MMF1921 Operational Research Project 2

Automated Asset Management System

by:

Lujia Yang 1002955563

Xiaolu Xiong 1001126202

May 2022

Abstract

In the digital age, businesses must manage and track more assets than ever before. Asset management systems help companies increase the predictability of asset value and allocate and manage assets efficiently to ensure their optimal capital utilization while reducing transaction costs. This project combined four different factor models (OLS, FF, LASSO, and BSS) with four optimization strategies (MVO, Robust MVO, CVaR, and RP) and investigated the portfolio performance over different periods and assets. The result implies a poor performance of the CVaR portfolio. RP and Robust MVO can improve the normal MVO, but RP requires more computational power than Robust MVO. Asset managers who target a higher Sharpe ratio and lower trading cost may prefer the system using BSS and Robust MVO.

Contents

1 Introduction

The application of mathematical optimization in portfolio construction has been prevalent in academia and industry since the introduction of modern portfolio theory (MPT) in Markowitz (1952). Meanwhile, Factor models have become popular in finance for the economic relevance of the factors and their ability to quantify different sources of an asset's systematic risk. Asset management firms use factor models to estimate asset return and correlation before making investment decisions. The mean-variance optimization (MVO) problem introduced by MPT analyses the trade-off between risk and return to construct optimal portfolios. These portfolios provide an optimal balance of risk and return relative to an investor's risk appetite. However, while MVO has been widely used as a quantitative tool in asset management, it has also been criticized for its susceptibility to estimation error in its input parameters - the asset expected returns and covariance matrix. Portfolio managers can improve the quality of the asset management system in two ways: 1.) increasing the accuracy of factor models when estimating the expected return and covariance matrix. 2.) enhance the optimization algorithms. A superior algorithmic trading system is expected to manage a portfolio to achieve a high Sharpe ratio, low turnover rate, and low running time. This project aims to design an automated asset management system based on monthly price and return data for equities and equities based factors listed in the following tables. Portfolios will be rebalanced every six months.

Table 1: List of assets by ticker

F	CAT	DIS	MCD	KO	PEP	WMT	C	WFC	JPM
AAPL	IBM	PFE	JNJ	XOM	MRO	ED	T	VZ	NEM

Table 2: List of factors

Market ('Mkt_RF')	Size ('SMB')	Value ('HML')	Short-term reversal ('ST_Rev')
Profitability ('RMW')	Investment ('CMA')	Momentum ('Mom')	Long-term reversal ('LT_Rev')

The model development process starts by utilizing four different factor models, including ordinary least squares (OLS), Fama-French (FF), Least Absolute Shrinkage and Selection Operator(LASSO), Best Subset Selection(BSS), to estimate asset returns and covariance matrix. Other than the MVO optimization strategy, we find three alternative strategies – Roubust MVO, CVaR, and Risk Parity (RP) to test whether these methods outperform the normal MVO. The next section will explain the concept and formulation of these models in detail. Additionally, to achieve the best estimation accuracy and the best portfolio value, we need to tune the model parameters and select an appropriate value, which will be discussed in the Parameter Selection Section.

After permuting and combining four factor models and four optimization strategies, we trained, tested, and validated 16 different algorithms and selected the best trading strategy, based on the elapsed time, portfolio Sharpe ratio, and average turnover rate. Section 4 will discuss the model selection process and demonstrate the asset allocation for the best portfolio.

2 Methodology

2.1 Factor models

Factor models attempt to explicitly explain the behaviour of a random variable either through a single factor, such as the capital asset pricing model or through a combination of multiple factors. This project will focus on modelling the asset excess return using four different factor models, including OLS, FF, LASSO and BSS. The excess return is obtained by subtracting the risk-free monthly rate provided with the factor data from our monthly asset returns.

As shown in Table 2, all the factors can be stemmed out from synthetic portfolios of assets with shared properties. It is essential for a portfolio manager to reduce the correlation risks among assets. Thus, the asset covariance matrix needs to be considered when doing portfolio optimization.

2.1.1 OLS model

The ordinary least squares (OLS) model, select the coefficient of a multi-variable linear function of all eight factors introduced in Table 2. This method aims to minimize the sum of the squares of the differences between the values of the stock's excess returns within the given data set. The multi-factor model is constructed using all eight factors listed in Table 1, for each asset $i = 1, \cdots, n; n = 20$. T is the total number of individual observations. The excess return for asset i using the OLS model is as below:

$$r_i - r_f = \alpha_i + \sum_{k=1}^{p=8} \beta_{ik} f_k + \epsilon_i$$

Here, r_i is the return for asset i; r_f is the risk-free rate; α_i is the intercept from regression; f_k is the return of factor k and β_{ik} is the corresponding factor loading; ϵ_i is the stochastic error term of the asset (idiosyncratic risk).

The coefficients of eight factors are chosen where the sum of squared residuals are minimized:

$$\min_{\boldsymbol{B}_i} ||\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i||_2^2$$

To simplify the problem which captures the difference between the observed assets returns and predicted returns by the OLS model, we denote the data set of intercept and observed factors in vector form using $\boldsymbol{X}$ ($\boldsymbol{X} \in \mathbb{R}^{T\times(p+1)}$). The regression coefficient matrix of intercept and factor loadings is represented by $\boldsymbol{B}_i$ ($\boldsymbol{B}_i \in \mathbb{R}^{(p+1)}$), while $||\cdot||$ calculates the squared Euclidean norm, which measures the distance of the vector coordinate from the defined starting point within the vector space.

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{1f} \end{bmatrix} = \begin{bmatrix} \boldsymbol{1} \ \boldsymbol{f1}...\boldsymbol{f8} \end{bmatrix}; \boldsymbol{B}_i = \begin{bmatrix} \alpha_i \\ \boldsymbol{V}_i \end{bmatrix} = \begin{bmatrix} \alpha_i \\ \beta_{i1} \\ ... \\ \beta_{i8} \end{bmatrix}; \boldsymbol{V} = \begin{bmatrix} \beta_{1,1} & ... & \beta_{20,1} \\ \cdots & \cdots & \cdots \\ \beta_{1,8} & ... & \beta_{20,8} \end{bmatrix}_{8\times 20}$$

Through calculation and expansion, we can derive the following formulas and can further derive the First-order Necessary Condition (FONC) respectively.

$$||\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i||_2^2 = (\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i)^T(\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i) = \boldsymbol{B}_i^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{B}_i - 2\boldsymbol{r}_i^T\boldsymbol{X}\boldsymbol{B}_i + \boldsymbol{r}_i^T\boldsymbol{r}_i$$

$$\min_{\boldsymbol{B}_i} ||\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i||_2^2 = \min_{\boldsymbol{B}_i} \boldsymbol{B}_i^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{B}_i - 2\boldsymbol{r}_i^T\boldsymbol{X}\boldsymbol{B}_i$$

Set $L(\boldsymbol{B}_i) = \boldsymbol{B}_i\boldsymbol{B}_i^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{B}_i - 2\boldsymbol{r}_i^T\boldsymbol{X}\boldsymbol{B}_i$, then:

$$\frac{\partial L(\boldsymbol{B}_i)}{\partial \boldsymbol{B}_i} = 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{B}_i - 2\boldsymbol{r}_i^T\boldsymbol{X} = 0$$

Therefore, given 20 assets ($n = 20$), we can obtain the optimal solution, $\boldsymbol{B}_i^*$ and $\boldsymbol{B}^*$, whereas $\boldsymbol{R}$ indicates the matrix of asset returns ($\boldsymbol{R} \in \mathbb{R}^{T\times n}$).

$$\boldsymbol{B}_i^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{r}_i; \boldsymbol{B}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{R} = \begin{bmatrix} \alpha^{*T} \\ \\ \boldsymbol{V}^* \end{bmatrix}_{9\times 20}$$

Note that the residual term can be written as $\epsilon_i = \boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i^*$, and we can use this vector of residuals to calculate the unbiased estimate of the residual variance $\sigma_{\epsilon_i^2} = \frac{1}{T-p-1}||\epsilon_i||_2^2$, whereas T is the total number of observations, p is the number of coefficients and $T-p-1$ is the degree of freedom.

In order to incorporate mean-variance optimization, the expected return can be written as $\boldsymbol{\mu} = \boldsymbol{\alpha} + \boldsymbol{V^T}\bar{\boldsymbol{f}}$, whereas $\bar{\boldsymbol{f}}$ is used to represent the expected factor returns ($\bar{\boldsymbol{f}} \in \mathbb{R}^p$). Similarly, the covariance matrix $\boldsymbol{Q}$ can be expressed as $\boldsymbol{Q} = \boldsymbol{V^T}\boldsymbol{F}\boldsymbol{V} + \boldsymbol{D}$, whereas the factor covariance effect is $\boldsymbol{F} = cov(\boldsymbol{f}); \boldsymbol{F} \in p \times p$ and the diagonal matrix of residual variances is written as $\boldsymbol{D} \in n \times n$.

$$\boldsymbol{D} = \begin{bmatrix} \sigma_{\epsilon_1^2} & 0 & \cdots & 0 \\ 0 & \sigma_{\epsilon_2^2} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \sigma_{\epsilon_{20}^2} \end{bmatrix}_{20\times 20}$$

.

2.1.2 FF model

In general, factor model can combine a range of factors to model security returns in the form of linear combination. The Fama–French (FF) three-factor model is a subset of the OLS model, where we use only the Market, Size, and Value factors from Table 2. The FF model is

$$r_i - r_f = \alpha_i + \beta_{im}(f_m - r_f) + \beta_{is}f_s + \beta_{iv}f_v + \epsilon_i$$

Here, r_i is the return of asset i; r_f is the risk-free rate; α_i is the intercept from regression; $f_m - r_f$ is the excess market return factor and β_{im} is its corresponding factor loading; f_s is the size factor and β_{is} is its corresponding factor loading; f_v is the value factor and β_{iv} is its corresponding factor loading; ϵ_i is the stochastic error term of the asset (idiosyncratic risk). FF model is similar to OLS model, so we also have similar formulation as below:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{1 f} \end{bmatrix} = \begin{bmatrix} \boldsymbol{1} \ \boldsymbol{f1}...\boldsymbol{f3} \end{bmatrix}; \boldsymbol{B}_i = \begin{bmatrix} \alpha_i \\ \boldsymbol{V}_i \end{bmatrix} = \begin{bmatrix} \alpha_i \\ \beta_{i1} \\ ... \\ \beta_{i3} \end{bmatrix}; \boldsymbol{V} = \begin{bmatrix} \beta_{1,1} \ ... \ \beta_{20,1} \\ \cdots \\ \beta_{1,3} \ ... \ \beta_{20,3} \end{bmatrix}_{8\times 20}$$

$$\boldsymbol{B}_i^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{r}_i; \boldsymbol{B}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{R} = \begin{bmatrix} \alpha^{*T} \\ \boldsymbol{V}^* \end{bmatrix}_{4\times 20}$$

The expected returns $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{Q}$ are obtained using the same approach as described in the previous section, OLS model.

2.1.3 LASSO model

Least Absolute Shrinkage and Selection Operator (LASSO), is an alternative to the subset selection method, which shrinks the regression coefficients toward zero by penalizing the regression model with a penalty term called L1-norm.

The constrained form of LASSO can be expressed as the following.

$$\min_{\boldsymbol{B}_i, y} ||\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i||_2^2$$
$$s.t. ||\boldsymbol{B}_i||_1 \leq s$$

Here, $||\boldsymbol{B}_i||_1 \leq s$ and $s \in \mathbb{R}_+$, which is a positive parameter that limits the size of the sum of our regression coefficients in LASSO model. In other words, there always exists some value of $s \geq 0$ and some value of $\lambda \geq 0$ such that the optimal solution B_i^* is equivalent for both constrained form and its penalized form of LASSO.

The penalized form of LASSO is calibrated for this project, which can be described as the following.

$$\min_{\boldsymbol{B}_i} ||\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i||_2^2 + \lambda||\boldsymbol{B}_i||_1$$

Note that $||\cdot||_1$ is used to represent the l_1 norm, which is continuous and convex, but not smooth everywhere.

In order to translate the lasso model to a quadratic programming problem, we need to consider that the penalized form is equivalent with the constrained form. Introducing an auxiliary variable $\boldsymbol{y} \in \mathbb{R}^{p+1}$, we have below problem.

$$\begin{aligned}
&\min_{\boldsymbol{B}_i, y} ||\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i||_2^2 + \lambda\mathbf{1^T y} \\
&s.t. \quad \boldsymbol{y} \geq \boldsymbol{B}_i \\
&\qquad \boldsymbol{y} \geq -\boldsymbol{B}_i
\end{aligned}$$

At optimality, $\boldsymbol{y}$ will be equal to $|\boldsymbol{B}_i|$. Therefore, for each asset class i, we can build a quadratic programming problem.

As

$$\begin{aligned}
||\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i||_2^2 + \lambda||\boldsymbol{B}||_i &= \boldsymbol{r}_i^T\boldsymbol{r}_i - 2\boldsymbol{r}_i^T\boldsymbol{X}(\boldsymbol{B}_i^+ - \boldsymbol{B}_i^-) + (\boldsymbol{B}_i^+ - \boldsymbol{B}_i^-)^T\boldsymbol{X}^T\boldsymbol{X}(\boldsymbol{B}_i^+\boldsymbol{B}_i^-) + \lambda\mathbf{1}^T(\boldsymbol{B}_i^+ + \boldsymbol{B}_i^-) \\
&= \boldsymbol{r}_i^T\boldsymbol{r}_i - 2\boldsymbol{r}_i^T\boldsymbol{X}\boldsymbol{B}_i + \boldsymbol{B}_i^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{B}_i + \lambda\mathbf{1}^T|\boldsymbol{B}_i|
\end{aligned}$$

while setting $|\boldsymbol{B}_i| = \boldsymbol{B}_i^+ + \boldsymbol{B}_i^-$, $\boldsymbol{B}_i = \boldsymbol{B}_i^+ - \boldsymbol{B}_i^-$ and $\boldsymbol{B}_i^+, \boldsymbol{B}_i^- \geq 0$.

That is, we can write $\boldsymbol{B}_i$ as $\begin{bmatrix}\boldsymbol{I} & -\boldsymbol{I}\end{bmatrix} \times \begin{bmatrix}\boldsymbol{B}_i^+ \\ \boldsymbol{B}_i^-\end{bmatrix}$, $|\boldsymbol{B}_i|$ as $\begin{bmatrix}\boldsymbol{I} & \boldsymbol{I}\end{bmatrix} \times \begin{bmatrix}\boldsymbol{B}_i^+ \\ \boldsymbol{B}_i^-\end{bmatrix}$, whereas $\boldsymbol{I}$ is an $p+1$ identity matrix, and p is the number of factors.

$$\begin{aligned}
&= \boldsymbol{r}_i^T\boldsymbol{r}_i - 2\boldsymbol{r}_i^T\boldsymbol{X}[\boldsymbol{I} - \boldsymbol{I}]\begin{bmatrix}\boldsymbol{B}_i^+ \\ \boldsymbol{B}_i^-\end{bmatrix} + ([\boldsymbol{I} - \boldsymbol{I}]\begin{bmatrix}\boldsymbol{B}_i^+ \\ \boldsymbol{B}_i^-\end{bmatrix})^T\boldsymbol{X}^T\boldsymbol{X}[\boldsymbol{I} - \boldsymbol{I}]\begin{bmatrix}\boldsymbol{B}_i^+ \\ \boldsymbol{B}_i^-\end{bmatrix} + \lambda\mathbf{1}^T[\boldsymbol{I}\ \boldsymbol{I}]\begin{bmatrix}\boldsymbol{B}_i^+ \\ \boldsymbol{B}_i^-\end{bmatrix} \\
&= (\begin{bmatrix}\boldsymbol{B}_i^+ \\ \boldsymbol{B}_i^-\end{bmatrix})^T[\boldsymbol{I} - \boldsymbol{I}]^T\boldsymbol{X}^T\boldsymbol{X}[\boldsymbol{I} - \boldsymbol{I}]\begin{bmatrix}\boldsymbol{B}_i^+ \\ \boldsymbol{B}_i^-\end{bmatrix} + (\lambda\mathbf{1}^T[\boldsymbol{I}\ \boldsymbol{I}] - 2\boldsymbol{r}_i^T\boldsymbol{X}[\boldsymbol{I} - \boldsymbol{I}])\begin{bmatrix}\boldsymbol{B}_i^+ \\ \boldsymbol{B}_i^-\end{bmatrix} + \boldsymbol{r}_i^T\boldsymbol{r}_i
\end{aligned}$$

Therefore , our quadratic programming is as follows.

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} + \boldsymbol{f}^T\boldsymbol{x} \\ &s.t. \quad \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \end{aligned}$$

where

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} \end{bmatrix}^T \boldsymbol{X}^T\boldsymbol{X} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} \end{bmatrix}^T$$

$$\boldsymbol{f} = -[\boldsymbol{I} - \boldsymbol{I}]^T\boldsymbol{X}^T\boldsymbol{r}_i + \frac{\boldsymbol{\lambda}}{2}[\boldsymbol{I}\ \boldsymbol{I}]^T\boldsymbol{1}$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{B}_i^+ \\ \boldsymbol{B}_i^- \end{bmatrix}$$

Based on the study and testing from porject 1, we observed there is a trade-off between adjusted-R^2 and the portfolio value. In the figure below, we can see the LASSO parameter $\lambda = 0.01$ is a reasonable choice, which balanced the model performance and portfolio return.

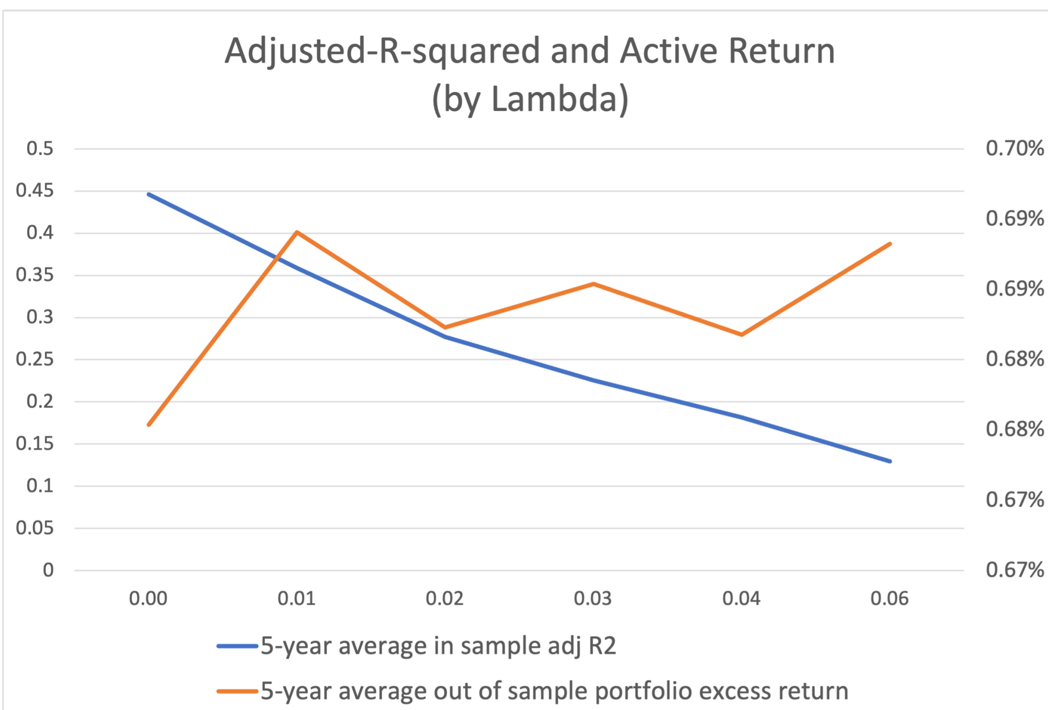


Figure 1: Comparison of Adjusted R-Square

2.1.4 BSS model

The constrained form of Best Subset Selection (BSS) model include all eight factors at the beginning. According to the selected parameter K, it will only select K factors to make prediction, in other words, there is only K factors with non-zero coefficient. The model is constructed by solving the following optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{B}_i} ||\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i||_2^2 + \lambda||\boldsymbol{B}_i||_0 \\ &s.t. \quad ||\boldsymbol{B}_i||_0 \leq K \end{aligned}$$

The l_0 norm can impose a cardinality limit on the coefficients, which is to limiting the number of factors used in the model. Consider adding the constraints for the BSS model, it can be expressed as

$$\begin{aligned} &\min_{\boldsymbol{B}_i, \boldsymbol{y}} ||\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i||_2^2 \\ &s.t. \quad L\boldsymbol{y} \leq \boldsymbol{B}_i \leq U\boldsymbol{y} \\ &\qquad \mathbf{1}^T\boldsymbol{y} \leq \boldsymbol{K} \end{aligned}$$

where $y_j \in \{0, 1\}$ for $j = 1, \cdots, p+1$, representing auxiliary binary variables. L and U are lower and upper bounds. K is the cardinality constraints ($K \in \mathbb{Z}_+$), which is a positive integer.

since $\boldsymbol{r}_i^T\boldsymbol{r}_i$ is a constant term and can be ignored

$$||\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i||_2^2 = \boldsymbol{B}_i^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{B}_i - 2\boldsymbol{r}_i^T\boldsymbol{X}\boldsymbol{B}_i + \boldsymbol{r}_i^T\boldsymbol{r}_i$$

$$\min_{\boldsymbol{B}_i, \boldsymbol{y}} ||\boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i||_2^2 \Longleftrightarrow \min_{\boldsymbol{B}_i, \boldsymbol{y}} \boldsymbol{B}_i^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{B}_i - 2\boldsymbol{r}_i^T\boldsymbol{X}\boldsymbol{B}_i$$

The above model can be converted to a quadratic programming problem which can be solved with quadprog().

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x}+\boldsymbol{c}^T\boldsymbol{x} \\ &s.t. \quad \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \end{aligned}$$

Let $\boldsymbol{H} = \boldsymbol{X}^T\boldsymbol{X} \in \mathbb{R}^{(2p+2)\times(2p+2)}$, $\boldsymbol{c}^T = -2\boldsymbol{r}_i^T\boldsymbol{X} \in \mathbb{R}^{(2p+2)\times 1}$.

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{B}_i \\ \boldsymbol{y}_i \end{bmatrix} \in \mathbb{R}^{(2p+2)\times 1}, \quad \boldsymbol{A} = \begin{bmatrix} -\boldsymbol{I} & L\boldsymbol{I} \\ -\boldsymbol{I} & U\boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ ... \\ 0 \end{bmatrix}_{(2p+2)\times 1}$$

And

$$\boldsymbol{Aeq} = \begin{bmatrix} \boldsymbol{0}_{1\times(p+1)} & \boldsymbol{1}_{1\times(p+1)} \end{bmatrix}, \quad \boldsymbol{beq} = \begin{bmatrix} K \\ ... \\ K \end{bmatrix}_{(p+1)\times 1}$$

Thus the above mixed-integer quadratic program (MIQP) can be solved in MATLAB with Gurobi functions. After obtaining $\boldsymbol{x}$, specifically $\boldsymbol{x}_i$ for each asset, we can further derive $\boldsymbol{B}_i = [\boldsymbol{1}_{(p+1)\times 1}\ \boldsymbol{0}_{(p+1)\times 1}]$ as well as $\boldsymbol{B} = [\boldsymbol{B}_1\ \boldsymbol{B}_2\ \cdots\ \boldsymbol{B}_{20}]$, and calculate $\boldsymbol{\mu} = \alpha^* + \boldsymbol{V}^{*T}\bar{\boldsymbol{f}}$ ($\bar{\boldsymbol{f}}$ represents the vector of expected factor returns) as well as $\boldsymbol{Q} = \boldsymbol{V}^{*T}\boldsymbol{F}\boldsymbol{V}^* + \boldsymbol{D}$, similar as what we derived in previous sections. Note that the residual term can be written as $\epsilon_i = \boldsymbol{r}_i - \boldsymbol{X}\boldsymbol{B}_i^*$, and we can use this vector of residuals to calculate the unbiased estimate of the residual variance $\sigma_{\epsilon_i^2} = \frac{1}{39}||\epsilon_i||_2^2$.

Based on the plot which also explained in porject 1, we can see the BSS parameter $k = 5$ is the best choice. When K > 5, adding another factor into the model will no longer increase the adjusted R^2, but the portfolio return drop significantly.

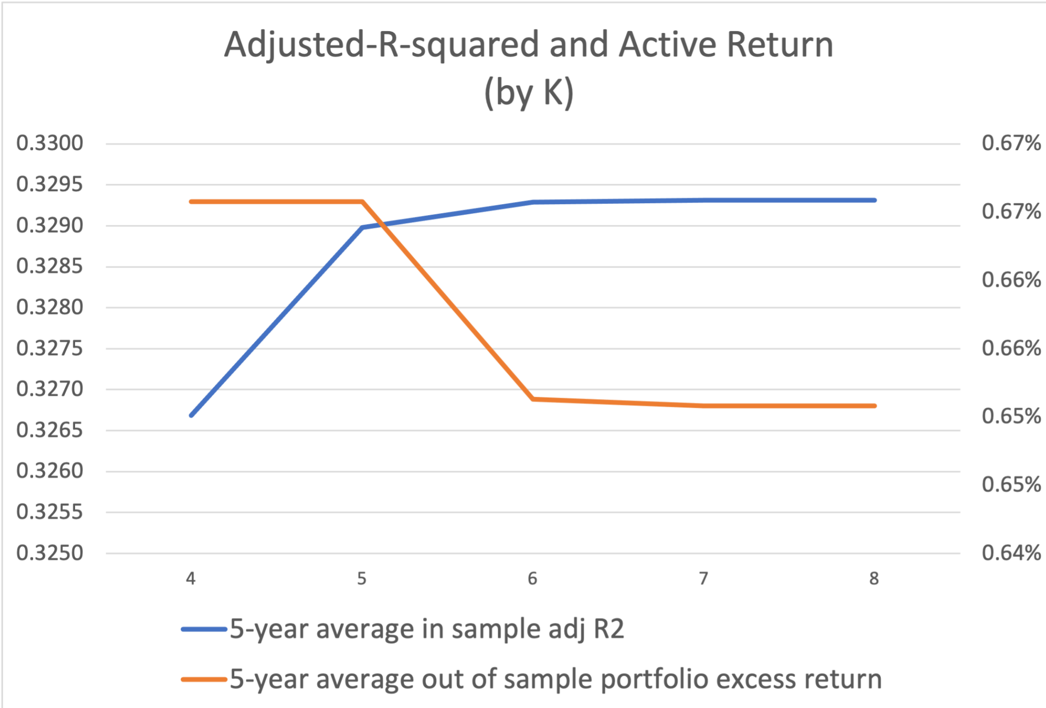


Figure 2: Comparison of Adjusted R-Square

2.2 Portfolio Optimization

Using the estimated result from the factor models as input for the optimization algorithms, we implement various methods, including Mean-Variance Optimization (MVO), Robust MVO, CVaR Optimization, and Risk Parity (RP). These optimization techniques differ in objectives and constraints, resulting in different asset allocation strategies. This layer of variation, in addition to the variation of factor models, potentially gives us better-performing portfolios.

2.2.1 Normal Mean-Variance Optimization

The mean-variance analysis is a tool that investors use to diversify risks while meeting the target asset return. To implement this approach, we can use the factor models to estimate the expected returns $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{Q}$ and use them as the input for Mean-

Variance Optimization (MVO) problem.

$$
\begin{aligned}
\text{minimize} \quad & \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} \\
s.t. \quad & \boldsymbol{\mu}^T \boldsymbol{x} \geq R \\
& \mathbf{1}^T \boldsymbol{x} = 1 \\
& \boldsymbol{x} \geq 0
\end{aligned}
$$

Here, $\boldsymbol{x} = [x_1 \ x_2 \ \cdots x_20]$, where $x_i \geq 0$ for $i = 1, 2, \cdots, 20$, which represents the weight of individual asset in the portfolio without allowing short-selling. Investors need to allocate asset appropriately to achieve the target rate of return. $\boldsymbol{x}$ is the weight allocate to each asset, the sum of which is 1.

Every 6 months, we re-calibrate and re-calculate our portfolio's expected returns $\boldsymbol{\mu}$ and the covariance $\boldsymbol{Q}$ by using preceding historical returns at the end of each year, and eventually calculate the asset allocation $\boldsymbol{x}$ for the optimal portfolios.

2.2.2 Robust Mean-Variance Optimization

The normal MVO is not stable. There is a large gap between the estimated and actual efficient frontier. Based on Chopra and Ziemba (1993), a widely accepted conclusion states that estimation errors during optimization can lead to over-concentrated portfolios. Errors in expected returns can impact ten times larger than errors in the covariance matrix. The robust optimization methods can introduce uncertainty sets around the estimated parameters to explicitly capture this estimation error during optimization. Moreover, it can introduce uncertainty regarding the choice of a probability distribution that is used to describe the asset returns. In this case, we applied the ellipsoidal uncertainty set ($\boldsymbol{\mu}^{true} = \{\boldsymbol{\mu}^{true} \in \mathbb{R}^n : (\boldsymbol{\mu}^{true} - \boldsymbol{\mu})^T \Theta^{-1} (\boldsymbol{\mu}^{true} - \boldsymbol{\mu}) \leq \epsilon_2^2\}$) around the estimation points to enhance the stability of algorithms and narrow the gap between the estimated and actual efficient frontier. ϵ_2 is the radius that bounds the standardized distance between $\boldsymbol{\mu}^{true}$ and $\boldsymbol{\mu}$. Θ is an n by n matrix that measures the uncertainty of estimated $\hat{\mu}$. We add two parameters while constructing the Robust MVO:

- **Parameter for penalized asset correlation** (λ): Robust optimization seeks to optimize a portfolio to the worst-case realization of the estimated parameters, so we add a parameter before $x^T\boldsymbol{Q}x$.

- **Confidence level** (α) determine the length of radius of the uncertainty set, where $\epsilon_2^2 = \chi_n^2(\alpha)$ and χ_n^2 is the inverse cumulative distribution function of the chi-squared distribution with n degrees of freedom. The uncertainty can be added as a penalty on the target returns as $\epsilon_2\|\Theta^{1/2}x\|_2 = \epsilon_2\sqrt{x^T\Theta x}$

We can have an optimal trade-off between the risk and penalized expected returns. The minimization problem are as follow:

$$\begin{aligned} \text{minimize} \quad & \lambda\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{\mu}^T x + \epsilon_2\|\Theta^{1/2}x\|_2 \\ s.t. \quad & \mathbf{1}^T\boldsymbol{x} = 1 \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

The optimization strategy used $\lambda = 0.01$ and $\alpha = 0.99$. The reason of choosing the parameter value will be explained in the parameter selection section.

2.2.3 CVaR Optimization

Conditional Value at Risk (CVaR) attempts to address the shortcomings of the VaR model, which measure the tail risk within an investment portfolio over a specific period. While VaR represents an extreme loss associated with a probability and a time horizon, CVaR is the expected loss beyond that worst-case threshold. Safer investments such as large-cap stocks or investment-grade bonds are likely to have a lower CVaR. Riskier assets, such as small-cap stocks, emerging market equities, or derivatives, can have large CVaRs. For portfolio managers, there is a trade-off between the high upside potential and low CVaRs.

Mathematically, we define VaR as below:

$$Let\ X = -r^T x\ as\ portfolio\ loss;$$

$$The\ Probability\ of\ portfolio\ loss\ less\ than\ \gamma\ is\ \psi(\boldsymbol{x}, \gamma) = \int_{f(\boldsymbol{x},r)\leq\gamma} p(\boldsymbol{r})d\boldsymbol{r}$$

$$VaR_\alpha(X) = min\{\gamma \in \mathbb{R} : P(X \geq \gamma) \leq 1-\alpha\} = min\{\gamma \in \mathbb{R} : \psi(\boldsymbol{x}, \gamma) \geq \alpha\}$$

Since VaR does not always respect diversification and optimizing VaR is a non-convex activity. Investors can optimize CVaR instead of VaR. Mathematically, we define CVaR as below:

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

We can let γ serve as a placeholder for VaR_α during optimization. The formula of CVaR can be written as:

$$F_\alpha(X, \gamma) = \gamma + \frac{1}{1-\alpha} \int (f(\boldsymbol{x}, \boldsymbol{r}) - \gamma)^+ p(\boldsymbol{r})d\boldsymbol{r}$$

By doing this, we modified the non-convex problem to a convex problem. $VaR_\alpha(x)$ is the γ that minimizes $F_\alpha(x, \gamma)$. The minimum value over γ of the function $F_\alpha(x, \gamma)$ is $CVaR_\alpha(x)$. Next we implement this function use a scenario-based representation. $\hat{r_s} \in \mathbb{R}^n$ is the realization of scenario s, where s = 1 ...S. Assume all scenarios are equally likely, and the loss function is linear $f(\boldsymbol{x}, \boldsymbol{\hat{r_s}}) = \boldsymbol{\hat{r_s}^T x}$. We can approximate $F_\alpha(x, \gamma)$:

$$\widetilde{F}_\alpha(X, \gamma) = \gamma + \frac{1}{(1-\alpha)S} \sum_{s=1}^{S} (f(\boldsymbol{x}, \boldsymbol{\hat{r_s}}) - \gamma)^+ p(\boldsymbol{r})d\boldsymbol{r}$$

Note that $(f(\boldsymbol{x}, \boldsymbol{\hat{r_s}}) - \gamma)^+$ is non-smooth, so we introduce an auxiliary variable Z_s for each

s. Finally, the CVaR optimization problem is:

$$
\begin{aligned}
\text{minimize} \quad & \gamma + \frac{1}{(1-\alpha)S}\sum_{s=1}^{S} Z_s \\
s.t. \quad & Z_s \geq 0 \ s = 1, ...S \\
& Z_s \geq f(\boldsymbol{x}, \boldsymbol{\hat{r}_s}) \ s = 1, ...S \\
& \boldsymbol{x} \in \chi
\end{aligned}
$$

The constrain set χ includes budget, target return, and other constrains influence portfolio weights. Based on the given dataset, we use the historical monthly data and the factor model to estimate the asset distribution and optimize CVaR. Investor need to select appropriate significance level (α) based on their investment goal and risk tolerance, which will explained in section 3.

2.2.4 Risk Parity Optimization

Risk parity overcomes the drawbacks of Markowitz portfolios by attempting to reduce the reliance on noisy estimated parameters which may mislead the optimization to rely on unreliable estimated asset returns. It eliminates the expected returns from the model. Therefore, it may be another substitution that improves the stability of the algorithms. The differences between risk contributions are minimized by this method which leads to the desired diversification trait, where resources are allocated based on the measure of risk where each asset contributes equal risk σ_i instead of equal weight. Risk parity optimization protects against overconcentration into individual assets.

The portfolio volatility ($\sigma_p = \sqrt{\boldsymbol{x^T Q x}}$) is a positive homogeneous function of degree k = 1. The marginal volatility contribution of asset i is:

$$
\frac{\partial \sigma_p}{\partial x_i} = \frac{(\boldsymbol{Qx})_i}{\sqrt{\boldsymbol{x^T Q x}}}
$$

where $(\boldsymbol{Qx})_i$ is the i^{th} element of vector $\boldsymbol{Qx}$. By Euler's theorem ($kf(\boldsymbol{x}) = \boldsymbol{x^T} \bigtriangledown f(\boldsymbol{x})$), we have:

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

The portfolio variance can be partitioned as below:

$$\sigma_p^2 = \boldsymbol{x^T Q x} = \sum_{i=1}^{n} x_i \frac{\partial \sigma_p^2}{\partial x_i} = \sum_{i=1}^{n} x_i (\boldsymbol{Qx})_i = \sum_{i=1}^{n} R_i$$

In this case, R_i is risk contribution per asset. Risk parity seeks a portfolio where $R_i = R_j$ for any i, j. Thus, based on least-squares approach, i.e., minimize the sum of squared differences, we constructed an initial non-convex formulation:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i (\boldsymbol{Qx})_i - x_j (\boldsymbol{Qx})_j)^2 \\
s.t. \quad & \boldsymbol{1^T x} = 1 \\
& \boldsymbol{x} \geq \boldsymbol{0}
\end{aligned}$$

The optimal solution is long-only, and it is a highly non-linear objective function, which is hard to find the gradient of this objective analytically. Therefore, we need a convex reformulation. By inspect the gradient of our objective function. $f(\boldsymbol{y}) = \frac{1}{2} \boldsymbol{y^T Q y} - \kappa \sum_{i=1}^{n} ln(y_i)$, which is a strictly convex function and the minimum is a unique global solution. $\boldsymbol{y} \in \mathbb{R}^n, \boldsymbol{y} > 0, \kappa > 0$. Then, we can find the gradient and set it to zero:

$$\begin{aligned}
& \bigtriangledown f(\boldsymbol{y}) = \boldsymbol{Qy} - \kappa \boldsymbol{y}^{-1} = \boldsymbol{0} \\
& where\ \boldsymbol{y}^{-1} = [\frac{1}{y_1} ... \frac{1}{y_n}]^T \\
& (\boldsymbol{Qy})_i = \frac{\kappa}{y_i}\ \forall i;\ y_i (\boldsymbol{Qy})_i = (y_j \boldsymbol{Qy})_j\ \forall i, j
\end{aligned}$$

The risk parity optimization problem can be reformulated as minimizing the difference in risk contributions:

$$\text{minimize} \quad f(\boldsymbol{y}) = \frac{1}{2}\boldsymbol{y^T Q y} - \kappa \sum_{i=1}^{n} ln(y_i)$$

$$s.t. \qquad \boldsymbol{y} \geq \boldsymbol{0}$$

By solving this convex problem, the optimal asset weights ($\boldsymbol{x}^*$) is unique and independent of the initial choice of $\kappa > 0$, where:

$$\boldsymbol{x}^* = \frac{y_i^*}{\sum_{i=1}^{n} y_i^*}$$

2.3 Performance Measure

2.3.1 R-squared (R^2)

The coefficient of determination, R^2, is based on the proportion of total variation of the outcomes explained by the model and can measure the model performance. Adjusted R-squared penalizes the model convexity and is a better measure of the quality of factor models. We use this measure while selecting the parameters' value for LASSO and BSS as mentioned in the factor model section.

$$SS_{\text{res}} = \sum_i (y_i - \hat{y}_i)^2 = \sum_i (r_i - \boldsymbol{X B}_i)^2 = \sum_i e_i^2$$

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2 = \sum_i (r_i - \bar{r})^2$$

$$Adjusted\ R^2 = 1 - \frac{SS_{\text{res}} \cdot (t - p - 1)}{SS_{\text{tot}} \cdot (t - 1)}$$

where n is the number of asset classes, p is the number of factors, and t is the number of observations.

2.3.2 Sharpe Ratio

To evaluate our model performance over three datasets (training, testing, and validation), we will calculate the portfolio's average return, variance, and sharp ratio for each model of each test period. Sharpe ratio is widely used for getting a risk-adjusted return, and it can help us to understand the model's risk-adjusted performance: a greater Sharpe Ratio indicates better risk-adjusted-performance.

$$\text{Sharpe Ratio} = \mathbb{E}[R_i - R_f]/\sigma_a$$

where R_i is the return of portfolio. R_f is the risk-free rate and σ_p is the standard deviation.

2.3.3 Average Turnover Rate

This rate indicates the number of times the stock in a portfolio buys or has been replaced each year. The higher the turnover rate means a higher trading frequency, which leads to a higher transaction cost.

2.3.4 Computational Runtime

The computational runtime is an important measure of system efficiency. The runtime analysis is a theoretical classification that estimates the increase in the running time of an algorithm when input size increases. In this project, we record the elapsed time to control the computational runtime of the system. The system with runtime issues only when it takes more than 5 minutes to run the algorithm.

2.3.5 Data and Model Scoring

For all datasets, including training, testing and validation, the first five years is reserved for calibration and estimation, and the rest of data will be used for the out-of-sample analysis. Dataset descriptions are as below:

Data	Period	Num of Asset	Factors
Dataset 1	Dec. 31,2001 - Dec. 31,2016	20 Assets	8 Factors in given table
Dataset 2	Dec. 31,1991 - Dec. 31,2006	30 Assets	8 Factors in given table
Dataset 3	Dec. 31,1999 - Dec. 31,2014	33 Assets	8 Factors in given table

Table 1: Dataset Description

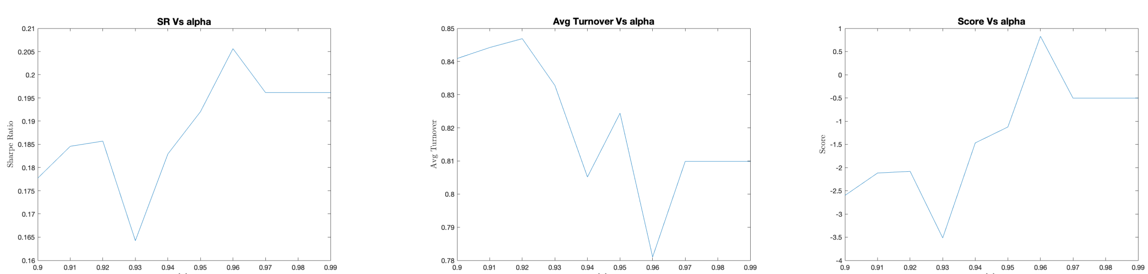
Based on the project assessment criteria, we make a score to roughly analyse the model quality, which helps us to rank the best model before deciding which algorithm to use in the final version of this system. We define a model score as:

$$\text{Score} = (0.8 \times \text{Sharpe Ratio} - 0.2 \times \text{Average Turnover Rate}) \times 100$$

3 Parameter Selection for Optimization Strategies

3.1 CVaR Parameter ($\alpha = 0.95$)

Value at Risk (VaR) is the minimum loss that a portfolio will match or exceed with a certain level of confidence α. Conditional Value-at-Risk (CVaR) measures the expected value of losses greater than or equal to VaR. In a CVaR optimization problem, intuitively, a higher confidence level for the investment loss estimation would lead to a less risky portfolio and probably result in lower portfolio returns. From the plot 3, we can find proof for part of our conjecture. When the confidence level is around 0.95, the portfolio's SR is around 0.205 which is a little greater than the SR = 0.195 when $\alpha = 0.99$. We noticed that the CVaR optimization method obtained results are dataset-specific according to trials on the three different datasets provided. The optimal solution to a CVaR problem with a lower confidence level can improve portfolio performance. However, the benefit from the lower confidence level is sensitivity to the thickness of the tails of the return distribution when a given level of risk is specified.Anderson et al. When the return distribution has heavier tails, the benefit would be more evident. As for this project, we don't have an assumption for the distribution of the returns, we decided to choose a relatively small α from a few commonly used αs, for instance $\alpha = 0.95, 0.975$ or 0.99. Therefore, $\alpha = 0.95$ is used when calibrating the CVaR Optimization.



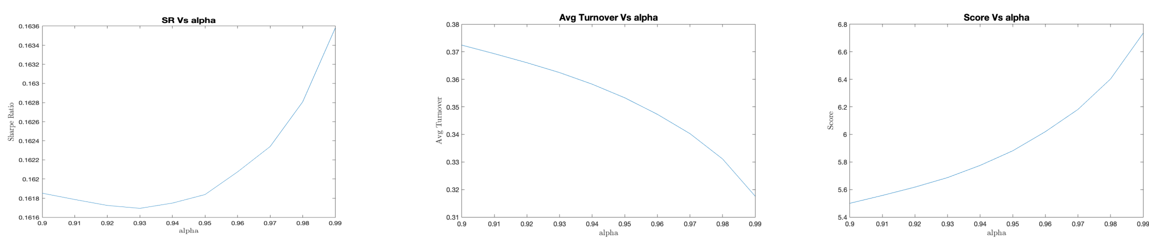
((a)) cVaR SR by α **((b))** cVaR Avg Turnover by α **((c))** cVaR Score by α

Figure 3: Finding Confidence Level for cVaR Optimzation

3.2 Robust MVO Parameter ($\alpha = 0.99$)

With an ellipsoidal uncertainty set, Robust Mean-Variance Optimization can incorporate uncertainty into the Mean-Variance optimization model. The ellipse's radius ϵ_2 is the radius that bounds the standardized distance between the model's mean return and the actual mean. The confidence level α of the robust MVO is related to the model's output as the ϵ_2^2 can be defined as $\chi_n^2(\alpha)$. Here χ_n^2 represents the inverse cumulative distribution function of the chi-squared distribution with n degrees of freedom. As the confidence level increases, the radius of the uncertainty set increases, meaning there is a heavier penalty to the target returns. As the uncertainty is better controlled compared to the case with a smaller penalty to the target returns, the model would be more stable. Hence, we expect our Sharpe ratio to increase.n Ceria [2006]

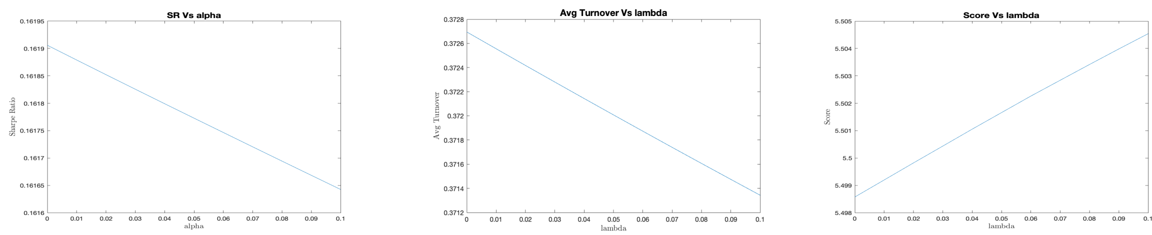
In addition, we can observe from the plot for trials of $\alpha = 0.90, 0.91, ...0.99$ with a fixed $\lambda = 0.02$ that the lower average turnover rate is an extra advantage with a higher confidence level. Thus, a higher confidence level is favourable for our studies; $\alpha = 0.99$ is chosen.



((a)) robustMVO SR by α ((b)) robustMVO Avg Turnover by α ((c)) robustMVO Score by α

Figure 4: Finding Confidence Level for robustMVO

3.3 Robust MVO Parameter ($\lambda = 0.01$)



((a)) robustMVO SR by λ

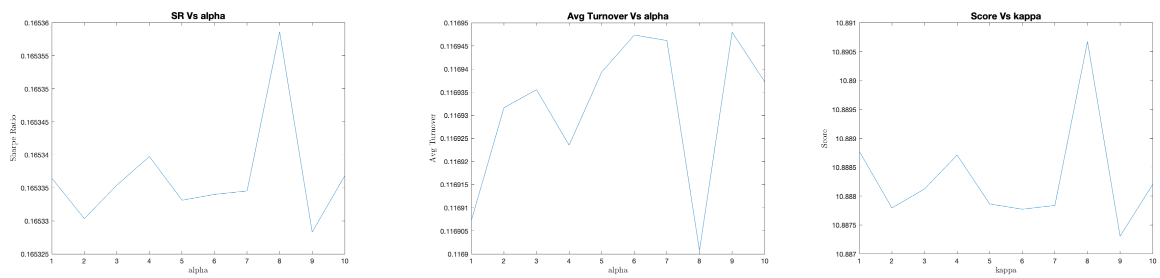
((b)) robustMVO Avg Turnover by λ

((c)) robustMVO Score by λ

Figure 5: Finding λ for robustMVO

λ is the parameter for penalized asset covariances that can control the risk factors in our model. The λ's effect on the model returns is tested with a fixed $\alpha = 0.90$. According to plot 5, as lambda goes up from 0.01 to 1 , the SR goes down from about 0.1619 to 0.1614. It is also noticeable that as λ increases, the average turnover rate decreases from around 0.3727 to 0.3713. The score of the model also increases with an increasing λ. As mentioned in the previous section, the score is calculated as (80%$\times$ Sharpe Ratio$-$20%$\times$ Avg Turnover Rate) $\times 100$. The score increase from $\lambda = 0.01$ to lambda $\lambda 1$ is about 0.006, which is minimal. From the previous analysis about confidence level α in robust MVO optimization, we know that with a fixed level of λ, as the confidence level decrease, the SR decreases. Therefore, the increase in Sharpe Ratio from a smaller λ is more valuable. $\lambda = 0.01$ that provided a better SR is chosen for further testing.

3.4 RP Parameter ($\kappa = 8$)



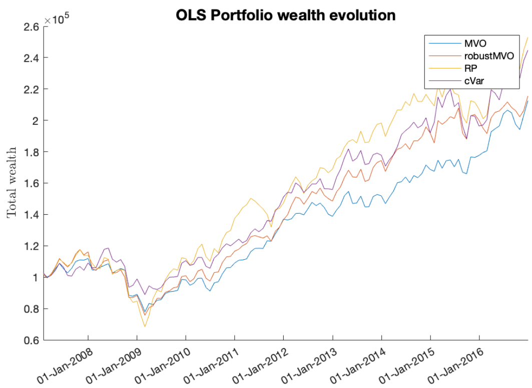
((**a**)) RP SR by κ ((**b**)) RP Avg Turnover by κ ((**c**)) RP Score by κ

Figure 6: Finding κ for RP

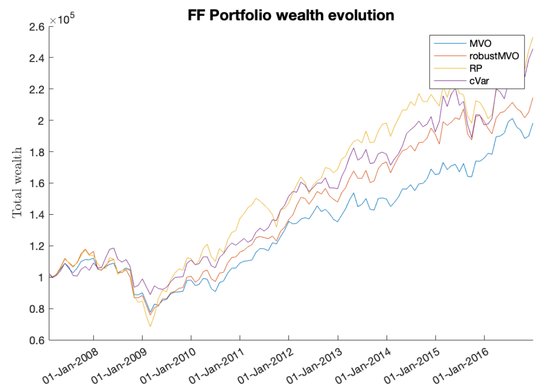
κ can be considered as a penalizing factor that controls in the objective function $\frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \kappa\sum_{i=1}^{n}\log(x_i)$ for the risk parity model. As κ increases, the part $\kappa \cdot \sum_{i=1}^{n}\log(x_i)$ increases. The effect of κ to the model's Sharpe Ratio, Average Turnover Rate and Score is tested with a set of κ ($\kappa = 1, 2, \cdots, 10$). Form the testing results 6, we didn't see a clear pattern for the changes of SR, Avg Turnover Rate or Score correspondingly. It appears that when $\kappa = 8$ the model performs surprisingly well and we have the highest SR and lowest avg turnover rate. So the optimum κ we decided to use in the further testing is $\kappa = 8$, where we can maximize the benefit from the risk diversification.

4 Performance of Portfolio Optimization Algorithms

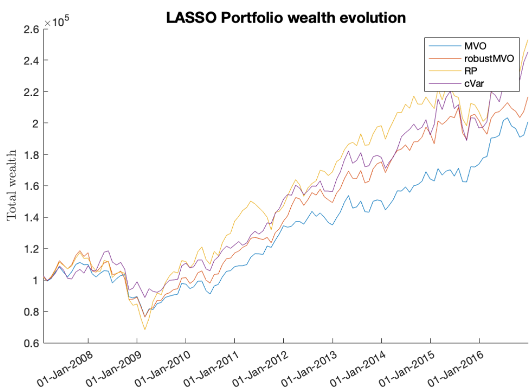
4.1 Performance Comparison: Dataset 1



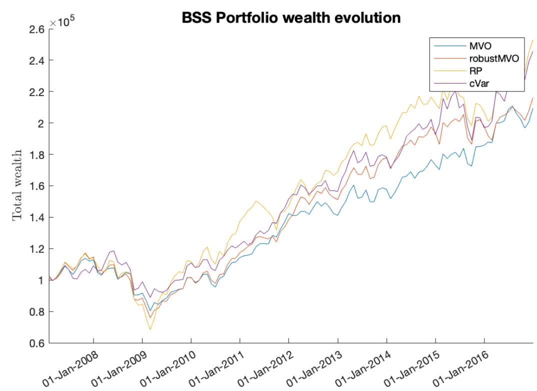
((a)) OLS Portfolio



((b)) FF Portfolio



((c)) LASSO Portfolio



((d)) BSS Portfolio

Figure 7: Compare four optimization strategies for each factor model in training

Parameter	Sharpe ratio				Avg. turnover				Elapsed time			
Optimization	MVO	Rob. MVO	CVaR	RP	MVO	Rob. MVO	CVaR	RP	MVO	Rob. MVO	CVaR	RP
OLS	0.1717	0.1619	0.1053	0.1654	0.4874	0.3726	0.9831	0.1169	0.8954	2.5395	1.3395	4.0037
FF	0.1522	0.1611	0.1053	0.1653	0.4462	0.3860	0.9831	0.1170	0.8411	2.5347	1.3725	3.6217
LASSO	0.1574	0.1629	0.1053	0.1654	0.3966	0.3921	0.9831	0.1169	2.6333	4.6428	3.0334	5.6402
BSS	0.1687	0.1673	0.1053	0.16531	0.5270	0.3335	0.9831	0.1169	1.3200	3.2411	1.9282	4.2331

Table 2: Dataset 1: Performance Metrics for Portfolios

The data in Dataset 1 is from December 31, 2001, to December 31, 2016. We applied a 3-year investment horizon. Based on the plot, we observe that the portfolio using RP as the optimization strategy outperforms the others no matter which factor model is used in the estimation stage. This is because the RP model eliminates the expected return as a model input, which is the major source of model error. The RP-related portfolio has a decent sharp ratio of around 0.16, with the lowest average turnover rate around 0.11. Moreover, RP and Robust MVO requires more computational resource than the other two optimization methods. LASSO is the most time-consuming factor model compared to the other three-factor models. On the other hand, the OLS portfolio has the lowest value, especially after the financial crisis (2008), with a moderate turnover rate and relatively high Sharpe ratio. It may be because of the overfitting issue in the OLS factor model.

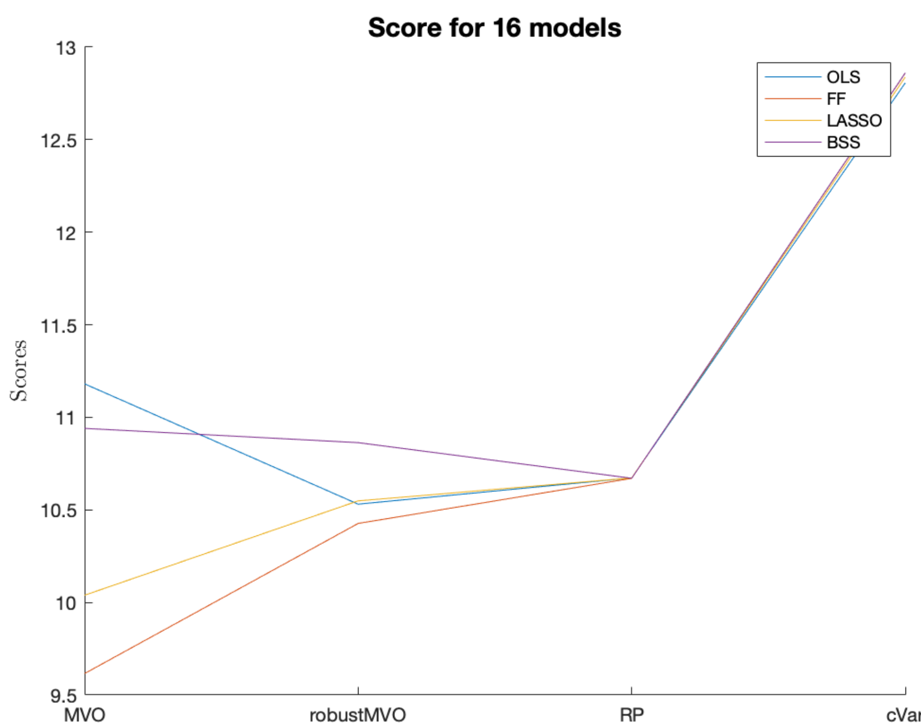
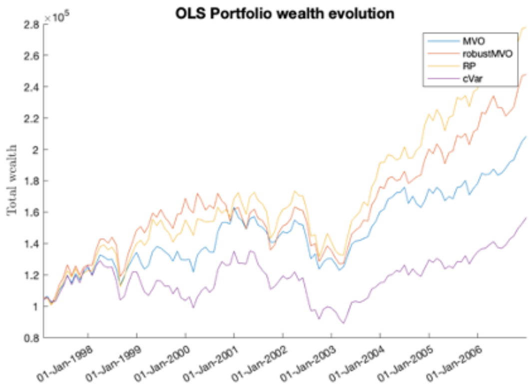


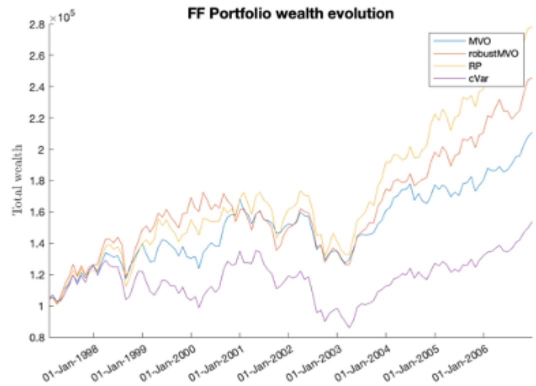
Figure 8: Dataset 1: Model Score

The score for the CVaR model is high because the scale for average turnover is larger than that of the Sharpe ratio. In other words, we cannot conclude the performance of the CVaR model is the best if only look at the model score in the figure. CVaR model has the lowest Sharpe ratio and high turnover rate. Since CVaR uses only the expected return from factor models, which is volatile and introduces more uncertainty than the covariance matrix, leading to a more fluctuating asset allocation. This explains the poor performance of the CVaR portfolio. Based on the portfolio value, MVO, Robust MVO, and RP portfolios have a similar performance during the recession period (2008 - 2009), while RP and Robust MVO significantly outperforms the other two models during normal years. The portfolio using MVO and OLS results in the best Sharpe ratio of 0.1717, average turnover of 0.4874, and highest model score. The second best portfolio is using Robust MVO and BSS, with a Sharpe ratio of 0.1673, and an average turnover rate of 0.3335. However, the elapsed time of this algorithm is around 3 times greater than the one using MVO and OLS.

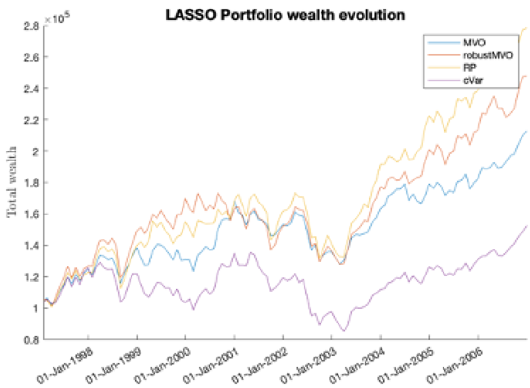
4.2 Performance Comparison: Dataset 2



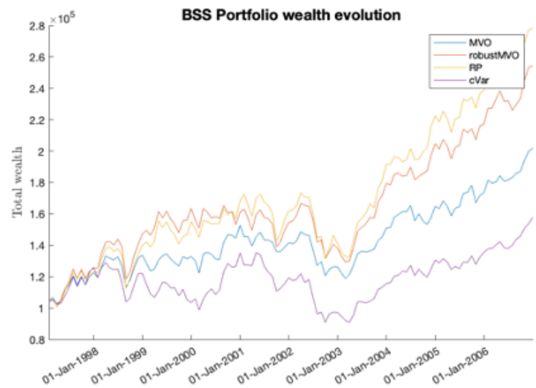
((a)) OLS Portfolio



((b)) FF Portfolio



((c)) LASSO Portfolio



((d)) BSS Portfolio

Figure 9: Compare four optimization strategies for each factor model in testing

Parameter	Sharpe ratio				Avg. turnover				Elapsed time			
Optimization	MVO	Rob. MVO	CVaR	RP	MVO	Rob. MVO	CVaR	RP	MVO	Rob. MVO	CVaR	RP
OLS	0.0810	0.1036	0.0127	0.1259	0.6486	0.2707	0.8019	0.0876	1.5800	3.9571	1.7447	7.6151
FF	0.0855	0.1019	0.0087	0.1261	0.5630	0.2856	0.7979	0.0881	1.0545	3.3504	1.5915	7.9141
LASSO	0.0876	0.1034	0.0069	0.1261	0.5422	0.2864	0.8039	0.0881	4.0304	6.7277	4.5998	11.2448
BSS	0.0727	0.1127	0.0142	0.1261	0.6729	0.2075	0.8542	0.0882	2.1062	4.6878	2.6139	8.3771

Table 3: Dataset 2: Performance Metrics for Portfolios

Dataset 2 is from December 31, 1991, to December 31, 2006, with 3 years horizon. In this case, we observed a superior performance of RP and Robust MVO strategy under four different factor models, with the Sharpe ratio around 0.126 and 0.1 respectively. However, the elapsed time to run RP is twice that of the Robust MVO. The investor needs to balance the computational time and the model performance when dealing with a large database. Similar to the result in training, CVaR has the lowest Sharpe ratio and the highest average turnover rate in the testing period, which under-performs the other optimization algorithms in both bull and bear markets. Moreover, by investigating the portfolio value plot, for either factor model, the gap among MVO, Robust MVO, and RP shrank during the recession and is wider when the market rebounded. The performance of Robust MVO and RP is closer and better than the others because they reduced the model risk and uncertainty in their own way. Robust MVO introduces an uncertainty set around the estimated return to absorb the model error and stabilized the optimization system, while RP removes the estimated return and only relies on the correlation matrix.

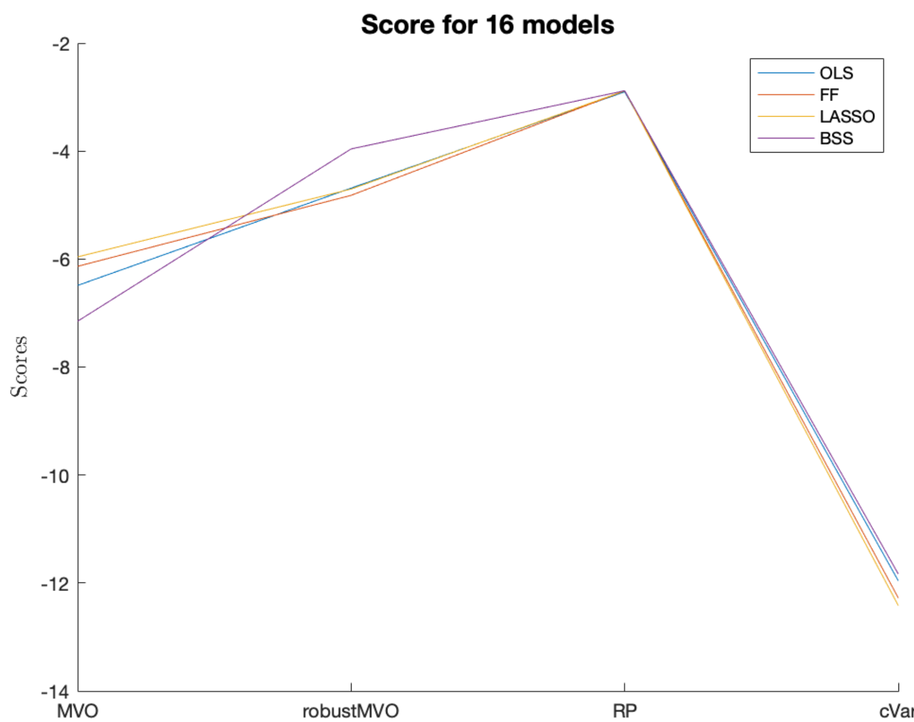
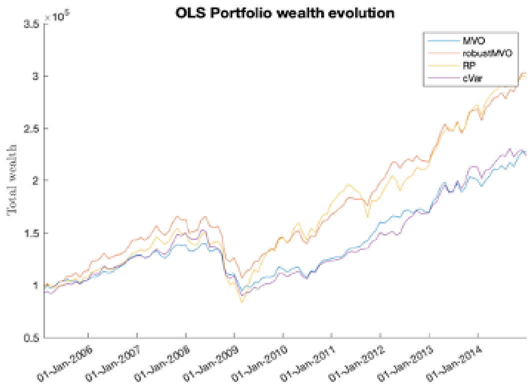


Figure 10: Dataset 2: Model Score

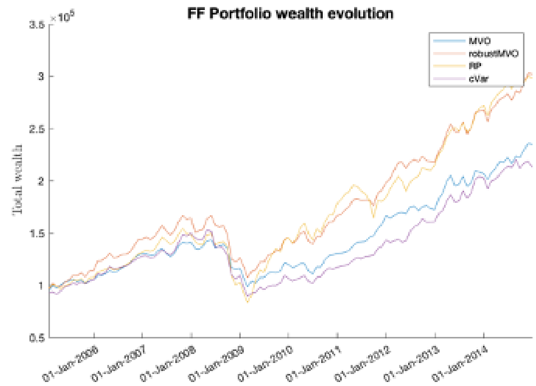
Additionally, since the dataset contains more assets, the elapsed time increase compared to the training stage. We observe the system using LASSO and RP requires 11.24 seconds, which is the highest. The running time for RP doubled when the number of assets increase from 20 to 30, which is higher compared to the other optimization algorithms.

Based on the plot for the model score, we can see RP has the highest score over all factor models, and the strategy combined with the Robust MVO and BSS has a high score similar to the score of RP. Since the elapsed time for Robust MVO and BSS is 4.68, which is much lower than any of the algorithms using RP. If considering runtime, we would prefer the Robust MVO and BSS algorithm at this stage. If investors aim to achieve a greater portfolio value, especially during the normal years, the system using RP and FF will be a reasonable choice. It can provide the highest Sharpe ratio 0.1261, and a relatively low average turnover (0.0881) and system runtime (7.9 seconds).

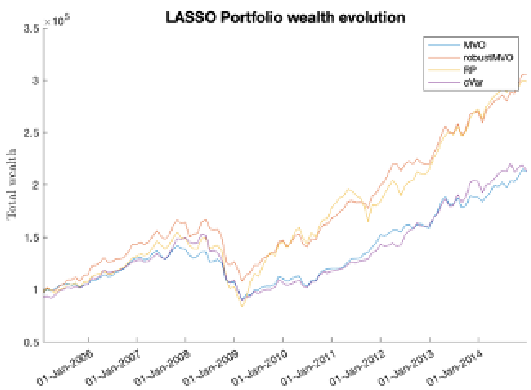
4.3 Performance Comparison: Dataset 3



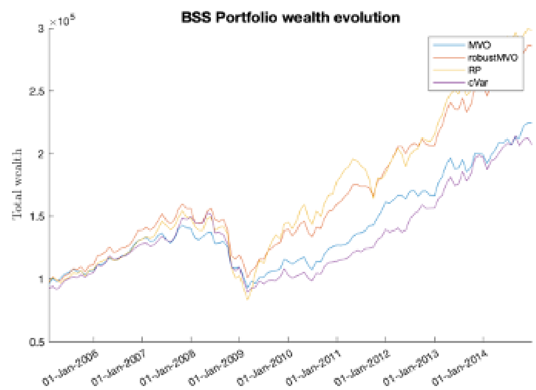
((a)) OLS Portfolio



((b)) FF Portfolio



((c)) LASSO Portfolio



((d)) BSS Portfolio

Figure 11: Compare four optimization strategies for each factor model in validation

Parameter	Sharpe ratio				Avg. turnover				Elapsed time			
Optimization	MVO	Rob. MVO	CVaR	RP	MVO	Rob. MVO	CVaR	RP	MVO	Rob. MVO	CVaR	RP
OLS	0.1928	0.2352	0.1805	0.1818	0.5500	0.3088	0.8317	0.1375	1.0882	3.6670	1.6449	7.0298
FF	0.2029	0.2348	0.1689	0.1815	0.5044	0.3204	0.8922	0.1376	1.1266	3.8989	1.7430	7.0179
LASSO	0.1732	0.2363	0.1686	0.1817	0.4477	0.3204	0.8861	0.1374	0.4477	0.3204	0.8861	0.1374
BSS	0.1884	0.2246	0.1605	0.1813	0.6016	0.2880	0.8794	0.1376	0.6016	0.2880	0.8794	0.1376

Table 4: Dataset 3: Performance Metrics for Portfolios

Here, the data is from December 31, 1999, to December 31, 2014, and the investment horizon is 3 years. Based on the analysis for previous datasets, regardless of the selection of factor models, the Robust MVO and RP strategies perform on par but obviously outperform the other two strategies. For dataset 3, the portfolio wealth for Robust MVO and RP are similar and at a high level, whereas the portfolio wealth for MVO and CVaR are close and lower in value. During the recession year (2000, and 2008), the portfolio value under different trading strategies is close. However, different from the training and testing, we can see the Robust MVO significantly outperforms RP under all factor models based on the figure and table. Although RP still has the lowest turnover rate, the Sharpe ratio of Robust MVO is approximately 0.05 higher than that of RP, while maintaining an acceptable turnover rate. The plot of the model score shows the score for Robust MVO combined with either of the four factor models is higher than the score for RP. Additionally, we found the Sharpe ratio is the highest (0.2363) when Robust MVO is paired with LASSO, and the turnover rate is the lowest (0.288) when Robust MVO is paired with BSS.

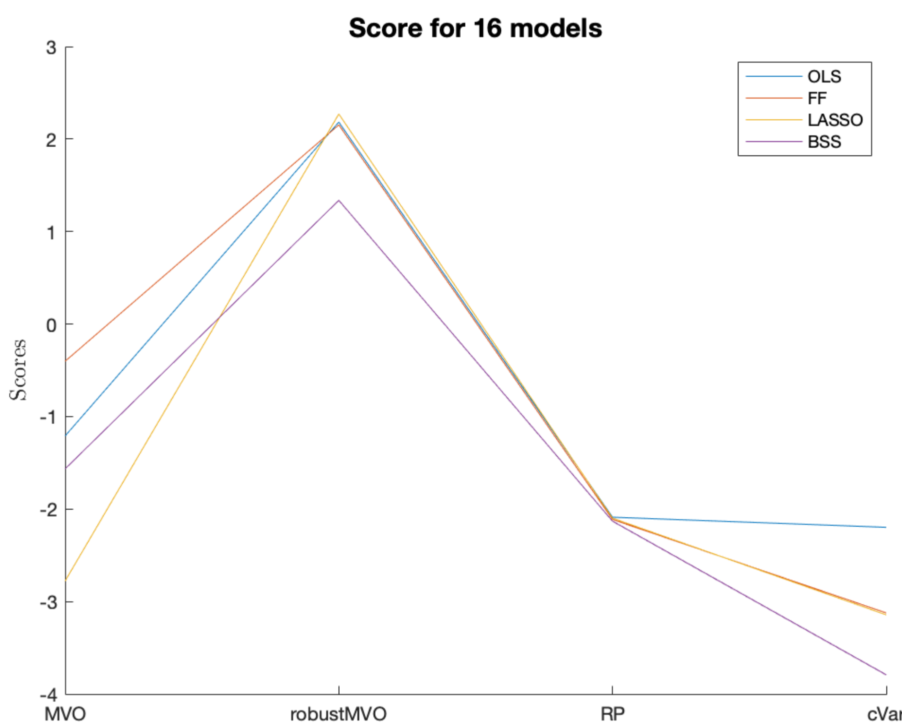


Figure 12: Validation: Model Score

4.4 Finalized Asset Allocation Strategy

Based on the average performance of the 16 different strategies, combining the BSS factor model and the Robust MVO optimization strategy results in the best portfolio under bear and bull market conditions. This section includes the analysis of the model selection and a detailed explanation of the final asset allocation strategy.

4.4.1 Compare Average Performance among All Models

Parameter	**Sharpe ratio**				**Avg. turnover**			
Optimization	MVO	Rob. MVO	CVaR	RP	MVO	Rob. MVO	CVaR	RP
OLS	0.1485	0.1669	0.0995	0.1577	0.5620	0.3174	0.8722	0.1140
FF	0.1468	0.1660	0.0943	0.1576	0.5045	0.3307	0.8911	0.1142
LASSO	0.1394	0.1675	0.0936	0.1577	0.4622	0.3329	0.8910	0.1141
BSS	0.1433	0.1682	0.0934	0.1576	0.6005	0.2764	0.9056	0.1142

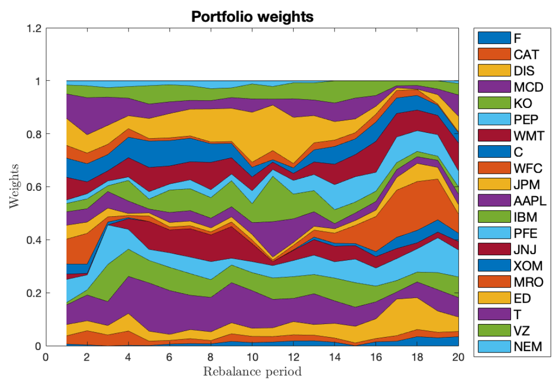
Table 5: Average Performance: Sharpe Ratio and Average Turnover

Parameter	**Elapsed time**				**Model socore**			
Optimization	MVO	Rob. MVO	CVaR	RP	MVO	Rob. MVO	CVaR	RP
OLS	1.1879	3.3879	1.5764	6.2162	1.1606	2.6761	-0.4497	1.8937
FF	1.0074	3.2614	1.5690	6.1846	1.0260	2.5866	-0.8461	1.8909
LASSO	3.5751	6.1019	4.2468	9.1516	0.4335	2.7051	-0.9074	1.8967
BSS	1.8940	4.5422	2.4914	7.3884	0.7408	**2.7468**	-0.9213	1.8857

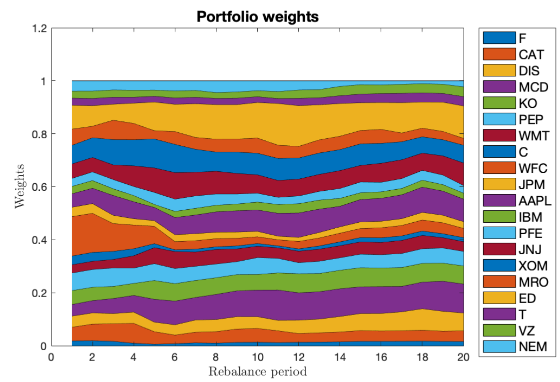
Table 6: Average Performance: Elapsed Time and Model Score

Based on the table, our final system use the BSS factor model and Robust MVO, and it has the highest model score 2.7468. The CVaR algorithm has the poorest performance as its performance is highly sensitive to the dataset. Although the average turnover for Robust MVO is higher than RP under different factor models, the average Sharpe ratio over training, testing, and validation is around 0.166 which is higher than that of RP. Considering the computational efficiency, the asset manager can save half of the time running the program when processing large databases if they use Robust MVO.

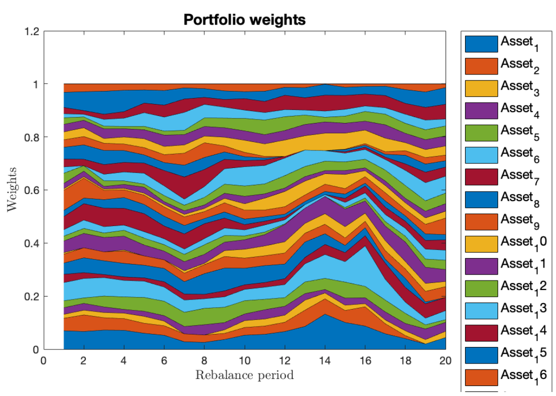
4.4.2 Asset Allocation Weights for Final Model



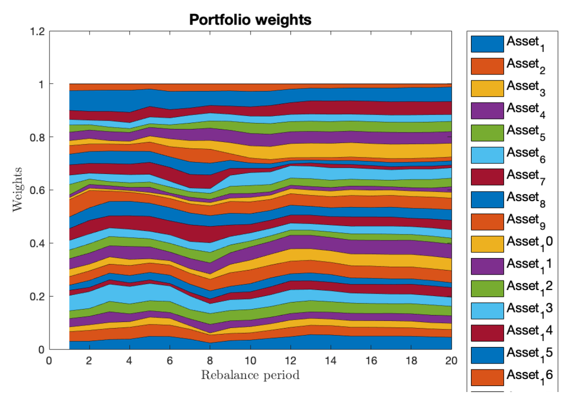
((a)) Dataset 1: 3 years horizon



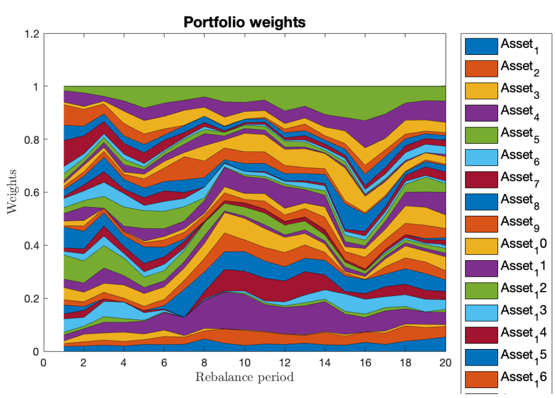
((b)) Dataset 1: 5 years horizon



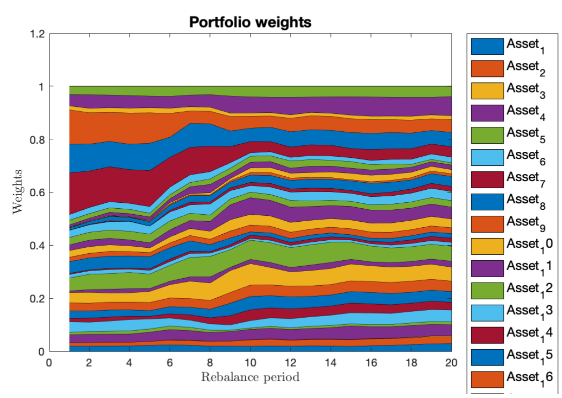
((c)) Dataset 2: 3 years horizon



((d)) Dataset 2: 5 years horizon



((e)) Dataset 3: 3 years horizon



((f)) Dataset 3: 5 years horizon

Figure 13: Asset Allocation for BSS-Robust MVO trading strategy

The final version of the automatic asset management system applied the BSS factor model and Robust Mean-variance optimization. We applied this strategy on both 3 years and 5 years investment horizons, with a 6-month balancing frequency. The following plot shows the portfolio asset allocation detail on the three datasets. As the time horizon increase, the portfolio weights become less volatile. In response to economic recessions and economic booms, our system can adjust asset allocation promptly. We can observe a clear inflection point in the graph. When a shorter term is chosen, the system's allocation of assets will be more volatile. This also means larger position adjustments and higher transaction costs. For a longer investment horizon, the magnitude of the asset adjustment will be relatively smaller. This is because the factor model used is more suitable for long-term value investing rather than short-term speculation. Figure (c.)-(f.) indicates a better risk diversification compared to Figure (a.)-(b.), as the dataset 1 only contain 20 assets, but the dataset 2 and 3 have around 30 assets to select.

5 Conclusion and Discussion

Investors measure an asset's risk and then compare it to the asset's expected return. The goal of the mean-variance optimization (MVO) is to maximize the return on investment in terms of risk. The traditional MVO strategy has model errors introduced by input variables – expected return and covariance matrix, predicted by factor models. One way to improve the asset allocation system is to select a factor model with the highest prediction accuracy. The four factor models are OLS, FF, LASSO, and BSS. The other way is to enhance the optimization strategy, reducing the dependence of the optimization algorithm on the factor model output, such as only relying on the covariance matrix (RP) or only letting the estimated return as the model input (CVaR). We can also define an ellipsoidal uncertainty set around the estimated return, which is the major source of uncertainty. Based on the previous study in project 1, for factor model, we set $\lambda = 0.01$ for LASSO, and $K = 5$ for BSS model. Then, we tested different parameters for each optimization strategies and set $\alpha = 0.95$ for CVaR, $\kappa = 8$ for RP, and $(\alpha = 0.99, \lambda = 0.01)$ for Robust MVO, which can maximize the system performance. After the training, testing, and validation, we conclude that Robust MVO and RP outperform the other two optimization methods. However, the runtime of RP is usually double the runtime of the other algorithms. CVaR under-performs the other methods may because it only considered the estimated return with more model error. The final system used the combination of the Robust MVO and BSS model, which provides the highest sharp ratio and the lowest turnover rate for all data sets, with the highest average score of 2.7468. Asset managers need to select an appropriate asset management strategy and an efficient trading automation system to achieve the target Sharpe ratio.

References

E. Anderson, H. Xu, and D. Zhang. Confidence levels for cvar risk measures and minimax limits. doi: http://www.optimization-online.org/DB_FILE/2014/01/4212.pdf.

S. n Ceria. Incorporating estimation errors into portfolio selection: Robust portfolio construction. *Journal of Asset Management*, 7(2):109–127, 2006. doi: 10.1057.