# **1 Introduction and Literature Review**

Though widely used in portfolio selection, mean-variance (MV) optimization has raised much debate since first introduced by Markowitz (1952). A significant limitation of MV optimization is it is very sensitive to errors in input parameters, particularly the expected return and the covariance matrix for the returns. Chopra and Ziemba (1993) examine the effect of misspecification in mean, variance and covariance in terms of cash equivalent loss under the assumptions that no short selling, a negative exponential utility function, a joint normal distribution of returns, and investors have constant absolute risk aversion. They find that loss resulting from misspecification in mean is the most substantial and errors in mean are even more important at higher risk tolerance levels. Specifically, errors in mean are around ten times as important as errors in variance, and 20 times as errors in covariance.

Attempts have been made to reduce the effects of estimation errors and reach a more stable and better mean-variance optimal portfolio. Ceria and Stubbs (2006) discuss robust optimization, a methodology that directly and explicitly considers estimation errors in unknown parameters in the optimization problem, and largely reduces the negative effect of errors in estimated expected returns. They argue that particularly for commonly used constraints, optimal portfolio calculated using misspecified expected return estimates gives significantly different weights as the true optimal portfolio and overestimates expected return. In addition, they introduce deviations of robust optimization that alleviate the ill-effects of estimation error more effectively than standard form.

We first tested effects of errors in inputs using Chopra and Ziemba’s (1993) data, and affirmed that errors in mean have a larger impact on optimal portfolio than errors in variance and covariance. We also conducted robust optimization and calculated efficient frontiers accordingly, and showed improvements in effects of errors in mean using standard robust optimization and covariance using shrinkage estimates of covariance matrix. Finally we backtested our results using selected ETFs.

# **2 Empirical Study and Results**

## **2.1 Data and Working Environment**

For empirical study, we used 2 datasets in this part. The first one is a simulated dataset, whose return series is generated according to the given mean vector and covariance matrix. The second one is collected from Yahoo Finance, consisting of monthly closing price data of 9 ETFs[1] from January 2011 to July 2020.

To deal with all of the optimization problems in our study, we applied the CVXOPT package for convex optimization based on the Python programming language.

## **2.2 The Effects of Errors in Inputs**

Similar to Chopra (1993), we used cash equivalent loss (CEL) to measure the effects of errors in means, variances and covariances separately. Assuming a negative exponential utility function, the cash equivalent (CE) value of the portfolio is calculated as the inverse function of the utility function:

where t is the risk tolerance parameter.

Using the original mean vector and covariance matrix to conduct Markowitz mean-variance optimization and get optimal portfolio o, we could calculate the cash equivalent value of this optimal portfolio, denoting as . By replacing the means by , where are independently standard normal distributed random numbers, we could get a new optimal portfolio x under changing mean vector and original covariance matrix. The cash equivalent value of portfolio x is calculated based on the new optimal weights and original inputs, denoting as . Then cash equivalent loss (CEL) from errors in means is calculated as:

Similarly, by changing variances from to and covariances from to separately, it would be easy for us to calculate cash equivalent loss (CEL) from errors in variances and covariances.

Exhibit 1 shows the mean cash equivalent loss over 100 iterations under different values of risk tolerance parameter and size of error using simulated dataset.

Compared to Chopra’s (1993) results, which shows that for a risk tolerance of 50, the importance of errors in means is 10 times as important as errors in variances, and 20 times as important as errors in covariances, we found that the effects on errors in means is much more important in our observation. Depending on the size of error, the importance of errors in means is 100 times, even up to millions of times, as important as the effects on errors in variances and covariances for a risk tolerance of 50.

As the risk tolerance increases, the importance of errors in means becomes much larger than the importance of errors in variances and covariances. For example, when risk tolerance is small, the importance of errors in means is about 10 times as important as the effects on errors in variances and covariances, while getting millions of times important when risk tolerance becomes larger.

Also, we found that the importance of errors in variance and covariances are very close to each other. And both of them tend to get smaller when the risk tolerance parameter increases.

**EXHIBIT 1**

**Mean Cash Equivalent Loss for Errors of Different Sizes and Different Risk Tolerance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| t (Risk Tolerance Parameter) | Parameter with Error | k = 0.05 | k = 0.10 | k = 0.15 | k = 0.20 |
| 0.5 | Means | 0.023941 | 0.049128 | 0.077237 | 0.102076 |
| 0.5 | Variances | 0.002839 | 0.00901 | 0.01528 | 0.018287 |
| 0.5 | Covariances | 0.003412 | 0.010755 | 0.020157 | 0.031821 |
|  |  |  |  |  |  |
| 1.0 | Means | 0.026043 | 0.058898 | 0.083466 | 0.114146 |
| 1.0 | Variances | 0.002597 | 0.008264 | 0.014138 | 0.022236 |
| 1.0 | Covariances | 0.003275 | 0.010036 | 0.018742 | 0.035428 |
|  |  |  |  |  |  |
| 5.0 | Means | 0.005611 | 0.031889 | 0.050273 | 0.084032 |
| 5.0 | Variances | 5.92E-08 | 0.000245 | 0.002268 | 0.006235 |
| 5.0 | Covariances | 7.53E-09 | 6.61E-08 | 0.000318 | 0.001127 |
|  |  |  |  |  |  |
| 10.0 | Means | 0.000631 | 0.017639 | 0.064921 | 0.065124 |
| 10.0 | Variances | 3.45E-08 | 1.07E-07 | 1.06E-07 | 4.44E-08 |
| 10.0 | Covariances | 1.49E-08 | 4.05E-08 | 1.22E-07 | 1.39E-07 |
|  |  |  |  |  |  |
| 50.0 | Means | 2.11E-07 | 0.010977 | 0.054027 | 0.075027 |
| 50.0 | Variances | 1.24E-09 | 6E-09 | 8.92E-09 | 1.42E-08 |
| 50.0 | Covariances | 2.47E-09 | 6.39E-09 | 7.23E-09 | 1.16E-08 |

## **2.3 Robust Optimization Implementation**

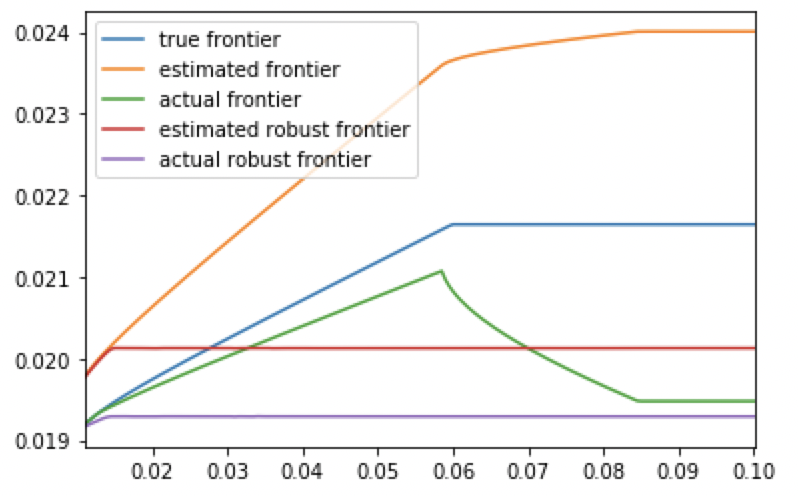
### **2.3.1 Efficient Frontier**

Referring to Ceria and Stubbs (2006), we calculated three types of frontiers: true frontier, estimated frontier, and actual frontier. True frontier is the frontier calculated using the true, but unknown, expected returns. Estimated frontier is the one given by expected returns estimates and the true covariance matrix. Calculating the expected return of the portfolio on the estimated frontier using true expected returns, and we can get the actual frontier using this return and true covariance matrix. According to this definition, the estimated frontier can lie anywhere, while the true frontier will always lie above the actual frontier. Also, the actual frontier is not necessarily concave because it is computed using the result from the estimated frontier instead of an optimization problem.

Using the data from Chopra’s paper, we applied mean-variance optimization and robust optimization and calculated their respective frontiers. We can see that the actual frontier lies below the true frontier, and the gap between estimated robust frontier and actual robust frontier is smaller than the one between true frontier and estimated frontier, meaning that robust optimization alleviates the effect of errors in estimation of expected returns.

**FIGURE 1**

**Markowitz Efficient Frontier and Robust Efficient Frontier**



### **2.3.2 Improvements on Effects of Error in Means**

As illustrated in 2.3.1, incorporating the estimation error into the portfolio construction process reduced its effect on the optimal portfolio. The predicted return for any given risk level was not exaggerated nearly as much, the actual robust frontiers are much closer to the true frontier than are the actual mean-variance frontiers. To further test the sensitivity of error for robust optimization, we apply the same methodology from 3.2 on robust optimization to see whether there are significant improvements of stability on effects of error.

Figure 2 shows the plots of mean cash equivalent loss over 200 iterations and size of error varied from 0 through 1 in steps of 0.05 using simulated dataset under different risk tolerance levels.

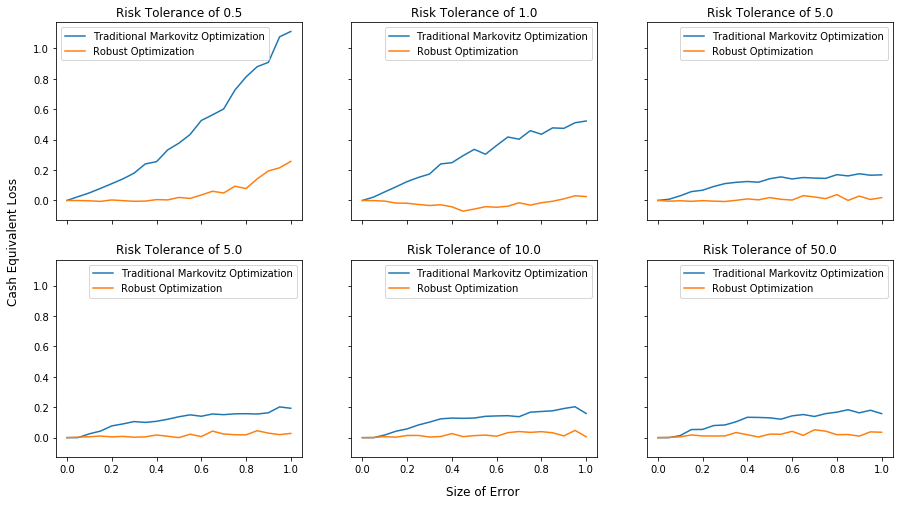
Robust optimization produced much lower cash equivalent loss than the mean-variance method under every risk tolerance level. And as the size of error increases, the cash equivalent loss didn’t increase significantly as the mean-variance method did, even staying relatively stable (less than 0.05) when the risk tolerance level was greater than 1.

Also, we found that the same as the mean-variance method, the cash equivalent loss of robust optimization tended to get smaller when the risk tolerance parameter increases, and was almost the same under different risk tolerance levels that were greater than 1.

As the results showed, robust optimization is much stabler and less sensitive to the change of mean and does overcome the ill-effects of classical portfolio optimization caused by the error-maximisation property.

**FIGURE 2**

**Mean Cash Equivalent Loss for Errors in Means through Markowitz and Robust Optimization**



## **2.4 Shrinkage Estimator of Covariance Matrix**

Statistical Shrinkage allows us to find the optimal trade-off between sample risk and model risk. In particular, we are not going to choose between either or another, but in fact, we mix them.

The idea is based on the average of two covariance matrix estimates, one with high sample risk and one with high model risk:

where is the factor-model based estimator of the covariance matrix, , which is instead an estimator of the covariance matrix which is not factor-model based, and is the shrinkage factor range is in (0,1).

If we use the ETFs data which is the same as the 3.5 back-test, and choose Fama-French three factor model to estimate the covariance matrix, we could easily see the reduced effect of covariance errors with shrinkage covariance model compared to the traditional Markovitz optimization. First of all, let us look at the asset allocation in Exhibit 2 (risk tolerance parameter= 1, iter\_num = 300):

**EXHIBIT 2**

**Optimal Portfolio Weights for Traditional Markovitz and Shrinkage Covariance Models**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | VB | VGLT | GLD | VOD | SHY | SPIB | VWO | DBC | VO |
| Traditional Markovitz | 0 | 0.164 | 0 | 0.836 | 0 | 0 | 0 | 0 | 0 |
| Shrinkage Covariance | 0 | 0.253 | 0 | 0.747 | 0 | 0 | 0 | 0 | 0 |

The result makes sense, since performing statistical shrinkage is equivalent to imposing constraints on weights. when doing shrinkage, we're actually shrinking the dispersion of the input parameter value, reducing the distribution of the inputs is equivalent to reducing the distribution of the outputs. In traditional Markovitz optimization, we put too much weight on the VOD, however, with shrinkage covariance model, the weight on VOD reduces to 0.747 from 0.836. Next, let us test the mean CEL for the two models in Exhibit 3:

**EXHIBIT 3**

**Mean Cash Equivalent Loss (CEL) in Covariance Error**

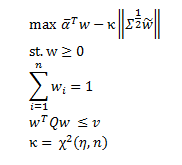
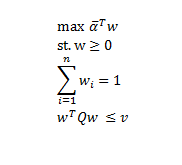
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | k = 0.05 | k = 0.10 | k = 0.15 | k = 0.20 |
| Traditional Markovitz | 0.000010 | 0.000040 | 0.000102 | 0.00025 |
| Shrinkage Covariance | 0.000005 | 0.000016 | 0.000044 | 0.00012 |

From the above result, we can see that with shrinkage covariance model, mean CEL in covariance error almost reduces to half of that with the traditional Markovitz optimization. Therefore, we can conclude that shrinkage covariance is an efficient method in robust covariance optimization.

## **2.5 Back-test Results**

In this part, we implemented both traditional mean-variance optimization and robust optimization on selected ETFs to construct portfolios. On the first transaction day during the back-test period through Feb. 2016 to Jun. 2020, we used the previous 60 months as the window length to estimate sample mean and sample covariance, and made changes on the portfolios according to the optimization results.

To make the portfolios more comparable, we set an upper bound for the variance of the portfolio in the constraints, meaning that we got the optimal weights of the underlying assets by solving the optimization problems below:



Where is the sample mean vector, is the sample covariance matrix, is the upper bound for portfolio variance, is the covariance matrix of the estimated expected mean vector, is the parameter controlling for the uncertainty of the estimated expected mean vector.

Setting and annualized portfolio volatility upper bound equals 0.05, 0.08, 0.10, 0.12, 0.15, we got the portfolio and analyzed their performances through metrics including annualized return, annualized volatility, Sharpe ratio, maximum drawdown, as shown in Exhibit 4:

**EXHIBIT 4**

**Performance Evaluation Results of Back-test Portfolios**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Volatility bound | Optimization method | Annualized return | Annualized volatility | Sharpe ratio | Maximum drawdown |
| 0.05 | mean-variance | 8.34% | 6.19% | 1.347 | 6.69% |
| 0.05 | robust | 5.97% | 5.08% | 1.174 | 6.98% |
| 0.08 | mean-variance | 12.27% | 9.13% | 1.343 | 10.38% |
| 0.08 | robust | 7.19% | 6.60% | 1.090 | 10.24% |
| 0.10 | mean-variance | 12.12% | 11.70% | 1.036 | 14.93% |
| 0.10 | robust | 7.98% | 7.26% | 1.098 | 11.10% |
| 0.12 | mean-variance | 11.06% | 13.29% | 0.832 | 19.16% |
| 0.12 | robust | 8.20% | 7.36% | 1.113 | 11.10% |
| 0.15 | Mean-variance | 10.91% | 13.71% | 0.796 | 19.91% |
| 0.15 | robust | 8.20% | 7.37% | 1.113 | 11.10% |

When the annualized portfolio volatility upper bound is small, the traditional mean-variance optimization portfolio performs better than the robust optimization portfolio. As the bound increases, the performance of the traditional mean-variance optimization portfolio exacerbates, and the robust optimization portfolio gradually stands out on the aspects of volatility, Sharpe ratio and maximum drawdown.

# **3 Conclusion**

We showed that traditional mean-variance optimization is sensitive to errors in input parameters. For a risk tolerance of 50, the importance of errors in mean can be larger than 100 times as important errors in variance and covariance. Errors in variance and covariance have a similar effect, and the importance of errors in mean becomes much larger than errors in variance and covariance as risk tolerance level increases.

Using robust optimization can alleviate the illness effects of estimation errors. The gap between estimated frontier and actual frontier calculated using robust optimization is smaller than the one using mean-variance optimization. Under every risk level we tested, robust mean optimization produced much lower cash equivalent loss than mean-variance optimization. And with shrinkage covariance model, mean cash equivalent loss in covariance error almost reduces to half compared to traditional mean-variance optimization.

Our backtested results show that when allowing low variance for the optimized portfolio, the mean-variance optimization portfolio performed better. But as allowing higher variance, robust optimization performed better in terms of volatility, sharpe ratio and maximum drawdown. We conclude that compared to traditional mean-variance optimization, robust optimization is more stable, less sensitive to changes of input parameters, and could result in a better optimized portfolio using real-world data.

[1] List of 9 chosen ETFs:

VOO: Vanguard S&P 500 ETF

VO: Vanguard Mid-Cap Index Fund ETF Shares

VB: Vanguard Small-Cap Index Fund ETF Shares

VWO: Vanguard FTSE Emerging Markets Index Fund ETF Shares

SHY: The iShares 1-3 Year Treasury Bond ETF

VGLT: The Vanguard Long-Term Government Bond ETF

SPIB: SPDR Portfolio Intermediate Term Corporate Bond ETF

DBC: Invesco DB Commodity Index Tracking Fund

GLD: SPDR Gold Shares