MIE377: Financial Optimization Models

Project 2 (Winter 2023)

Group Members: Harvi Karatha, Ziana Suleman, James Li

Results from testing out multiple mean-variance optimization models in combination with a ordinary least squares method with a focus on sparse factor models were compiled throughout this project. Based on the resulting Sharpe Ratios, the Ridge Regression with the Sharpe Ratio maximization model was chosen to be the most optimal and usable combination. With the provided dataset, the resulting Sharpe Ratio for dataset 1, 2, 3 is 0.20594, 0.12121, and 0.21119 respectively. The average turnover rate for dataset 1, 2, 3 is 0.48375, 0.6486, 0.55004 respectively.

---

1 Introduction

The purpose of this project is to design an algorithmic trading system that maximizes the Sharpe Ratio and minimizes average turnover. Leveraging concepts seen throughout the course, our team decided to edit the existing code to achieve 2 main goals: prioritizing the most relevant factors and optimizing the Sharpe Ratio. Using sparse factor models and different portfolio optimization methods, we were able to compare and contrast different models to determine the model that has the best and most consistent performance under different market conditions.

2 Methodology

2.1 Key Takeaways from Project 1 on changing the initial data range

Our results and analysis from project 1 allowed us to reflect upon components of our model that we should keep and components that we should change. One thing that we found was successful from project 1 was changing the initial data range from 3 years to 5 years. This was useful because we were able to use more data for estimating our μ and Q. More data improves the error of our variance estimate while the error of our mean estimate remained the same.

2.2 Choosing OLS, LASSO, or Ridge in OLS.m

2.2.1 Key Takeaways from Project 1

Another thing that we did not change for project 2 was our OLS.m function because in project 1, we accounted for the original OLS file, LASSO, and ridge regression using k-fold cross validation. From course content, we learned that sparse factors models decide how many factors should be included to make sure the optimization model is not over fitted. A sparse model uses less factors which also makes it easier to interpret. The penalized regression model of BSS utilizes the l_0 norm which is not convex therefore BSS is not convex. The l_0 norm requires multiple indicator functions in our optimization model and it is difficult to address computationally which is why we decided to not implement BSS in OLS.m. As stated in project 1, ridge regression was our most favourable factor model method because the implementation of k-fold cross validation helps to prevent over fitting. Since our model will be tested on unseen datasets, this method of prioritizing factors is the most favourable and is the one we will heavily consider in testing for project 2. While we will still consider the ordinary least squares method and LASSO, our final model will be an MVO paired with ridge regression because it will prevent over fitting our data.

2.3 Choosing Optimization Model in MVO.m

2.3.1 Key changes in project 2

For this project, we decided to consider a different set of MVOs to see how they impact the Sharpe Ratio and average turnover. In Appendices 6.3 and 7.2, we can see that the most impactful models are the Sharpe Ratio are the Sharpe Ratio optimization and risk adjusted lambda. Based on our conclusions from project 1, the Sharpe Ratio optimization is independent of parameters whereas the risk-adjusted MVO was dependent on a lambda parameter which is why we decided to only take into consideration the Sharpe Ratio optimization from project 1. The new models we decided to implement were Robust MVO with box and ellipsoidal uncertainty, Risk-Adjusted Return, Risk Parity optimization, CVaR optimization, and Michaud Resampling.

2.3.2 Sharpe Ratio Optimization

The naive, non-convex version of Sharpe Ratio optimization goes from the following:

$$\begin{aligned} \max_{x} \quad & \frac{\mu^{\mathbf{T}}\mathbf{x} - \mathbf{r_f}}{\sqrt{\mathbf{x^T Q x}}} \\ \text{s.t.} \quad & \mathbf{1^T x = 1} \\ & \mathbf{x \geq 0} \end{aligned}$$

to a convex form in which MATLAB's `quadprog()` can solve:

$$\begin{aligned} \min_{y,\kappa} \quad & \mathbf{y^T Q y} \\ \text{s.t.} \quad & \mathbf{1^T y} = \kappa \\ & \mu^{\mathbf{T}}\mathbf{y = 1} \\ & \kappa, \mathbf{y \geq 0} \end{aligned}$$

The Sharpe Ratio MVO was anticipated to be the most successful MVO due to its focus on specifically optimizing the Sharpe Ratio. Since the weighting of assets would be wholly determined on how to best maximize the Sharpe Ratio, the Sharpe Ratio optimization focuses on turning a non-quadratic problem into a manageable convex optimization problem by redefining the variables.

In terms of Sharpe maximization, we minimize κ and y. We have also defined κ such that $y = \kappa x$.

2.3.3 Sharpe Ratio Optimization with robust term

In general, Robust MVO quantifies the uncertainty of estimating parameters and incorporates it into our optimization model to minimize it. Using a robust term and reformulating the Sharpe Ratio optimization from above, we will be able to account for parameter uncertainty in our model. This model focuses on maximizing the Sharpe Ratio while incorporating uncertainty into the model. We decided to implement this model because the Sharpe maximization above does not account for uncertainty in the parameters and if this uncertainty is too large, it may be pointless to optimize at all. We decided to include both box uncertainty and ellipsoidal uncertainty because their μ_{true} vectors are different and we wanted to see how the different vectors would impact the Sharpe Ratio and the average turnover.

Box Uncertainty: Robust MVO with box uncertainty says that the μ_{true} vector containing the true expected returns lies somewhere within a 'box' centered around the estimate μ. We can reformulate the Sharpe Ratio optimization to include the box uncertainty robust MVO term as follows:

$$
\begin{aligned}
\min_{y,} \quad & \mathbf{y^T Q y} \\
\text{s.t.} \quad & (\mu - \mathbf{r_f})^{\mathbf{T}} - \delta^{\mathbf{T}} \mathbf{z} \geq \mathbf{1} \\
& \mathbf{z} \geq \mathbf{y} \\
& \mathbf{z} \geq -\mathbf{y} \\
& \mathbf{y} \geq \mathbf{0}
\end{aligned}
$$

Ellipsoidal Uncertainty: Realizations in the corners of box uncertainty are usually quite extreme and ellipsoidal uncertainty does not take into account the corners of the box. We can reformulate the Sharpe Ratio optimization to include the ellipsoidal uncertainty robust MVO term as follows:

$$
\begin{aligned}
\min_{y,} \quad & \mathbf{y^T Q y} \\
\text{s.t.} \quad & (\mu - \mathbf{r_f})^{\mathbf{T}} - \delta^{\mathbf{T}} \mathbf{z} \geq \mathbf{1} \\
& \mathbf{z} \geq \mathbf{y} \\
& \mathbf{z} \geq -\mathbf{y} \\
& \mathbf{y} \geq \mathbf{0}
\end{aligned}
$$

2.3.4 Risk Adjusted Return MVO

The risk-adjusted return MVO focuses on minimizing the variance and maximizing returns (two factors that heavily impact the sharpe ratio). For the purposes of this project, the risk-adjusted return MVO will be used in conjunction with the turnover constraints. Additionally, the λ was set to 1 based on prior testing.

$$
\min_{x} \; x^T Q x - \mu^T x
$$

2.3.5 Risk Parity Optimization

The goal of risk parity optimization is to maximize risk diversification by equalizing the risk contribution of each asset. We chose to implement this model because it diversifies our risk meaning if one asset under performs, the overall portfolio is less likely to have significant losses. Since the Sharpe Ratio assesses the risk adjusted performance of a portfolio, we wanted to see how equalizing the risk contribution of each asset impacts the Sharpe Ratio.

The original risk parity function presented in lecture contained a non-linear objective function. By reformulating the objective into a convex function, we can calculate our asset weights using the following optimization problem:

$$
\begin{aligned}
\min_{y} \quad & \mathbf{\frac{1}{2} y^T Q y - c \sum_{i=1}^{n} \log(y_i)} \\
\text{s.t.} \quad & \mathbf{y} \geq \mathbf{0}
\end{aligned}
$$

2.3.6 CVaR Optimization

In terms of CVaR Optimization, this model focuses on minimizing downside risk and the potential of loss in the Profit and Loss distribution function, $f(x, r_s)$. With z as auxiliary variables for each scenario and r_s as a realization for scenario s, this model ensures that there is α confidence that the loss will not exceed the value at risk. The CVaR model involves the following optimmization model:

$$
\begin{aligned}
\min_{x,z,\gamma,} \quad & \gamma + \frac{\mathbf{1}}{((\mathbf{1}-\alpha)\mathbf{S}} \sum_{\mathbf{s=1}}^{\mathbf{S}} \mathbf{z_s} & \\
\text{s.t. } & \mathbf{z_s} \geq \mathbf{0} & \text{s} = 1 \text{ to S} \\
& \mathbf{z_s} \geq \mathbf{f}(\mathbf{x}, \mathbf{r_s}) - \gamma & \text{s} = 1 \text{ to S}
\end{aligned}
$$

2.3.7 Michaud Resampling

Michaud resampling is a robust optimization technique for portfolio optimization that was used to generate an efficient frontier that minimizes the impact of uncertainty within the data used. It is implemented within the mean-variance optimizer, samples T points from a normal distribution with mean μ and variance Q, and generates an efficient frontier for this subset of data. After running it through N iterations, we take the average of the efficient frontiers to generate an efficient frontier that should be adverse of risk.

For this project, the uncertainty lies within the parameter estimation, specifically μ and Q. Although we use other techniques to minimize uncertainty, utilizing this technique with a high number of simulations addresses the intrinsic uncertainty of the estimated parameter to ensure the portfolio is robustified. We drew samples from the normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{Q})$ and run 3 simulations of the optimizers, depending on the optimizer used. All entries of the randomly sampled data needed to be ≥ -1 to ensure the `geomean(sampleRets + 1)` function would run; to prevent this, a condition was enforced to resample the data with the same parameters until all entries are ≥ 1. Finally, after the weight matrix is compiled for each simulation, the mean is taken to obtain the optimal asset weights for the portfolio.

2.4 K-Fold Cross-Validation

This section will only provide a quick summary of the details from Project 1 (for a more detailed explanation, see Appendix 6.1) and improvements made to the implementation of 10-fold cross-validation.

10-fold cross-validation was used to estimate an appropriate penalty term for the ridge regression algorithm. However, some implementation changes were tested in the following ways:

- To improve runtime, the λ selection process was optimized by vectorizing the λ errors. In specific, the Sharpe Ratio error of executing 10-fold validation for **all** λ's were determined at once, returning as a vector where we choose the λ with the smallest error term.
- Michaud resampling was implemented **directly** within the 10-fold CV. In this way, the 10-fold CV can run on sampled returns data and determine the best λ on a consistent and randomly sampled dataset. In this fashion, the optimizers did not have to individually be altered to account for Michaud resampling and instead the general cross-validation code would resample for us.

2.4.1 Minimizing Average Turnover

A main component that was missing from our first project was a constraint that specifically focused on minimizing the average turnover. In project 1, we were very focused on maximizing the Sharpe Ratio that we neglected to consider how to minimize the average turnover at the same time. For project 2, this was one of our main goals and utilizing course content on limiting our transaction costs, we decided to implement similar constraints for minimizing average turnover. Based on the MVO slides and information we were given, there are a few methods of effectively implementing this minimization that were compared depending on the effectiveness. The first method is including the following formulation for our constraints for each of the n assets:

$$\mathbf{x_i} - \mathbf{x_i^0} \leq \lambda * \mathbf{1}$$
$$\mathbf{x_i^0} - \mathbf{x_i} \leq \lambda * \mathbf{1}$$

This is a situation where λ can be set as a integer value, which is the standard for the max change that can occur for each asset. The second method is minimizing the y_i using the following constraints and adding a y_i into the objective function.

$$\mathbf{x_i} - \mathbf{x_i^0} \leq \mathbf{y_i}$$
$$\mathbf{x_i^0} - \mathbf{x_i} \leq \mathbf{y_i}$$

3 Analysis from Training, Validation, and Testing

3.1 Analysis Criteria

The given project handout emphasized two testing criteria: Sharpe Ratio and Average Turnover. However, it was also emphasized that 80% of the "score" for each trial will be placed upon the Sharpe Ratio maximization. Thus, this is one of the main aspects that the model aimed to maximize. Although there is a heavy emphasis on maximizing the Sharpe Ratio, there is a concern of over fitting the model to the provided dataset, which is why a cross-validation step is needed for the penalty term selection. This step allows the model to appropriately adapt to each dataset and dynamically select the parameter based on the losses. In the following section, we will analyze and test ridge regression with our optimization models (Sharpe Ratio Optimization, Robust MVO, Risk Parity, CVaR, Michaud Resampling) to determine the models that have that improve the Sharpe Ratio and the average turnover ratio. The best models will then be compared to choose which one has the best and most consistent performance.

3.2 Ridge Regression Training Set

Since factor returns were given as part of the dataset, it was apparent that optimizing the factor coefficients was necessary in order to optimize the model. This led to choosing between two sparse factor models, LASSO Regression ($\ell 1$) and Ridge Regression ($\ell 2$), since the $\ell 0$ model is not convex and cannot be solved using MATLAB. Ultimately, it was found that Ridge Regression is easier to implement through MATLAB, using the `ridge()` function. Additionally, utilizing the unconstrained version of Ridge Regression allows to adjust the penalty term (λ) for different re-balancing periods:

$$\min_{\beta^{(\mathbf{i})}} ||\mathbf{r}^{(i)} + \mathbf{X}\beta^{(i)}|| - \lambda||\beta^{(i)}||$$

where the adjusting of the penalty term will be implemented using k-fold cross-validation.

3.2.1 K-Fold Cross-Validation

There was no new testing conducted on the 10-fold cross-validation. For the testing of the previous project submission, where different sampling lengths were utilized for parameter estimation, see Appendix 7.1.

3.3 Comparing Different Optimization Models w/ Ridge Regression

3.3.1 Baseline OLS and MVO

We summarized the baseline values in the table below as a reference for how our values have changed from the baseline project files that were given.

Dataset	MVO Method	Sharpe	Average turnover
1	Baseline	0.17171	0.48735
2	Baseline	0.081036	0.6486
3	Baseline	0.19279	0.55004

Table 1: Sharpe Ratio Maximization without turnover constraint and Ridge Regression

3.3.2 Optimizing Sharpe Ratio MVO:

The table below includes the results of altering MVO.m to optimize the Sharpe Ratio with ridge regression. This is based on the following optimization model with minimization:

$$\begin{aligned} \min_{y,\kappa} \quad & \mathbf{y^T Q y} \\ \text{s.t.} \quad & \mathbf{1^T y} = \kappa \\ & \mu^{\mathbf{T}}\mathbf{y} = \mathbf{1} \\ & \kappa, \mathbf{y} \geq \mathbf{0} \\ & y = \kappa x \end{aligned}$$

Dataset	MVO Method	Sharpe	Average turnover
1	Sharpe w/o turnover constraint	0.20735	0.43638
2	Sharpe w/o turnover constraint	0.108	0.66934
3	Sharpe w/o turnover constraint	0.17726	0.56511

Table 2: Sharpe Ratio Maximization without turnover constraint and Ridge Regression

Our results show that the Sharpe Ratio has increased for datasets 1 and 2 with a slight decrease for dataset 3. The average turnover have decreased for dataset 1 and increased for both dataset 2 and 3.

Minimizing average turnover was something that we focused on addressing in project 2 which is why we also re-conducted analysis for optimizing the Sharpe Ratio with an additional turnover constraint. The results from testing this using ridge regression are summarized in the table below:

Dataset	MVO Method	Sharpe	Average turnover
1	Sharpe w/ turnover constraint	0.21099	0.47944
2	Sharpe w/ turnover constraint	0.12337	0.72613
3	Sharpe w/ turnover constraint	0.19074	0.5552

Table 3: Sharpe Ratio Maximization with turnover constraint and Ridge Regression

When compared to the Sharpe Ratio optimization without a turnover constraint, the Sharpe Ratio has increased for all datasets and our average turnover has decreased for dataset 3. When compared to the given model, the Sharpe Ratio for each dataset has significantly increased and the average turnover has decreased for dataset 3.

3.3.3 Robust MVO: box and ellipsoidal uncertainty

In terms of the Robust MVO using box and ellipsoidal uncertainty, we tested both implementations using Ridge Regression. Table 4 displays the results of our testing for all three datasets:

Dataset	MVO Method	Sharpe	Average turnover
1	Sharpe w/ Box uncertainty	0.17367	3.20E-09
2	Sharpe w/ Box uncertainty	0.11591	6.44E-09
3	Sharpe w/ Box uncertainty	0.18606	6.18E-09
1	Sharpe w/ ellipsoidal uncertainty	0.1737	2.90E-09
2	Sharpe w/ ellipsoidal uncertainty	0.11587	7.08E-09
3	Sharpe w/ ellipsoidal uncertainty	0.18607	6.51E-09

Table 4: Sharpe Ratio Maximization with robust penalty term and Ridge Regression

Our results show that the ellipsoidal and box uncertainty have very similar Sharpe Ratios with slightly different average turnovers. Each combination performs better than the original files given where the Sharpe Ratio has slightly increased for dataset 1 and 2 and slightly decreased for dataset 3. The average turnover for all three datasets has significantly decreased.

3.3.4 Risk Parity Optimization:

We tested our risk parity optimization using Ridge regression and table 5 displays the results of our testing for all three datasets:

Dataset	Sharpe	Average turnover
1	0.1737	6.81E-09
2	0.11586	3.74E-09
3	0.18609	1.58E-09

Table 5: Risk Parity Optimization and Ridge Regression

Our results show that each combination performs better than the original files given where the Sharpe Ratio has slightly increased for dataset 1 and 2 and slightly decreased for dataset 3. The average turnover for all three datasets has significantly decreased. These results are also very similar Robust MVO which could be due to the constraint for average turnover.

3.3.5 Risk Adjusted Return

When the lambda value is larger, the risk-adjusted return MVO focuses on minimizing the variance: with the larger λ, the objective function attempts to minimize the variance to lower the impact of the larger λ. However, with a fixed chosen lambda for prior testing, the turnover rate constraint was added to create the following results.

MVO Method	OLS Method	Turnover Constraint	Average Sharpe	Average turnover
Risk-Adjusted	Original	Y	0.16865	5.37%
Risk-Adjusted	Ridge	Y	0.12490	3.52%

MVO	OLS	Turnover(Y/N)	Avg Sharpe	Avg Turnover
cVar + Sharpe	Original	yes - both	0.12614	8.00%
cVar + Sharpe	Ridge	yes - both	0.17205	45.68%
cVar + Equal Weights ($\lambda = 0.0001$)	Original	yes - cVar	0.15628	5.06%
cVar + Equal Weights ($\lambda = 0.0001$)	Ridge	yes - cVar	0.15628	5.06%
cVar + Equal Weights ($\lambda = 0$)	Original	yes - cVar	0.15582	4.86%
cVar + Equal Weights ($\lambda = 0$)	Ridge	yes - cVar	0.15582	4.86%

3.3.6 CVaR

Using CVaR involves combining this algorithm with other models, which was used whenever CVaR returned an empty vector. An if-else statement was added following the execution of the CVaR code to ensure if an empty vector was returned, the other optimizer would execute and return an optimal weight vector. Using the turnover rate constraints, the λ is varied with 0.0001 and 0 providing the best values with this method of implementation: $x_0 - x \leq \lambda$ and $x - x_0 \leq \lambda$. Equal weights and the Sharpe Ratio were chosen to be used with cVaR because of their easier implementation.

3.3.7 Michaud Resampling

For testing with Michaud resampling, the results below only used dataset 1 since this section is so computationally heavy. Note that the statistics below may not be reproducible since Michaud Resampling will not necessarily sample the same data to estimate μ and Q in the formulations. In addition, running Michaud resampling alongside 10-fold cross-validation is computationally heavy, with a runtime of > 10 minutes, which is why the runtime will be listed as a metric to consider in the table below. An estimation of the total number of computations needed is as follows: 10 years of rebalancing (180 months - 60 months of parameter estimation), meaning 20 rebalancing periods $\rightarrow$ 3 simulations per rebalancing period, 100 lambda values per simulation, 10 folds split per lambda value, and the execution of the optimizer per fold split. Overall, this totals to about 60000 iterations of running the optimizer, which undoubtedly is computationally heavy.

$\{N, T\}$	Optimizer	Sharpe Ratio	Average Turnover	Runtime (s)
{3, 100}	Robust MVO (ellipsoidal)	0.1737	3.42×10^{-9}	683.34
{5, 250}	Robust MVO (ellipsoidal)	0.1737	3.22×10^{-9}	1074.49
{3, 100}	Robust MVO (box)	0.1737	3.41×10^{-9}	618.11
{5, 250}	Robust MVO (box)	0.1737	3.18×10^{-9}	981.65
{3, 100}	MVO	0.18714	0.60528	260.19
{5, 250}	MVO	0.19742	0.62644	472.28
{3, 100}	CVaR ($\alpha = 95\%$)	0.17212	0.00210	609.32
{5, 250}	CVaR ($\alpha = 95\%$)	0.17212	0.00179	1367.12
{3, 100}	Risk-Parity	0.1737	6.81×10^{-9}	845.49
{5, 250}	Risk-Parity	0.1737	6.807×10^{-9}	1457.19

Table 6: Results after using Michaud Resampling for {N=3, T=100} and {N=5, T=250}.

Based off these metrics, it can be noted that in comparison to the cross-validation with no Michaud resampling, the Sharpe Ratio is not improved at all with a nearly identical turnover. A plausible explanation for this is that the turnover constraint in each of these models is too harsh, resulting in no changes made to the portfolio for each iteration. This essentially nullifies the effect of resampling the data since no changes can be made and thus using new data is pointless. Thus, Michaud resampling was not considered to be implemented.

4 Discussion and Conclusion

4.1 Choosing our Final Model

Dataset	MVO Method	Sharpe	Average turnover
1	Sharpe w/ turnover constraint	0.21099	0.47944
2	Sharpe w/ turnover constraint	0.12337	0.72613
3	Sharpe w/ turnover constraint	0.19074	0.5552

Table 7: Sharpe Ratio Maximization with turnover constraint and Ridge Regression

Based on the results, our best model is using our Sharpe Ratio optimization with a turnover constraint and ridge regression. This model produces the best results for the Sharpe Ratio by far when compared to the other models that we have implemented. Although the average turnover is not as low as we expected, it is lower than the project files given as well as it is lower for dataset 3 compared to the Sharpe Ratio optimization model without a turnover constraint.

Another benefit of the Sharpe Ratio optimization is that it is independent of parameters which allows it to be generalized and adaptable to other datasets. With the use of k-fold cross validation, we will be choosing the best lambda to create a factor model for each dataset that reduces over fitting.

Ideally, we would have found a better model from the ones that were tested to minimize turnover while maximizing sharpe but the models that did minimize turnover significantly decreased the sharpe ratio. Since the sharpe ratio is more important in the model, we decided that it would be better to choose the model that maximized the sharpe ratio the most.

4.2 Comparison of Original Model, Project 1 Final Model, Project 2 Final Model

The portfolio wealth evolution for dataset 1 for the original model, project 1 model, and our chosen project 2 model are as follows:

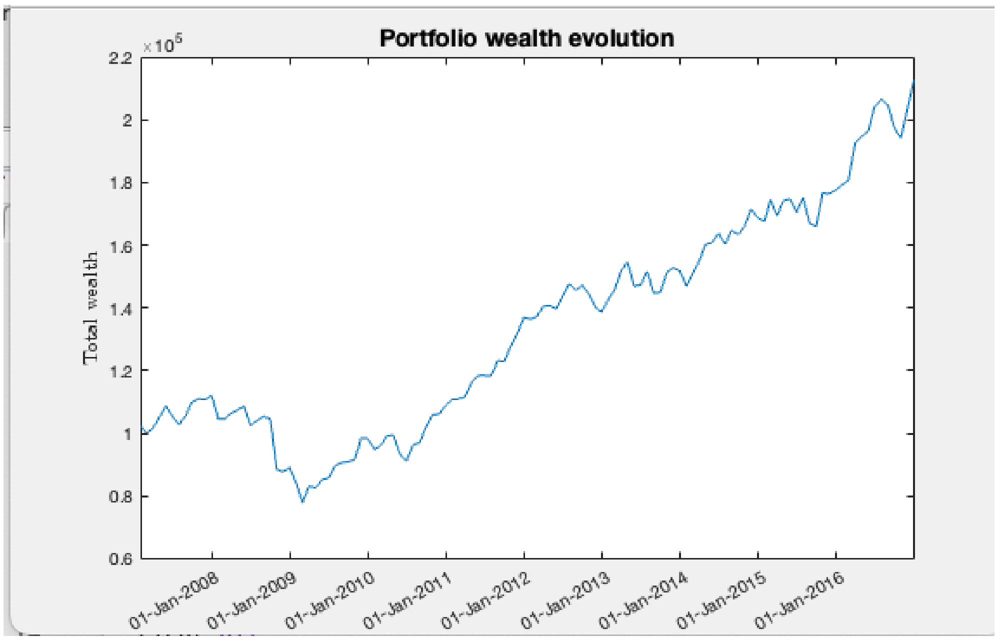

Figure 1: Portfolio Wealth Evolution for Original Model - Sharpe Ratio = 0.17171, average turnover = 0.48735

When comparing the portfolio wealth evolution graphs from the project 2 model with our original model and the project 1 model, our total wealth has increased over the time period from 2008 to 2016. By January 2016, our project 2 model has exceeded 2.5×10^5 in total wealth compared to the original model which reached approximately 1.7×10^5 in total wealth by January 2016. We can also see that our project 2 model is closer to approaching 3×10^5 than our project 1 model. It is also evident that the slope of a trendline of the portfolio wealth evolution graph has increased in our project 2 model: this shows that our total wealth is increasing at a faster rate compared to the original model and our project 1 model.

The portfolio weight for the original model, project 1 model, and our chosen project 2 follow are as follows:

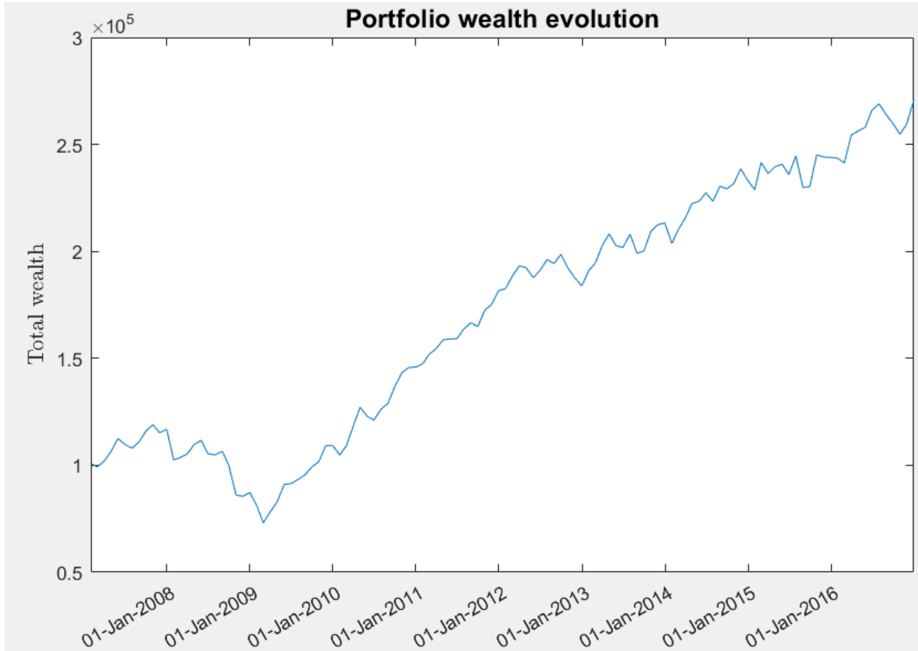

Figure 2: Portfolio Wealth Evolution for the Project 1 Final Model

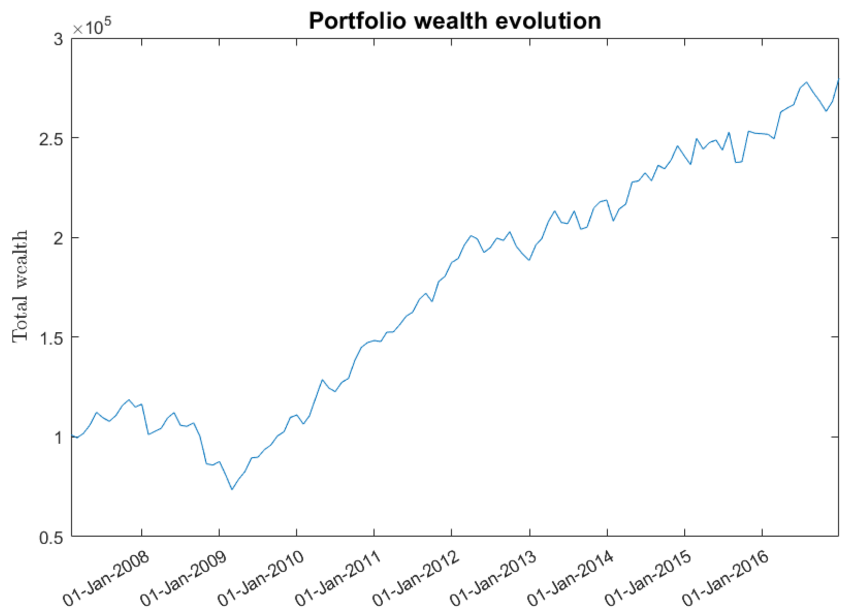

Figure 3: Portfolio Wealth Evolution for the Project 2 Final Model

When directly comparing Figure 4, Figure 5, and Figure6, the portfolio weights rebalancing is visually less extreme with fewer spikes in reweighting for the 12 to 20 rebalance period in Figure 5 and 6 compared to Figure 4. The original model has consistently many spikes which are indicative of major immediate re-weightings between periods, especially in periods 1 to 6. Other than T and AAPL in Figure 5 and 6, which have major spikes in the rebalance periods from 4 to 9, the overarching majority of Figure 5 and 6 is clearly more smooth in direct contract to Figure 4. This is a testament to our efforts to lower the average turnover rate.

From the original model to our project 2 model, there are significantly fewer sharp spikes compared to the original model. This is a testament to the added turnover constraints. Since the sharp ratio maximization prioritizes more than just the variance unlike the original model, there are visibly

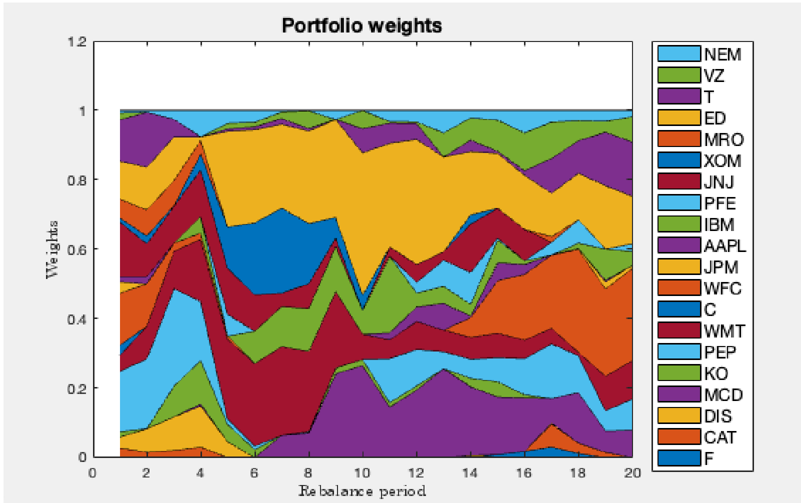

Figure 4: Portfolio Weights for original model Sharpe Ratio = 0.17171, average turnover = 0.48735

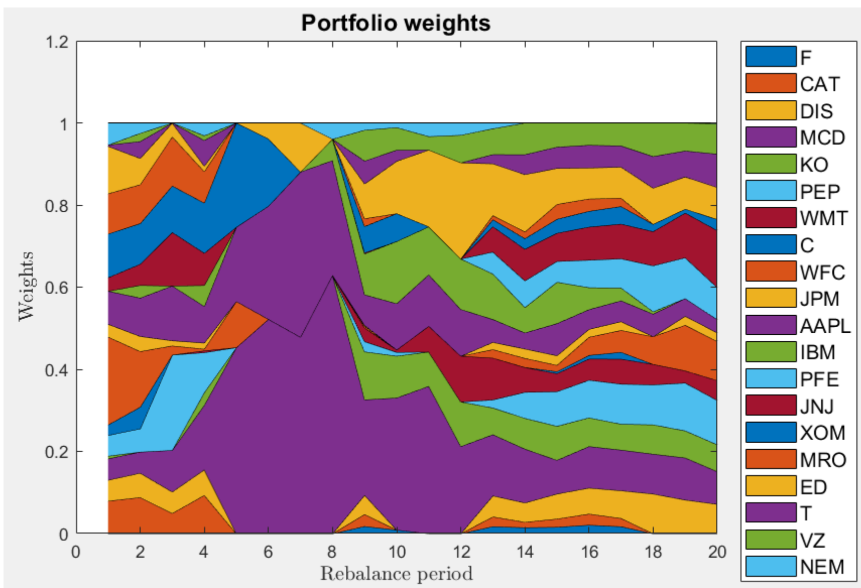

Figure 5: Portfolio Weights for the project 1 final model.

very different weightings for each of the asset classes when comparing the models as well.
From the project 1 model to the project 2 model: The improvement between project 1 and project 2 is quite subtle due to the usage of the same model. However, due to the turnover rate constraints, the major spikes that are clearly visible in project 1 model are slightly more rounded off in the project 2 model. Other than the decrease in prominent spikes, there are not many other visible differences.

4.3 Conclusion

After testing multiple combinations using different factor models and optimization models, we believe that the model that will perform the best and most consistent under different market conditions is the model that uses ridge regression and Sharpe Ratio optimization with a turnover constraint. Our implementation of ridge regression using k-fold cross validation allows the best penalty term λ to be chosen with each dataset. This will allow us to calculate μ and Q that will be used in our optimization formulation to consistently maximize the Sharpe Ratio and reduce average turnover.

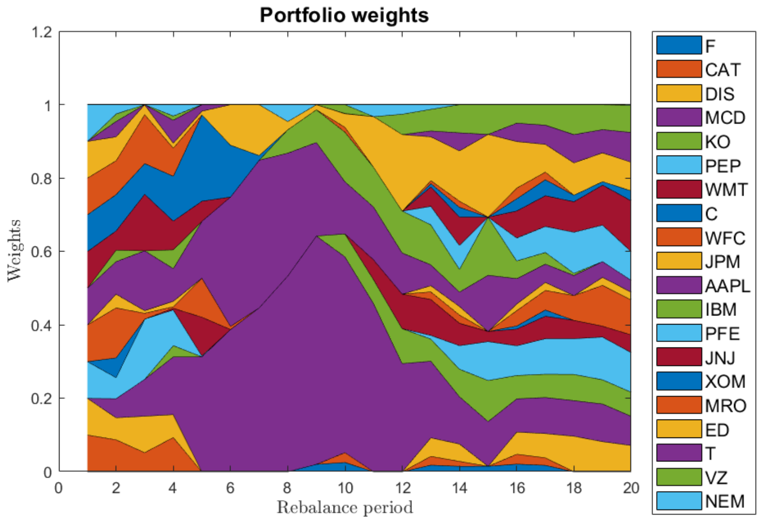

Figure 6: Portfolio Weights for the project 2 final model.

There are a few weaknesses and improvements that can be made to our model. First of all, the turnover constraint does not always lower the turnover as it should. Another weakness is that we did select a very similar model to project 1 but our other models that were tested did not maximize the sharpe ratio as well as our chosen model. The main issue with our other models were that we had a lot of difficulty minimizing the turnover while maintaining a good sharpe ratio. Our next steps would be to find another constraint that successfully minimizes turnover while without lowering the Sharpe Ratio.

5 Appendix

5.1 Appendix A: Full Dataset testing Excel Sheet

MVO	OLS	Turnover Constrained (Yes/No)	Average Sharpe	Average Turnover
Sharpe Ratio Maximization	Ridge	UNDER BEST ONE SO FAR	0.18207	46.58%
Sharpe Ratio Maximization	Ridge	yes - sharpe M1	0.17205	45.68%
cVar + Sharpe	Ridge	yes - cVar yes - Sharpe	0.17205	45.68%
Sharpe Ratio Maximization	Ridge	yes - sharpe M2	0.17056	45.81%
Sharpe Ratio Maximization	Original	yes - sharpe M1	0.16941	48.23%
Risk-Adjusted	Original	yes - Risk	0.16865	5.37%
Sharpe Ratio Maximization	Original	yes - sharpe M2	0.16742	48.21%
Sharpe Ratio Maximization (from project 1)	Ridge	no	0.16420	45.88%
Robust MVO- ellipsoidal uncertainty	Original	yes	0.15855	3.96%
Risk Parity	Ridge	yes	0.15855	3.96%
Risk Parity	LASSO	yes	0.15855	3.96%
Robust MVO- box uncertainty	Ridge	yes	0.15855	3.96%
Robust MVO- ellipsoidal uncertainty	Ridge	yes	0.15855	3.96%
Risk Parity	Original	yes	0.15854	3.96%
Robust MVO- box uncertainty	Original	yes	0.15852	3.96%
cVar + Equal Weights	Original	yes - cVar	0.15824	4.75%
cVar + Equal Weights	Ridge	yes - cVar	0.15824	4.75%
cVar + Equal Weights (Lambda = 0.0001)	Original	yes - cVar	0.15628	5.06%
cVar + Equal Weights (Lambda = 0.0001)	Ridge	yes - cVar	0.15628	5.06%
cVar + Equal Weights (Lambda = 0)	Original	yes - cVar	0.15582	4.86%
cVar + Equal Weights (Lambda = 0)	Ridge	yes - cVar	0.15582	4.86%
cVar + Equal Weights	LASSO	yes - cVar	0.15127	6.08%
Original	Original	no	0.14851	45.86%
Original	Ridge	no	0.13824	30.47%
cVar + Sharpe	Original	yes - cVar yes - Sharpe	0.12614	8.00%
Risk-Adjusted	Ridge	yes - Risk	0.12490	3.52%
Original	LASSO	no	0.10695	46.69%
Sharpe Ratio Maximization (from project 1)	LASSO	no	0.04366	70.71%

Figure 7: Average turnover and average Sharpe over the three datasets

Sharpe (D1)	Turnover Rate (D1)
0.21166	0.47762
0.20151	0.47909
0.20151	0.47909
0.20052	0.48282
0.21033	0.52154
0.20972	1.32E-09
0.2092	0.52726
0.20735	0.43638
0.1737	2.77E-09
0.1737	6.81E-09
0.1737	1.21E-08
0.17367	3.20E-09
0.1737	2.90E-09
0.17369	6.81E-09
0.1737	8.42E-17
0.1737	8.24E-17
0.17407	0.002
0.17407	0.002
0.1737	7.48E-17
0.1737	8.03E-17
0.15682	0.012518
0.17171	0.48735
0.17217	0.30487
0.16529	8.58E-17
0.20026	2.56E-09
0.13947	0.56553
0.043491	0.9149

Figure 8: Average turnover and Sharpe for dataset 1

Sharpe (D2)	Turnover Rate (D2)	Sharpe (D3)	Turnover Rate (D3)
0.12337	0.72503	0.21119	0.47862
0.10568	0.66712	0.20895	0.50883
0.10568	0.66712	0.20895	0.50883
0.10222	0.66967	0.20895	0.50954
0.11061	0.67108	0.1873	0.56715
0.10009	0.042905	0.19615	0.0033143
0.10644	0.67745	0.18661	0.55629
0.108	0.66934	0.17726	0.56511
0.11589	4.53E-09	0.18607	6.50E-09
0.11586	3.74E-09	0.18609	1.58E-09
0.11587	2.72E-09	0.18607	3.77E-09
0.11591	6.44E-09	0.18606	6.18E-09
0.11587	7.08E-09	0.18607	6.51E-09
0.11586	4.06E-09	0.18608	1.59E-09
0.11494	0.031721	0.18607	7.94E-17
0.11494	0.031721	0.18607	8.22E-17
0.11187	0.035189	0.1829	0.0089626
0.11187	0.035189	0.1829	0.0089626
0.11157	0.032659	0.18218	5.93E-03
0.11157	0.032659	0.18218	0.005931
0.1161	0.033523	0.1809	0.046039
0.081036	0.6486	0.19279	0.55004
0.075392	0.37164	0.16716	0.40415
0.12261	0.10526	0.090517	0.088797
0.070115	0.0159	0.10433	4.64E-09
0.025342	0.61114	0.15605	0.58394
0.03089	0.98712	0.056596	0.88287

Figure 9: Average turnover and Sharpe for datasets 2 and 3

6 Appendix B: Project 1 Full Section Descriptions

6.1 Appendix B.1: K-Fold Cross-Validation

The penalized version of the regression models are being used in order to better adjust for the purpose of the model (see equation 4). As such, a λ penalty term needs to be properly selected to fit both the dataset and the purpose of the model. k-fold cross-validation is utilized for the λ selection by evaluating the score of the model on a specific λ; the cross-validation takes 100 λ values from $\{10^{-6}, 10^2\}$. By splitting the dataset into k folds, 1 fold is used as a testing dataset and $k-1$ folds are used as a training dataset. In a loop, each of the k folds will be used as a testing set with the other folds being used as a training set, and the model will be evaluated using the total error for each of the k iterations.

The loss function is used to evaluate the quality of the model by comparing the model using the training dataset versus the testing dataset. Initially, the loss was evaluated by determining the total estimation error of the test return for all k folds across all assets and time periods:

$$\mathcal{L} = \sum_{j=1}^{k} \sum_{t=1}^{T} \sum_{i=1}^{n} \mathbf{r}_{tk}^{(i)} - \mathbf{X}_{tk}\beta^{(i)}$$

whereas the final model implemented a loss function that determined the Sharpe Ratio loss, which would ultimately outrank the previous loss function for Sharpe Ratio maximization.

$$\mathcal{L} = \frac{\prod_{i=1}^{n}(\mu_i + 1) - 1}{\sigma_i}$$

6.2 Appendix B.2: OLS.m factor models from Project 1

6.2.1 Ordinary Least Squares

In the given OLS.m file, we are conducting Ordinary Least Square to estimate the coefficients of our factor model. The formula used to find the coefficients using OLS is shown below.

$$\mathbf{B} = (X^T X)^{-1} X^T Y \tag{1}$$

With OLS, we create our factor model using all 8 factors, which means the model that is being constructed may be too closely tailored to the dataset given, or being overfit. This can lead to inaccurate predictions, which cannot perform well for all new types of data.

6.2.2 Selecting Sparse Factor Models

There are 3 different types of sparse factor models to improve our factor model coefficients: Best Subset Selection (BSS), LASSO, and Ridge Regression which are shown below:

$$\min_{B_i} \; ||(r_i - XB_i)||_2^2 + \lambda ||B_i||_0 \tag{2}$$

$$\min_{B_i} \; ||(r_i - XB_i)||_2^2 + \lambda ||B_i||_1 \tag{3}$$

$$\min_{B_i} \; ||(r_i - XB_i)||_2^2 + \lambda ||B_i||_2 \tag{4}$$

The penalized regression model of BSS utilizes the l_0 norm which is not convex therefore BSS is not convex. The l_0 norm requires multiple indicator functions in our optimization model and it is difficult to address computationally which is why we decided to not implement BSS in OLS.m.
The penalized LASSO model utilizes the l_1 norm which is continuous and convex but not smooth and the penalized Ridge Regression model utilizes the l_2 norm which is continuous, convex, and smooth. Ridge Regression does not eliminate any of the factors and instead it determines the coefficients based on the significance of a factor. LASSO does shrink the parameters and conducts factor selection automatically which is why we decided to implement LASSO and Ridge Regression and compare their results with each optimization model to determine the model with the best performance.
Note in equations (3) and (4), there is a penalty term λ which will be addressed in section 2.4 and how we will properly select this penalty term.

6.3 Appendix B.3: MVO.m models from Project 1

6.3.1 Mean Variance Optimization

$$\min_{x} \; x^T Q x$$

The baseline mean variance optimization aided in showcasing how optimizing with a objective function focused on variance minimization still provides an adequate Sharpe Ratio. Additionally, this is the standard convex mean-variance optimization problem with a weight constraint and target return constraint.

6.3.2 MVO with Adjusted Target Return

Using the baseline MVO, the target return constraint is modified to ensure that the portfolio's return exceeds the mean of a defined quantile of the asset returns. Once the asset returns are sorted, the quantile parameter is defined as a percentage between 0% and 100%: Then, the mean of the sorted asset returns from the smallest to the specific percentile of asset returns is used for the return constraint.

$$\begin{aligned} \min_x \ & x^T Q x \\ \text{s.t. } & \mu^T x \geq Return \end{aligned}$$

With this in mind, this provides a more nuanced look at asset returns to see if lowering the necessary portfolio return can improve the Sharpe Ratio.

6.3.3 Risk Adjusted Return MVO

$$\min_x \ \lambda x^T Q x - \mu^T x$$

When the lambda value is larger, the risk-adjusted return MVO focuses on minimizing the variance: with the larger λ, the objective function attempts to minimize the variance to lower the impact of the larger λ. With a small λ value, the formula shifts to maximizing returns because the most negative element of the function would be the mean return component. This risk-adjusted return MVO was considered due to its ability to balance/reconcile mean return maximization and mean variance minimization depending on which one is of more impact on the resulting Sharpe Ratio.

7 Appendix C: Testing from Project 1

7.1 Appendix C.1: K-Fold Cross-Validation Testing

In the k-fold cross-validation, there were many parameters that were varied during the testing process. These include the number of folds, $k \in \{5, 6, 10\}$, the number of months of data used for parameter estimation, $\{36, 48, 60\}$, and using different loss functions to evaluate the cross-validation. Comparing the two loss functions:

Next, it was noted that more months of data used in parameter estimation led to an improved Sharpe Ratio. This trend is easily explainable by the fact that despite returning an ex post Sharpe Ratio, there is no sufficient historical data for portfolio returns and variances. Thus, estimations for these parameters must be made; the more accurate the parameter estimation, the better the ex post Sharpe Ratio. The following data will present results of using 5-fold, 6-fold and 10-fold cross-validation against 36, 48 and 60 months of parameter estimation in an ordered pair designating {Sharpe Ratio, Average Turnover}.

One parameter that was changed but saw little difference in results was the number of λ's tested, varying from 50 up to 1000. It was noted that the cross-validation would lead to very similar λ's

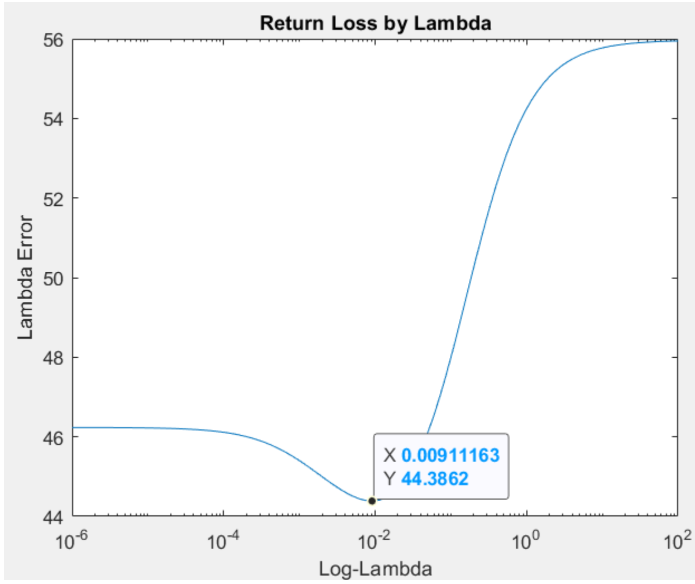

Figure 10: Semilog Plot of Lambdas - Sharpe Ratio = 0.2023, average turnover = 0.48557

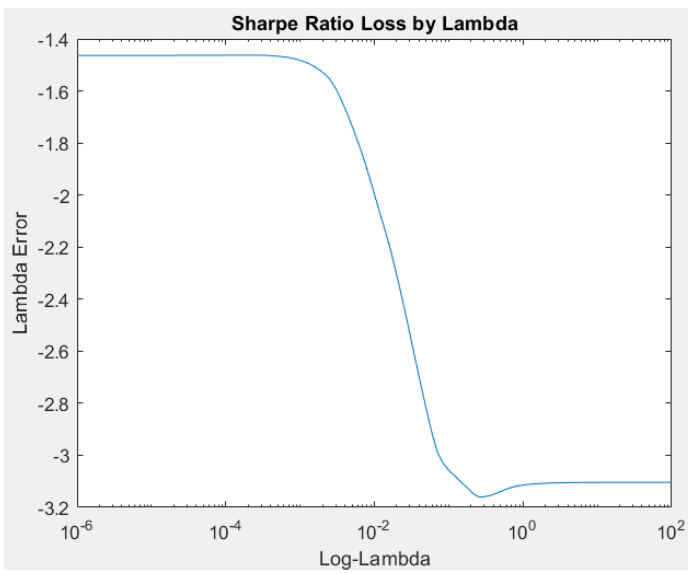

Figure 11: Semilog Plot of Lambdas - Sharpe Ratio = 0.20756, average turnover: 0.43819

	$t = 36$	$t = 48$	$t = 60$
$k = 5$	{0.16998, 0.6221}	{0.13146, 0.51599}	{0.20588, 0.64156}
$k = 6$	{0.16483, 0.62683}	{0.1419, 0.58575}	{0.18689, 0.48391}
$k = 10$	{0.15182, 0.67025}	{0.15553, 0.62967}	{0.20756, 0.43819}

Table 8: Outcomes when varying number of months used for parameter estimation (t) and number of folds for cross-validation (k); output given as {SR, AVG. TO}.

being chosen despite the higher density of points. In addition, the number of folds to split the data into was chosen to ideally be multiples of the number of observations since it would yield an equal number of observations within each subset of data.

7.2 Appendix C.2: All testing from Project 1

OLS Method	MVO Method	Sharpe Ratio	Average Turnover
OLS	Baseline	0.17171	0.48735
OLS	Sharpe	0.16245	0.75314
OLS	Quantile 5%	0.15815	0.70606
OLS	Quantile 10%	0.15815	0.70606
OLS	Quantile 20%	0.15816	0.70606
OLS	Quantile 50%	0.15815	0.70606
OLS	Quantile 75%	0.15815	0.70606
OLS	Quantile 90%	0.15815	0.70606
OLS	Risk-Adjusted Lambda = 10^{-3}	0.029822	0.73684
OLS	Risk-Adjusted Lambda = 10^{-2}	0.026513	0.73442
OLS	Risk-Adjusted Lambda = 10^{-1}	0.096704	0.71141
OLS	Risk-Adjusted Lambda = 1	0.096704	0.71141
OLS	Risk-Adjusted Lambda = 10	0.14708	0.7055
OLS	Risk-Adjusted Lambda = 10^{2}	0.16536	0.48961
OLS	Risk-Adjusted Lambda = 10^{3}	0.17164	0.48728
OLS	Risk-Adjusted Lambda = 10^{4}	0.1717	0.48736
Lasso	Baseline	0.13947	0.56553
Lasso	Sharpe	0.043491	0.9149
Lasso	Quantile 5%	0.18459	0.42364
Lasso	Quantile 10%	0.18459	0.42364
Lasso	Quantile 20%	0.18459	0.42364
Lasso	Quantile 50%	0.18459	0.42363
Lasso	Quantile 75%	0.18064	0.42185
Lasso	Quantile 90%	0.16505	0.46323
Lasso	Risk-Adjusted Lambda = 10^{-3}	0.12133	0.94691
Lasso	Risk-Adjusted Lambda = 10^{-2}	0.10547	0.95833
Lasso	Risk-Adjusted Lambda = 10^{-1}	0.080505	1.0371
Lasso	Risk-Adjusted Lambda = 1	0.080505	1.0371
Lasso	Risk-Adjusted Lambda = 10	0.13803	0.5601
Lasso	Risk-Adjusted Lambda = 10^{2}	0.13935	0.56492
Lasso	Risk-Adjusted Lambda = 10^{3}	0.13947	0.5655
Ridge	Baseline	0.17217	0.30487
Ridge	Sharpe	0.20756	0.43819
Ridge	Quantile 5%	0.16347	0.43876
Ridge	Quantile 10%	0.16347	0.43876
Ridge	Quantile 20%	0.16347	0.43876
Ridge	Quantile 50%	0.16347	0.43876
Ridge	Quantile 75%	0.16347	0.43876
Ridge	Quantile 90%	0.16347	0.43876
Ridge	Risk-Adjusted Lambda = 10^{-3}	0.19314	0.52632
Ridge	Risk-Adjusted Lambda = 10^{-2}	0.19314	0.52632
Ridge	Risk-Adjusted Lambda = 10^{-1}	0.21442	0.38308
Ridge	Risk-Adjusted Lambda = 1	0.21047	0.3539
Ridge	Risk-Adjusted Lambda = 10	0.14217	0.51525
Ridge	Risk-Adjusted Lambda = 10^{2}	0.1799	0.29192
Ridge	Risk-Adjusted Lambda = 10^{3}	0.1719	0.3026

Table 9: Sharpe Ratio Turnover Rate