Automated Asset Management System

Daniel Zhuang, Yara Nassar

Introduction

The purpose of this investigation is to develop an automated asset management system to determine optimal weights for a portfolio of assets. The objective is to maximize portfolio Sharpe Ratio (SR) and minimize average turnover at every rebalancing date, where the SR is deemed to be more important. There are two key states in the project. First, based on past data, we obtain estimates of the mean and covariance of each asset. These are inputs to the portfolio optimization method which is the second stage. The second stage consists of testing different optimization methods and refining them. The final algorithm combines various elements of our analysis in these stages and incorporates risk parity optimization and portfolio cardinality reduction methods.

Our broad goal is to develop an algorithm that is generalizable. The regression stage should provide accurate predictions of input parameters under different dataset characteristics, and the portfolio optimization stage should perform well in various market scenarios.

The three datasets used comprise of time series for 20, 30, 33 assets and 8 factors. Qualitatively, they cover a blend of different market regimes, including the 2008 recession and the periods of high growth before and after the recession. The final model achieved SR of 0.2190, 0.1234, and 0.1814, and an average turnover of 0.1913, 0.1952, and 0.1752.

In the report, “high performance” in general refers to a high Sharpe Ratio (SR) and low average turnover.

Background Information

For ease of reference, the following table assigns commonly used variables and their meaning, which will be used consistently in the report.

Variable Name	Variable Meaning	Variable Name	Variable Meaning
μ	Estimate of expected return of assets	Q	Covariance matrix of assets
x	Asset weights	L	Turnover limit at every rebalancing
x_i^o	Weight of asset i prior to rebalancing	n	Total number of available assets
λ	Risk-aversion parameter	R	Minimum mean portfolio return
m	Maximum number of allowed assets in portfolio	k	Ridge regression regularization parameter

Table 1: Variables and their corresponding meaning.

Risk Parity Optimization

Risk parity optimization finds asset weights that have the equal risk contributions to the overall portfolio. Since the original problem is non-convex, we utilized a convex reformulation to ensure that optimal solutions found are global. The model does not allow for shorting, and is formulated as:

$$\min_{y} \frac{1}{2} y^T Q y - C \sum_{i=1}^{n} \ln (y_i)$$

$$s.t.\ y \geq 0$$

The result is independent of C, which can be any positive real number. The portfolio weights are calculated by:

$$x_i = \frac{y_i}{\sum_{j=1}^{n} y_j}$$

$$for\ i = 1, 2, \dots, n, \quad y_i, x_i \in \mathbb{R}$$

Conditional Value-at-Risk (CVaR) Optimization

Conditional Value-at-Risk is the expected value of the losses in the distribution tail of a certain probability α. It measures the average of the extreme values which are beyond the Value-at-Risk (VaR) point, the point of minimum loss which we have a probability of $1 - \alpha$ of matching or exceeding.

CVaR is defined by the equation:

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

Where x is the vector of portfolio weights, r is the random vector of asset returns, $p(r)$ is the probability density function of the returns, $f(x,r)$ is the loss of a portfolio with weights x and returns r, and α is the probability level.

CVaR optimization aims to minimize the downside risk only by minimizing $CVaR_\alpha(x)$. Therefore, the optimization problem is:

$$\min_{x} CVaR_\alpha(x)$$

$$\text{s.t. } x \in X$$

where X is the set of constraints and $VaR_\alpha(x) = min\{\gamma \in \mathbb{R}: \Psi(x, \gamma) \geq \alpha\}$

As the loss and probability density functions are challenging to estimate given the data available, we use S historical scenarios. By allowing γ to be a placeholder auxiliary variable for VaR_α in the optimization, the problem becomes:

$$\min_{x,z,\gamma} \gamma + \frac{1}{(1-\alpha)S} \sum_{s=1}^{S} z_s$$

$$s.t. z_s \geq 0 \; for \; all \; s = 1, \ldots, S$$

$$z_s \geq -\hat{r}_s^T x - \gamma \; for \; all \; s = 1, \ldots, S$$

$$\mu^T x \geq R$$

$$1^T x \geq 1$$

Where α is the confidence level of our CVaR model. We set $\alpha = 0.95$. z is an auxiliary variable that represents the losses that exceed VaR_α. $\hat{r}_s$ are the historical returns of the assets, where we treat every scenario as equally likely.

Sharpe Ratio Optimization

The Sharpe ratio is a financial performance metric that calculates the risk-adjusted excess return of a financial instrument. Therefore, it allows for the comparison of different instruments with different risk and return levels.

The Sharpe Ratio of a portfolio calculates the excess return received for every additional unit of volatility using the formula:

$$SR_P = \frac{E[r_p] - r_f}{\sqrt{var(r_p - r_f)}} = \frac{\mu^T x - r_f}{\sqrt{x^T Q x}}$$

Sharpe ratio optimization aims to find the portfolio weights that maximize the Sharpe ratio. Therefore, the problem becomes:

$$\max_x \frac{\mu^T x - r_f}{\sqrt{x^T Q x}}$$

$$\text{s.t. } x \in X$$

where X is the set of constraints.

However, the objective function is not linear or quadratic, so the expression is transformed by introducing an auxiliary variable y. The resulting formulation is:

$$\min_{y,\kappa} y^T Q y$$

$$\text{s.t. } (\mu - r_f)^T y = 1$$

$$1^T y = \kappa$$

$$\kappa \geq 0$$

To recover the optimal portfolio weights, we use the equation:

$$x_j^* = \frac{y_j^*}{\kappa^*} = \frac{y_j^*}{\sum_{i=1}^n y_j^*}$$

Robust Sharpe Optimization

Ex Ante Sharpe optimization uses estimates of μ and Q. There is uncertainty in these estimates which can be accounted through robust optimization. The estimation of Q is generally stable, so we only consider the uncertainty in μ by creating an uncertainty set. We created an ellipsoidal uncertainty set because values in the corners of the box uncertainty set are extreme.

Ellipsoidal Uncertainty Set:

$$\mu^{true} \in \{\mu^{true} \in \mathbb{R}^n : (\mu^{true} - \mu)^T \Theta^{-1} (\mu^{true} - \mu) \leq \varepsilon_2^2\}$$

The robust optimization problem becomes:

$$\min_{y,} y^T Q y$$

$$\text{s.t. } (\mu - r_f)^T y - \varepsilon \left\| \Theta^{1/2} y \right\|_2 \geq 1$$

$$1^T y \geq 0$$

Where $\Theta = \frac{1}{T} diag(Q)$ and $\varepsilon = \sqrt{\chi_n^2(\alpha)}$ and $\chi_n^2(\alpha)$ is the chi-squared distribution with n degrees of freedom and level of confidence α.

We recover the optimal portfolio weights as in the normal Sharpe optimization.

Resampling

An alternative to introducing robustness in the Sharpe optimization problem to account for the uncertainty in μ is to conduct portfolio resampling. The algorithm involves estimating μ and Q then randomly generating T samples by assuming that the returns follow a normal distribution $r \sim \mathcal{N}(\mu, Q)$. Then, μ_2 and Q_2 are calculated from each sample and are used in the optimization of the Sharpe ratio to produce an optimal weight portfolio x. We then calculate the average weight of the optimal portfolios produced from every sample.

Asset Reduction

The large number of candidate assets and the presence of highly correlated assets can diminish the value of diversification. Moreover, due to noise in parameter estimation, the portfolio often has high turnover rate. To reduce the cardinality of the portfolio, we used a variation of mean variance tracking error reduction formulation. The method finds the portfolio with a reduced set of m assets that minimizes the portfolio deviation from an original portfolio without cardinality restriction, which weights $x_{unrestr}$.

$$\min_x x^T Q x - 2x_{unrestr}^T Q x$$

$$s.t. 1^T x = 1$$

$$1^T y \leq m$$

$$x_i \leq y_i$$

$$0 \leq x \leq 1$$

$$-\mu^T x \leq -\mu^T x_{unrestr}$$

$$1^T z \leq L$$

$$x_i - x_i^0 \leq z_i$$

$$x_i^0 - x_i \leq z_i$$

$$for\ i = 1, 2, \ldots, n, \qquad y_i \ \epsilon \ \{0, 1\}, \qquad x_i, z_i \ \epsilon \ \mathbb{R}$$

The objective function has been simplified from $(x^T - x_{unrestr}^T)Q(x^T - x_{unrestr}^T)$ by removing constant terms. To control turnover, which may be high when optimal solution alternates between two similar assets, a limit on the total turnover from the previous portfolio is imposed. The formulation ensures that the resulting portfolio does not have a lower expected return than the unrestricted portfolio. As this may lead to infeasible solutions in combination with the turnover constraint, the final algorithm includes potential adjustments to the maximum turnover when necessary.

Factor Models

Assuming that returns of our assets follow a factor model structure allows us to estimate μ and Q using linear regression. The factor model that this analysis is based on is:

$$r_i = \alpha_i + \sum_{k=1}^{p} \beta_{ik} f_k + \varepsilon_i$$

Where r_i is the return of asset i, f_k is the factor return, β_{ik} is the factor loading obtained from regression, and ε_i is the idiosyncratic residual assumed to have a mean of 0.

Estimation of Model Inputs

Model inputs μ and Q are estimated by performing regression using different methods to find the factor loadings β_{ik}. The regression models used are the same as those we developed and tested in Project 1. However, we only used the models that were successful in producing good results, so we excluded PCA.

Ordinary Least Squares (OLS) Regression

Regression method that fits a linear model to a set of data by minimizing the sum of squared errors in the fit. The linear model fit to the data is the factor model (XX). The dependent variable is the return of each asset, the factor returns are the predictors, and the factor loadings are the coefficients.

Regularization

Regularization is applied to the regression model to make it more generalizable to unseen data. It helps prevent overfitting to the training data which would produce a coefficient vector that does not predict new data very well.

To regularize we used Lasso regression which introduces the penalty term $k\|\beta\|_1$ and ridge regression which has the penalty term $k\|\beta\|_2^2$. Both methods penalize large values of β which makes the model more generalizable to unseen data. Lasso regression tends to result in a sparse β while ridge regression results in small values of β.

Weighted Least Squares (WLS) Regression

Data that has non-constant variance violates an important assumption of linear regression which would produce inaccurate results. Moreover, more recent data may be more relevant to prediction. We used weighted least squares regression to solve this issue by weighting the sum of squared errors by the term $w_i = e^{\tau*(number\ of\ observations\ in\ the\ past)}$, for $\tau < 0$, which results in weights exponentially decaying with time. The lscov function in Matlab was used to implement this regression.

Evaluation of Regression Accuracy

The same regression models as Project 1 were used, so testing for regression model accuracy was not repeated. For completeness, the following metrics for estimation error were used.

Percentage Difference

This is a measure of the percentage difference between the estimated and the actual mean and covariance calculated using the formula, using absolute value. Q_i denotes an entry on the covariance matrix **Q**.

$$d_i^1 = \left|\frac{\mu_{i,actual} - \mu_{i,estimated}}{\mu_{i,actual}}\right|$$

$$d_i^2 = \left|\frac{Q_{i,actual} - Q_{i,estimated}}{Q_{i,actual}}\right|$$

The maximum difference among all assets and the average difference are then calculated.

$$d_{max}^1 = \max(d_i^1)$$

$$d_{max}^2 = \max(d_i^2)$$

$$d_{avg}^1 = \frac{\sum_{i=1}^{n} d_i^1}{n}$$

$$d_{avg}^2 = \frac{\sum_{i=1}^{n^2} d_i^2}{n}$$

Root-mean-square (RMS)

We also used the root-mean-square (RMS) measure to evaluate the error between predicted and actual return. The predicted return was calculated based on the actual values factor returns and estimations of factor loadings. The square difference was averaged across all observations (N) and for all n assets.

$$RMS = \sqrt{\frac{\sum_{i=1}^{nN}\left(r_{it,predicted} - r_{it,actual}\right)^2}{nN}}$$

Model Development

Splitting Data

Three separate datasets were used to test and validate the models. We divided the dataset into a training and testing set, whenever a regression algorithm or portfolio optimization model was tested. Similar to Project 1, for the estimation of parameters such as regression coefficients for a factor models and regression algorithm parameter, the training data was used. The test data was the most recent 6 months and training data in sliding window of 1, 2, and 3 years and from the start of available data, preceding the 6 months were investigated. This ensured that the training data was separate from testing data.

When the portfolio optimization model was tested for its Sharpe Ratio and turnover, the training data was tested among the most immediate observations in 1-, 2-, and 3-years sliding window and from the start of available data. The testing data was the upcoming six months of data, during which no rebalancing can occur.

Summary of Regression Algorithm (Project 1)

The algorithms that had good testing results in Project 1 were used to estimate μ and Q values. These algorithms are OLS, lasso regression, ridge regression, and weighted least squares regression. It was previously determined that out of the methods tested, for mean estimations, errors were lowest when using ridge regression (k = 10 – 100) and WLS (τ = (-2) – (-1.5)). For covariance estimations, ridge regression with k = 100-1000 performed best. This suggested that different optimization parameters could be used for estimations of $\boldsymbol{\mu}$ and $\mathbf{Q}$, which was investigated later in the model development process. Based on these results, a final regression algorithm that estimates an appropriate regression parameter at every rebalancing was developed in Project 1. A detailed summary of the algorithm can be found in the Final Model Algorithm section.

Pairwise Testing of Combinations of Regression and Portfolio Optimization Algorithms

While the developed regression algorithm was used final estimation of factor loadings, other regression methods were used to obtain a detailed assessment of the viability of each portfolio optimization model.

The regression algorithms were paired with the Sharpe Ratio, CVaR, and risk parity optimization models. The resulting Sharpe ratio and turnover ratio were recorded in Table 2. The regression hyperparameters that were found to produce the best testing accuracy in Project 1 were used. For completeness, we included results from modifications to the SR optimization, robustification under an ellipsoidal uncertainty set and resampling, which occurred in the next step of the analysis. Further discussion these improvements will follow in the next section.

Optimization Method	λ	Regression Method	Hyperparameter	Sharpe Ratio – Dataset 1	Sharpe Ratio – Dataset 2	Sharpe Ratio – Dataset 3	Average Turnover – Dataset 1	Average Turnover – Dataset 2	Average Turnover – Dataset 3
Sharpe Optimization	N/A	OLS	N/A	0.1625	0.0419	0.2209	0.7531	1.0105	0.7353
		Ridge	$k = 0.15$	0.1621	0.0410	0.2211	0.7517	1.0113	0.7347
		Lasso	Optimized in Model	0.0668	-0.0062	0.1122	0.9478	0.9983	0.9775
		Weighted LS	$\tau = -0.5$	0.1869	0.1426	0.1898	1.2686	1.3637	1.2111
Robust Sharpe Optimization	0.001	OLS	N/A	0.1842	0.0877	0.1964	0.3659	0.4565	0.5392
		Ridge	$k = 0.15$	0.1846	0.0870	0.2065	0.3457	0.4614	0.5111
		Lasso	Optimized in model	0.1557	0.1027	0.1677	0.2587	0.3483	0.3706
		Weighted LS	$\tau = -0.5$	0.2254	0.1231	0.1899	0.8637	1.0714	0.9408
	0.01	LS	N/A	0.1718	0.0979	0.2371	0.3946	0.3435	0.4611
		Weighted LS	$\tau = -0.5$	0.1836	0.0646	0.2156	1.4164	1.9410	1.7096
	0.1	OLS	N/A	0.1662	0.0993	0.2366	0.4209	0.3735	0.5042
Sharpe Optimization + Resampling	N/A	OLS	N/A	0.1581	0.0909	0.2111	0.5312	0.6603	0.5819
		Ridge	$k = 0.15$	0.1526	0.0845	0.2011	0.5614	0.6386	0.5572
CVaR	N/A	N/A	N/A	0.1721	0.0465	0.1920	0.8710	0.9724	0.8463
Risk Parity	N/A	OLS	N/A	0.1698	0.1282	0.1772	0.1541	0.0891	0.1932
		Ridge	Optimized in model	0.1721	0.1231	0.1819	0.1131	0.1184	0.1371

Table 2: Table of the Sharpe Ratio and Turnover Rate for different pairs of algorithms and for each of the three datasets.

While no method clearly outperformed other methods across all datasets, testing revealed the relative performance of some methods over others. The combination that produced the best average Sharpe Ratio across all datasets is performing Sharpe optimization with Weighted LS regression, $SR_{avg} = 0.1731$. However, this combination also had a high average turnover rate across the three trials (1.2795). The combination that produced the best balance of Sharpe ratio and turnover rate is risk parity and with ridge regression, $SR_{avg} = 0.1590$ and average turnover of 0.1271. CVaR optimization had high turnover with a lower SR compared to other methods.

Model Refinement

Some key observation were made in the initial testing of the optimization algorithms.

1. **Inconsistent performance across different datasets:** The relative performance of optimization methods was not consistent. For example, the risk parity portfolio had lower SR than CVaR for dataset 1 but had significantly higher SR for dataset 2.
2. **High turnover rate:** The CVaR and optimal SR portfolios had high turnover rate. The turnover rate of the risk parity portfolio was higher when the number of candidate assets grew (i.e., dataset 3).
3. **Uncertainty in mean estimates:** The SR portfolio is dependent on the estimate of the mean return, which has a high degree of estimation error that was observed in testing.

Robust optimization:

Two methods were used to introduce robustification to optimization, taking into consideration the uncertainty in the estimation of μ. The first was using an ellipsoidal uncertainty set. This produced a significantly improved Sharpe Ratio for dataset 1 across all regression methods, and a better turnover rate for all data sets. We found that taking into consideration uncertainty in μ produced more stable portfolios across all datasets. Similarly, improvement in the turnover rate due to robustification was observed in Project 1 when applied with MVO. In this project, we found that applying robustification with Sharpe optimization results in a similar improvement. The results are summarized in Table 1.

The second method of robustification is resampling asset return data from the normal distribution based on estimated μ and Q. Initial estimates of μ and Q found using OLS regression and ridge regression. Using these estimates, 100 samples from the return distribution $\mathcal{N}(\mu, Q)$ were obtained, and portfolio weights were obtained from SR optimization. This was repeated 50 times, and the average portfolio weight was taken as the final weights. This method also reduced the turnover rate across all datasets from [0.7531 1.0105 0.7353] for Sharpe Ratio optimization with OLS to [0.5312 0.6603 0.5819]. However, the reduction in turnover was not as large as the variation with the ellipsoidal uncertainty set, and also resulted in an undesirable reduction in SR.

Asset Reduction

The issues observed in the risk parity portfolio, low average SR, and a growing turnover rate for portfolios with larger cardinality, were addressed by reducing the number of assets through the mean variance tracking error. We investigated the effect of reducing the number of assets to $n - 10$ and $\left\lceil \frac{n}{2} \right\rceil$. It was found that the SR improved significantly to be greater than 0.2 for datasets and 1 and 3 when cardinality was restricted at $n - 10$. Reducing to $\left\lceil \frac{n}{2} \right\rceil$ was deemed to be too excessive – the SR only improved slightly to around 0.19. Moreover, the runtime for dataset 2, which had 30 assets, was too long, indicating this reduction to be too computationally intensive when the initial cardinality was high.

A consequence of reducing cardinality was a substantial increase of average turnover to over 0.7 for all three datasets. Analysis of the portfolio weights indicated the cause to be the optimization algorithm selecting between highly correlated assets at each rebalancing. Therefore, we imposed a constraint on the maximum allowable turnover. Pairing this constraint with the constraint that the resulting portfolio must have a greater expected return than the unrestricted portfolio could render the problem infeasible. While this did not occur for the three datasets in any of the rebalancing moments, the algorithm is designed to progressively raise the turnover threshold.

	Sharpe Ratio (SR) Dataset			Average Turnover Dataset		
Maximum Turnover (L)/Dataset	1	2	3	1	2	3
0.15	0.17425	0.1184	0.1825	0.1258	0.1486	0.1406
0.20	0.2316	0.1233	0.1815	0.1902	0.1966	0.1994
0.25	0.2164	0.1193	0.2012	0.2383	0.2325	0.2467
0.30	0.22018	0.1150	0.18933	0.24141	0.26895	0.2132
Unrestricted	0.2330	0.1094	0.1884	0.2651	0.3839	0.2326

Table3: Risk parity portfolio performance with different restrictions on turnover and reducing the number of allowable assets by 10.

Portfolio performance peaked, across the three datasets, by taking an initial value of L of 0.20-0.25 (Table 3). The testing revealed that turnover rate could be controlled with little impact on SR. Allowing for a higher turnover did not improve portfolio SR.

Changes to Testing Data Length

In Project 1, all available data was used to estimate an appropriate regression algorithm parameter (e.g., k for ridge regression). Once this was determined, all available data was used to estimate the factors loadings for the estimation of Q and μ. This was justified as ridge and WLS regression had lower prediction error when all data was used compared to sliding window. However, we hypothesized that selecting a shorter training period for algorithm parameter estimation could improve the estimation accuracy, as this would account for changing characteristics of the data over time that impacts the performance of regularization. Moreover, improved performance of a regularization algorithm, when measured by the errors from its estimations, does not necessarily translate to improved performance of the overall portfolio. However, as errors were lowest when more data was used, factor loadings would continue to be estimated by all available data. We tested this by varying the length of data used to predict the factor loadings and regression parameters, while keeping the portfolio optimization model fixed.

		Sharpe Ratio (SR)			Average Turnover		
Method	**Dataset**	1	2	3	1	2	3
Project 1 Regression Algorithm (Optimized Ridge and WLS)	Regression parameter: 3 years Factor Loading: Start of data	0.2188	0.1120	0.1814	0.1734	0.1883	0.1704
	Regression parameter: 3 years Factor Loading: 3 years	0.1801	0.1196	0.1772	0.1913	0.1952	0.1752
	Regression parameter: Start of data Factor Loading: Start of data	0.2081	0.1120	0.1761	0.1634	0.1871	0.1705

Optimized Ridge Regression	Regression parameter: 3 years Factor Loading: Start of data	0.2190	0.1233	0.1814	0.1734	0.1876	0.1704
	Regression parameter: 3 years Factor Loading: 3 years	0.1810	0.1152	0.1772	0.1634	0.1876	0.1704
	Regression parameter: Start of data Factor Loading: Start of data	0.2082	0.1080	0.1725	0.1497	0.1853	0.1743

Table 4: Average portfolio performance for the three datasets, using different time durations for the estimation of regression parameters and factor loading. The tests were conducted for the final regression algorithm in Project 1 and ridge regression, optimizing for parameter k. Portfolio weights were determined using the risk parity and reduced asset algorithm ($L = 0.2$). The best results were consistently obtained using ridge regression, using 3 years of data to estimate regression parameter and all available data to estimate factor loadings.

For the three datasets, the best performance was achieved when all available data was used to estimate the factor loadings, but when a sliding window of 3 years was used to determine the regression parameter. Since errors were lowest for three years compared to other durations of sliding window, it was selected as the duration in the final model.

Moreover, the testing revealed that using ridge regression with separate estimations of k performed better than the mix of WLS and ridge (Table 4), and had improved performance compared to fixing the value of k. Notably, the algorithm was superior to using ordinary least squares regression, which had a Sharpe Ratio of 0.1484, 3.6364, and 0.1810. Therefore, ridge regression was selected as the method for regression. A detailed description of the algorithm can be found in the Final Model Algorithm section.

Final Model Results and Selection

The best results for the Robust SR optimization portfolio with an ellipsoidal uncertainty set were for the one with $\lambda = 0.01$ and OLS regression. The SR for this combination is [0.1718 0.0979 0.2371] and the average turnover rate is [0.3946 0.3435 0.4611]. The SR optimization portfolio with an ellipsoidal uncertainty set, using WLS regression ($\lambda = 0.001$), also had high SR [0.2254 0.1231 0.1899], but the turnover was particularly high (>0.85) for all tests.

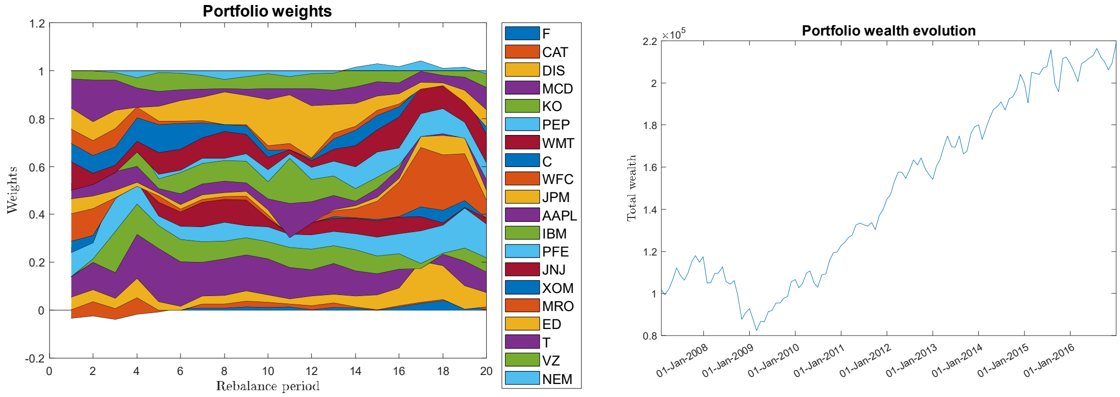


Figure 1: Portfolio weights and wealth for the Sharpe Optimization portfolio with ellipsoidal uncertainty set and OLS regression.

The best results for SR optimization with resampling is the one with OLS regression. The SR for this combination is [0.1581 0.0909 0.2111] and the average turnover rate is [0.5312 0.6603 0.5819]. The runtime for each dataset was 38s, 40s, and 44s.

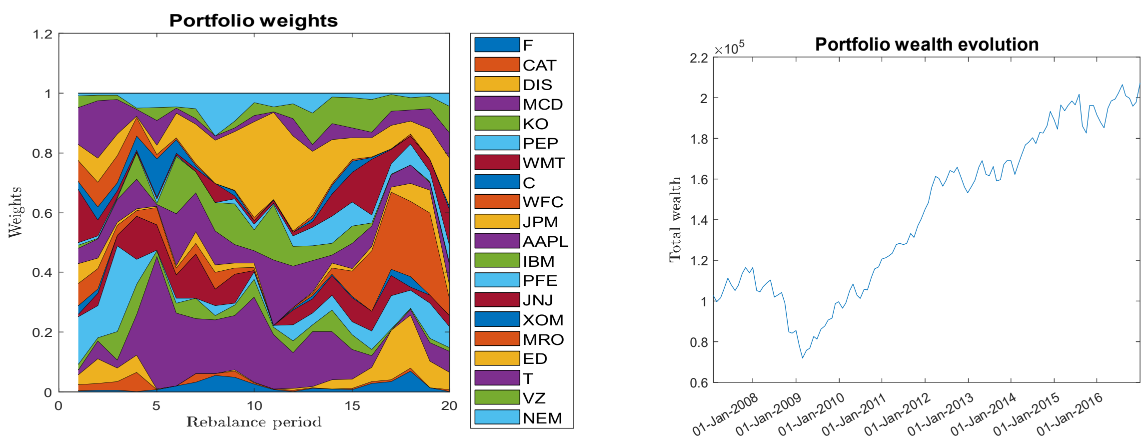


Figure 2: Portfolio weights and wealth for the Sharpe Optimization portfolio with resampling and OLS regression.

The CVaR portfolio produced a SR of [0.1721 0.0465 0.1920] and a turnover rate of [0.8710 0.9724 0.8463]. The runtime for each dataset was [1.1028 0.2899 0.2708].

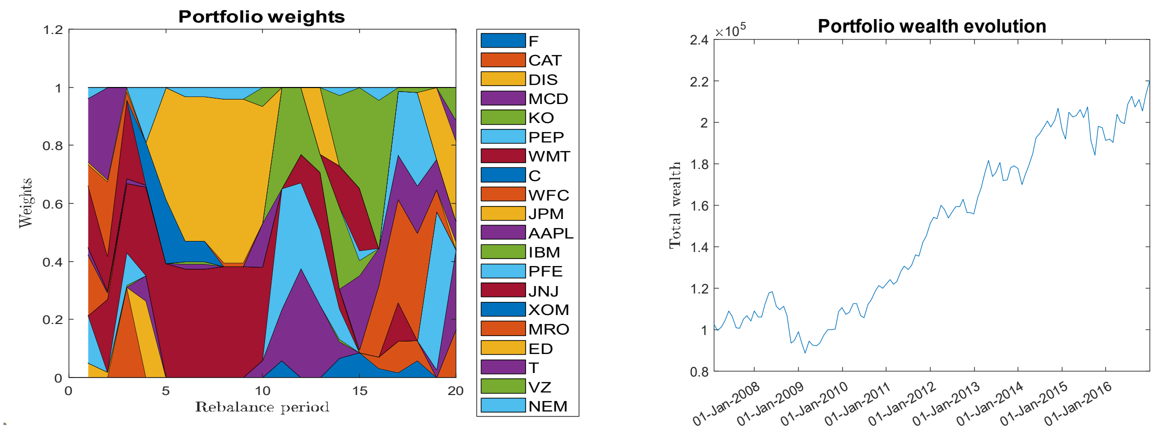
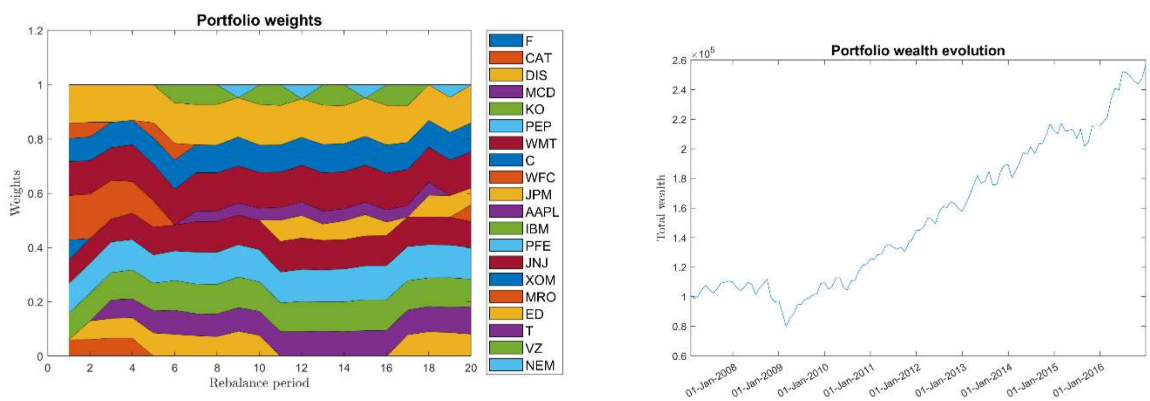


Figure 3: Portfolio weights and wealth for the CVaR optimization portfolio.

The risk parity portfolio, with the number of assets reduced by 10, achieved the best results overall with $L = 0.20$. The SR was [0.2190 0.1234 0.1814] and the average turnover was [0.1733 0.1876 and 0.1704] for the three datasets. The runtime each dataset was 20s, 145s, and 349s.



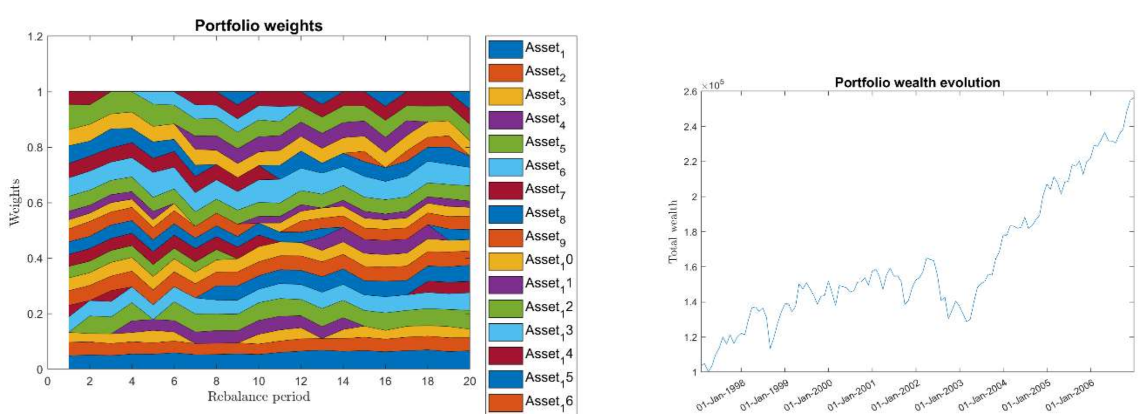


Figure 4: Portfolio weights and wealth for the enhanced risk parity portfolio. First and second rows depict dataset 1 and 2, respectively. Notably, the drawdown during the 2008 crisis did not go below 80%, and the portfolio further captured the high rate of growth in the years 2003 – 2007 and post 2011.

Given the consistent performance across all datasets and the stable behaviour of the results, the risk parity model with cardinality reduction was selected as the final model.

Final Model Algorithm

At every rebalancing, the following stages are implemented. The section on the estimation of regression parameters is adapted from Project 1.

Estimation of Regression Parameters

1. Denote observations occurring within 6 months (inclusive) to be testing data, and observations prior to that as training data. Specify the length of training data be from the time at which the data is available up to 6 months prior (exclusive, so that it does not overlap with testing data).
2. Estimate the optimal method and parameter (k for ridge) for $\boldsymbol{\mu}$ estimation.
    a. Find parameter and method that minimizes RMS using Matlab's fminbnd parameter. The search range for k = [0.5, 150]. RMS metric is calculated by the root-mean-square of the absolute value of the difference between the actual return of the asset and the returns determined by the factor loadings and the actual factor returns, across the test period.
3. Estimate the parameter (k in ridge regression) for **Q** estimation.
    a. Find k that minimizes RMS using Matlab's fminbnd parameter. The search range for k = [0.5, 150].

Estimation of Factor Loadings

Use the parameter determined previously to estimate the factor loadings to estimate $\boldsymbol{\mu}$ and **Q** separately. This will be our best estimate of these two values.

Risk Parity Optimization

The weights for the risk parity portfolio are determined using convex reformulation of the risk parity problem, through the fmincon function.

Portfolio Cardinality Reduction

We impose a cardinality constraint of $\max(10, n - 10)$, with the floor to ensure some diversification in the portfolio. Gurobi is used to solve the mean variance tracking error minimization problem. The algorithm further imposes a turnover limit $L = 0.2$. Should the constraints render the problem infeasible, the limit will be raised by increments of 0.05, up to a maximum of $L = 0.4$. At that point, the turnover will be deemed too high, and the same weights from the previous rebalancing will be used.

Discussion

Different regression coefficients were used to estimate μ and Q, with higher k coefficients used for Q. Since Q is generally more stable over time, the increased regularization could be helpful for raising the out of sample predictive ability of the regression model. In contrast, means estimations are more unstable, so retaining more of the information in regression led to lower prediction error. While most required as an input to the risk parity portfolio, the estimation was used in other models, such as SR optimization, so increasing prediction accuracy for all parameters was important so models can be compared at a more equal footing. We highlight our finding that using sliding window and all available data to calibrate regression parameters and estimate factor loadings, respectively, improved portfolio performance. Changes in data characteristics over time makes different levels of regularization more effective, and this could be assessed with the sliding window approach. However, for the final prediction of Q, a stable parameter, using as much data as available helps to increase prediction accuracy.

Sharpe Ratio optimization directly addressed the objective of determining the portfolio weights that maximize the SR (i.e., finding the portfolio that maximizes the excess returns and minimizes the accompanied volatility). However, this volatility is measured by the standard deviation of the returns which may understate tail risk. Additionally, minimizing variance causes the portfolio to be averse to desired deviations from the mean. Moreover, as expected returns greatly fluctuate over time, the method tends to result in higher turnover as it seeks to capture the "optimal" assets for the period. This phenomenon is compounded by the noise in the predicted values. As a result, a high turnover rate was observed for all three datasets.

To address this issue, introducing robustness through ellipsoidal uncertainty set and resampling reduced the turnover rate for the Sharpe optimization. However, it is still much higher than the risk parity method. Also, the portfolio weights are less stable over time, as seen in Figures 1 and 2, likely due to factors including the instability of the mean estimates and the need to frequently change the composition due to different market conditions that emerged in the test dataset. However, since the risk parity method is based on asset risk, a more stable value, the result is a more stable portfolio.

CVaR optimization minimizes the downside risk of the portfolio by minimizing the expected value of losses greater than or equal to VaR. This is a better approach than minimizing the standard deviation as in Sharpe Optimization since it does not make the portfolio disincline towards positive deviations away from the mean. However, the traditional implementation did not minimize the turnover rate, so it results in a high turnover rate across all datasets. Moreover, the loss function was estimated using historical returns and the probability density function was assumed to be uniform. This formulation did not fully capture the behaviour of the assets considered, which potentially led to its weaker performance.

The basis of risk parity optimization is to evenly distribute risk contributions among the assets in the portfolio. The resulting portfolio can be more resistant to large perturbations to portfolio value. The risk exposure is greater than the traditional mean variance portfolio but less than the equally weighted portfolio. It is appropriate for both bull and bear market environments, so long as the estimation of Q factors in ongoing changes asset covariances. Our error testing in Project 1 showed large deviations from the estimated mean return and realized values. Risk parity optimization relies on estimations of asset covariances, which are relatively more stable, and does not require any predictions of returns. A disadvantage that it does not allow the portfolio to shift weights to take advantage of large returns predicted in some assets. While other models, such as Sharpe Ratio optimization, enables this, they had lower SR, likely due to the low reliability of mean estimations. Moreover, due to the stability of covariances over time, a lower turnover rate was observed for both the original and improved risk parity formulation compared to SR optimization, even with robustification. The portfolio composition can be seen in Figure 4, which is more stable than the other methods in Figures 1 and 2.

As the number of candidate assets may be large, we explored methods to reduce the number of assets while retaining features and distributions of risk in the original optimization solution. The reduction of portfolio cardinality can be viewed as a form of regularization in which we simplified the portfolio presented by traditional risk parity. The first round of optimization ensured balanced risk distributions, while the second round represented that distribution with fewer assets that continued to capture the systematic risk profiles of the market. As Figure 3 illustrates, this strategy was effective for both bear (2008 recession) and bull (pre and post recession periods) environments. Interestingly, we found that introducing a reasonable lower limit on turnover of 0.20 – 0.25 improved SR in many cases. While this behaviour should be further investigated, it is believed that the benefits come from reducing overfitting in the context of portfolio weights.

Limitations

Several limitations of the analysis and model must be mentioned.

- **Noise in parameter estimation:** While our final model does not require estimations of the mean, it remains dependent on the estimated value of Q. The covariance relationships between assets cannot be determined with certainty, although we have selected the regression method with the smallest out-of-sample error and optimized for the regularization parameter.
- **Selection of optimization method:** During the analysis, the risk parity model was selected as it consistently had the highest SR and low turnover. However, we cannot make any conclusions on the efficacy of the optimization method itself, as the performance of other models was degraded by their dependence on the estimated mean return which had high uncertainty.
- **Generalizability to different datasets and assets:** The model was validated using three datasets, each containing time periods of market growth and downturn. While the risk parity methodology and the robust optimization techniques tested help increase the stability of the portfolios in a variety of market environments, results will differ depending on the dataset used. For example, if the original assets were all highly uncorrelated, the method of reducing cardinality may be less effective, as there may be increased value with holding as many assets as possible.

Next Steps

The noted limitations and our analysis suggest possible areas to improve the algorithm.

- **Blended portfolio optimization:** The goal of the current project was to develop a portfolio optimization method that performs well on different market environments. Our approach was two -pronged. First, we improved the accuracy of parameter estimation, considering the changing features of data, to increase the reliability of model inputs. Second, enhancing risk parity optimization, by constructing a portfolio that had equal allocation of risk and regularized for the assets as well. We believe performance could be further improved by a portfolio determined by a gradient of the different methods investigated in this study, which could be explicitly adjusted for depending on the market regime. For example, during periods of acute market growth, optimizing for SR directly may allow the portfolio to better capture the returns, while the risk parity portfolio could be better served during a market downturn. This type of analysis requires explicit focus on the analysis of market signals, such as macroeconomic variables [1]. Controlling portfolio turnover when the desired optimization method shifts, as noted in our analysis, would be a challenge. The use of a graduated limit on turnover, similar to our final model, could be used to smoothen out turnover.

- **Dynamic determination of portfolio cardinality:** Asset reduction can be further improved by determining an optimal number of assets to hold considering the market environment and the covariance relationships between the assets (i.e., whether an asset can be represented by another). A possible approach would be constructing portfolios with various cardinality ex post some time in the past, and selecting the cardinality that corresponds to the highest observed Sharpe ratio in the historical period. Moreover, for larger datasets, the runtime of the algorithm will increase. Therefore, it may be beneficial to explore alternate formulations of the problem, such as maximizing portfolio correlation.
- **Estimation and generation of return scenarios:** More robust methods of generating the return space, based not only observed data but market indicators. The current analysis used many assumptions about return data. Specifically, our CVaR implementation assumed a return space based on historical returns alone and a uniform distribution. The improved SR optimization uses different estimates of μ and Q, but these are resampled from the same distribution defined by the initial estimation of these parameters. While these are beyond the scope of the current analysis, generating higher quality inputs would further improve the results obtained in portfolio optimization.

References

[1] S.-S. Chen, "Predicting the bear stock market: Macroeconomic variables as leading indicators," *Journal of Banking & Finance,* vol. 33, no. 2, pp. 211-223, 2009.

Appendix

The following items are attached in the zip folder. They are placed in another folder and will not be involved with the algorithm.

- Regression files for WLS, OLS, ridge, and lasso regression.
- Implementation of CVaR, Sharpe Ratio optimization, SR optimization with ellipsoidal uncertainty and resampling.