

Title: Hierarchical Risk Parity

URL: https://en.wikipedia.org/wiki/Hierarchical_Risk_Parity

PageID: 78681967

Categories: Category:Machine learning, Category:Machine learning algorithms, Category:Portfolio theories

Source: Wikipedia (CC BY-SA 4.0). Content may require attribution.

Hierarchical Risk Parity (HRP) is an advanced investment portfolio optimization framework developed in 2016 by Marcos López de Prado at Guggenheim Partners and Cornell University . HRP is a probabilistic graph-based alternative to the prevailing mean-variance optimization (MVO) framework developed by Harry Markowitz in 1952, and for which he received the Nobel Prize in economic sciences . HRP algorithms apply discrete mathematics and machine learning techniques to create diversified and robust investment portfolios that outperform MVO methods out-of-sample . HRP aims to address the limitations of traditional portfolio construction methods , particularly when dealing with highly correlated assets. Following its publication, HRP has been implemented in numerous open-source libraries, and received multiple extensions.

Key Features

HRP portfolios have been proposed as a robust alternative to traditional quadratic optimization methods, including the Critical Line Algorithm (CLA) of Markowitz. HRP addresses three central issues commonly associated with quadratic optimizers: numerical instability, excessive concentration in a small number of assets, and poor out-of-sample performance.

HRP leverages techniques from graph theory and machine learning to construct diversified portfolios using only the information embedded in the covariance matrix. Unlike quadratic programming methods, HRP does not require the covariance matrix to be invertible. Consequently, HRP remains applicable even in cases where the covariance matrix is ill-conditioned or singular—conditions under which standard optimizers fail.

Monte Carlo simulations indicate that HRP achieves lower out-of-sample variance than CLA, despite the fact that minimizing variance is the explicit optimization objective of CLA. Furthermore, HRP portfolios exhibit lower realized risk compared to those generated by traditional risk parity methodologies. Empirical backtests have demonstrated that HRP would have historically outperformed conventional portfolio construction techniques.

Algorithms within the HRP framework are characterized by the following features:

Machine Learning Approach : HRP employs hierarchical clustering, a machine learning technique, to group similar assets based on their correlations. This allows the algorithm to identify the underlying hierarchical structure of the portfolio, and avoid that errors spread through the entire network.

Risk-Based Allocation : The algorithm allocates capital based on risk, ensuring that assets only compete with similar assets for representation in the portfolio. This approach leads to better diversification across different risk sources, while avoiding the instability associated with noisy returns estimates.

Covariance Matrix Handling : Unlike traditional methods like Mean-Variance Optimization, HRP does not require inverting the covariance matrix. This makes it more stable and applicable to portfolios with a large number of assets, particularly when the covariance matrix's condition number is high.

The Problem: Markowitz's Curse

Portfolio construction is perhaps the most recurrent financial problem. On a daily basis, investment managers must build portfolios that incorporate their views and forecasts on risks and returns. Despite the theoretical elegance of Markowitz's mean-variance framework, its practical

implementation is hindered by several limitations that undermine the reliability of solutions derived from the Critical Line Algorithm (CLA). A principal concern is the high sensitivity of optimal portfolios to small perturbations in expected returns: even minor forecasting errors can result in significantly different allocations (Michaud, 1998). Given the inherent difficulty of producing accurate return forecasts, numerous researchers have advocated for approaches that forgo expected returns entirely and instead rely solely on the covariance structure of asset returns. This has given rise to risk-based allocation methods, among which risk parity is a widely cited example (Jurczenko, 2015).

While eliminating return forecasts mitigates some instability, it does not eliminate it. Quadratic programming techniques employed in portfolio optimization require the inversion of a positive-definite covariance matrix, meaning all eigenvalues must be strictly positive. When the matrix is numerically ill-conditioned—that is, when the ratio of its largest to smallest eigenvalue (its condition number) is large—matrix inversion becomes unreliable and prone to significant numerical errors (Bailey and López de Prado, 2012).

The condition number of a covariance, correlation, or any symmetric (and thus diagonalizable) matrix is defined as the absolute value of the ratio between its largest and smallest eigenvalues in modulus. The figure on the right presents the sorted eigenvalues of several correlation matrices; the condition number is represented by the ratio of the first to last eigenvalues in each sequence. A diagonal correlation matrix, which is equal to its own inverse, exhibits the minimum possible condition number.

As the number of correlated (or multicollinear) assets in a portfolio increases, the condition number rises. At high levels, this leads to severe numerical instability, whereby slight modifications in any matrix entry may result in drastically different inverses. This phenomenon, often referred to as Markowitz's curse, encapsulates the paradox wherein increased correlation among assets heightens the theoretical need for diversification, yet simultaneously increases the likelihood of unstable optimization outcomes. Consequently, the potential benefits of diversification are frequently overshadowed by estimation errors.

These problems are exacerbated as the dimensionality of the covariance matrix increases. The estimation of each covariance term consumes degrees of freedom, and in general, a minimum of $\frac{1}{2}N(N+1)$ independent and identically distributed (IID) observations is required to estimate a non-singular covariance matrix of dimension N . For example, constructing an invertible covariance matrix of dimension 50 necessitates at least five years of daily IID observations. However, empirical evidence suggests that the correlation structure of financial assets is highly unstable over such extended periods. These difficulties are highlighted by the observation that even naïve allocation strategies—such as equally weighted portfolios—have frequently outperformed both mean-variance and risk-based optimizations in out-of-sample tests (De Miguel et al., 2009).

The Solution: Hierarchical Risk Parity

The HRP algorithm addresses Markowitz's curse in three steps:

Hierarchical Clustering : Assets are grouped into clusters based on their correlations, forming a hierarchical tree structure.

Quasi-Diagonalization : The correlation matrix is reordered based on the clustering results, revealing a block diagonal structure.

Recursive Bisection : Weights are assigned to assets through a top-down approach, splitting the portfolio into smaller sub-portfolios and allocating capital based on inverse variance.

Step 1: Hierarchical Clustering

Given a $T \times N$ matrix of asset returns X , where each column represents a time series of returns for one of N assets over T time periods, a hierarchical clustering process can be used to construct a tree-based representation of asset relationships. First, we compute the $N \times N$ correlation matrix $\rho = \rho_{i,j}$, $i, j = 1 \dots N$, where $\rho_{i,j} = \text{corr}$

(X_i, X_j) $\{\displaystyle \rho_{i,j} = \mathrm{corr}(X_i, X_j)\}$. From this, a pairwise distance matrix $D = d_{i,j}$ $\{\displaystyle D = \{d_{i,j}\}$ is defined using the transformation:

This distance function defines a proper metric space, satisfying non-negativity, identity of indiscernibles, symmetry, and the triangle inequality.

Next, a secondary distance matrix $\tilde{D} = \{\tilde{d}_{i,j}\}$ is computed, where each entry measures the Euclidean distance between the distance profiles of two assets:

While $d_{i,j}$ $\{\displaystyle d_{i,j}\}$ reflects correlation-based proximity between two assets, $\tilde{d}_{i,j}$ $\{\displaystyle \tilde{d}_{i,j}\}$ quantifies dissimilarity across the entire system, as it depends on all pairwise distances.

Hierarchical clustering proceeds by identifying the pair (i, j) $\{\displaystyle (i, j)\}$ with the smallest value of $\tilde{d}_{i,j}$ $\{\displaystyle \tilde{d}_{i,j}\}$ (for $i \neq j$ $\{\displaystyle i \neq j\}$), and forming a new cluster $u[1] = (i, j)$ $\{\displaystyle u[1] = (i, j)\}$. The distance between this new cluster and all remaining items is updated using a linkage criterion. One example is the single linkage (nearest neighbor) method:

The algorithm is repeated recursively: the pair with minimum distance is clustered, the matrix \tilde{D} $\{\displaystyle \tilde{D}\}$ is updated, and clustering continues until all items belong to a single cluster. This produces $N - 1$ $\{\displaystyle N - 1\}$ clusters and a binary tree (dendrogram) that encodes the hierarchical structure.

The clustering process is recorded in a linkage matrix $Y \in \mathbb{R}^{(N-1) \times 4}$ $\{\displaystyle Y \in \mathbb{R}^{(N-1) \times 4}\}$, where each row $(y_{m,1}, y_{m,2}, y_{m,3}, y_{m,4})$ $\{\displaystyle (y_{m,1}, y_{m,2}, y_{m,3}, y_{m,4})\}$ represents a clustering step: $y_{m,1}$ $\{\displaystyle y_{m,1}\}$ and $y_{m,2}$ $\{\displaystyle y_{m,2}\}$ are the merged clusters, $y_{m,3}$ $\{\displaystyle y_{m,3}\}$ is the distance between them, and $y_{m,4}$ $\{\displaystyle y_{m,4}\}$ is the total number of original items included in the cluster. HRP accepts a wide range of clustering metrics and linkage criteria. For further discussion, see Rokach and Maimon (2005), Bruualdi (2010), and the documentation for SciPy's hierarchical clustering tools.

Step 2: Quasi-Diagonalization

At this stage, the rows and columns of the covariance matrix are reordered such that the largest values are concentrated along the diagonal. This process is referred to as a quasi-diagonalization of the covariance matrix. Unlike full diagonalization, this method does not require a change of basis. The resulting structure places similar investments near each other and dissimilar investments farther apart, facilitating block-wise portfolio construction. (See illustrative examples on the right)

The procedure relies on the structure of the linkage matrix generated during hierarchical clustering. Each row in the linkage matrix Y $\{\displaystyle Y\}$ represents the merger of two branches into a cluster. To obtain the ordering of original items, the final row $(y_{N-1,1}, y_{N-1,2})$ $\{\displaystyle (y_{N-1,1}, y_{N-1,2})\}$ is recursively expanded by replacing clusters with their constituents, in the order dictated by the hierarchical tree. This produces a sorted list of indices corresponding to the unclustered items.

This ordering is used in the final step of the HRP algorithm to allocate capital proportionally to the hierarchical structure of asset relationships.

Step 3: Recursive Bisection

This final stage of the Hierarchical Risk Parity (HRP) algorithm computes portfolio weights using the quasi-diagonal covariance matrix. When the covariance matrix is diagonal, inverse-variance weighting is optimal (see Appendix 16.A.2). HRP uses this insight in both bottom-up and top-down directions:

Bottom-up : estimate the variance of a cluster using inverse-variance weights.

Top-down : split capital between clusters inversely proportional to their estimated variances.

The recursive algorithm proceeds as follows:

Initialize a list L with all asset indices: $L = \{1, 2, \dots, N\}$.

Assign initial weights: $w_n = 1$ for all assets.

While any subset in L contains more than one element: Bisect the subset into two halves: L_1 and L_2 . For each subset L_j , compute its covariance matrix V_j and define weights proportional to the inverse of its diagonal entries. Normalize those weights so they sum to 1, and compute the variance of the cluster as $w' V w$. Compute a split factor $\alpha = 1 - (V_1 / (V_1 + V_2))$. Rescale the weights of L_1 by α and of L_2 by $1 - \alpha$.

Bisect the subset into two halves: L_1 and L_2 .

For each subset L_j , compute its covariance matrix V_j and define weights proportional to the inverse of its diagonal entries.

Normalize those weights so they sum to 1, and compute the variance of the cluster as $w' V w$.

Compute a split factor $\alpha = 1 - (V_1 / (V_1 + V_2))$.

Rescale the weights of L_1 by α and of L_2 by $1 - \alpha$.

This ensures that all weights remain between 0 and 1 and sum to 1 at all times.

The following Python function implements this process, where `sortx` is the list of ordered assets returned by Step 2 (Quasi-Diagonalization), `cov` is the covariance matrix:

This stage completes the HRP algorithm. It runs in $O(\log N)$ time in the best case and $O(N)$ in the worst case. The result is a portfolio allocation that respects hierarchical structure while being robust to estimation error.

Numerical Example

In this section, we compute HRP portfolio allocations on the sample correlation matrix used earlier, and compare them to two alternative standard methods:

A minimum-variance portfolio computed using quadratic optimization, specifically the Critical Line Algorithm (CLA). This is the only solution on the efficient frontier that does not depend on expected returns. See Bailey and López de Prado (2013) for an implementation of CLA.

A traditional risk parity portfolio based on inverse-variance weights (IVP).

All portfolios are subject to standard constraints: non-negativity ($0 \leq w_i \leq 1$) and full investment ($\sum_{i=1}^N w_i = 1$). The condition number of the covariance matrix used in this example is approximately 150.93, which is not particularly high and thus does not unduly penalize CLA.

The table on the right shows several notable results. CLA allocates 92.66% of the total portfolio weight to the top five assets, whereas HRP allocates 62.57% to the same group. CLA assigns zero weight to three assets; without the non-negativity constraint, those allocations would have been negative. HRP offers an intermediate allocation structure between the concentrated nature of CLA and the equalizing tendency of IVP.

The concentration observed in CLA is a result of its objective to minimize total portfolio variance. However, the resulting portfolios display similar risk profiles, with standard deviations of $\sigma_{\text{HRP}} = 0.4640$ and $\sigma_{\text{CLA}} = 0.4486$. Thus, CLA reduces risk only marginally while significantly reducing diversification. In this example, the top five positions dominate the CLA portfolio, exposing it to greater vulnerability in adverse market conditions compared to the more balanced HRP allocation.

Monte-Carlo Simulations

While the in-sample variance of the Critical Line Algorithm (CLA) portfolio is lower than that of the Hierarchical Risk Parity (HRP) portfolio in the earlier numerical example, the portfolio with minimum in-sample variance is not necessarily optimal out-of-sample. To evaluate robustness, a Monte Carlo simulation can be conducted, consistent with the methodology in López de Prado (2018).

The simulation procedure consists of the following steps:

Generate 10 time series of Gaussian returns, each with 520 observations (equivalent to two years of daily data), zero mean, and a standard deviation of 10%. To reflect realistic market behavior, random shocks and a random correlation structure are applied to the data, consistent with jump-diffusion models such as Merton (1976).

Using a rolling window of 260 observations (one year), compute portfolio weights under three methodologies: HRP, CLA (minimum-variance), and the Inverse-Variance Portfolio (IVP). Portfolios are re-estimated and rebalanced every 22 observations (monthly frequency).

Calculate the out-of-sample returns of the three portfolios over the subsequent periods. This process is repeated 10,000 times.

As expected, the average out-of-sample returns are close to zero across all methods. The key differences emerge in the variances of the out-of-sample returns: $\sigma_{\mathrm{CLA}}^2 = 0.1157$, $\sigma_{\mathrm{IVP}}^2 = 0.0928$, and $\sigma_{\mathrm{HRP}}^2 = 0.0671$.

Although CLA is designed to minimize portfolio variance, its out-of-sample performance shows the highest variance, exceeding that of HRP by 72.47%. This finding aligns with earlier results from De Miguel et al. (2009). Based on these figures, HRP improves the out-of-sample Sharpe ratio of a CLA strategy by approximately 31.3%. While IVP benefits from assuming a diagonal covariance matrix, its variance remains 38.24% higher than HRP's. These differences are especially important for risk parity strategies, which often employ significant leverage (see Bailey et al., 2014).

The experiment can be repeated with different parameters, including larger asset universes, more frequent or severe shocks, varying correlation structures, or the inclusion of rebalancing costs. Notably, the cumulative transaction costs associated with frequent CLA rebalances can materially degrade performance over time.

The mathematical explanation for HRP's outperformance over CLA and IVP is non-trivial. However, an intuitive rationale can be offered: CLA's concentration makes it vulnerable to idiosyncratic shocks, while IVP's disregard for asset correlations makes it sensitive to systematic shocks involving correlated groups. HRP mitigates both risks by balancing diversification across individual assets and across hierarchical clusters. This results in more robust allocations. For a detailed mathematical proof, see Antonov et al. (2024).

Advantages

HRP algorithms offer several advantages over the (at the time) MVO state-of-the-art methods:

Improved diversification : HRP creates portfolios that are well-diversified across different risk sources. [1]

Robustness : The algorithm has shown to generate portfolios with robust out-of-sample properties.

Flexibility : HRP can handle singular covariance matrices and incorporate various constraints.

Intuitive approach : The clustering-based method provides an intuitive understanding of the portfolio structure. [2]

By combining elements of machine learning, risk parity, and traditional portfolio theory, HRP offers a sophisticated approach to portfolio construction that aims to overcome the limitations of conventional methods.

References