Vikram Rawal **MIE377 Project 2** April 11, 2024

Introduction

The financial optimization project is an opportunity to build on insights from Project 1 by designing, and implementing, and testing various algorithmic strategies designed to maximize the geometric mean of the Sharpe ratio of the portfolio while minimizing its turnover. This report will delineate the mathematical motivation behind the trading strategies, the implementation and testing of the strategies in MATLAB, and the reasoning through which the optimal model was selected. Furthermore, the report expounds on the strategies tested with regards to next steps and highlights a few of the algorithms' limitations.

The crux of the problem is as follows: we are given three datasets, with each dataset containing a set of asset returns over some time period and a corresponding set of factor returns (for eight factors) over said time period. The objective is to utilize the portfolio management techniques explored in the course to build a portfolio management algorithm which successfully maximizes the Sharpe ratio and minimizes the turnover of the portfolio. Recall that $SR = \frac{\mu - r_f}{\sigma}$ and that Average Turnover $= \frac{1}{T-1}\sum_{t=2}^{T} |x - x_{t-1}|$.

We note that the models used either calculated x directly or were some combination of the following:

1. A model that estimates one or both of the parameters μ and Q using the asset and factor returns as inputs (we call these *type one* models).
2. A model that determines an optimal portfolio x using the parameters μ, Q, and x_0 as inputs (we call these *type two* models).

Therefore, we will first introduce the models used and then outline the reasoning by which they were combined to yield the best-performing model.

Methodology and Model Development

The model which we began with (given in the starter code) was a combination of using OLS (Ordinary Least Squares) to estimate the mean μ and the covariance matrix Q of the given assets, and then solving the standard mean variance optimization problem using the estimates to determine the optimal portfolio x at the end of every six-month period. The Ordinary Least Squares method determines μ and Q as follows:

$$B = \begin{bmatrix}\alpha^T & V\end{bmatrix}^T = (X^T X)^{-1} X^T R$$
$$\mu = \alpha + V^T f$$

$$Q = V^T F V + D$$

Where $X = [1 \;\; F]$ is the matrix of factor returns, R is the matrix of asset returns, α is the intercept vector, V is the matrix of factor loadings, F is the factor covariance matrix, and D is the diagonal matrix of residual variances. The resulting μ and Q are then used as inputs to the following quadratic program (note that short-selling is allowed, as is reflected by the absence of an $x \geq 0$ constraint):

$$\begin{aligned} &min \; x^T Q x \\ &s.t. \; \mu^T x \geq R \\ &\quad 1^T x = 1 \end{aligned}$$

And solving this optimization problem yields the updated portfolio x.

Robust Mean-Variance Optimization Model

As noted in lecture and as seen by running the baseline model, the standard MVO model is highly sensitive to minute changes in the input parameters μ and Q. Over a long time period, this renders the model susceptible to high turnover and non-optimal portfolio. Therefore, we sought to improve the model by using an elliptical uncertainty set as follows:

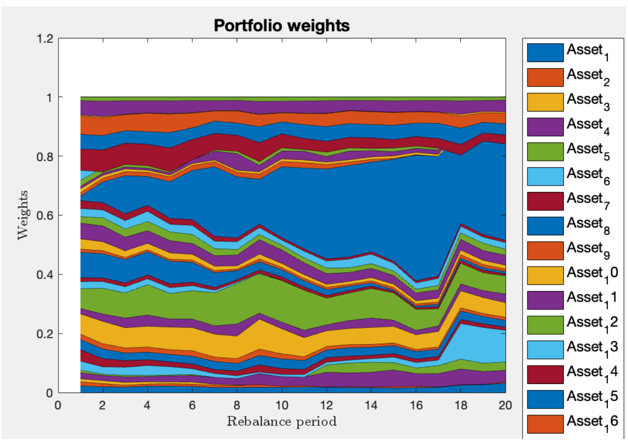
$$\begin{gathered} min \; x^T Q x - (\mu^T x - \varepsilon_2 ||\Theta^{1/2} x|_2) + c||x - x_0||_1 \\ s.t. \; \mu^T x \geq R \\ 1^T x = 1 \\ lb \leq x \leq ub \end{gathered}$$

where $\Theta = \frac{diag(Q)}{T}$ and $\varepsilon_2 = \sqrt{\chi(2, n)}$, where $\chi(2, n)$ is the chi-squared distribution for n dimensions (here, n assets), and where $R = mean(\lambda\mu)$ (with λ being some constant determined through testing). Creating an ellipsoidal uncertainty set, as opposed to the box uncertainty set explored earlier in the course, allows the model to avoid the suboptimal "corners" which can lead to imbalances in the asset weights (i.e.: too large a weighting of some assets and too small a weighting of some assets). Furthermore, we constrained our mean portfolio return to be greater than the mean return of the assets, and imposed lower and upper bound constraints that were determined through testing. Finally, the $c||x - x_0||_1$ is a turnover penalty term that was initially defined as a constraint, but was put into the objective function to offer the algorithm more flexibility.

Using a combination of OLS and Robust MVO with an elliptical uncertainty set, we iterated to find the optimal parameter models of $\lambda = 1.19$, $c = 1$, $lb = -1$, and $ub = 1$.This model led to the following Sharpe ratios and turnovers, and the portfolio weights and performance over time for Dataset 3 are shown in Figures 1.2 and 1.3:

	Dataset 1	Dataset 2	Dataset 3	Average
Sharpe ratio	0.20064	0.10796	0.21206	**0.17355**
Turnover	0.027472	0.038154	0.035707	**0.03378**

Figure 1.1: Sharpe ratio and Turnover performance for Robust Ellipsoidal MVO model



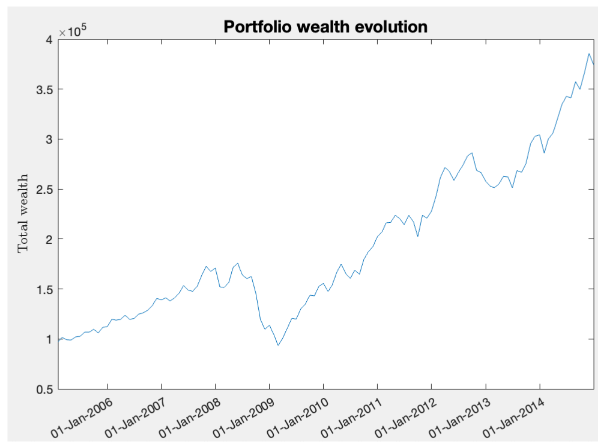


Fig. 1.2: Robust MVO Portfolio Weights **Fig. 1.3:** Robust MVO Portfolio Wealth Evolution

---

Convex Reformulation of Risk-Parity Model

Looking to improve on our previous model, we next attempted to use a risk-parity model with inputs of μ and Q (again calculated using OLS) in place of the Robust MVO model to calculate the optimal portfolio x. The goal of the standard risk-parity model is to *equalize* the *risk contribution per asset*, with the idea that if the risk is spread out evenly over the entire asset universe, the portfolio will be relatively low-risk while maintaining an expected return that is near (if not above) the mean asset return.

Given the portfolio volatility of $\sigma_p = \sqrt{x^T Q x}$, we apply Euler's theorem for positive homogeneous functions of degree $k = 1$ to arrive at

$$\sigma_p = \sum_{i=1}^{n} \frac{(Qx)_i}{\sqrt{x^T Q x}}$$

where $(Qx)_i$ is the i^{th} element of the vector Qx. Similarly, applying Euler's theorem with $k = 2$ to the marginal variance contribution of asset i to the portfolio variance, we find that

$$\frac{d\sigma_p^2}{dx_i} = 2(Qx)_i.$$

But we note that we can now expand to write $\sigma_p^2 = x^T Qx$ as the sum of $\frac{1}{2}\sum_{i=1}^{n} x_i \cdot 2(Qx)_i = \sum_{i=1}^{n} R_i$, from which it follows that $R_i = x_i(Qx)_i$ is the contribution of asset i to the portfolio variance. By definition, risk-parity necessitates that the R_i values are all equal, and so we use least squares to minimize the squared differences between them:

$$min \sum_{i=1}^{n}\sum_{j=1}^{n}(x_i(Qx)_i - x_j(Qx)_j)^2$$

And we note that in order to hope to obtain a single optimal portfolio, we *must* eliminate short-selling—otherwise the x_i and x_j could vary infinitely, even under the $1^T x = 1$ budget constraint. Finally, to implement the model, we use a rather unmotivated convex reformulation:

$$f(y) = \frac{1}{2}\sum_{i=1}^{n} y^T Qy - c\sum_{i=1}^{n} ln(y_i)$$

Noting that when the gradient of f is 0, we will have $Qy - \frac{c}{y} = 0$, which implies that for every asset i, $y_i(Qy)_i = c$, thus satisfying the risk-parity constraint because the R_i values are constant. Also, we note that since the budget constraint is dropped to accommodate the convex reformulation, we must reaccount for it after solving for y by normalizing the vector such that its elements sum to 1. The convex risk-parity optimization problem implemented with the turnover penalty included in the objective function is therefore:

$$min \frac{1}{2}\sum_{i=1}^{n} y^T Qy - c_1\sum_{i=1}^{n} ln(y_i) + c_2 || \frac{y_i^*}{\sum_{i=1}^{n} y_i^*} - x_0 ||$$
$$s.t.\ y \geq 0$$

From which we retrieve the optimal asset weights by computing $x_i^* = \frac{y_i^*}{\sum_{i=1}^{n} y_i^*}$. Arbitrarily choosing $c_1 = 2$, we noted that changing c_1 had an incredibly miniscule effect on the Sharpe ratio (because since $x*$ does not depend on c_1, neither does σ, and the changes in $\mu - r_f$ are small). The portfolio performance was as follows:

	Dataset 1	Dataset 2	Dataset 3	Average
Sharpe ratio	0.16931	0.12533	0.17699	**0.15721**
Turnover	0.13292	0.11794	0.17927	**0.14338**

Figure 2.1: Sharpe ratio and Turnover performance for Risk-Parity model

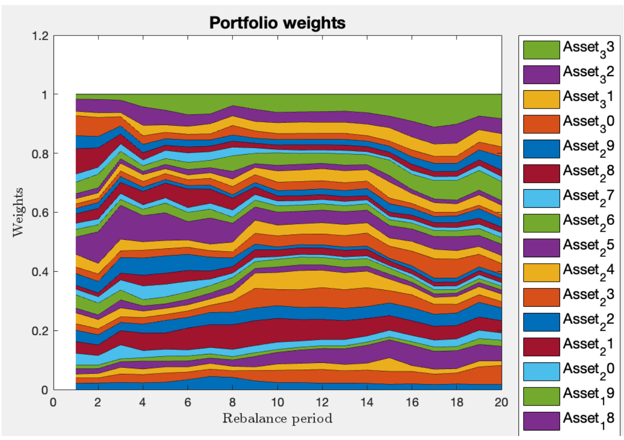


Fig. 2.2: Risk-Parity Portfolio Weights

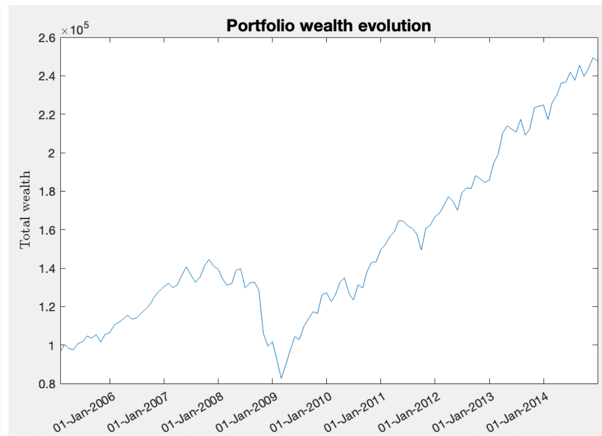


Fig. 2.3: Risk-Parity Portfolio Wealth Evolution

Black-Litterman Model

We next tested a modified version of the Black-Litterman Model, whose outputs μ and Q were used on the Standard MVO model and the Robust MVO model developed earlier to output an optimal portfolio x. The goal of the Black-Litterman model is to use the μ corresponding to the efficient market portfolio x_{mkt}, which encodes information about market sentiment and other usually non-quantifiable factors onto μ.

We first use the following risk-adjusted MVO program to solve for μ using the First-Order Necessary Condition (FONC) applied to $max\ x_{mkt}^T \pi - \frac{\lambda}{2} x_{mkt}^T Q x_{mkt}$:

$$\nabla f(x) = 0 \Rightarrow \mu = \lambda Q x_{mkt}$$

Then, given the risk aversion coefficient of $\lambda = \frac{E[r_{mkt}] - r_f}{\sigma_{mkt}^2}$, we rearrange to find that $E[r_{mkt}] = \mu_{raw}^T x_{mkt} - r_f$, where μ_{raw} is the mean over each asset of the geometric means of the raw data over time. Note, however, that since we were not given x_{mkt} or the market capitalizations needed to calculate it, we used the (slightly crude) approximation of

$x_{mkt} = \frac{\mu_{raw}}{||\mu_{raw}||_1}$. After calculating Q as the covariance matrix of the excess returns, we rearrange to find that $\lambda = \frac{\mu_{raw}^T x_{mkt} - r_f}{x_{mkt}^T Q x_{mkt}}$ (but note that since we are given excess returns, we do not have to subtract r_f when implementing the algorithm). Finally, $\pi = \lambda Q x_{mkt}$, and since we have no investor views (given that we are working entirely with predetermined data), we simply use $\mu = \pi$ and apply mean-variance optimization.

We first examine the performance of the Black-Litterman model with Standard MVO:

	Dataset 1	Dataset 2	Dataset 3	Average
Sharpe ratio	0.19728	0.12282	0.21025	**0.17678**
Turnover	0.20437	0.15715	0.2337	**0.19841**

Figure 3.1: Sharpe ratio and Turnover performance for BL with Standard MVO

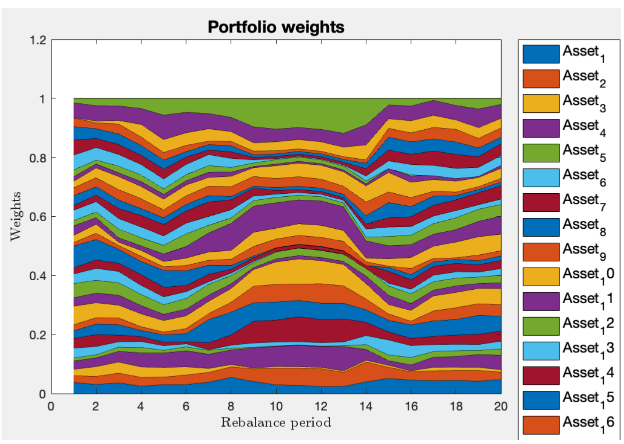


Fig. 3.2: BL w/ SMVO Portfolio Weights

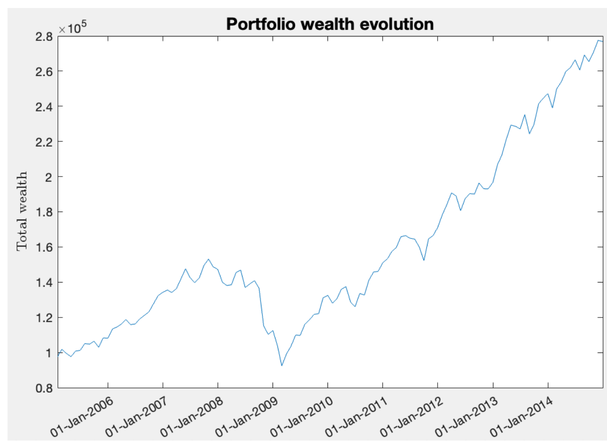


Fig. 3.3: BL w/ SMVO Portfolio Wealth Evolution

And the performance of the Black-Litterman model with our previously implemented Robust MVO is as follows:

	Dataset 1	Dataset 2	Dataset 3	Average
Sharpe ratio	0.19091	0.079923	0.20096	**0.15726**
Turnover	0.043962	0.048296	0.064627	**0.05230**

Figure 4.1: Sharpe ratio and Turnover performance for BL with Robust MVO

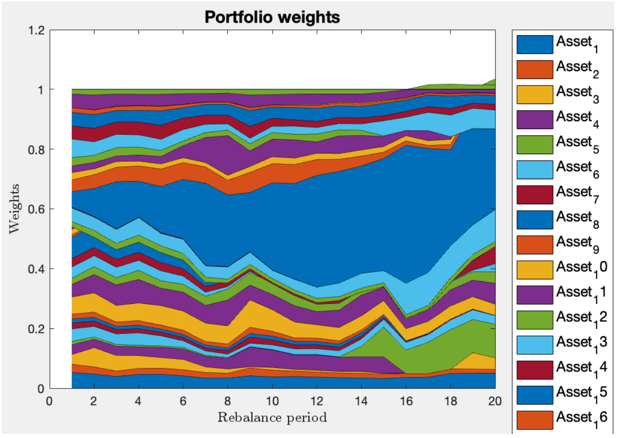


Fig. 3.2: BL w/ RMVO Portfolio Weights

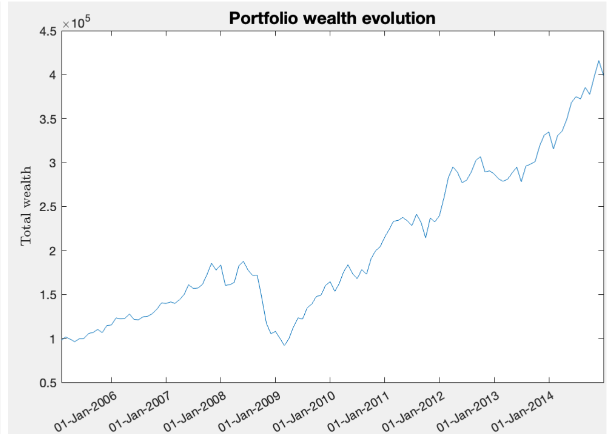


Fig. 3.3: BL w/ RMVO Portfolio Wealth Evolution

Ex Ante Sharpe Ratio Maximization Model

While this model did not turn out to be one of the top models when implemented in MATLAB (regardless of the method used to get μ and Q), we employed one of the strategies which were thought up for this model for a future model, and so the derivation is presented below for posterity.

We seek to find the portfolio x which maximizes the *ex ante* Sharpe Ratio, subject to the budget constraint, the fulfillment of a minimum return, and upper and lower bounds:

$$\begin{aligned} max \ \frac{\mu^T x - r_f}{\sqrt{x^T Q x}} \\ s.t. \ 1^T x &= 1 \\ Ax &\leq b \\ lb \leq x &\leq ub \end{aligned}$$

The problem is that the objective function is very obviously non-linear. To this end, we define $\kappa = \frac{1}{(\mu - r_f)^T x}$ for a given feasible portfolio x and let $y = \kappa x$. Rearranging, it follows that we must now maximize $\frac{1}{\sqrt{y^T Q y}}$, which is mathematically equivalent to minimizing $y^T Q y$. Therefore, the problem, now in quadratic form, is

$$\begin{aligned} min \ y^T Q y \\ s.t. \ (\mu - r_f)^T x = 1 \\ 1^T y = \kappa \end{aligned}$$

$$Ay \leq b \cdot \kappa$$
$$\kappa \geq 0$$

After which, having again ignored the budget constraint in our reformulation, we recover our optimal portfolio weights by calculating $x_i^* = \frac{y_i^*}{\sum_{i=1}^{n} y_i^*}$.

A strategy that was found to be useful in improving the performance of this model was putting the return constraint inside the objective function, as follows:

$$min\ y^T Q y - (b \cdot \kappa - Ay)_+$$
$$s.t.\ (\mu - r_f)^T x = 1$$
$$1^T y = \kappa$$
$$\kappa \geq 0$$

This strategy relaxed the constraint on the return and allowed the model to sacrifice minor improvements to the Sharpe ratio in favour of drastic turnover decreases.

CVaR Optimization Model

The CVaR optimization model aims to minimize the downside risk $CVAR_\alpha(x)$ as calculated by the integral of $Var_\alpha(x)$, a figure representing the minimum loss we may suffer (with probability $1 - \alpha$). Mathematically, we have that CVaR is the *expected value* of VaR, which fails to account for the magnitude by which we may exceed a given minimum loss:

$$CVaR_\alpha(x) = \frac{1}{1-\alpha} \int_{f(x,r) \geq VaR_\alpha(x)} f(x,r)p(r)dr$$

Where $f(x, r)$ is the loss of x for a realization r of the random returns, and $p(r)$ is the probability distribution function of the random returns. Since this function is defined in terms of $VaR_\alpha(x)$, we use the variable γ as its placeholder and treat it as a constant to arrive at

$$F_\alpha(x, \gamma) = \gamma + \frac{1}{1-\alpha} \int (f(x,r) - \gamma)_+ (p(r))dr$$

Since the integral equals 0 when $f(x, r) < \gamma = VaR_{\alpha}(x)$. Since this new function increases more and more rapidly for increasing values of x and γ, it is strictly convex, and so our optimization problem becomes

$$\begin{aligned} & min\ F_{\alpha}(x, \gamma) \\ & s.t.\ \ x \in \chi \end{aligned}$$

Where χ is used to impose any remaining auxiliary constraints. Finally, we make a simplification using an approximate, discrete representation of $p(r)$ and a corresponding discrete loss function $f(x) =- r_s^T x$, so that F becomes a sum:

$$F_{\alpha}(x, \gamma) \simeq \gamma + \frac{1}{(1-\alpha)S} \sum_{s=1}^{S} (- r_s^T x - \gamma)$$

And using a dummy variable z_s for each $s \in \{1 \dots S\}$, we finally arrive at the following linear optimization problem:

$$\begin{aligned} min\ \gamma + \frac{1}{(1-\alpha)S} \sum_{s=1}^{S} z_s + c||x - x_0||_1 & \\ s.t.\ \ z_s \geq 0 \quad \forall\, s \in \{1 \dots S\} & \\ z_s \geq \ - r_s^T x - \gamma \quad \forall\, s \in \{1 \dots S\} & \\ x \in \chi & \end{aligned}$$

Where we have used our insights from the previous models to add the turnover constraint into the objective function of the problem. The resulting portfolio performance is as follows:

	Dataset 1	Dataset 2	Dataset 3	Average
Sharpe ratio	0.17032	0.10629	0.15312	**0.14324**
Turnover	0.049204	0.18927	0.079461	**0.10598**

Figure 5.1: Sharpe ratio and Turnover performance for CVaR with Turnover Constraint

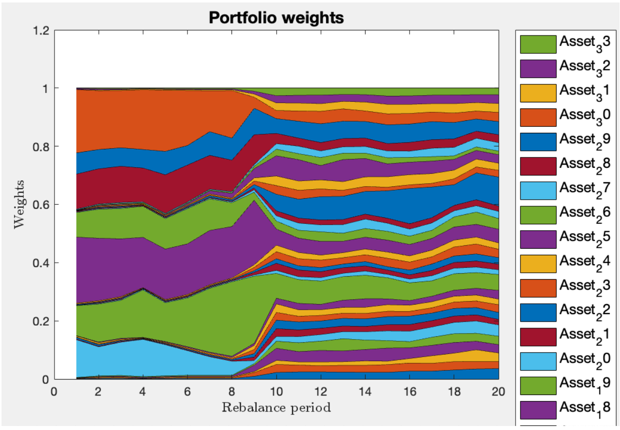


Fig. 5.2: CVaR Portfolio Weights

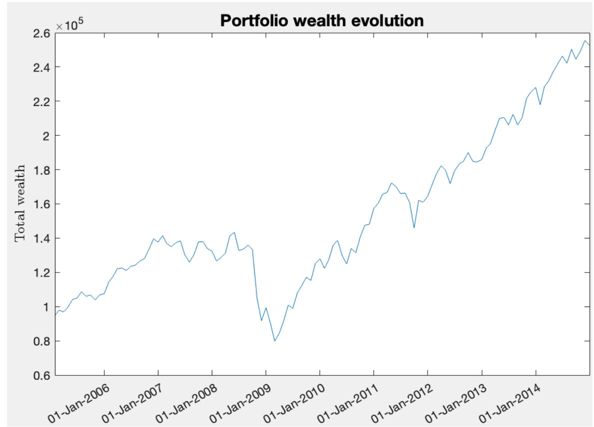


Fig. 5.3: CVaR Portfolio Wealth Evolution

Principal Component Analysis Model

The final model we tested was a modified version of the Principal Component Analysis (PCA) model, which was utilized in place of the standard OLS algorithm to provide our type two algorithms with more accurate estimates for μ and Q.

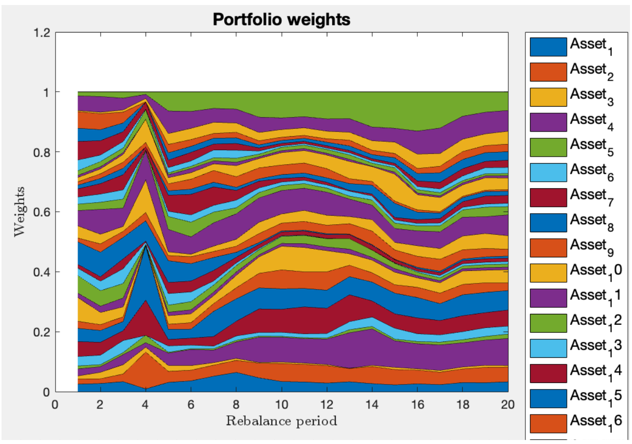
We recall that as per our factor model, we can model the mean return as the sum of an intercept term α and a linear combination of the factor returns f_t. Given our data matrix R of asset returns, we calculate the centered data matrix $R' = R - (1)(\frac{1}{T}R^T 1)^T$ and use R' to estimate the biased covariance matrix as $Q = \frac{1}{T}(R')^T(R')$, which we can decompose as $Q = \Gamma\Lambda\Gamma^T$.

Noting that the eigenvalues of this matrix are the variances of the assets, we seek a model (using OLS) that will explain as much of the variance as possible while reducing the dimensionality of the problem. This is done by removing a certain number of columns from the $P = R\Gamma$ matrix with indices corresponding to the indices of Λ with the smallest entries along the diagonal (i.e.: the smallest eigenvalues).

While a fixed p value (i.e.: the number of factors to be kept in the model) can be used, we employed a technique with a threshold of 0.001 determined through trial and error, where columns with a corresponding eigenvalue of $\lambda > 0.001$ would be kept in the updated data matrix. To keep a minimum number of factors in the model, we set $p = max\{3, n_\lambda\}$, where n_λ is the number of columns satisfying the aforementioned constraint. After obtaining the updated data matrix, OLS was performed as usual to calculate estimates for μ and Q, which were fed into the type two algorithms, as discussed previously. The algorithm's performance with the Standard MVO, Robust MVO, and Risk-Parity models (all of type two) is shown below:

	Dataset 1	Dataset 2	Dataset 3	Average
Sharpe ratio	0.18449	0.11603	0.19551	**0.16534**
Turnover	0.24476	0.15677	0.27973	**0.22709**

Figure 6.1: Sharpe ratio and Turnover performance for PCA/SMVO Model



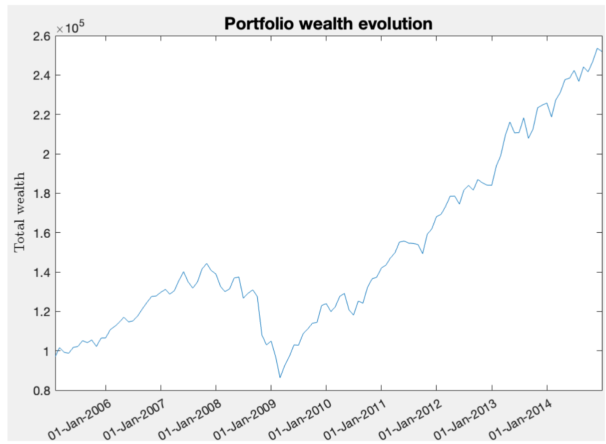
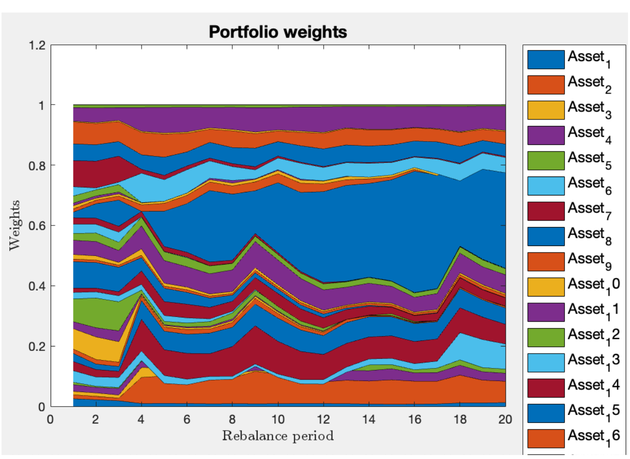


Fig. 6.2: PCA/SMVO Portfolio Weights **Fig. 6.3:** PCA/SMVO Portfolio Wealth Evolution

	Dataset 1	Dataset 2	Dataset 3	Average
Sharpe ratio	0.18806	0.11324	0.23885	**0.18005**
Turnover	0.089789	0.03154	0.092238	**0.071189**

Figure 7.1: Sharpe ratio and Turnover performance for PCA/RMVO Model



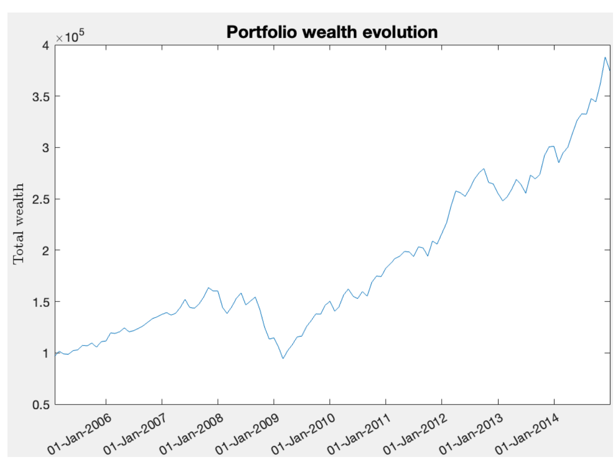


Fig. 7.2: PCA/RMVO Portfolio Weights **Fig. 7.3:** PCA/RMVO Portfolio Wealth Evolution

	Dataset 1	Dataset 2	Dataset 3	Average
Sharpe ratio	0.1683	0.10538	0.16979	**0.14782**
Turnover	0.21364	0.12633	0.25788	**0.19928**

Figure 8.1: Sharpe ratio and Turnover performance for PCA/RP Model

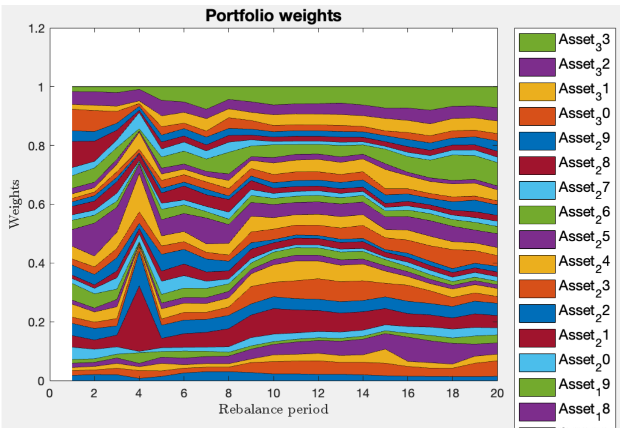


Fig. 8.2: PCA/RP Portfolio Weights

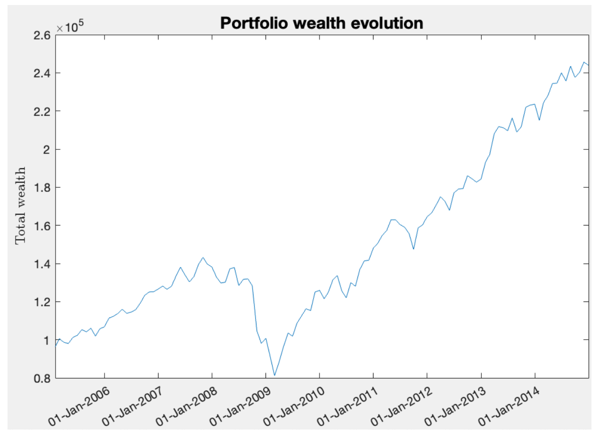


Fig. 8.3: PCA/RP Portfolio Wealth Evolution

Optimal Model Selection

Through the raw Sharpe ratio and turnover values, it's evident that our best model is either the Robust Ellipsoid MVO model with OLS ($RS = 0.17355$, $T = 0.03378$) or the Robust Ellipsoid MVO model with PCA ($RS = 0.18005$, $T = 0.071189$). To decide between these two models, we make use of the formula used to assign marks (naturally), which calculates a score as follows:

$$Score = 0.8 \cdot SR + 0.2 \cdot (1 - T)$$

We obtain:

$$Score_{RMVO/OLS} = 0.8 \cdot 0.17355 + 0.2 \cdot (1 - 0.03378) = 0.332084$$
$$Score_{RMVO/PCA} = 0.8 \cdot 0.18005 + 0.2 \cdot (1 - 0.071189) = 0.3298002$$

Therefore, our best model is the Robust Ellipsoid MVO model with a turnover constraint, and using Ordinary Least Squares (OLS) to estimate the values of μ and Q.

Conclusion

To conclude, this report holistically described and evaluated the implementation of various algorithmic trading strategies designed to maximize the average Sharpe ratio of a portfolio while minimizing its average turnover. In terms of next steps, I'd like to experiment with testing parameter values more thoroughly, perhaps even using scikit-learn or another machine learning library to determine the parameter values for which various strategies would perform best on the training data. I could only try so many values by hand, and I'm quite sure that I haven't found the most optimal trading strategy given the confines of this project.

Another path of exploration might be to use the asset and factor data to generate random asset prices using Monte-Carlo simulations, and then use the results of those simulations to improve the precision of the algorithms. In particular, my CVaR algorithm would likely have benefited from a non-linear formulation, and using Monte-Carlo simulation results to generate a probabilistically accurate distribution function $p(r)$ of the random returns seems like a promising idea.
