Financial Math Mini-Project (Part 3)

Ahmer Nadeem Khan

14 April 2025

1 Introduction

In this part of the mini-project, we want to take values of a representative set of stocks in the US Stock Market over a short past period and compute the Global Minimum Variance Portfolio given by this historical data and the results of Portfolio theory. In Part 2, we used the Ledoit-Wolf Single Factor Estimator. In this part, we additionally use the James Stein Eigenvector (JSE) shrinkage on the eigenvector estimation, and compare to the crude results.

2 Objectives and Methodology

Note that this section is very similar to Part 1, and Part 1 can be consulted for details. Our objectives changes are the following:

1. We consider the top 400 stocks (by weight) in the index S&P500.
2. We obtain weekly adjusted closing prices of each stock for the past 27 weeks, starting from the current date. This means getting the adjusted closing prices starting on 03/28/2025 ("Today") and retrieving data for the preceding 26 weeks. Note that "Today" corresponds to market close on last Friday. Then, close prices are obtained for the required Fridays before. We consider adjusted close prices because these account for dividend payments, stock splits, and other variations.
3. We compute the covariance matrix, the holdings vector for the Portfolio C (GMV Portfolio), the return, variance, and the standard deviation (risk) of the portfolio using the following formulas:

   $$\mathbf{V} = \frac{(\mathbf{r}-\mathbf{f})(\mathbf{r}-\mathbf{f})^\top}{n} \qquad \Sigma = (\lambda^2 - l^2)\,\mathbf{h}\mathbf{h}^\top + \left(\frac{n}{p}\right) l^2\,\mathbf{I}$$

   $$\mathbf{h}_c = \frac{\Sigma^{-1}\,\mathbf{e}}{\mathbf{e}^\top\,\Sigma^{-1}\,\mathbf{e}} \qquad f_c = \mathbf{h}_c^\top \mathbf{f} \qquad \sigma_c^2 = \mathbf{e}^\top\,\Sigma\,\mathbf{e}$$

   where $\mathbf{e} = \mathbf{1}$ is a vector of ones, $\mathbf{V}$ is the covariance matrix, $\mathbf{h}_c$ is the holdings vector of the Portfolio C, $\mathbf{r}$ is the matrix of returns, σ_c^2 is the variance of the portfolio C, and the expected returns vector is $\mathbf{f} = \mathbb{E}[\mathbf{r}]$.

   Σ is an estimator for the covariance matrix, where λ^2 is the leading eigenvalue of V, $\mathbf{h}$ is the corresponding unit eigenvector, $\mathbf{I}$ is the identity matrix of size n, and l^2 is defined as

   $$l^2 = \frac{\mathbf{tr}(V) - \lambda^2}{n-1}$$

   where $\mathbf{tr}(V)$ is the trace of V. This is the Single Factor estimation.
4. We then perform the same computations and present and compare our results.

Please note the following observations:

- We have $p = 400$, the number of assets i.e. 400 stocks.
- We have $n = 26$, the number of time periods over which we have returns, i.e. 26 weeks.
- The size of the matrix of returns, $\mathbf{r}$, is $p \times n$, where each row corresponds to the returns of the i-th asset over the specified period. For our case this becomes 400×26.
- The various issues with data collecting and pre-processing were covered in the last part.

2.1 Covariance Estimator Matrix Σ and JSE Shrinkage

For our setting, our returns matrix $\mathbf{r}$ has size 400×26 where $p = 400$ is the number of assets and $n = 26$ is how often we sample the historical returns (weekly, for 26 weeks). This means when we form the covariance matrix

$$\mathbf{V} = \frac{(\mathbf{r} - \mathbf{f})(\mathbf{r} - \mathbf{f})^\top}{n}$$

The resulting matrix has size 400×400, but since $p > n$ in this case, the covariance matrix is rank-deficient, i.e it has rank at max 26 $\implies$ $\textbf{rank}(\mathbf{V}) \leq 26$. In particular this means that $\mathbf{V}$ is not invertible; which is why our earlier expressions to compute the holdings vectors and associated quantities cannot be used. We instead estimate the covariance matrix with

$$\Sigma = (\lambda^2 - l^2)\,\mathbf{h}\mathbf{h}^\top + \left(\frac{n}{p}\right) l^2\,\mathbf{I}$$

where

- λ^2 is the leading eigenvalue of $\mathbf{V}$.
- $\mathbf{h}$ is the corresponding leading eigenvector of V.
- l^2 is defined as

  $$l^2 = \frac{\textbf{tr}(V) - \lambda^2}{n-1}$$

  This means that l^2 is the average of the eigenvalues of $\mathbf{V}$ smaller than λ^2 (since it is the leading eigenvalue).

This is the single-factor market model, and is called the **Ledoit-Wolf Single-Factor Shrinkage Estimator**. We proved certain results about Σ in the last part:

Proposition 2.1. *Σ is a spd matrix, and therefore, invertible.*

Proposition 2.2. *$\boldsymbol{h}$ is the the leading eigenvector of Σ.*

JSE is a shrinkage estimator that improves on h by having a lower squared error with high probability and leading to better estimates of covariance matrices for use in quadratic optimization [1]. The JSE estimator h^{JSE} is defined by shrinking the entries of h toward their average $m(h)$:

$$h^{\text{JSE}} = m(h)\mathbf{1} + c^{\text{JSE}}(h - m(h)\mathbf{1}),$$

where the shrinkage constant c^{JSE} is

$$c^{\text{JSE}} = 1 - \frac{\nu^2}{s^2(h)},$$

where

$$s^2(h) = \frac{1}{p}\sum_{i=1}^{p} (\lambda h_i - \lambda m(h))^2$$

is a measure of the variation of the entries of λh around their average $\lambda m(h)$, and ν^2 is equal to the average of the nonzero smaller eigenvalues of S, scaled by $1/p$,

$$\nu^2 = \frac{\text{tr}(S) - \lambda^2}{p \cdot (n-1)}.$$

The improvement provided by JSE is substantiated theoretically and numerically. The improved estimator then becomes:

$$\Sigma_{\text{JSE}} = (\lambda^2 - \ell^2)\frac{h^{\text{JSE}}(h^{\text{JSE}})^\top}{|h^{\text{JSE}}|^2} + \left(\frac{n}{p}\right)\ell^2 I$$

3 Results and Conclusions

I will first simply display the required values and variables obtained from the code. All quantities are annualized. I used graphs and charts instead of vector lists, wherever appropriate.

Metric	**PCA Estimate**	**JSE Estimate**
Minimum Variance Portfolio Return	-0.146815	-0.146907
Minimum Variance Portfolio Variance	0.000501	0.000501
Minimum Variance Portfolio Std Dev	0.022376	0.022388

We observe numerically that the variance for the JSE estimate goes up and the returns go down. Theoretically, we know this is statistically a better estimate for the minimum variance portfolio then before. We can visually the shrinkage on the eigenvector through these plots:

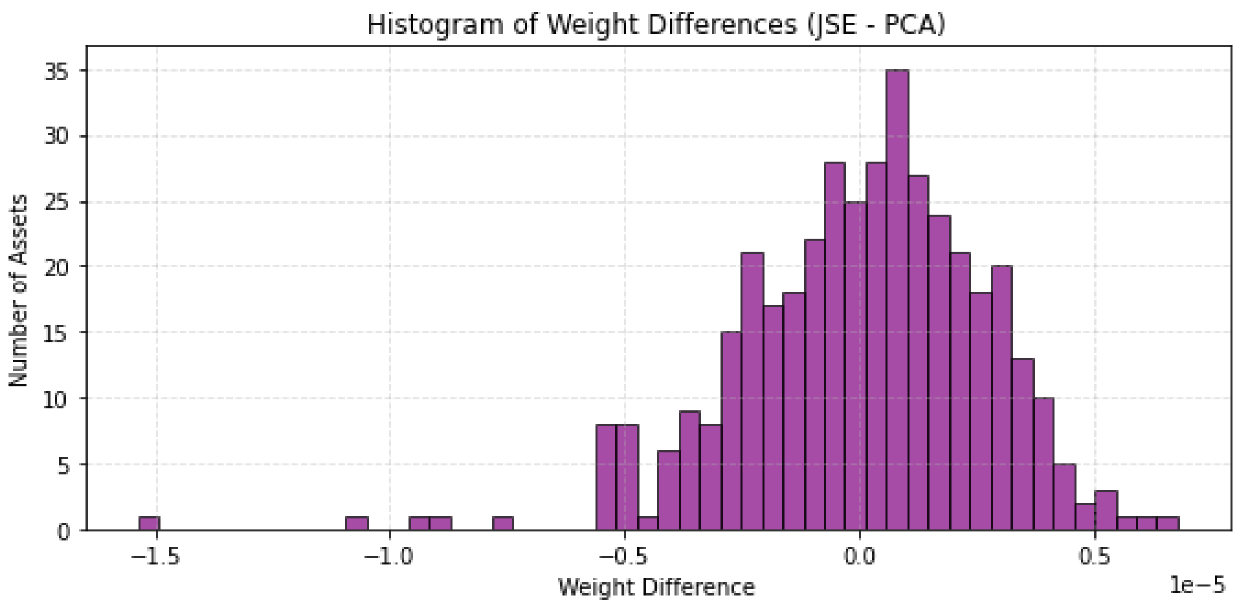
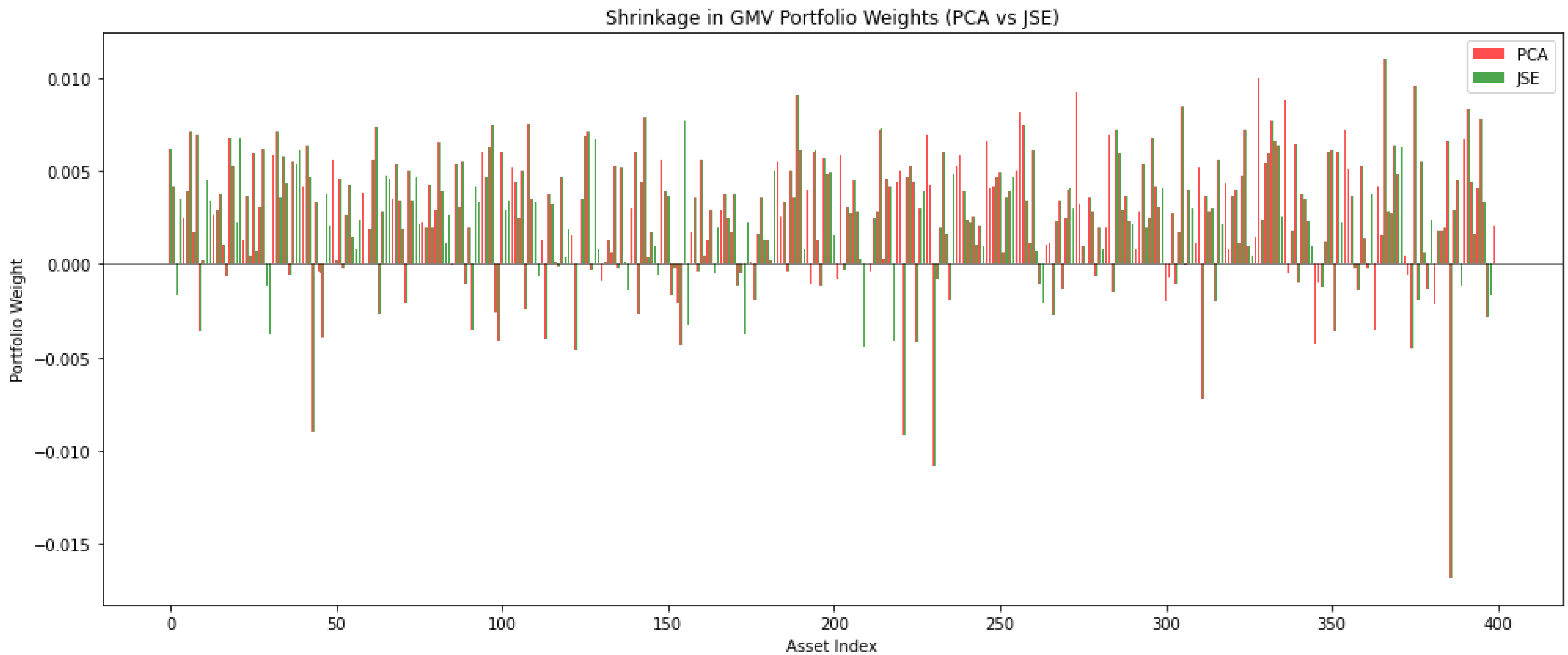


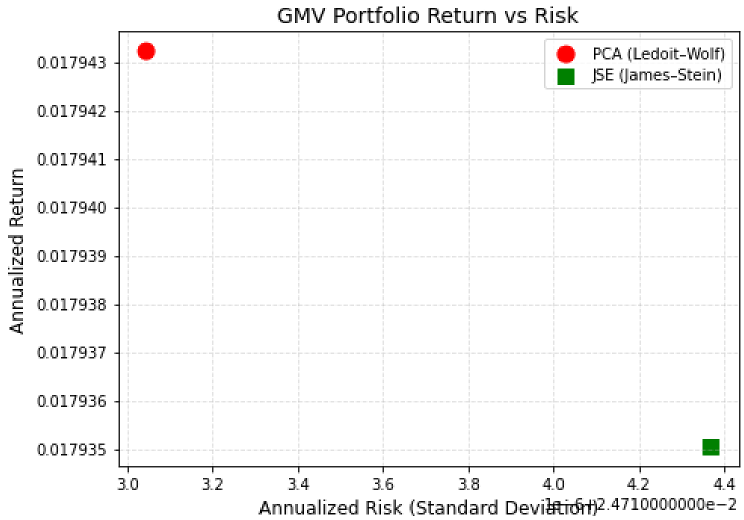
Figure 1: Caption

The difference is more clear from this statistical plot, but we can also plot the weights themselves:


Shrinkage in GMV Portfolio Weights (PCA vs JSE)
PCA
JSE
Portfolio Weight
0.010
0.005
0.000
−0.005
−0.010
−0.015
Asset Index
0
50
100
150
200
250
300
350
400

4 2 Year Horizon with Daily Data

The difference between the JSE and PCA estimator portfolio's is very minimal for the two yearly data, but we do observe a larger variance for the JSE, and lower returns:



Conclusions: We observed shrinkage using JSE of the main eigenvector of the sample covariance matrix for both the shorter time span of 26 weeks and also the larger 2 year timeline with daily data points (though this obviously has very high correlation). To fully ascertain the improvement of JSE, we would require more thorough testing using simulations and statistical properties, to see better results for portfolio variances (as was done in the cited paper). Here, we know that the larger variances we observe (and lower returns) are probably better estimates for our portfolios). Comparing the portfolio differences with the individual stocks on a Risk-Return diagram does not have enough resolution to see the differences, but this was still included in the notebook file as a plot (the higher resolution zoom is shown above).

References

[1] Lisa R. Goldberg and Alec N. Kercheval. James–stein for the leading eigenvector. *Proceedings of the National Academy of Sciences*, 120(2):e2207046120, 2023.

5 Code Appendix

JSE Shrinkage Estimator

```
# Eigen decomposition (cov_matrix is symmetric)
eigvals, eigvecs = np.linalg.eigh(cov_matrix)
lambda_sq = eigvals[-1]
h = eigvecs[:, -1]  # unit eigenvector (length p)

# Average of h
mh = np.mean(h)

# nu² = (tr(S) - lambda²) / (p(n-1))
trace_S = np.trace(cov_matrix)
nu_sq = (trace_S - lambda_sq) / (p * (n - 1))

# s²(h) = lambda² * Var(h) * p
s_sq_h = lambda_sq * np.var(h) * p

# Shrinkage constant
c_JSE = 1 - (nu_sq / s_sq_h)

# Compute h_JSE
h_JSE = mh + c_JSE * (h - mh)

# l² (as in Ledoit-Wolf): average of smaller eigenvalues
ell_sq = (trace_S - lambda_sq) / (n - 1)

# Final JSE covariance estimator:
# sigma_JSE = (lambda² - l²) * h_JSE h_JSE.T / |h_JSE|² + (n/p) * l² * I
norm_sq_h_JSE = np.sum(h_JSE ** 2)
term1 = (lambda_sq - ell_sq) * np.outer(h_JSE, h_JSE) / norm_sq_h_JSE
term2 = (n / p) * ell_sq * np.eye(p)
Sigma_JSE = term1 + term2

# Output summary
print("JSE Covariance Estimate Shape:", Sigma_JSE.shape)
print("lambda² =", lambda_sq)
print("l² =", ell_sq)
print("nu² =", nu_sq)
print("c_JSE =", c_JSE)
```

Note: Visualization code is available in the Jupyter notebook file *mini_project_2.ipynb*.