1(t)$=$NW(t)(1 + E(r_{t+1}))$ Where $NW(t)$ is networth at time t. f_1 is the expected networth at the end of the next period.

$f_2(t)$=$-\lambda NW(t)\sigma(t)$ f_2 is a risk cost. We take off λ standard deviations from our expected return. The higher the risk aversion λ, the lower the quantile we are maximizing.

$f_3(t)$=$-pNW(t)TC(t)$ Where p is a penalty factor and $TC(t)$ is the turnover caused by reallocating from $t-1$ to t. f_3 is a transaction cost. We set p to different levels for testing purposes. We optimize over Ω using a given p and then use our real transaction costs to calculate networth.

p is a quenching factor. Imposing a higher p will limit the amount of trading done. This reflects our confidence in the estimation procedure: if it were perfect, we wouldn't have to restrain trading.

$f_4(t) = -C_{concentration}(N_{concentration} - \frac{1}{H})$ **if** $\frac{1}{H} < N_{concentration}$ and 0 otherwise. f_4 is a concentration cost.

3-factor function

We start by designing a 3-factor function, using the performance, cost and variance indicators:

$$f(t) = NW(t)(1 + E(r_{t+1}) - \lambda\sigma(t) - pTC(t))$$

λ Increasing λ translates into increasing investor risk aversion.

p Increasing p means setting stronger transaction costs. Portfolios will be re-allocated less often.

We scan a variety of (λ, p) values. To avoid data snooping, we do this over three distinct periods. Only if a pair of values (λ, p) emerges as dominant over the three periods can we suppose that there is an optimal search function.

4-factor function

We add the diversification indicator to our function

$$f(t) = NW(t)(1 + E(r_{t+1}) - \lambda\sigma(t) - pTC(t) - C(N - \tfrac{1}{H})) \text{ if } \tfrac{1}{H} < N$$

$$f(t) = NW(t)(1 + E(r_{t+1}) - \lambda\sigma(t) - pTC(t)) \text{ if not}$$

Increasing C means that we force the portfolio towards the variance adjusted equally weighted portfolio.

4.4 Procedure

We define the periods over which we carry out testing:

- Period 1: January 1st 1995 to January 1st 1999
- Period 2: January 1st 1999 to January 1st 2003
- Period 3: January 1st 2003 to January 1st 2007

We test every function over all three periods and compute the Sharpe and Calmar Ratios. We compile the results in table 4.1.

For the 3-factor function, there is no single best score. We plot surfaces of Sharpe Ratio and Calmar Ratio as a function of (λ, p). Figures 4.1, 4.2 and 4.3 show these surfaces.

4.5 Results

The equally weighted portfolio dominates all the other portfolios in every period in terms of Sharpe ratio. In terms of Calmar ratio, higher order moment methods perform better in some periods. There is no best strategy overall.

If we look at the SR and CR surfaces, we see large flat planes of (λ, p) settings. In these subspaces, little or no trading was done beyond the initial allocation. Overly risk-averse investors don't trade. The same goes for applying too high a penalty constraint.

Sharpe Ratio			
Strategy	Period 1	Period 2	Period 3
Equally weighted	1.33	-0.26	0.84
Equally weighted, then free	1.27	-0.30	0.80
Value-weighted portfolio	1.19	-0.30	0.80
Minimum variance	1.12	-0.38	0.59
Minimum semi-variance	1.19	-0.42	0.57
Minimum fourth centered moment	0.87	-0.35	0.76
Minimum eighth centered moment	0.78	-0.32	0.73

Calmar Ratio			
Strategy	Period 1	Period 2	Period 3
Equally weighted	0.75	-0.17	0.79
Equally weighted, then free	0.69	-0.20	0.71
Value-weighted portfolio	0.63	-0.20	0.71
Minimum variance	0.50	-0.21	0.56
Minimum semi-variance	0.57	-0.19	0.51
Minimum fourth centered moment	0.41	-0.19	0.89
Minimum eighth centered moment	0.39	-0.18	0.86

Table 4.1: Sharpe and Calmar Ratio for each strategy over each period

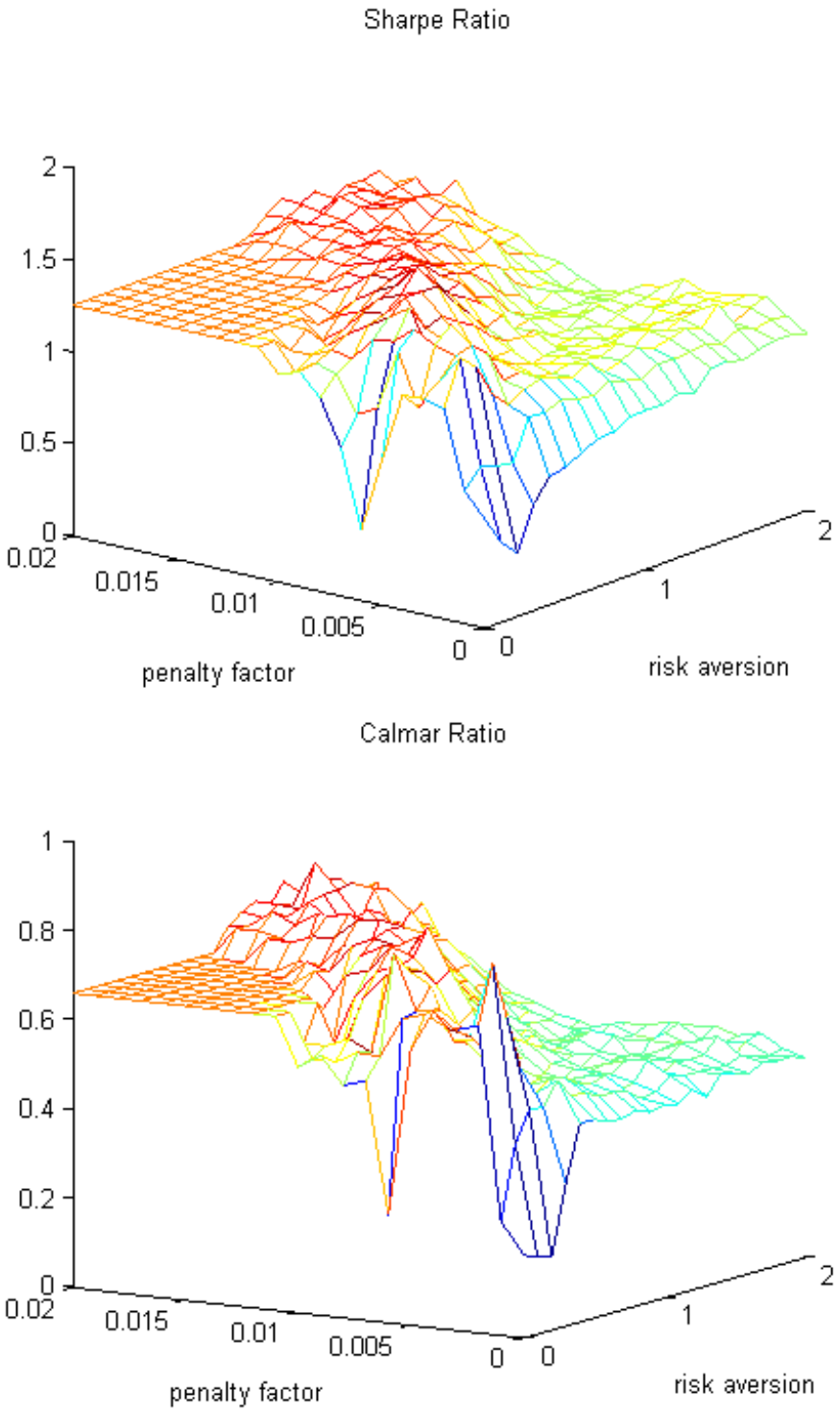

Fig. 4.1: Period 1: Sharpe and Calmar Ratio as a function of (λ, p)

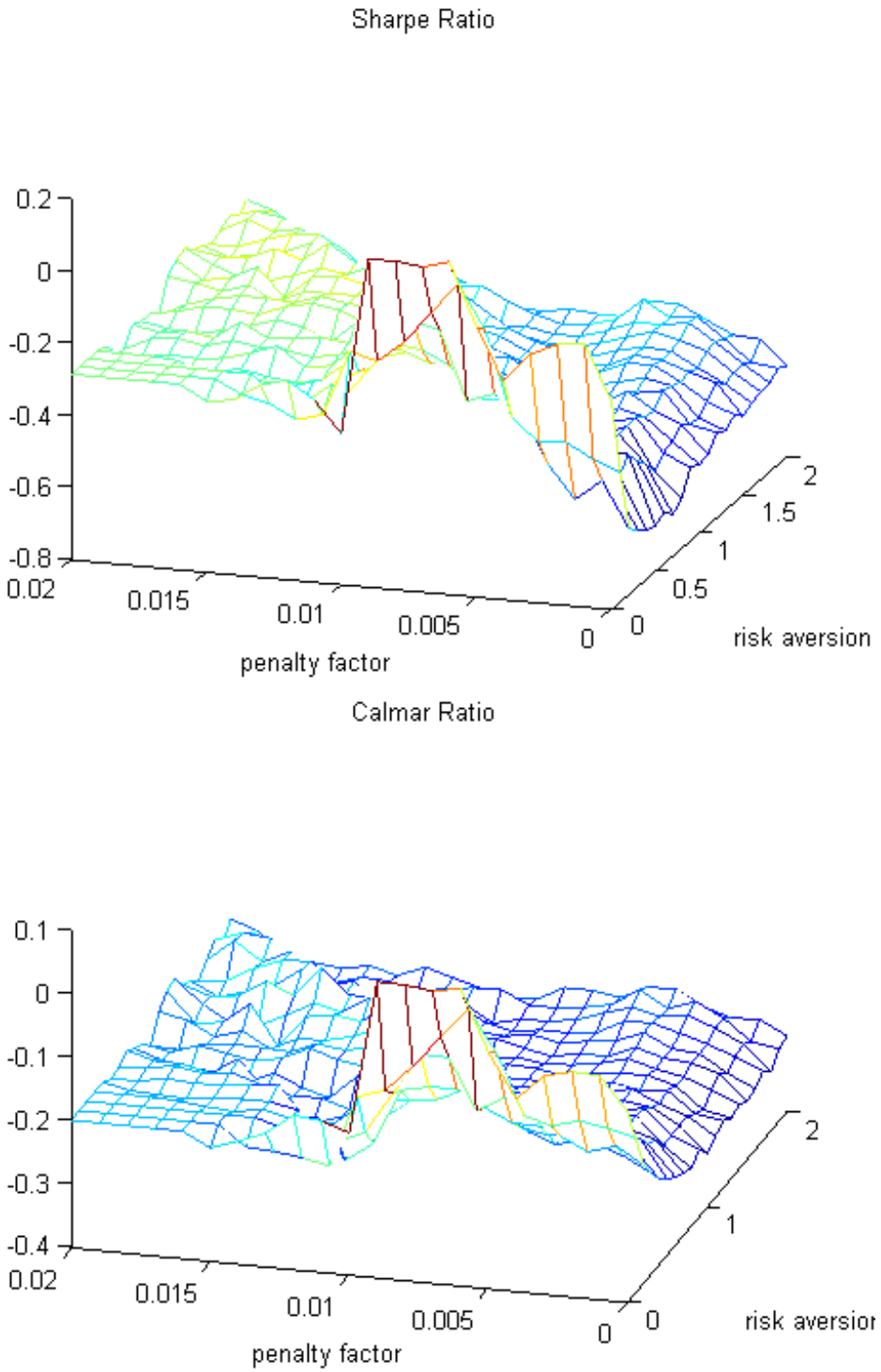

Fig. 4.2: Period 2: Sharpe and Calmar Ratio as a function of (λ, p)

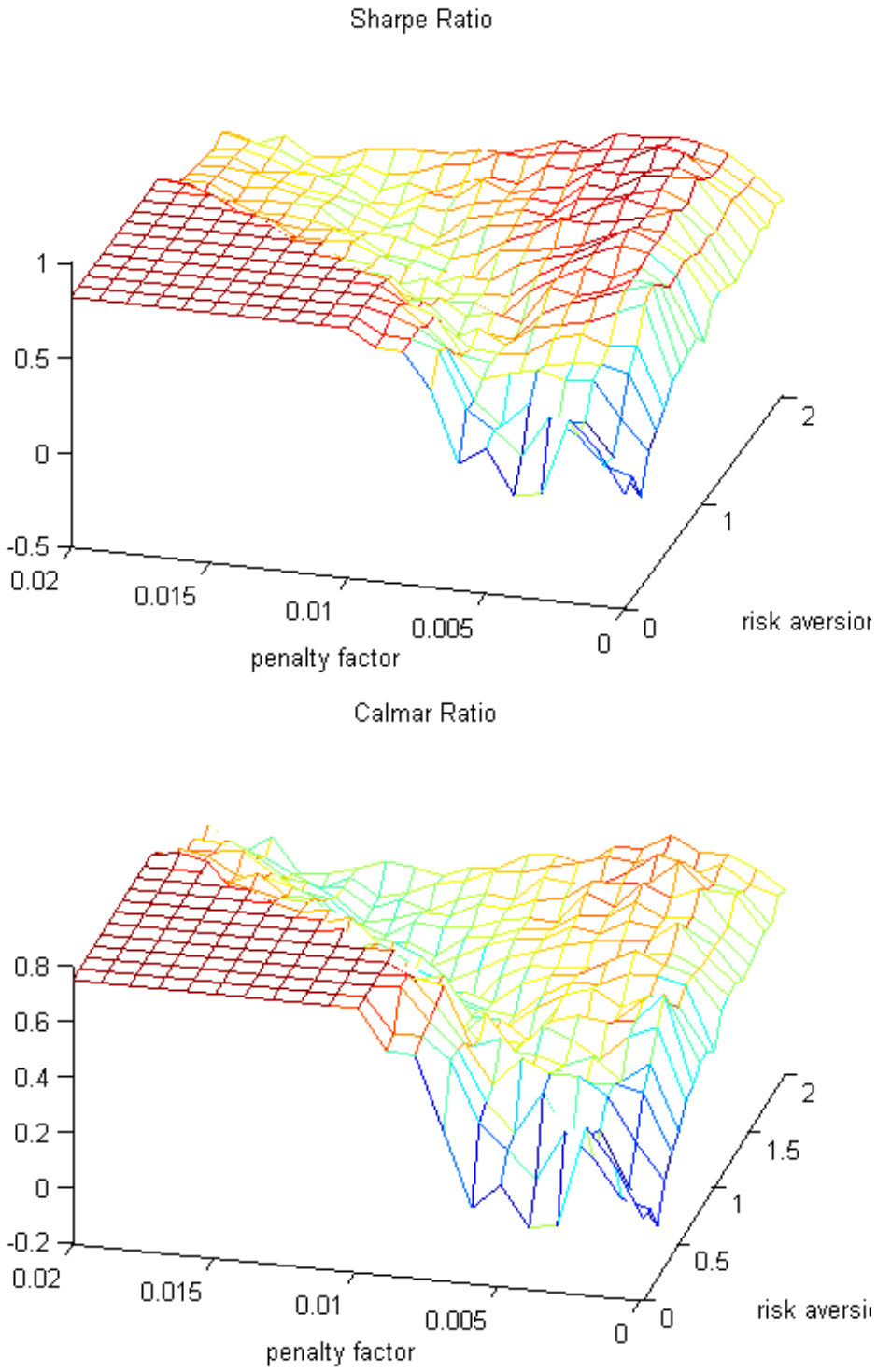

Fig. 4.3: Period 3: Sharpe and Calmar Ratio as a function of (λ, p)

Chapter 5

The illusion of control

5.1 How good ideas sometimes fail

For all its flaws, mean-variance has one great advantage: it is intuitive and easy to use, even with no technical baggage. The framework is easy to explain and it makes investment sense. Investors can choose between two factors they wish to emphasize: more risk (in the form of variance) or more expected return. It's a tradeoff. For this given level of expected risk, you get this level of expected return.

We set out to build upon this framework and develop a more complete function. To our performance and risk factors, we added a cost factor and a new *palette* of risk factors, as well as a concentration factor. For each factor, we provided different settings, hoping to find the best functional form.

- More risk?
- Yes please.
- Would that be variance or drawdown risk, sir?
- A bit of both please.
- And would you like turnover costs with that as well?
- Why of course. I don't mind if I do.

This was to lead us to better portfolio control. It made investment sense and each factor had a purpose. A few tests later, we come to the same conclusion others have reached before us. All of our *strategies* are beaten by the simple equally weighted approach. Over different time periods, no strategy outperforms the market portfolio. We might as well invest in the whole market indifferently. We would get better results and explanations are much easier.

5.1.1 Simplicity ueber alles

How would we go about presenting our investment strategies to a potential client?

In the equally weighted case, we'd simply say: "spread out your risk" and buy into the whole market. Simple.

Our functional form is more complicated. By changing the associated weightings $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$, we change f:

$$f = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4$$

With our approach, it is very hard to explain how each factor affects the global outcome. Individually, they all make sense. Taking one performance factor and one risk factor, we can define a tradeoff between them. But how do you explain anything beyond that?

What does the $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ weighting represent? While a 1% change makes sense for our optimizer, can we really say we'd rather have a $(11, 9, 8, 72)$ rather than a $(10, 10, 10, 70)$ weighting in f?

The SR and CR surfaces show that such slight changes make a difference. How should we explain such large shifts in performance other than by overfitting and random luck?

More importantly, how do you convince an investor that your more complicated and under-performing strategy is actually the better choice?

5.2 Starting from scratch: a simple test for an easy problem

Our risk factors and performance measures are quite good for describing portfolios ex-post. Once we have realized returns, or when we are working in-sample, they find very good portfolios. They just don't work out of sample.

To illustrate our problem, we run a simple test, which we could/should have run at the beginning of our study. Up until now, we have made no simplifying assumptions about our return series. We have worked only with real data and said nothing about the underlying processes. If we knew the underlying process governing the returns, then we should be able to build portfolios with our eyes closed, right?

We suppose that returns follow a multivariate normal distribution. We generate synthetic data with known mean vector of returns r and covariance matrix V. We run a simple minimum variance portfolio optimizer and look at the weights we get over time with different settings.

- We consider a fixed-length rolling window spanning six months.
- We consider a fixed-length rolling window spanning five years.
- We consider a window of increasing length. Starting with a six-month window, at each time-step, we add a data point, therebye increasing our sample size and the precision of our estimator.

Figure 5.1 shows the optimal weights over period 1 for the minimum variance portfolio in the first two cases, with fixed length rolling windows.

Results In the first case, the optimal weights don't converge. They seem to randomly change over time. In the second case, the weights seem to fluctuate around some value. We could say that a six month sample is too small an estimation. Five years is a lot larger and yet we see that asset weights still fluctuate very strongly.

The third setting, shown in figure 5.2, gives us an idea of how the procedure behaves over time. At the end of the test, our estimation is done over 1000 data points, roughly 20 years of synthetic data. The weights still haven't converged to a unique portfolio.

What if ? If we used an f-like function, would we get better results? We based f on the assumption that our input parameters were correct. Keeping the same estimators, we expect our portfolio to fluctuate with the estimation. Imposing a strong p constraint, we could probably quench these fluctuations but would have no idea whether we chose the true best portfolio or not. We could just as well fix ourselves on a bad portfolio and not re-allocate because of costs.

What now ? If the procedure doesn't work with an easy function and synthetic data which we control, why should it work for a more complicated function and real data, the underlying process of which we don't know?

Our estimators are simply too naive. Supposing our data was governed by a multivariate normal distribution, how large a time-window would we need to get good estimates? This depends what we mean by good. It's easy to get confidence intervals for our estimators of returns and covariance matrix. What isn't easy is to quantify the sensitivity of the optimization process to the uncertainty in the estimators.

More precisely, if we have a confidence interval for our estimators, can we get a confidence interval for the resulting weights?

5.3 Room for improvement

We look at possible solutions to our problems, suggested by the litterature and our experience.

5.3.1 Estimating returns

We have used historical returns as estimators for future returns. As we've seen, this is a bad estimator. Papers in the litterature have shown as much.

Jorion (1986) [13] proposes other classes of estimators which do a better job.

Other methods include shrinking the vector of expected returns towards an empirical value, following asset managers' beliefs. Some propose ignoring returns all together.

5.3.2 Estimating the covariance matrix

We have only used naive estimators of V. We take the historical covariance matrix, which is a bad estimator and very unstable. There is a whole field which deals with developing better, more stable estimators of V. Good references include Jorion (1986) and Ledoit (2003) [14].

The idea here is to shrink V towards some specific, stable matrix. In most cases, the sample V can be projected into a smaller subspace of admissible (positive definite) matrices. This is a field in itself. More applied methods include reducing the number of correlation parameters we have to estimate, by considering subspaces of assets (industry groups for instance) instead of individual stocks.

In a perfect world, we would have developed suitable estimators, then run our procedure.

5.3.3 Resampling methods

Richard Michaud is the pioneer and leader in the domain of *portfolio resampling* [15]. The idea behind his method is that realized returns are very noisy. Since the optimization procedure is very unstable with respect to small variations in input parameters, we should optimize our portfolio over sets of similar return series. On average, noise should be evened out.

Starting with our original return series, we generate new series adding small amounts of noise to the original series. We then run the procedure over all series. This gives us a set of different optimal portfolios for a same level of expected returns. We take the average over all optimal portfolios, which should be more stable with respect to errors in the input data.

5.3.4 Problems in the GA

In our selection process, we put convergence before diversification. We use the best member to generate new members, which means we are generating preferentially in a neighbourhood of the currrent best member. In doing so, we might be missing good solutions elsewhere.

At the same time, this is the only way to get convergence of the population. If we use a random member to generate new members, we will have to define some other criteria for stopping the GA. For instance, we can count the time since the last fitness gain. If this time is too long, then we stop the run.

We would have liked a procedure where each individual is switched only if it's better than it's predecessor and if it isn't too close to any other population member. For this we need a measure of distance and some sort of diversity preference.

5.4 Le mot de la fin

Despite our efforts, we haven't come up with a ground-breaking method for building portfolios. We have, however, grasped the importance of uncertainty in the whole process. Whatever method is applied, we stress the importance of confidence intervals and sensitivity tests. Portfolio optimization methods should come with clearly defined limits. We've seen that even in straightforward cases, using synthetic data and simple functions, results still aren't what we expect.

Part of this underperformance we've attributed to noise. It's hard to say how much information and noise we find in our data series. Using raw data doesn't work. To palliate this, a whole field has developed, dedicated to sifting through noise and finding *true* information. Doing this requires developing models and making simplifying assumptions. How much information do we truly find and how much is just enhanced noise or fabrication? It is always a trade-off between model simplicity and real-world constraints. By shrinking covariance matrices, we move away from real data. At the same time, we get cleaner results, more stable input parameters and better theory.

For practitioners, the main issue will be back-testing: a method produces superior returns or it doesn't. If it works, then little else matters.

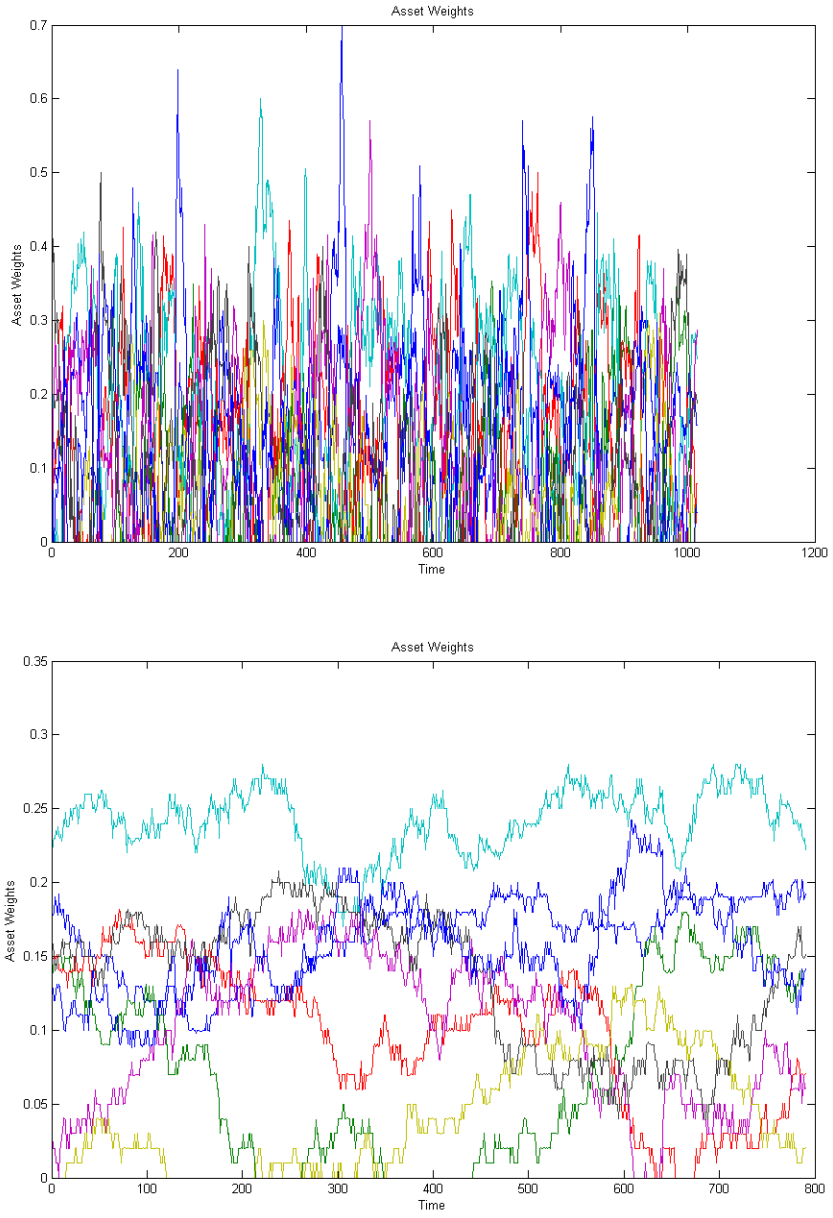

Fig. 5.1: Asset weights over time, as given by the minimum variance solver. Top: six-month sample window. Bottom: five-year sample window

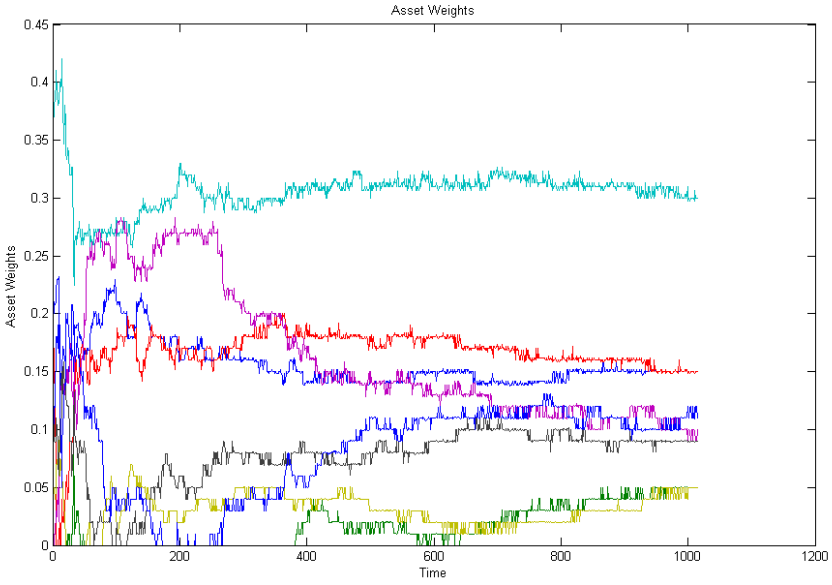

Fig. 5.2: Asset weights over time, as given by the minimum variance solver. The sample window increases in size at each time-step, reaching twenty years at the end of the test.

Appendix A

Diversification Index

We use a diversification index as a factor in our objective function. Considering a portfolio $x = (\omega_1, \omega_2, \ldots, \omega_n)$ and the following notations:

- Weight of asset $i = \omega_i$
- Variance of returns of asset $i = \sigma_i^2$
- $\forall i, \omega_i \geq 0$
- $\Sigma\omega_i = 1$

We define the "variance adjusted weight" of asset i:

$$p_i = \frac{\omega_i^2 \sigma_i^2}{\Sigma \omega_j^2 \sigma_i^2}$$

So the sum of variance adjusted weights over all assets is one:

$$\Sigma_{i=1..n} p_i = 1$$

We define the diversification index H in the following manner:

$$H = \Sigma_{i=1..n} p_i^2 \tag{A.1}$$

If all funds have the same variance σ_i^2 and a similar allocation $\frac{1}{n}$ then $H = \frac{1}{n}$

If our portfolio is made up of one asset, then $H = 1$. In the same way, if one asset is responsible for all the variations in portfolio returns (a portfolio made of one riskless treaury note and one stock), then $H = 1$.

H captures the notion of "volatility adjusted diversification", and $\frac{1}{H}$ can be interpreted as the number of "diversified" funds in the portfolio.

Bibliography

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

[2] Harry M. Markowitz. Foundations of portfolio theory. *The Journal of Finance*, 46(2):469–477, 1991.

[3] Michael J. Best and Robert R. Grauer. On the sensitivity of mean-variance-efficient portfolios to changes in asset means: Some analytical and computational results.

[4] Richard C. Green and Burton Hollifield. When will mean-variance efficient portfolios be well diversified? *The Journal of Finance*, 47(5):1785–1809, 1992.

[5] Bernd Scherer. Portfolio resampling: Review and critique. *Financial Analysts Journal*, 58(6):98–109, November 2002.

[6] E. J. Elton and M. J. Gruber. *Modern portfolio theory and investment analysis*. J. Wiley and Sons, 1995.

[7] Robert C. Merton. An analytic derivation of the efficient portfolio frontier. *The Journal of Financial and Quantitative Analysis*, 7(4):1851–1872, 1972.

[8] A. Mcneil, R. Frey, and P. Embrechts. *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton University Press, 2005.

[9] Hiroshi Konno and Hiroaki Yamazaki. Mean-absolute deviation portfolio optimization model and its applications to tokyo stock market. *Management Science*, 37(5):519–531, 1991.

[10] D. Sornette and Y. Malevergne. Higher moment portfolio theory: Capitalizing on behavioral anomalies of stock markets. *Journal of Portfolio Management*, 2005.

[11] Rainer Storn and Kenneth Price. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4):341–359, December 1997.

[12] V. Demiguel, L. Garlappi, and R. Uppal. *How Inefficient is the 1/N Asset-Allocation Strategy?* 2005.

[13] P. Jorion. Bayes-stein estimation for portfolio analysis. *The Journal of Financial and Quantitative Analysis*, 21(3):279–292, September 1986.

[14] O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, February 2004.

[15] R. O. Michaud. The markowitz optimization enigma: Is optimized optimal. *Financial Analysts Journal*, 1989.