DSA4212 Portfolio Optimisation

Introduction

Objective

This portfolio optimisation project aims to apply the mean-variance optimisation framework to construct an optimal asset portfolio. The focus is on maximising returns for a specified risk level, balancing growth with controlled risk exposure. This approach is practical for investors seeking higher returns without exceeding risk limits. The project involves calculating expected returns, volatilities, and correlations from historical data, followed by optimisation to determine portfolio weights that achieve this risk-return balance.

Background

Mean-variance optimisation, a key concept in modern portfolio theory, constructs portfolios that maximise expected returns for a given risk or minimise risk for a target return. By balancing risk (variance) and return, it identifies efficient portfolios along the "efficient frontier." This framework supports informed asset allocation by incorporating expected returns, volatilities, and correlations, thus enhancing risk diversification. However, it is sensitive to input estimates and relies on historical data, which may not accurately predict future performance. This project implements the framework and tests solution stability under different scenarios.

Approach Overview

This project is structured into three stages:

1. **Data Collection**: Gather historical financial data for a diversified asset set.
2. **Optimisation**: Implement mean-variance optimisation using the CVXPY library to determine optimal portfolio weights. Visualise the results on an efficient frontier plot.
3. **Stability Analysis**: Test the robustness of the optimised portfolios and provide insights into how changes in market conditions may impact portfolio performance.

Data Collection

Data Source and Timeframe

The historical financial data for this project was obtained using the `yfinance` library in Python, which provides an efficient and reliable way to access stock price and other financial data. The time frame chosen spans from 1 January 2005 to 1 October 2024. This period incorporates data from various market cycles, including major events like the 2008 Global Financial Crisis and the 2020 COVID-19 pandemic, both of which had significant impacts on global markets. Including these periods ensures that the optimisation reflects a range of market conditions, providing a more robust basis for portfolio construction.

Asset Selection

The portfolio was constructed to include a diversified set of assets, focusing on U.S.-listed stocks. This selection aimed to cover a wide range of sectors, ensuring that the portfolio was not overly concentrated in any one industry. The chosen assets span the following sectors: Technology, Healthcare, Consumer Staples, Consumer Discretionary, Financials, Industrials, Utilities, Communication Services, Energy, Bonds, REITs, and Commodities. This sectoral diversity helps to manage risk by spreading exposure across different parts of the economy, which tend to perform differently in various market environments.

Data Processing

Before the data could be used for analysis, several preprocessing steps were taken:

1. **Data Cleaning**: Missing values or irregularities in price data, such as "NaN" entries or outliers, were handled. For missing values, forward or backward filling methods were used, as they can provide continuity without distorting the data.
2. **Adjusting for Dividends and Splits**: The data from `yfinance` already considers dividends and stock splits through adjusted closing prices. This adjustment ensures that the return calculations are accurate and reflect the true growth in value.

Calculations

Daily Returns and Covariance Matrix

- For each asset, its daily returns are calculated by taking the percentage change in adjusted closing prices, thereby capturing its day-to-day performance.
- Using the daily returns, I then calculated the covariance matrix. This matrix represents the degree to which each pair of assets moves in relation to one another, which is essential for understanding diversification benefits within the portfolio.

Expected Returns

- I calculated the expected daily return for each asset by taking the mean of its daily returns over the entire time frame.
- To annualise these expected returns, I multiplied each asset's expected daily return by 252, the approximate number of trading days in a year. This annualisation step provides a clearer view of each asset's average yearly return, a key input for portfolio optimisation.

Portfolio Expected Returns

- The portfolio's expected return was calculated by taking the weighted sum of the individual assets' annual expected returns. This was achieved by multiplying each asset's annualised expected return by its corresponding weight in the portfolio and then summing these values. This result reflects the overall expected return for the portfolio, based on the asset allocation.

Portfolio Expected Risk

- Portfolio risk was calculated using the portfolio variance formula:

$$\sigma_p^2 = \omega^T \Sigma \omega$$

where ω is the vector of asset weights in the portfolio, and Σ is the annualised covariance matrix of returns.

- Taking the square root of the portfolio variance provided the portfolio's standard deviation, representing its overall risk. This measure of risk reflects the combined volatility of all assets in the portfolio, adjusted for correlations, and helps assess the expected range of fluctuations in portfolio returns.

Optimisation Framework

Formulation of the Mean-Variance Optimisation Problem

Objective: Maximise the expected return of the portfolio, given a target level of risk. Mathematically, we can express this as:

$$\textit{maximise } \omega^T \mu$$

subject to:

$$\begin{aligned} \omega^T \Sigma \omega &\leq \sigma^2_{target} && \text{[Constrain risk to target]} \\ \Sigma \omega_i &= 1 && \text{[Sum of weights = 1]} \\ w_i &\geq 0 \quad \textit{for all } i && \text{[No short selling]} \end{aligned}$$

where:

- ω represents the vector of asset weights.
- μ is the vector of expected returns for each asset.
- Σ is the covariance of asset returns.
- σ_{target} is the target risk level, set to 8% in this project.

This formulation ensures that the optimisation process seeks the highest possible return for a portfolio with an overall risk (standard deviation) not exceeding 8%.

Implementation

- **Objective**: The objective function is set to maximise the portfolio's expected return, defined as the dot product of the asset weights ω and the expected returns μ.
- **Constraints**:
  - **Risk Constraint**: The portfolio variance (calculated as $\omega^T \Sigma \omega$) is constrained to be less than or equal to σ_{target} , where σ_{target} is 8% (or 0.08 in decimal form). This constraint ensures that the portfolio's risk does not exceed the specified target.
  - **Weight Sum Constraint**: The sum of all asset weights is constrained to equal 1. This ensures that the entire budget is invested across the selected assets without any additional borrowing.
  - **Non-negativity Constraint**: All weights w_i are constrained to be non-negative, meaning no short-selling is allowed. This is a practical constraint for risk-averse investors or funds that prefer long-only positions.

- **Parameters**:
    - **Target Risk Level**: The target risk for this optimisation is set at 8%, guiding the model to find the optimal return achievable within this risk constraint.
    - **Expected Returns and Covariance Matrix**: The expected returns vector μ and the covariance matrix Σ serve as the primary inputs for defining the risk-return characteristics of the portfolio.

Efficient Frontier

The efficient frontier was constructed by adjusting the risk constraint in each optimisation run to identify the maximum achievable return at that specific risk level.

By plotting the resulting optimised portfolios across different risk levels, I generated the efficient frontier—a graphical representation of the trade-off between risk and return. Each point on the efficient frontier represents an optimal portfolio, where any increase in return is matched by an increase in risk.

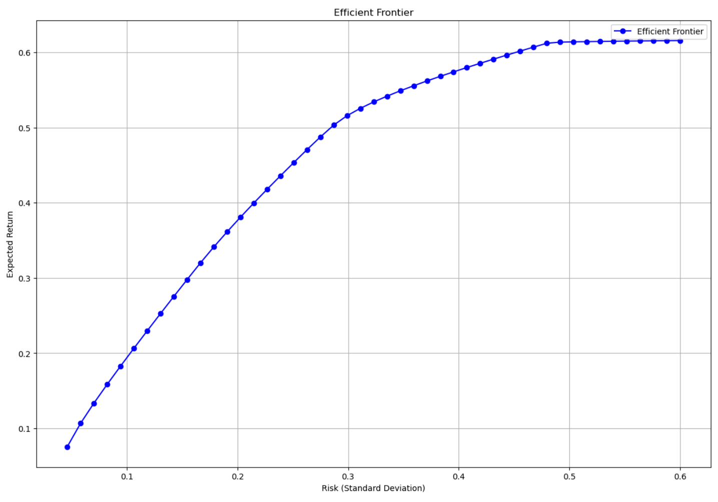

Figure 1: Efficient frontier plot at 8% risk target

The efficient frontier illustrates the set of optimal portfolios. Portfolios lying on the frontier are considered efficient because, for a given level of risk, they offer the maximum possible return. Conversely, for a desired level of return, they carry the minimum risk. Any portfolio below the frontier is suboptimal, as it would either have higher risk for a given return or a lower return for a given risk level.

In this project, the efficient frontier plot also highlights the specific point corresponding to the 8% risk target, showing the optimal return achievable under this risk constraint.

Stability of Solutions and Limitations of CVXPY

Stability Across Time Periods

I assessed the stability of the mean-variance optimised portfolio across five distinct market periods: **Pre-2008 Financial Crisis** (2005-2007), **Global Financial Crisis** (2008-2009),

Post-Crisis Recovery (2010-2015), **COVID-19 Pandemic** (2020-2021), and **Recent Period** (2022-2024). Comparing portfolio weights across these periods highlighted the model's sensitivity to different economic conditions, especially during high-volatility events, providing insights into how mean-variance optimisation adapts to market fluctuations.

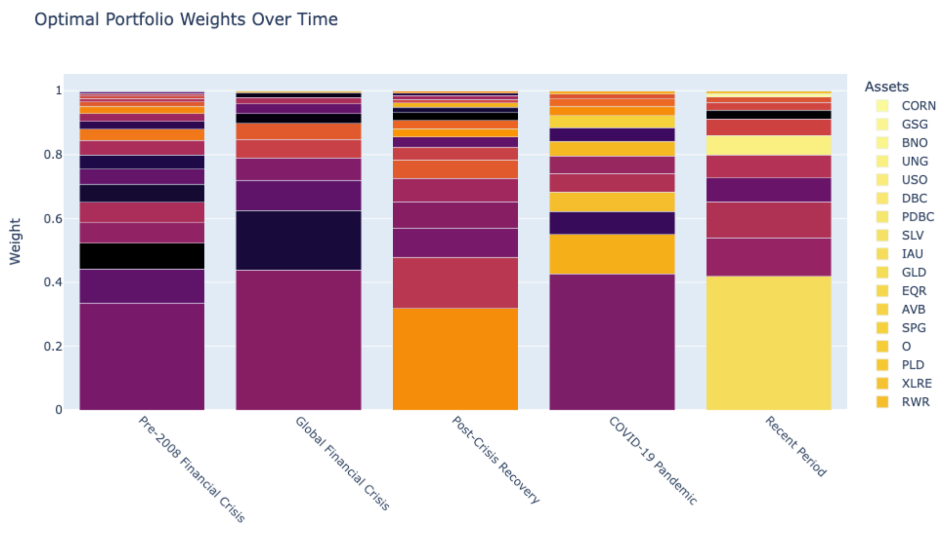

Figure 2: Optimal portfolio weights across various time periods

The portfolio weights shifted significantly across different periods, adapting to changes in expected returns and risk. During times of high market volatility, such as the Global Financial Crisis and COVID-19 Pandemic, the model favoured defensive assets, reflecting a cautious approach. In contrast, in recovery phases like the Post-Crisis Recovery, the allocation was more balanced, leveraging diversified growth opportunities.

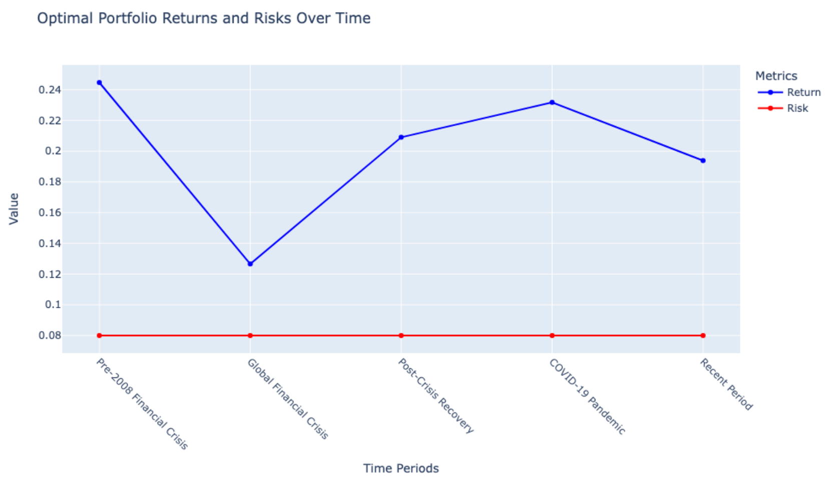

Figure 3: Annual returns and risk level across various time periods

The target risk constraint of 8% was consistently met across all periods (as seen in the stable red line on Figure 3), though expected returns varied considerably. High-volatility periods, like the Global Financial Crisis, showed lower returns, while more stable phases, such as the Pre-2008 Financial Crisis and Post-Crisis Recovery, displayed higher returns. These findings indicate that mean-variance optimisation can control risk but remains highly sensitive to shifts in expected returns and covariances, necessitating frequent rebalancing to maintain optimal allocations. This sensitivity suggests that portfolios using this method may require active management, particularly during volatile periods, to adapt to evolving market conditions.

Stability With Random Perturbations

To evaluate the stability of my mean-variance optimised portfolio, I conducted a systematic perturbation analysis by introducing progressively larger amounts of random Gaussian noise to the daily returns data. This allowed me to simulate different levels of market fluctuations and observe the portfolio's response.

1. **Introducing Randomness**: Gaussian noise with a mean of zero and varied standard deviations were generated, with each standard deviation (perturbation level) representing the intensity of simulated market fluctuations. Lower levels reflected minor noise, while higher levels introduced substantial randomness.
2. **Simulating Market Fluctuations**: At each perturbation level, this noise was applied to the daily returns to create “perturbed returns,” adding controlled variability to assess the portfolio’s behaviour in increasingly volatile conditions.
3. **Recomputing Portfolio Weights**: For each set of perturbed returns, I recalculated the expected returns and covariance matrix—key inputs for the portfolio optimisation—and re-ran the optimisation to determine the new weights that maximise returns under the same constraints.
4. **Comparing Results**: I measured the average and maximum changes in weights between the baseline (non-perturbed) weights and those optimised with perturbed data, providing insight into the portfolio's sensitivity to different levels of market noise.

By systematically increasing the perturbation level, I was able to evaluate the portfolio’s resilience to simulated market fluctuations, offering a clearer picture of its robustness under varying levels of uncertainty.

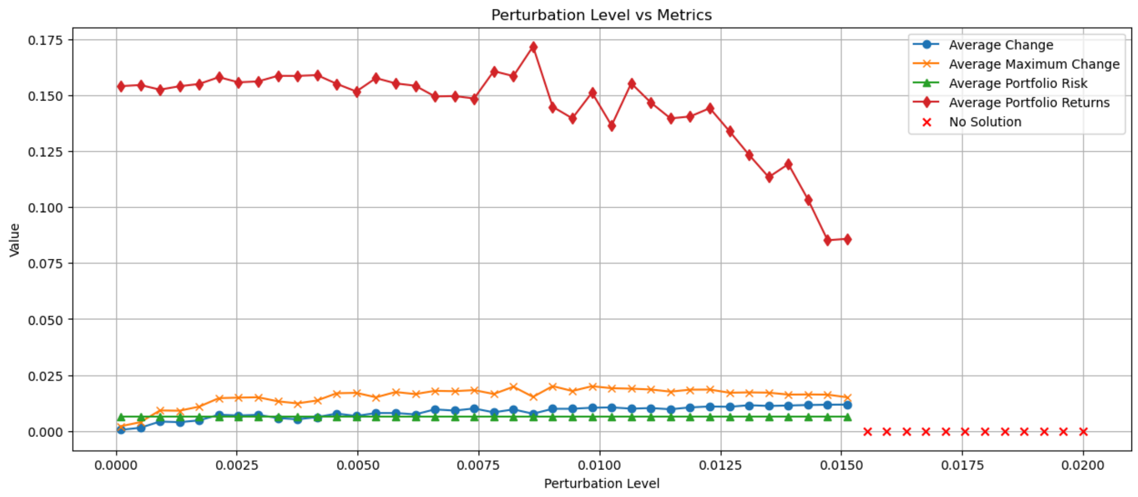

Figure 4: Impact of perturbation level on portfolio metrics

At low perturbation levels (below ~0.0125), the optimisation process performed reliably, with CVXPY consistently finding feasible solutions. In this range, the portfolio metrics—namely,

Average Change, **Average Maximum Change**, **Average Portfolio Risk**, and **Average Portfolio Returns**—displayed minimal variation across different perturbation levels. This stability demonstrates the robustness of the mean-variance optimisation framework under minor market variations, which is desirable in real-world scenarios.

As the perturbation level increased beyond a certain threshold (~0.0150), the model started encountering infeasibility issues, marked by a series of red "X" symbols on the plot. These markers denote perturbation levels where CVXPY was unable to find a feasible solution.

At these higher perturbation levels, I observed the error message: `Forming a nonconvex expression quad_form(x, indefinite)`. This error indicates that the covariance matrix derived from the perturbed returns data was no longer positive semidefinite (PSD), a requirement for the convex optimisation framework in CVXPY. Once the covariance matrix became indefinite, the `quad_form` function could not process it to calculate portfolio risk, leading to an infeasible problem. This requirement limits CVXPY's usability under conditions of high market volatility or data noise, where non-PSD matrices are more likely.

There is a workaround to this by using the `cp.psd_wrap()` function which is designed to interpret a matrix as positive semidefinite (PSD) in the context of optimisation, but it does not make the matrix actually PSD. If the matrix is indefinite or has negative eigenvalues, `cp.psd_wrap()` will still lead to an error. This limitation means that CVXPY does not have a built-in feature to automatically "adjust" a matrix to be PSD, which is often necessary in portfolio optimisation when dealing with perturbed or noisy covariance matrices.

Attempts to Stabilise the Portfolio

Method	Description	Limitations
Exponential Shrinkage Only	Used exponential weighting with a decay factor to stabilise the covariance matrix by capturing recent trends.	Exponential shrinkage alone was too conservative, underestimating the risk. This led to a concentration in high-return but high-risk assets without adequate diversification.
L2 Regularisation + Exponential Shrinkage	Applied both L2 regularisation to control weights and exponential shrinkage for the covariance matrix.	Shrinkage effect overpowered L2 regularisation, leading to insufficient diversification. Shrinkage underestimated risk, concentrating weight on assets with higher returns.
L2 Regularisation + Dynamic regularisation strength + Soft Risk Constraint	Used L2 regularisation with a soft risk constraint to penalise exceeding target risk. Additionally, the regularisation strength (λ) was adjusted dynamically based on the perturbation level.	Dynamically adjusting the regularisation strength allowed for more flexible control over weight constraints, which helped stabilise the portfolio as perturbation levels rose. This resulted in realistic risk levels and better diversification. The trade-off was that returns increased as perturbations rose, possibly due to riskier assets gaining more weight as portfolio volatility increased.

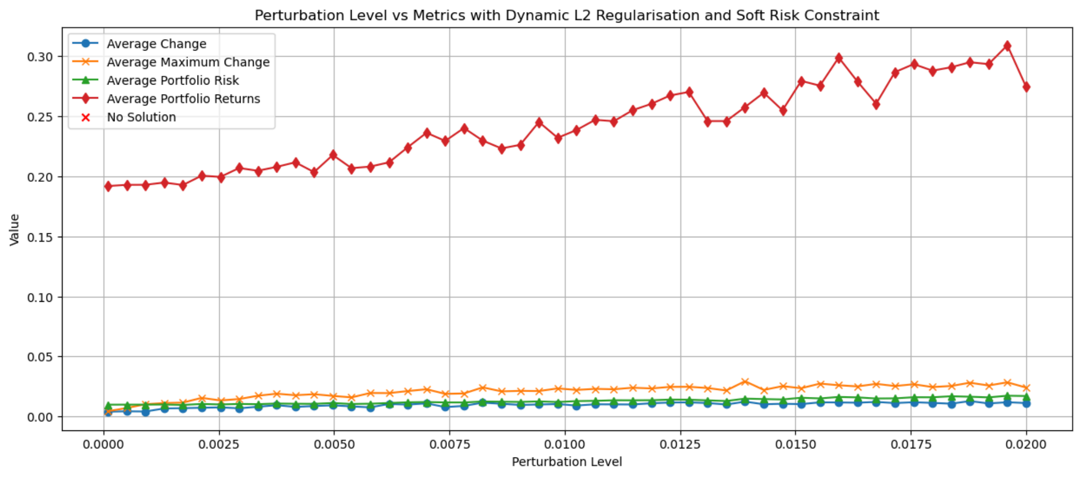

Figure 5: Perturbation analysis using L2 Regularisation + Soft Risk Constraint + Dynamic regularisation strength

Discussion

CVXPY works well for **convex**, stable scenarios with a positive semidefinite (PSD) covariance matrix and **minimal data noise**, making it suitable for periodic rebalancing in stable market conditions.

In contrast, for **high-volatility** or **non-convex cases**, robust optimisation or alternative methods that can handle non-PSD matrices may be more effective.

Suggested Improvements and Alternatives

- **Robust optimisation**: Using robust optimisation can enhance stability under uncertainty, making portfolios less sensitive to data noise.
- **Data Smoothing**: Smoothed or aggregated data, such as moving averages, can improve matrix stability and reduce non-PSD issues.
- **PSD Preprocessing**: Adjust the covariance matrix to be PSD before input, using eigenvalue correction or nearest PSD projection.

Conclusion

Overall, CVXPY provides a solid foundation for traditional portfolio optimisation tasks. However, adapting to real-world market conditions may require either robust optimisation techniques or alternative frameworks to ensure stability and flexibility. This study reinforces the need to carefully select optimisation tools based on specific market conditions and problem requirements, balancing the simplicity of convex frameworks like CVXPY with the adaptability needed for complex, volatile scenarios.

References

Zeng, S. (2020, September 10). The efficient frontier in python. Medium. https://medium.com/@zeng.simonl/the-efficient-frontier-in-python-a1bc9496a0a1

Welcome to CVXPY 1.5 - CVXPY 1.5 documentation. (n.d.). https://www.cvxpy.org/

Yfinance. PyPI. (n.d.). https://pypi.org/project/yfinance/

Towards Data Science. (2019). A gentle introduction to L2 regularization (Ridge regression). https://towardsdatascience.com/a-gentle-introduction-to-l2-regularization-ridge-regression-3ea4f33e0d12

Brownlee, J. (2017). How to use weight regularization to reduce overfitting of deep learning models. Machine Learning Mastery. https://machinelearningmastery.com/weight-regularization-to-reduce-overfitting-of-deep-learning-models/

Helmenstine, A. M. (2020). Exponential decay definition. ThoughtCo. https://www.thoughtco.com/exponential-decay-definition-2312215