Portfolio Optimisation: Modern Portfolio Theory and Hierarchical Risk Clustering

QT Systematic Research Group 1
Autumn 2023

Contents

1 Introduction to Modern Portfolio Theory

Modern Portfolio Theory, abbreviated in this paper as MPT, is a theorised framework for constructing a portfolio based on its expected returns and risk-level, based mostly on Markowitz's writings, *Portfolio Selection*, and *Portfolio Selection Efficient Diversification*. In the former, he explains the stages behind selecting a portfolio:

1. The first stage involves observation of financial analysis on the performance of prospective securities. This concludes with making a prediction on the future performances of the available securities.[5]
2. The next makes relevant beliefs about future performances and ends with constructing a portfolio.[5]

2 Definitions and Calculations in Portfolio Theory

2.1 Key Assumptions of MPT

1. **Investor Rationality:** Investors are rational and will purely aim to maximise their portfolio's expected utility.
2. **Efficient Markets:** All investors have access to the same information, and this information is reflected in asset prices.
3. **Normal Distribution of Returns:** This simplifies the calculation of risk as variance or standard deviation.
4. **Fixed Risk-Free Rate**
5. **Single-Period Investment Horizon:** we consider a single-period investment horizon, which simplifies the analysis.
6. **Quantifiable Risk:** This is defined by variance and/or standard deviation of asset returns.
7. **No Taxes or Transaction Costs**
8. **Unlimited Short Selling:** Investors can sell securities short without restriction, and there are no constraints on the divisibility of assets.

2.2 Calculating Returns

Return is often considered the primary motivating force and the principal reward in any investment process. It is calculated based on the change in the value of an investment over a period of time. The return from an investment is typically expressed as a ratio of the final value of the investment S_t at time T to the initial value S_0 at time $T = 0$, minus one. The formula is given by:

$$\text{Return} = \frac{S_t - S_0}{S_0}, \tag{1}$$

where S_0 represents the stock price at the beginning of the investment period $T = 0$, and S_t denotes the stock price at the end of the investment period $T = t$. This formula will yield the total return of the investment over the period from $T = 0$ to $T = t$.

2.3 Matrix Formulation

Consider a set of securities where each security i has a series of returns over T time periods. We can represent this as a matrix R, where each row corresponds to a security r_i, and each column corresponds to a time period t:

$$R = \begin{bmatrix} — & \mathbf{r_1} & — \\ — & \mathbf{r_2} & — \\ & \vdots & \\ — & \mathbf{r_n} & — \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1T} \\ r_{21} & r_{22} & \cdots & r_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nT} \end{bmatrix} \tag{2}$$

Here, r_{ij} represents the return of security i at time j, with i indexing the security and j indexing the time period. The row $r_i = (r_{i1}, r_{i2}, \ldots, r_{iT})$ is the time series of returns for security i across all T periods.

2.4 Expectation and Variance

In the context of financial returns, the *expectation* of a random variable, often a security's return, is its expected or average value over a specified period. The *variance* measures the spread or dispersion of the returns around this expected value.

Given a vector of returns for security i over T time periods, $\mathbf{r_i} = (r_{i1}, r_{i2}, \ldots, r_{iT})^\top$, the expected return of security i is denoted by

$$E[r_i] = \sum_{j=1}^{T} p_j r_{ij} = \mu_i \tag{3}$$

This is viewed as a weighted average, where p_j represents the probability or weight assigned to the j-th time period, and r_{ij} is the return of security i in that period.

NB: Later on in this text we will use μ_i to refer to the expected returns of security i.

The variance of the return of security i, denoted by $\mathrm{Var}(r_i)$, measures the spread of the returns around the expected value over time T time periods and is defined as:

$$\mathrm{Var}(r_i) = \sum_{j=1}^{T} p_j (r_{ij} - E[r_i])^2, \tag{4}$$

where $E[r_i]$ is the expected return of security i.

2.5 Covariance

Covariance is a measure of how two securities' returns move together. A positive covariance implies that the securities' returns move in the same direction, while a negative covariance indicates that they move in opposite directions. The covariance between securities i and k is denoted by $\mathrm{Cov}(r_i, r_k)$.

The covariance between two securities i and k can be calculated using their individual returns and the respective probabilities or weights of these returns:

$$\mathrm{Cov}(r_i, r_k) = \sum_{j=1}^{T} p_j (r_{ij} - E[r_i])(r_{kj} - E[r_k]), \tag{5}$$

where r_{ij} and r_{kj} are the returns of securities i and k in the j-th scenario or period, respectively, and $E[r_i]$ and $E[r_k]$ are their expected returns.

This measure is crucial in portfolio theory as it helps in understanding how different securities interact with each other within a portfolio, influencing the overall risk and return characteristics.

2.6 Portfolio Return

The expected return of a portfolio is determined by the weighted sum of the expected returns of the individual assets in the portfolio. If we have a portfolio consisting of n different assets, with each asset i having an expected return $E[r_i]$ and a weight w_i in the portfolio, the expected return of the portfolio, denoted as μ_p, is calculated as follows:

$$\mu_p = \sum_{i=1}^{n} w_i \cdot E[r_i] = \mathbf{w}^\top E[R] \tag{6}$$

where:

- μ_p is the expected return of the portfolio,
- w_i is the weight of asset i in the portfolio (such that the sum of all weights equals 1), and
- $E[r_i]$ is the expected return of asset i.

To better visualise this, we have:

$$\mu_p = \mathbf{w}^\top E[R] = \mathbf{w}^\top E\left[\begin{bmatrix} — & \mathbf{r_1} & — \\ — & \mathbf{r_2} & — \\ & \vdots & \\ — & \mathbf{r_n} & — \end{bmatrix}\right] = \begin{bmatrix} w_1 & w_2 & \cdots w_n \end{bmatrix} \begin{bmatrix} — & E[\mathbf{r_1}] & — \\ — & E[\mathbf{r_2}] & — \\ & \vdots & \\ — & E[\mathbf{r_n}] & — \end{bmatrix} \tag{7}$$

2.7 Portfolio Risk and the Covariance Matrix

The risk of a portfolio is influenced by the volatility of individual assets and the correlation between the returns of these assets. The covariance matrix, denoted as Σ, is a square matrix that contains the covariances corresponding to all pairs of assets in the portfolio. The diagonal elements of Σ represent the variances of each asset's returns, while the off-diagonal elements represent the covariances between the returns of different assets.

$$\Sigma = \begin{bmatrix} \text{Var}(r_1) & \text{Cov}(r_1, r_2) & \cdots & \text{Cov}(r_1, r_n) \\ \text{Cov}(r_2, r_1) & \text{Var}(r_2) & \cdots & \text{Cov}(r_2, r_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(r_n, r_1) & \text{Cov}(r_n, r_2) & \cdots & \text{Var}(r_n) \end{bmatrix} \tag{8}$$

Here, $\text{Var}(r_i)$ is the variance of the return of asset i, and $\text{Cov}(r_i, r_j)$ is the covariance between the returns of assets i and j. The covariance matrix is symmetric since $\text{Cov}(r_i, r_j) = \text{Cov}(r_j, r_i)$. Also, note that the $\text{Cov}(r_i, r_i) = \text{Var}(r_i)$.

To calculate the total risk of a portfolio, we use the portfolio's weights and the covariance matrix of the asset returns. The portfolio's risk is quantified by the standard deviation of its returns, which is the square root of the variance. The variance of the portfolio's return is calculated as follows:

$$\sigma_p^2 = \mathbf{w}^\top \Sigma \mathbf{w} \tag{9}$$

where σ_p^2 represents the variance of the portfolio's returns, $\mathbf{w}$ is a vector of the portfolio weights, and Σ is the covariance matrix of the returns. We visualise this as such:

$$\sigma_p^2 = \mathbf{w}^\top \Sigma \mathbf{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix} \begin{bmatrix} \text{Var}(r_1) & \text{Cov}(r_1, r_2) & \cdots & \text{Cov}(r_1, r_n) \\ \text{Cov}(r_2, r_1) & \text{Var}(r_2) & \cdots & \text{Cov}(r_2, r_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(r_n, r_1) & \text{Cov}(r_n, r_2) & \cdots & \text{Var}(r_n) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \tag{10}$$

The total risk of the portfolio, or the portfolio's standard deviation, is then:

$$\sigma_p = \sqrt{\mathbf{w}^\top \Sigma \mathbf{w}} \tag{11}$$

Given a portfolio with weights $\mathbf{w} = [w_1, w_2, \ldots, w_n]^\top$ and a covariance matrix Σ, we can compute the total risk by performing the matrix multiplication followed by taking the square root of the result.

2.8 Understanding the Sharpe Ratio

The Sharpe Ratio is a metric used to assess the performance of an investment by adjusting for its risk. The ratio is the average return earned in excess of the risk-free rate per unit of volatility or total risk. It allows investors to analyse how much greater a return one is obtaining in exchange for the additional volatility endured by holding a riskier asset. The Sharpe Ratio is calculated as follows:

$$\text{Sharpe Ratio} = \frac{E[R_p] - r_f}{\sigma_p} \tag{12}$$

where:

- $E[R_p]$ is the expected return of the portfolio,
- r_f is the risk-free rate of return, which is the return on an investment with zero risk,
- σ_p is the portfolio's risk.

A higher Sharpe Ratio indicates better risk-adjusted performance of the portfolio. It is important to note that the Sharpe Ratio can be used to compare the risk-adjusted returns of portfolios across different types of investments.

3 Optimisation

3.1 Efficient Frontier

The Efficient Frontier represents the set of portfolios that provide the best possible expected return for a given level of risk, or the least possible risk for a given level of expected return.

Portfolios that lie on the Efficient Frontier are considered to be 'efficient' in the sense that no other portfolio exists with a higher expected return at the same level of risk or with a lower risk at the same level of expected return.

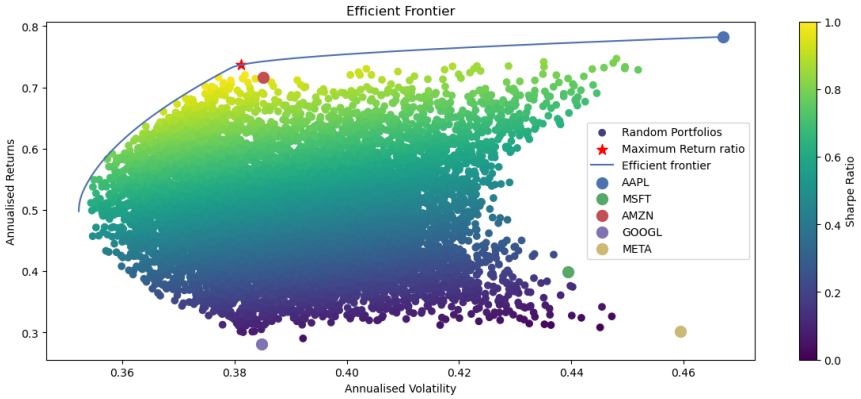

Figure 1: Efficient Frontier with maximum Sharpe marked

3.2 Mean-Variance Optimisation

Mean-Variance Optimisation (MVO) focuses on constructing portfolios that maximise expected return for a given level of risk, or equivalently, minimise risk for a given level of expected return. This concept hinges on the quantification of risk through the variance (or standard deviation) of portfolio returns and the expected return as the portfolio's mean return.

Mathematically, MVO can be expressed as an optimisation problem. Given a set of n assets with expected returns $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)^\top$ and a covariance matrix Σ representing the risks and correlations between the returns of these assets, the objective is to find the optimal portfolio weights $\mathbf{w} = (w_1, w_2, \ldots, w_n)^\top$ that solve the following problems:

$$\begin{aligned} \underset{\mathbf{w}}{\text{maximise}} \quad & E[R_p] = \mathbf{w}^\top \boldsymbol{\mu} \\ \text{subject to} \quad & \mathbf{w}^\top \Sigma \mathbf{w} \leq \sigma^2_{\text{target}}, \\ & \sum_{i=1}^{n} w_i = 1, \\ & w_i \geq 0, \ i = 1, \ldots, n. \end{aligned} \tag{13}$$

or its equivalent:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimise}} \quad & \sigma^2 = \mathbf{w}^\top \Sigma \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^\top \boldsymbol{\mu} \geq \mu_{\text{target}}, \\ & \sum_{i=1}^{n} w_i = 1, \\ & w_i \geq 0, \ i = 1, \ldots, n. \end{aligned} \tag{14}$$

Since both structures are equivalent, we will follow general literature conventions and focus on problems of the second structure, minimising risk.

3.3 Additional Inequality Constraints

Realistically, portfolio managers will implement additional limits in their investments to ensure diversification, which plays an additional factor in risk. If a portfolio takes on a substantial loss due to a few assets, then it lacks diversification. This is because the portfolio with the smallest risk, can potentially consist of a single security from the pool, and portfolio returns will depend solely on that single security. We reflect this by adding additional inequality constraints to the weights of the portfolios, (in machine learning, this is a crude form of regularisation):

$$
\begin{aligned}
\underset{\mathbf{w}}{\text{minimise}} \quad & \sigma^2 = \mathbf{w}^\top \Sigma \mathbf{w} \\
\text{subject to} \quad & \mathbf{w}^\top \boldsymbol{\mu} \geq \mu_{\text{target}}, \\
& \sum_{i=1}^{n} w_i = 1, \\
& w_i \geq l_i,\ w_i \leq \mu_i,\ i = 1, \ldots, n.
\end{aligned}
\tag{15}
$$

Where we denote lower and upper bounds with l and u. The non-negativity constraint $w_i \geq 0$ can also be relaxed to allow for short selling.

3.4 Computational Approaches

It is generally possible to use any quadratic solver to the MVO problem, but the added inequality constraints and portfolio manager needs for an efficient frontier visualisation requires more specific approaches. The usual method of Lagrange multipliers cannot be used due to the inequalities present.

Traditional gradient-based algorithms offered by open-source and commercial optimisers are typically exhibit sensitivity to boundary constraints, are overly inefficient, and may not reach feasible or close-to-optimal solutions, as noted by Bailey [1].

Markowitz and Todd [4] proposed the critical line algorithm (CLA), that is specifically designed for inequality-constrained portfolio optimisation problems, and guarantees an exact solution found after a given number of iterations. It not only finds a single solution, but also computes the entire efficient frontier solution.

Broadly, Markowitz took a 'divide-and-conquer' approach to a constrained problem by formulating it as a set of unconstrained problems. Markowitz introduced the idea of a *turning point*, a solution vector w^* that has in its vicinity another solution vector that has different unconstrained free assets. In the regions of the solution space, away from the turning points, inequality constraints effective become trivial with respect to the free assets. Computing the solution reduces to solving an unconstrained Lagrangian on the free assets. Markowitz focused on computing the optimal portfolio for each turning point, arguing *the efficient frontier can be simply derived as a convex combination between any two neighbour turning points.* [4]

This report does not cover the CLA algorithm and its arguments for optimality.

4 Limitations of MPT

4.1 Covariance Matrix Instability

As described in previous chapters, many risk-based asset allocation models require a sample covariance matrix as the input parameter. In practice, the true covariance matrix varies differently from the sampled covariance matrix. Lediot and Wolf [3] state that the general convention of taking historical returns should be avoided, as a sampled covariance matrix tends to have extreme values caused by an extreme amount of error. This is the curse of dimensionality, where for an increase in N variables, there must be an even greater increase in T observations to ensure that our sample covariance matrix is a good-enough approximator for true covariance. When the concentration ratio $q = N/T$ is close to 1, the sample covariance matrix is very likely to be noisy.

These errors cause the optimiser to gravitate towards the most unreliable constraints, a phenomenon known as 'error maximisation'. [6].

4.2 Markowitz's Curse

Other general weaknesses also stem from having groups of highly correlated securities, causing the covariance matrix to potentially become rank-deficient. Due to the complex structure of the covariance matrix that links every security with each other, small deviations in forecasted returns can cause high variance in portfolios produced by CLA.

All factors lead to what is called Markowitz's Curse: having more correlated investments results in a higher need for diversification, which causes more unstable solutions, and greater estimation errors that result offset the benefits obtained from diversification.
In the next topic we will discuss López de Prado's approach in addressing these issues by utilising a different way of viewing relationships between each security [2].

5 Hierarchical Risk Parity

5.1 Motivation & Preliminaries

5.1.1 Why do we Cluster?

- We are aiming to **classify data into groups by characteristics**
- Data are sometimes **unlabelled**, from which we need to decide which variables are used to classify them
- Sometimes the classifications **variables are unclear/unknown**

5.1.2 Hierarchical Clustering

- Seeks to create **different level of precision**(hierarchies) between clusters of data
- Enables convenient selection of clusters by their "distance".
- Distance between clusters are defined by:
  - Metric: distance between data points
  - Linkage Criterion: data points that is used to measure "distance" between clusters
    N.B., most hierarchical clustering algorithms very similar, differ by metric and linkage criterion
- HCAs is performed in 2 ways, iteratively:
  - Agglomerative: start with individual data points, and group together step-by-step
  - Divisive: start with one cluster, and divide into sub-clusters iteratively
- The "hierarchies" of clusters are represented by a Dendrogram, recording merging and splitting

5.1.3 Hierarchical Risk Clustering/Parity

Hierarchical Risk Parity is approach proposed by López de Prado [2] that addresses the problems within the covariance matrix structure.

López de Prado argues that a covariance (or correlation matrix) is too complex to be fully processed and analysed, and instability grows disproportionately as the number of assets increases.

Instead, taking a hierarchical approach simplifies a lot of unnecessary relationships, cutting down the number of securities that could be potential subsitutes for each other.

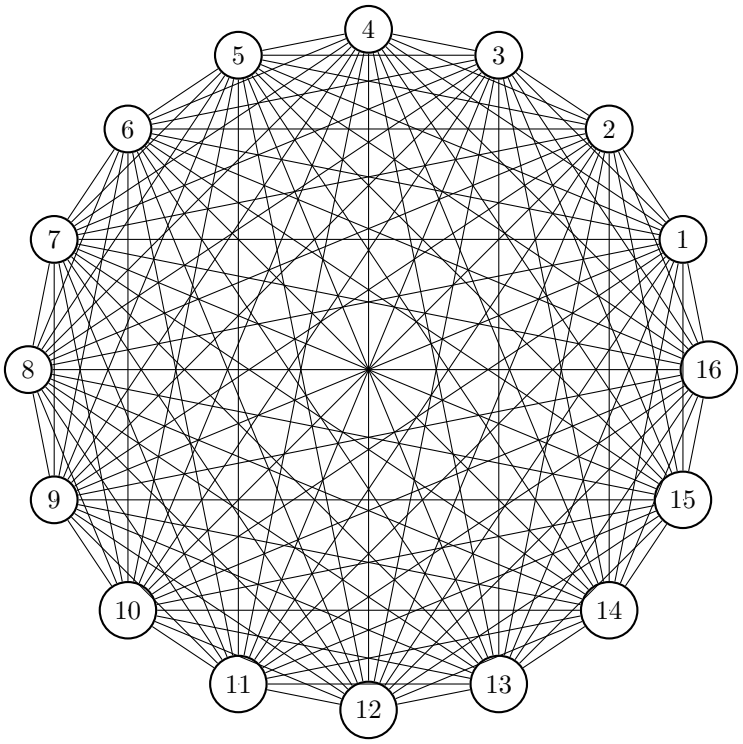

Figure 2: A visualisation of relationships for a covariance matrix of 16 nodes – a complete graph with every possible pairwise combination.

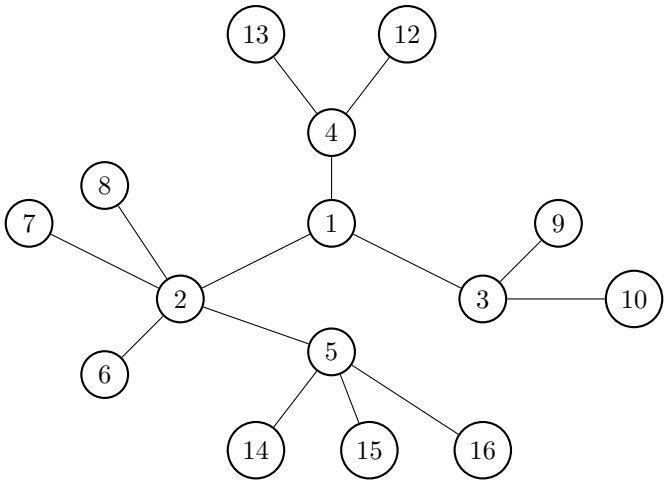

Figure 3: Visualisation of a covariance matrix of size 16 represented by a tree structure

Through HRP, we are able to identify common risk factors, drivers, or sources that affect multiple risks, and more importantly, discover emerging or hidden risks that are not captured by conventional risk frameworks or methods.

Given a covariance matrix Σ for n assets, defined as:

$$\Sigma = \begin{bmatrix} \text{Var}(r_1) & \text{Cov}(r_1, r_2) & \cdots & \text{Cov}(r_1, r_n) \\ \text{Cov}(r_2, r_1) & \text{Var}(r_2) & \cdots & \text{Cov}(r_2, r_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(r_n, r_1) & \text{Cov}(r_n, r_2) & \cdots & \text{Var}(r_n) \end{bmatrix}, \tag{16}$$

we first convert this matrix to a correlation matrix $\mathbf{P}$, where each element ρ_{ij} is the Pearson correlation coefficient between the returns of assets i and j. The correlation matrix is derived by standardising the covariance matrix (*cor* denotes the operation of converting the covariance matrix into the correlation matrix):

$$\mathbf{P} = [\rho_{ij}]_{i,j=1}^{n} = \text{cor}(\Sigma) \quad \rho_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii} \cdot \Sigma_{jj}}} \tag{17}$$

$$\text{correlation}(i, j) = \frac{\text{covariance}(i, j)}{\sqrt{\text{variance}(i) \times \text{variance}(j)}} \tag{18}$$

Subsequently, we define the distance matrix $\mathbf{D}$ with elements d_{ij} that quantify the dissimilarity between each pair of assets, calculated as:

$$\mathbf{D} = [d_{ij}]_{i,j=1}^{n} \quad d_{ij} = \sqrt{\frac{1}{2}(1 - \rho_{ij})}. \tag{19}$$

The distance matrix $\mathbf{D}$ has several important properties:

- Non-negativity: $d_{ij} \geq 0$
- Coincidence: $d_{ij} = 0$ if and only if $i = j$
- Symmetry: $d_{ij} = d_{ji}$
- Subadditivity: $d_{ik} \leq d_{ij} + d_{jk}$

These properties ensure that $\mathbf{D}$ is a proper metric space, suitable for the clustering step in HRP.

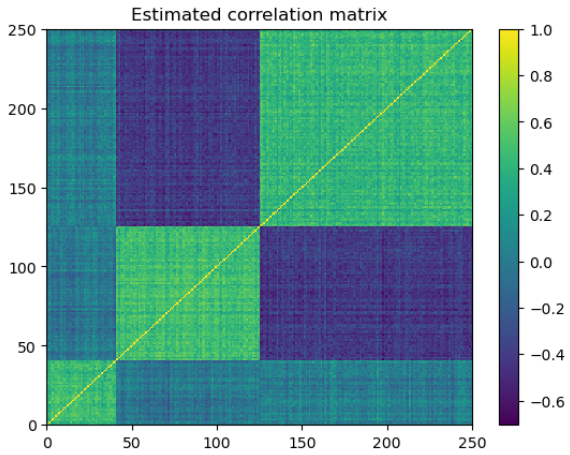

Figure 4: Covariance Matrix

5.2 Clustering and Portfolio Allocation

Using the distance matrix $\mathbf{D}$, we can apply a hierarchical clustering algorithm to group assets. The goal is to pair assets into clusters based on their distance, which reflects the correlation of their returns. This process can be visualised using a dendrogram, which shows the hierarchy of clusters.

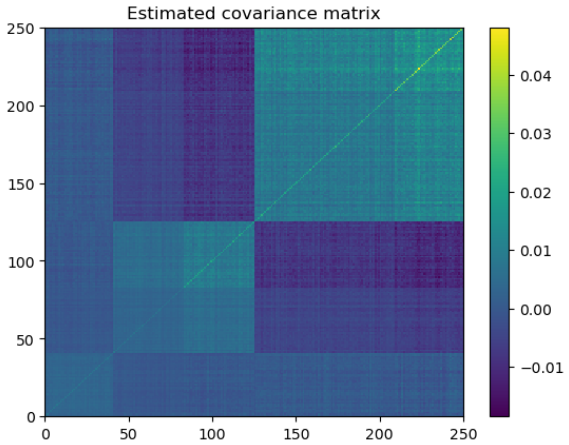

Figure 5: Correlation Matrix

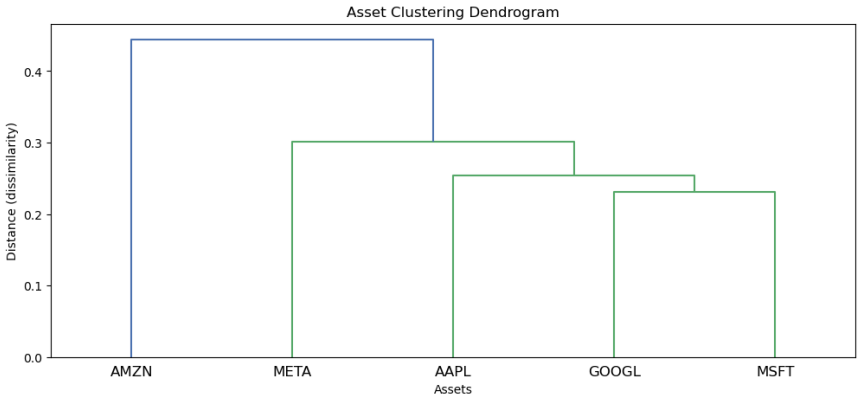

Figure 6: Example dendrogram

Once the assets are clustered, we can allocate capital to each cluster based on the inverse-variance method, which allocates more capital to clusters with lower variance. This method aims to construct a diversified portfolio that minimises the expected volatility.

5.3 Metrics & Linkage Criterion

Consider a simple example where we have computed the distances between three assets. The clustering step would involve finding the pair of assets (represented by column vectors) with the smallest distance:

$$\{i^*, j^*\} = \arg\min_{(i,j)} \mathbf{D} = \arg\min_{(i,j)} \{d_{ij}\}, \tag{20}$$

and then clustering them together. This initial cluster is denoted as $U[1]$. The process continues iteratively, clustering the next closest pair of assets or asset clusters until all assets are grouped into a single cluster.

We want to compute the distances between each column vector of the distance matrix. This is an augmented distance matrix function $\widetilde{\mathbf{D}}$, a function of the whole correlation matrix, a distance of distances in $\mathbf{D}$, which provides a measure of dissimilarity between two assets i and j across the entire portfolio. The distance function is given by:

$$\widetilde{\mathbf{D}}(i,j) = \sqrt{\sum_{k=1}^{N} (\mathbf{D}(k,i) - \mathbf{D}(k,j))^2} \tag{21}$$

The process begins by forming the first cluster from the pair of assets that are closest together. This is represented by:

$$U[1] = \arg\min_{i,j} \widetilde{\mathbf{D}}(i,j) = (i^*, j^*) \tag{22}$$

where U is the set of clusters.

Once the first cluster is formed, we need to define a distance metric between a cluster and unclustered items, so the distance matrix can be updated. This is a **linkage criterion**. The distance between the first clustered item $U[1]$ and any other asset i is computed as:

$$\widetilde{\mathbf{D}}(i, U[1]) = \min(\widetilde{\mathbf{D}}(i^*, i), \widetilde{\mathbf{D}}(j^*, i)) \tag{23}$$

where (i^*, j^*) are the assets in the first cluster $U[1]$ – this is also called the nearest-point algorithm. This process is iteratively applied, updating the distance matrix at each step, until all assets are clustered into a single group. More generally, for any cluster $(c_1, c_2, c_3, \ldots, c_n)$, the distance between a cluster and any other asset i is:

$$\widetilde{\mathbf{D}}(i, (c_1, c_2, c_3, \ldots, c_n)) = \min(\widetilde{\mathbf{D}}(i, c_1), \widetilde{\mathbf{D}}(i, c_2), \ldots, \widetilde{\mathbf{D}}(i, c_n)) \tag{24}$$

NOTE: other common linkage criterions:

- **MAX**(complete-linkage): noise proof; but biased towards globular clusters, and in occasions, it might break larger clusters
- **Group Average**: noise proof; but biased towards globular clusters
- **Ward's Method**(minimum-variance): noise proof; but biased towards globular clusters
- **Centroid Distance**: generally less popular

Reasons why **MIN**/single-linkage is chosen:

1. PROS: More robust in separating non-elliptical shapes as long as the gap between the two clusters is not small.
   This is essential as we do not know the distribution of risk factors
2. CONS: No defence against noise between clusters

5.4 Clustering Process Example

Let's consider an example with three assets and their respective distance matrix $\mathbf{D}$, and augmented distance matrix $\widetilde{\mathbf{D}}$.

$$\mathbf{D} = \begin{bmatrix} 0 & .3873 & .6325 \\ .3873 & 0 & .7746 \\ .6325 & .7746 & 0 \end{bmatrix} \Rightarrow \widetilde{\mathbf{D}} = \begin{bmatrix} 0 & 0.5659 & 0.9747 \\ 0.5659 & 0 & 1.1225 \\ 0.9747 & 1.1225 & 0 \end{bmatrix} \tag{25}$$

Step 1: Forming the First Cluster

We identify the pair of assets with the smallest distance:

$$U[1] = \arg\min_{i,j} \mathbf{D}_{ij} \text{ where } i \neq j \tag{26}$$

For our example, $U[1] = (1, 2)$ since they have the smallest Euclidean distance of 0.8138.

Step 2: Updating the Distance Matrix

With the first cluster $U[1]$, we update the distance matrix using the linkage criterion:

$$\widetilde{\mathbf{D}}(i, U[1]) = \min(\mathbf{D}(i, i^*), \mathbf{D}(i, j^*)) \tag{27}$$

where (i^*, j^*) are the assets in cluster $U[1]$. The updated distance vector is:

$$\widetilde{\mathbf{D}}(i, U[1]) = \begin{bmatrix} \min(0, 0.5659) \\ \min(0.5659, 0) \\ \min(0.9747, 1.1225) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.9747 \end{bmatrix} \tag{28}$$

Once the initial cluster is formed, the distance matrix $\widetilde{\mathbf{D}}$ is updated to reflect the distances of the remaining assets to this new cluster.

After the first cluster $U[1] = (1, 2)$ is formed, we update the distance matrix by adding the cluster, getting:

$$\widetilde{\mathbf{D}}_{i,j}^{new} = \begin{bmatrix} 0 & 0.5659 & 0.9747 & 0 \\ 0.5659 & 0 & 1.1225 & 0 \\ 0.9747 & 1.1225 & 0 & 0.9747 \\ 0 & 0 & 0.9747 & 0 \end{bmatrix} \tag{29}$$

Then, we remove the clustered rows and columns, getting:

$$\widetilde{\mathbf{D}}_{i,j}^{final} = \begin{bmatrix} 0 & 0.9747 \\ 0.9747 & 0 \end{bmatrix} \tag{30}$$

Step 3: Forming Subsequent Clusters

The process continues recursively. We select the next pair of assets with the minimum distance from the updated distance matrix and form a new cluster. This is appended to the set of existing clusters, and the distance matrix is updated accordingly.

Completion of the Clustering Process

The clustering process is complete when all assets are included in a single cluster. At this point, the algorithm stops. The final cluster $u[N-1]$ contains all the original items, and the hierarchical clustering is fully determined. In our example with three assets, we have:

$$U[2] = (3, U[1]) \rightarrow \text{Stop} \tag{31}$$

At this point, we have a complete hierarchical clustering of assets, which can be represented as a dendrogram and used for portfolio allocation in the HRP framework.

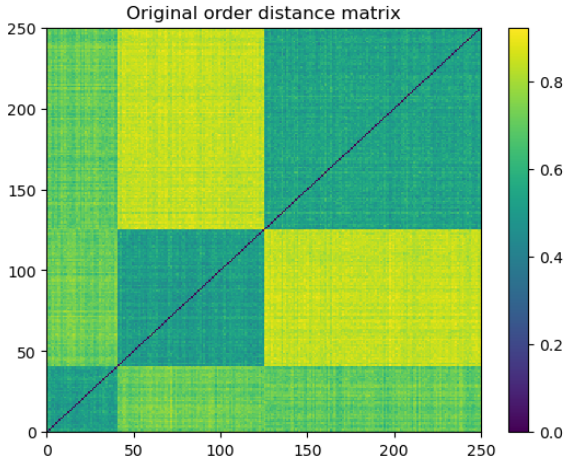

Figure 7: Distance Matrix created from comparing distances between securities, also simplifying distances with clusters formed in the process.

5.5 Quasi-Diagonalisation

Quasi-diagonalisation is a process applied after the hierarchical clustering stage in the Hierarchical Risk Parity (HRP) approach. The aim is to reorganise the covariance matrix so that the largest values lie along the diagonal (top left to bottom right). This has the effect that similar investments are grouped together, and dissimilar investments are spaced further apart.

Broadly, this re-arranges the securities to match the hierarchical structure from the previous step.

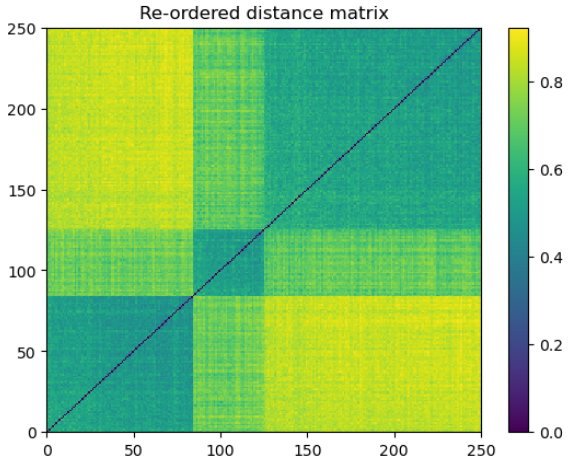

Figure 8: After quasi-diagonalisation.

5.6 Recursive Bisection

Theorem: The inverse allocation for a portfolio is optimal when the covariance matrix is diagonal.

Consider our existing portfolio optimisation problem:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimise}} \quad & \sigma^2 = \mathbf{w}^\top \Sigma \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^\top \mathbf{1} = 1 \\ & w_i \geq 0, \ i = 1, \ldots, n. \end{aligned} \tag{32}$$

Note we have denoted the constraint $\sum_{i=1}^{n} w_i = 1$ as taking the dot product of the weight vector with the ones vector $\mathbf{1}$.

A diagonal covariance matrix implies the returns of all securities are uncorrelated. Then taking the inverse of that diagonal matrix gives the reciprocal of all values along the diagonal, which is the optimal weighting for the securities.

$$\Sigma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 9 \end{bmatrix} \Rightarrow \Sigma^{-1} = \begin{bmatrix} 1/1 & 0 & 0 \\ 0 & 1/5 & 0 \\ 0 & 0 & 1/9 \end{bmatrix} \tag{33}$$

We get the minimum variance portfolio solution

$$\mathbf{w} = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}^\top \Sigma \mathbf{1}} \tag{34}$$

Which can be written as (for a portfolio of n securities)

$$\mathbf{w_n} = \frac{\Sigma_{n,n}^{-1}}{\sum_{i=1}^{N} \Sigma_{i,i}^{-1}} \tag{35}$$

When $N = 2$,

$$\mathbf{w}_1 = \frac{\frac{1}{\Sigma_{1,1}}}{\frac{1}{\Sigma_{1,1}} + \frac{1}{\Sigma_{2,2}}} = 1 - \frac{\Sigma_{1,1}}{\Sigma_{1,1}, \Sigma_{2,2}}$$

Extending this, we use recursively bisect the quasi-diagonalised matrix and assign weights based on the variance of each subset.

1. **Initialisation**: The algorithm starts with a list of all assets L_0 and assigns a unit weight to each asset, indicating that initially, the portfolio is equally weighted across all assets.

2. **Checking for Completion**: If any list L_i in the list of assets L contains only one asset, the bisection for that list is considered complete. This is because a single asset cannot be split further.

3. **Recursive Bisection Process**: For each list with more than one asset:

   (a) The list L_i is divided into two subsets $L_i^{(1)}$ and $L_i^{(2)}$, ensuring that the original order of assets is preserved. The median of the list is typically used to determine the bisection point.

   (b) For each subset j define the variance of each subset as the quadratic form $\tilde{\Sigma}_i^{(j)} \equiv \tilde{\mathbf{w}}_i^{(j)\top} \Sigma_i^{(j)} \tilde{\mathbf{w}}_i^{(j)}$ where $\Sigma_i^{(j)}$ is the covariance matrix between the subsets of the bisection. The weight vector is $\tilde{\mathbf{w}}_i^{(j)}$ is found by normalising the inverse of the diagonal of the covariance matrix by the trace of its inverse, $\tilde{\mathbf{w}}_i^{(j)} = \text{diag}(\Sigma_i^{(j)})^{-1} \frac{1}{\text{trace}(\text{diag}(\Sigma_i^{(j)})^{-1})}$

   (c) Compute the split factor α_i using the variance of the subsets. $\alpha_i = 1 - \frac{\tilde{\Sigma_i^{(1)}}}{\tilde{\Sigma_i^{(1)}} + \tilde{\Sigma_i^{(2)}}}$ and $0 < \alpha_i < 1$

   (d) Adjust the weights of the assets in each subset by multiplying by the split factor α_i for weights in L_i^1 and $1 - \alpha_i$ for weights in L_i^2, ensuring that the weights remain normalised and the total sum of weights equals 1.

4. **Iteration**: The process is iterated by going back to the second step until all assets are individually accounted for in the list L.

6 Performance Comparison of HRP and CLA

In our testing, the `PyPortfolioOpt` library is utilised to rebalance a similar selection of S&P500 securities in two portfolios, one using hierarchical risk parity (HRP) and the other using the critical line algorithm (CLA).

The `yfinance` library is then used to obtain data from 2021 onwards to late 2023 at the time of this report's writing, and split the in-sample data to the 2021-2022 period and the out-of-sample data to the 2022-2023 period.

Portfolios are initialised to the optimised weights from in-sample data in aggregate, then we look into the out-of-sample data in a rolling fashion to rebalance the portfolio every 22 days (equivalent to a trading month). This mirrors López de Prado's approach [2], but instead of using Monte Carlo simulations to generate a random price walk, historical data is used instead.

The volatility of portfolio returns from the in-sample data is saved, the volatility of the out-of-sample returns are calculated and compared.

In Figure 9 and Figure 10 we observe that CLA allocates weights on less securities which makes it vulnerable to sudden price shocks. HRP tries to diversify investments, but tries to respect the correlation structure, so it fares better to systemic shocks in security prices.

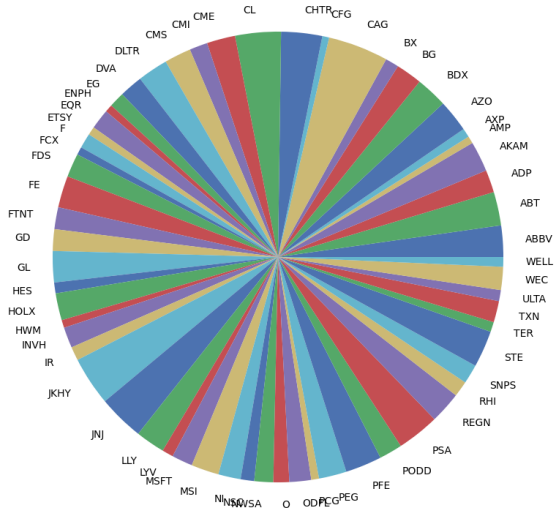

Figure 9: HRP usually produces a diverse portfolio of similar weights.

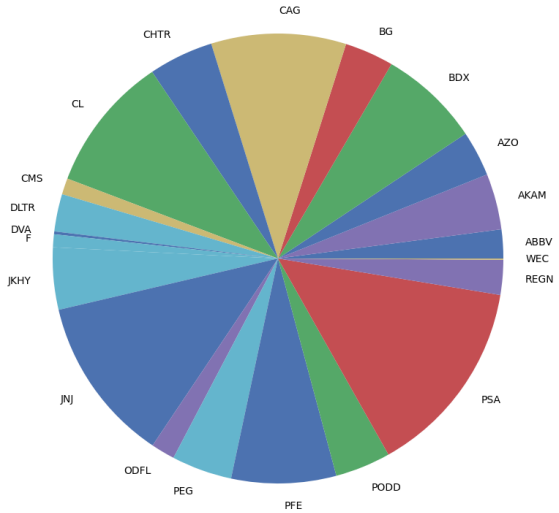

Figure 10: CLA usually produces a less diverse portfolio of varying weights.

In Figure 11 we see in this backtest that HRP provides superior returns and volatility compared to HRP for a large number of securities (60 in total). It is interesting to note that while CLA find the lowest in-sample volatility portfolio, a lower out-of-sample volatility is achieved by HRP.

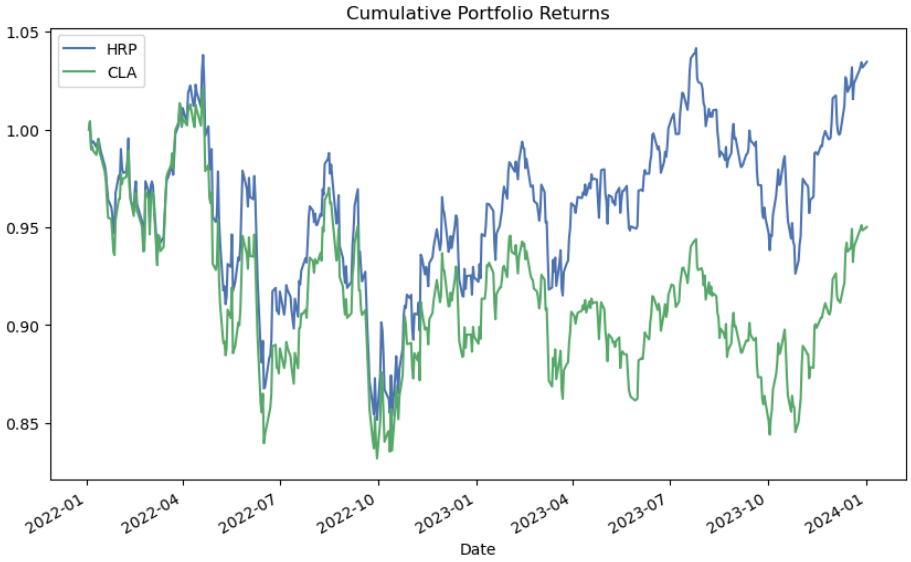

Figure 11: Portfolio returns for HRP and CLA optimisers.

	HRP	CLA
In-Sample Volatility (Annualised)	13.8%	11.9%
Out-of-Sample Volatility (Annualised)	16.6%	17.1%

References

[1] David H. Bailey and Marcos Lopez De Prado. An open-source implementation of the critical-line algorithm for portfolio optimization. *SSRN Electronic Journal*, 2013.

[2] Marcos López de Prado. *Advances in Financial Machine Learning*. John Wiley & Sons, 2018.

[3] Olivier Ledoit and Michael Wolf. Honey, I Shrunk the Sample Covariance Matrix. *The Journal of Portfolio Management*, 30(4):110–119, 2004.

[4] H. M. Markowitz and G. P. Todd. *Mean Variance Analysis in Portfolio Choice and Capital Markets*. John Wiley and Sons, New York, USA, 1st edition, 2000.

[5] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[6] Richard O. Michaud and Robert O. Michaud. *Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation*. Oxford University Press, 2008.