TEXAS McCombs
The University of Texas at Austin
McCombs School of Business

Project 2 - Advanced Corporate Finance

Humphrey Hui

Peyton Gibbs

Jack Feen

Overview

In this project, we investigated the different effects of a sample covariance matrix vs Ledoit-Wolf shrinkage for a given stock portfolio's weights, returns, and volatility applied to a global minimum variance portfolio. Additionally, we will investigate how changing the backtesting lengths affects the performance differences between the Ledoit-Wolf and sample covariance methods.

Methodology

To start off, we wrote the function data_and_constraints_acquisition(). First, we queried the user to collect the stock ticker symbols they wanted to add to their portfolio. Then, we wrote a SQL query that collected the date and monthly return for each stock between Jan 1, 2010 and Dec 31, 2024 from CRSP.MSF in WRDS.

Once this information was consolidated into a dataframe, we reindexed to include all months to flag missing data as NaN, making sure to use the last day of the trading month. We then ran a check for any series of three consecutive months that was missing data (i.e. an entire quarter) or more than 10% of the months were missing, and prompted users to use a different stock if the check failed. We also filled in any missing months with 0, assuming that there was no change in the stock price during those months. Once all the stocks passed this check, we stored the dataframe of monthly returns as crsp_wide.

Finally, we created a dictionary to keep track of the user's constraints on the stock weights (e.g. they did not want a certain stock to exceed 75% or fall below -10%). We had the user pick from four options: both minimum and maximum constraints, only a minimum constraint, only a maximum constraint, and no constraints. We stored this information as a dictionary optimization_constraints. Finally, the function would return crsp_wide and optimization constraints.

After this, we developed functions to optimize the Global Minimum Variance Portfolio of our stocks, using sample covariance and Ledoit-Wolf, which shrinks stock covariances to adjust for noise and increases performance on out-of-sample stock data. We first calculate demeaned returns and the covariance matrix to use as parameters to optimize our data with CVXPY. We used the quadratic solver within this package. In our stress testing, we found an example where small floating point rounding errors made a non-symmetric covariance matrix - so we used an LLM to create a function that enforces symmetry.

Finally, we made a function that uses a rolling period of months to re-optimize our portfolio on a month-to-month basis. In our analysis, we looked into the effects of different backtesting periods and the choice of Ledoit-Wolf vs Sample Covariance on our portfolios out of out-of-sample returns and weights. While computing returns on different lookback periods for backtesting, we ensured that we calculated these tests on the same period of data, even between lookback periods of differing sizes.

Parameter Choices

We chose to collect the monthly returns from Jan 1, 2010 to Dec 31, 2024 as this provided us with 15 years of data - more than enough to compute rolling backtesting periods from a few months to multiple years. Furthermore, we limited the stock to U.S. common stocks listed on NYSE, AMEX, or NASDAQ to ensure data quality and avoid introducing unnecessary noise to the model via securities that are not standard.

Next, we chose to exclude stocks with three or more consecutive months of missing data or more than 10% of the total monthly returns missing. We felt that these checks helped maintain the quality of our data and ensure that the assumption we made for missing months (0% change) was not overused. Instead, we would rather just ask the user for a new stock if the checks failed for better stock data.

During the GMV calculations, we chose to use CVXPY to optimize our GMV portfolio, with the restrictions being what the user passed for minimum and maximum weights. We used the Minimize, Problem, and Solve functions of CVXPY with the solver being the OSQP method. This is the splitting quadratic program solver, as it handles complex quadratic problems with linear constraints. For our shrinkage method, we used the Ledoit-Wolf method. We chose this method because we were looking to reduce the sample noise and overfitting on the test data.

Finally, we allowed the user to input the rolling backtesting period (in months) for analysis. On our variance and cumulative return plots, we used a list of periods ranging from 4 to 60 months, incrementing by 4 months each time.

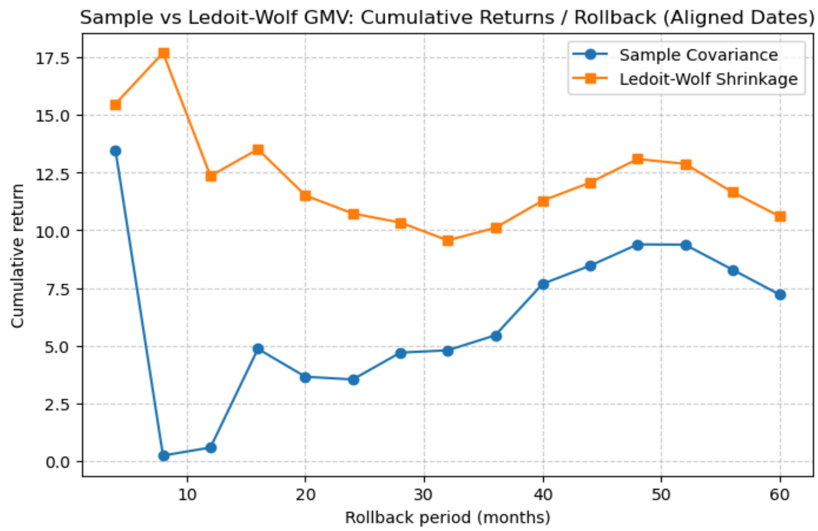
Challenges Faced

During our initial testing of the data_and_constraints_acquisition() function, we found that 30% of Apple's monthly returns were missing, which did not make sense. Further investigation found that we were using the last date of each month when CRSP.msf returns the last trading day, and were only capturing returns where the last trading day was the same as the last day of the month. As such, we adjusted our code accordingly to align with the last trading day instead, solving this problem and accurately capturing all returns.

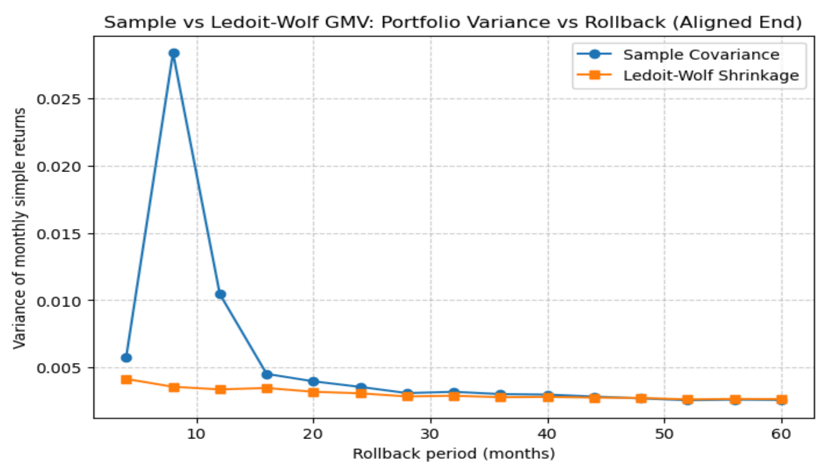
Another challenge we faced was during the GMV calculations, as we were getting a floating-point symmetry error when doing matrix calculations. We were able to solve this by forcing the square matrix to be symmetric, and we clipped tiny negative eigenvalues to ensure we obtain a positive semidefinite (PSD) matrix.

Results Analysis

The Ledoit Wolf, designed to shrink noise, has better performance on out-of-sample data across all rollback periods, showing the effectiveness of the model.

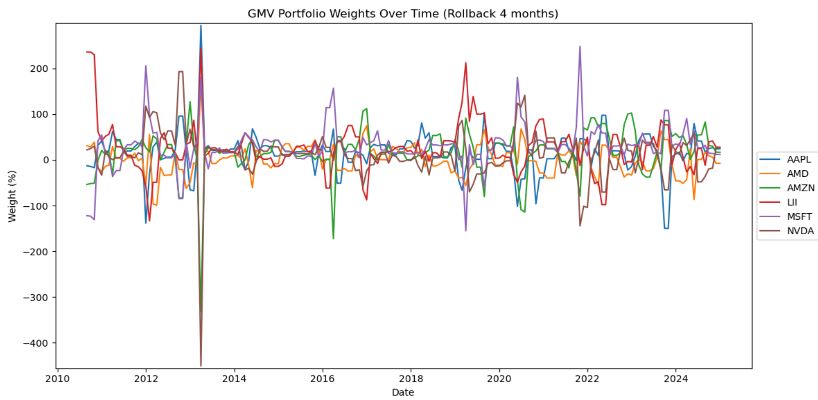


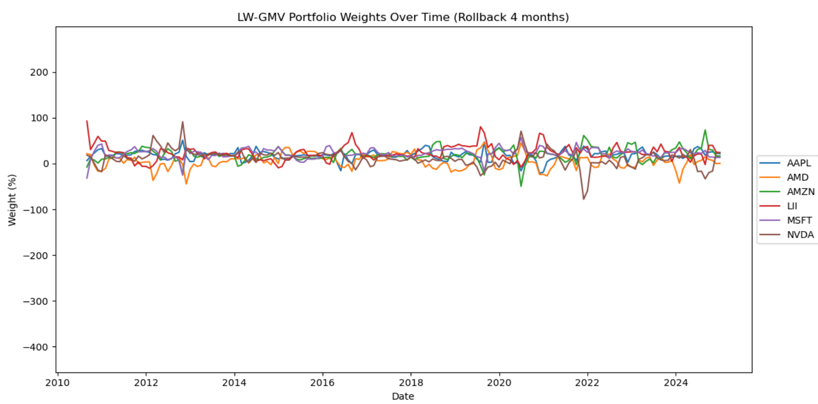
As expected, the sample covariance method starts off with extreme variance that converges to the Ledoit-Wolf variance as we increase the number of rollback periods. Portfolio variance continues to decrease as far as 60 months out for a rollback period.



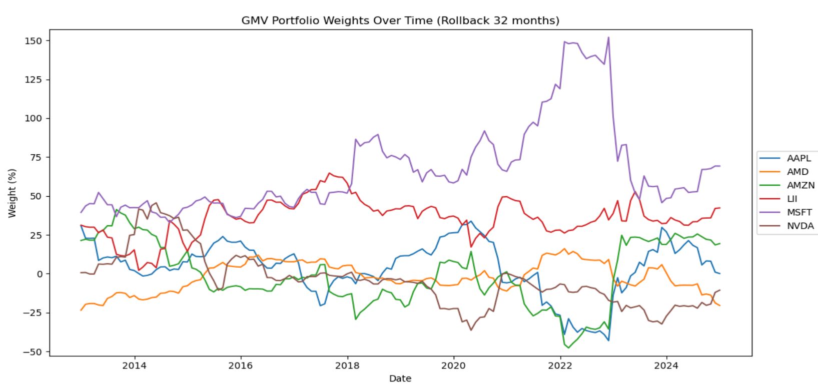
We tested the portfolio weights on a rollback period of 4 months and 32 months. As expected, the Ledoit-Wolf decreased the variance of portfolio weights drastically, especially for the 4-month lookback period with insufficient rank for 6 stocks. At the end of 2024, the 4-month lookback and a portfolio weight standard deviation of 0.13 for sample covariance and 0.08 Ledoit-Wolf. The 32-month rollback had a standard deviation of 0.34 and 0.25, respectively.

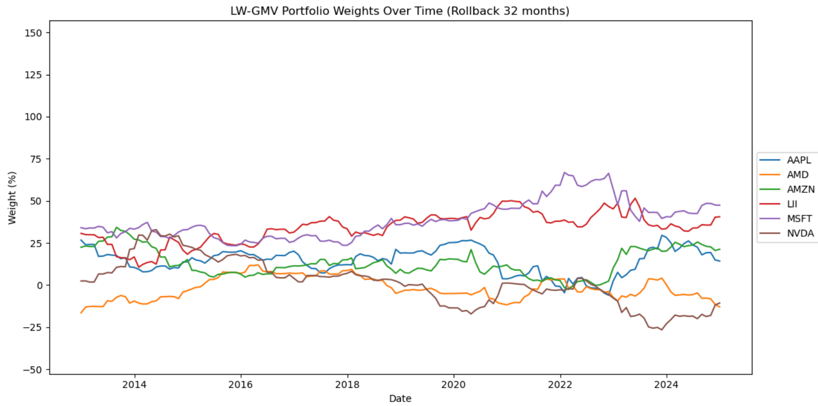
Ledoit-Wolf vs Sample Covariance portfolio weights: 4-month lookback



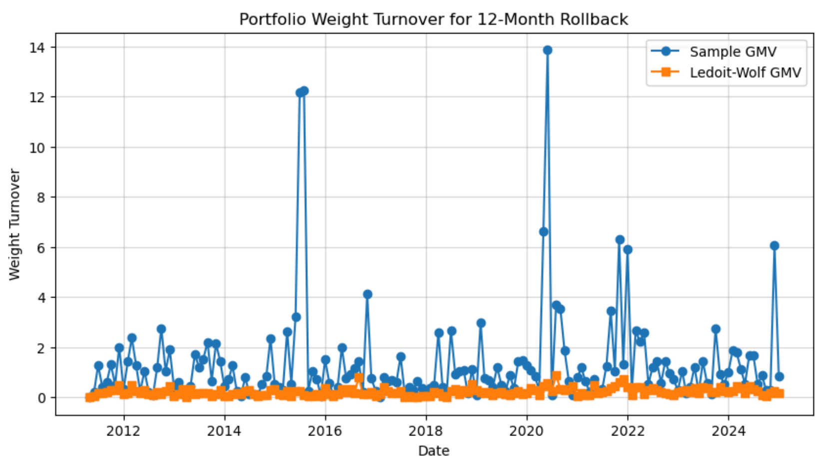


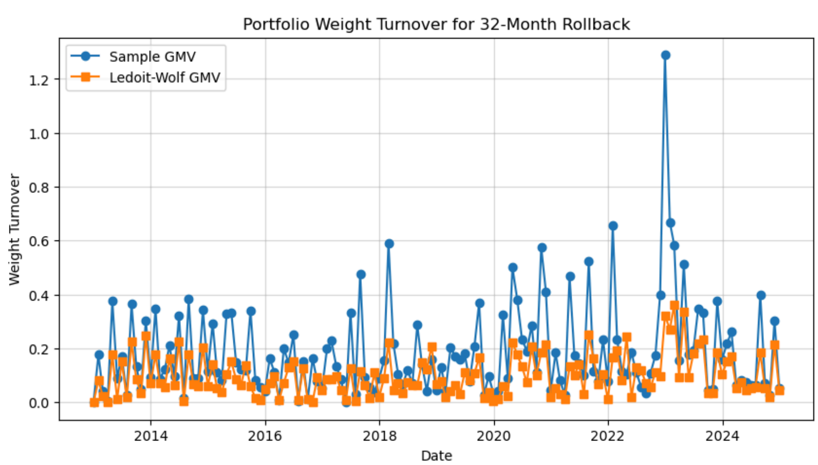
Ledoit-Wolf vs Sample Covariance portfolio weights: 32-month lookback





While visually obvious from the previous charts, plotting the sum of absolute differences in portfolio weights from month to month shows how drastic the effects of Ledoit-Wolf are. For the 32-month lookback period, the average monthly turnover in the Ledoit-Wolf portfolio was half that of the sample covariance, and for smaller lookback periods, those results magnify drastically.





Conclusion

The higher expected return on out-of-sample data, lower volatility, and stability of weights show that the Ledoit-Wolf shrinkage method is an effective portfolio optimization technique. While Ledoit-Wolf is designed for the shrinkage of large portfolios with insufficient returns data, it excels even while working with smaller portfolios with sufficient rank.