Portfolio Optimization

December 15, 2024

1 Introduction

This report explores the implementation of the mean-variance portfolio optimization framework. Historical data of 74 stocks will be used to illustrate how returns can be maximised for a given level of risk or how risk can be minimized for a given return level. Data collection and analysis was done using Python 3 and several libraries.

2 Data Collection

Historical data of 100 stocks from NASDAQ-100 index was retrieved using the 'yfinance' library which contains stock data provided by Yahoo Finance. Stocks with NA values from all analysed time periods were removed, leaving 74 stocks for analysis. The time period used for the initial analysis is 31 January 2019 to 31 December 2023.

Daily returns are likely to have more noise which may obscure the trend while yearly returns have much fewer data points which may provide inaccurate estimates of variance. Therefore, we use monthly returns as a good compromise to accurately estimate returns and risk. Since the data provided by 'yfinance' is daily, we retrieve the price on the last day of each month and calculate monthly returns using the following formula:

$$i^{th} \text{ month return} = \frac{\text{Close Price on last day of } i^{th} \text{ month}}{\text{Close Price on last day of } (i-1)^{th} \text{ month}} - 1 \tag{1}$$

Some data is shown for illustration:

Table 1: Monthly Returns

Ticker Date	AAL	AAPL	ABBV	ABT	ACN
2019-01-31	0.113983	0.055154	-0.129081	0.008987	0.088930
2019-02-28	-0.003914	0.040315	-0.013078	0.063579	0.050993
2019-03-31	-0.108616	0.097026	0.017037	0.029889	0.090718
2019-04-30	0.076196	0.056436	-0.014890	-0.004754	0.037780
2019-05-31	-0.203335	-0.127573	-0.033757	-0.043112	-0.025182

3 Preparing Data for Optimization

Next, we calculate the necessary quantities for portfolio optimization; Expected returns are calculated by taking the mean of the monthly returns of each stock. Volatilities are calculated by taking the standard deviation of the monthly returns. Covariance between 2 stocks X_1 and X_2 is calculated using the following formula:

$$\mathrm{Cov}(X_1, X_2) = \mathrm{E}[X_1X_2] - \mathrm{E}[X_1]\mathrm{E}[X_2] \tag{2}$$

Then, the covariance matrix is as follows:

$$\text{Covariance Matrix} = \begin{bmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_{100}) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) & \cdots & \mathrm{Cov}(X_2, X_{100}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_{100}, X_1) & \mathrm{Cov}(X_{100}, X_2) & \cdots & \mathrm{Cov}(X_{100}, X_{100}) \end{bmatrix} \tag{3}$$

While the empirical covariance matrix can be useful, it provides pairwise correlations, which may be more volatile to changes and contain noise. To reduce this volatility, a shrinkage estimator could be used instead, to improve the stability of the covariance matrix.

In this paper, we will explore the use of the Ledoit-Wolf shrinkage which shrinks the covariance matrix towards a target, in our case a diagonal matrix. This reduces the effect of noise since the off-diagonal elements are shrunken towards 0. This stabilizes the covariance matrix and reduces the effect of noise on the covariance matrix. The formula is shown below. α is calculated such that it minimizes the Mean Squared Error between the estimated and the real covariance matrix.

$$\Sigma_{\text{shrunk}} = \alpha T + (1 - \alpha)\Sigma \tag{4}$$

$$\alpha = \frac{\mathrm{tr}\left((\Sigma - T)^2\right) - \mathrm{tr}\left(\Sigma^2\right)}{\mathrm{tr}\left((\Sigma - T)^2\right) - N \cdot \mathrm{tr}\left(T^2\right)} \tag{5}$$

where:

- Σ_{shrunk} is the shrunk covariance matrix,
- T is the target matrix
- Σ is the empirical covariance matrix,
- α is the shrinkage intensity, which minimizes the estimation error.
- N is the number of assets

There are numerous other ways to estimate the covariance matrix. Some of these include: the Oracle Approximating Shrinkage which uses a different approach to find α, use of factor models such as the Capital Asset Pricing Model which decomposes risk into different components and machine learning methods which use neural networks to predict covariance matrices. Later on, we will compare the stability of portfolios formed using empirical covariance and shrunk covariance.

4 Mean-Variance Optimization

Now we implement and solve the mean-variance optimization to find how returns can be maximised for a given level of risk or how risk can be minimized for a given return level. For our purposes, we assume that there is no short selling. Therefore, all the components of our weight vector will be greater than or equal to 0. Lastly, we want to ensure that we utilize all our resources which means the sum of the components of the weight vector will be equal to 1. We use the CVXPY Python library to solve our optimization problems.

4.1 Framing the Optimization Problem

Let:

- $\mu \in \mathbb{R}^n$ be the vector of expected returns of the n stocks.
- $\Sigma \in \mathbb{R}^{n \times n}$ be the covariance matrix of returns. The covariance matrix is positive semi-definite.
- $w \in \mathbb{R}^n$ be the vector of portfolio weights.
- R be the expected return of the portfolio.

Therefore,

$$\text{Portfolio Variance} = w^\top \Sigma w \tag{6}$$
$$\text{Portfolio Return R} = w^\top \mu \tag{7}$$

4.2 Minimizing Variance

To minimize variance, we set up the optimization problem as follows:

$$\min_w [w^\top \Sigma w] \tag{8}$$

subject to the constraints:

$$\begin{aligned} & w \geq 0 \\ & w^\top \mathbb{1} = 1 \end{aligned} \tag{9}$$

This is a convex optimization problem since the covariance matrix Σ is positive semi-definite. We use the CVXPY library in Python to solve for w which gives us the minimum variance portfolio.

4.3 Minimizing Variance for a given return

Similar to 4.2, we simply add the constraint of the portfolio return R being greater than or equal to a given return. The optimization problem is as follows:

$$\min_w [w^\top \Sigma w] \tag{10}$$

subject to the constraints:

$$\begin{aligned} & w \geq 0 \\ & w^\top \mathbb{1} = 1 \\ & R \geq R_1, (R_1 \text{ is the given minimum return}) \end{aligned} \tag{11}$$

This is also a convex optimization problem for the same reasons as in 4.2. We use CVXPY to solve this as well.

4.4 Maximizing Return for a Given Variance

Now, we want to maximize the return for a given minimum variance. Thus, we frame the optimization problem as follows:

$$\max_w [w^\top \mu] \tag{12}$$

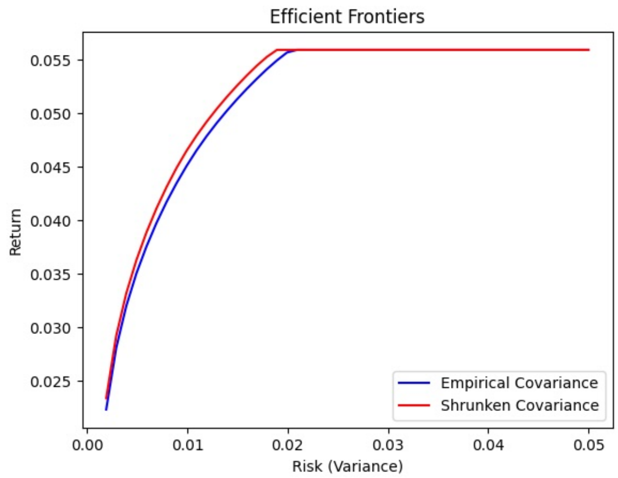
subject to the constraints:

$$\begin{aligned} & w \geq 0 \\ & w^\top \mathbb{1} = 1 \\ & w^\top \Sigma w \leq V, (V \text{ is the given maximum variance}) \end{aligned} \tag{13}$$

Here, we are maximizing a linear function, with a constraint of a maximum variance, which is a convex function. Overall, this makes the problem a convex optimization problem. Again, we use CVXPY to solve this.

4.5 Efficient Frontier

The efficient frontier represents the set of optimal portfolios for different risk-return trade-offs. Intuitively, we might expect risk and return to be linearly related, with greater risk giving rise to greater return. However, as shown in this plot, we can observe that plotting the efficient frontier portfolio return against variance of the portfolio gives us a parabola. Here, we compare the Efficient Frontiers of the shrunken covariance portfolios and empirical covariance portfolios.



It should be mentioned that in this case, the efficient frontier formed when using the Ledoit-Wolf shrunken covariance was largely similar to the one obtained when using the empirical covariance. However, this is not always the case.

4.6 Stability of Solutions

Now, we study how stable the solutions are by comparing the weights of the minimum variance portfolio between different time periods. We compare the weights by finding the euclidean distance between the weight vectors to see how far apart the solutions are. To have a point of reference to know how large the distances are, we compare to the maximum possible euclidean distance between 2 portfolios. The maximum distance would be when one portfolio has all its weight on one stock while the second portfolio has all its weight on a different stock. Then, the euclidean distance between these 2 portfolios is given by:

$$\text{Maximum Euclidean Distance } D_{max} = ||e_i - e_j|| = \sqrt{2} \tag{14}$$

We compare 3 different 5-year periods to see how stable the solutions are: 2019-2023, 2014-2018, 2009-2013. We also compare the stability of the portfolios formed using the shrunken covariance and the empirical covariance.

$$T_1 \text{ vs } T_2 = ||w_{2019-2023} - w_{2014-2018}|| = 28.7\% \text{ of } D_{max} \tag{15}$$
$$T_1 \text{ vs } T_2 \text{ (Ledoit-Wolf)} = ||w_{2019-2023} - w_{2014-2018}|| = 19.0\% \text{ of } D_{max} \tag{16}$$
$$T_2 \text{ vs } T_3 = ||w_{2014-2018} - w_{2009-2013}|| = 28.1\% \text{ of } D_{max} \tag{17}$$
$$T_2 \text{ vs } T_3 \text{ (Ledoit-Wolf)} = ||w_{2014-2018} - w_{2009-2013}|| = 16.4\% \text{ of } D_{max} \tag{18}$$
$$T_1 \text{ vs } T_3 = ||w_{2019-2023} - w_{2009-2013}|| = 22.8\% \text{ of } D_{max} \tag{19}$$
$$T_1 \text{ vs } T_3 \text{ (Ledoit-Wolf)} = ||w_{2019-2023} - w_{2009-2013}|| = 20.1\% \text{ of } D_{max} \tag{20}$$

Overall, we can see that there is a moderate variation between the minimum variance portfolio weights which suggests that there are somewhat significant changes in the markets even between 5-year periods.

We also observe that the Ledoit-Wolf shrinkage stabilizes the minimum variance portfolio between the time periods. This would mean that less re-balancing of the portfolio is needed if covariance is shrunken using Ledoit-Wolf. This would likely lead to lower transaction costs in practice. Furthermore, given the efficient frontier curves observed in **4.5**, we know that the Ledoit-Wolf shrinkage generally allows for equal/better returns. Taking these equal/better returns into account as well as the lower transaction costs, using the Ledoit-Wolf shrinkage allows us to generate greater net profit with our portfolio.

5 Conclusion

In this paper, we analysed stock data of 74 stocks from NASDAQ-100. We used the most recent time period 2019-2023 to illustrate the mean-variance portfolio optimization framework for various optimization objectives. We examined 2 ways to form the covariance matrix, the empirical covariance and the shrunken covariance formed using the Ledoit-Wolf shrinkage. We then plotted the efficient frontier which is the set of optimal portfolios for different risk-return trade-offs which depicted a parabola. We observed that the efficient frontiers of portfolios formed using the different covariance estimation methods were similar. Next, we compared the stability of the minimum variance portfolio weight vector over 3 time periods and found that there was moderate variation. Finally, we found that using the Ledoit-Wolf shrinkage for the covariance matrix allowed us to have a more stable portfolio which would likely lead to better profits.

References

[gru23] gru.ai. How to get a complete and up-to-date list of ticker symbols. https://stackoverflow.com/questions/78364453/how-to-get-a-complete-and-up-to-date-list-of-ticker-symbols-for-any-yahoo-financ, 2023. Accessed: 2024-10-23.

[Gun22] Gregory Gundersen. Geometry of the efficient frontier. https://gregorygundersen.com/blog/2022/01/09/geometry-efficient-frontier/#:~:text=Figure%20II%20graphs%20%5Bthis%5D%20frontier,the%20bullet%20is%20a%20hyperbola., 2022. Accessed: 2024-10-23.

[Sl24] Scikit-learn. Ledoitwolf. https://scikit-learn.org/1.5/modules/generated/sklearn.covariance.LedoitWolf.html, 2024. Accessed: 2024-11-08.