# Hierarchical Risk Parity (HRP): A First-Principles Approach to Portfolio Construction

Based on the work of Marcos López de Prado (2016)
Expanded Educational Treatment

November 12, 2025

**Abstract**

This document provides a comprehensive, first-principles explanation of Hierarchical Risk Parity (HRP), a revolutionary portfolio construction method that addresses fundamental problems in traditional Markowitz optimization. We build understanding from foundational concepts in probability, linear algebra, and graph theory, progressively developing the mathematical machinery needed to understand why HRP represents a paradigm shift in portfolio management. This treatment is designed in the style of rigorous mathematical education platforms, emphasizing conceptual understanding before technical implementation.

## Contents

**1 Part I: Foundations - Building the Intuition . . . 3**
1.1 What is a Portfolio? . . . 3
1.2 Risk and Return: The Two Pillars . . . 3
1.3 Portfolio Return and Risk . . . 4
1.4 Covariance: The Key to Understanding Portfolio Risk . . . 5
1.5 Correlation: A Normalized Measure of Co-movement . . . 6
1.6 The Fundamental Goal: Diversification . . . 7

**2 Part II: Classical Portfolio Theory - Markowitz Optimization . . . 9**
2.1 The Markowitz Mean-Variance Framework . . . 9
2.2 The Efficient Frontier . . . 9
2.3 The Minimum Variance Portfolio . . . 10
2.4 Why Matrix Inversion? . . . 11

**3 Part III: The Problems with Classical Optimization . . . 12**
3.1 The Condition Number: A Measure of Stability . . . 12
3.2 Why Are Covariance Matrices Ill-Conditioned? . . . 13
3.3 Two Sources of Instability . . . 13
3.3.1 Noise-Induced Instability . . . 13
3.3.2 Signal-Induced Instability . . . 14
3.4 Practical Manifestations: The Three Failures . . . 14

4 **Part IV: Mathematical Prerequisites for HRP** **16**
4.1 Distance Metrics and Metric Spaces . . . . . 16
4.2 From Correlation to Distance . . . . . 16
4.3 Graph Theory Basics . . . . . 17
4.4 Hierarchical Clustering . . . . . 18
4.5 From Geometry to Topology: The Key Insight . . . . . 19

5 **Part V: Hierarchical Risk Parity - The Complete Algorithm** **20**
5.1 Overview and Intuition . . . . . 20
5.2 Step 1: Tree Clustering . . . . . 20
5.3 Step 2: Quasi-Diagonalization . . . . . 21
5.4 Step 3: Recursive Bisection . . . . . 21
5.5 Complete HRP Algorithm Summary . . . . . 23
5.6 A Concrete Example . . . . . 23

6 **Part VI: Analysis and Performance** **26**
6.1 Comparison with Other Methods . . . . . 26
6.1.1 Critical Line Algorithm (CLA) / Minimum Variance . . . . . 26
6.1.2 Inverse Variance Portfolio (IVP) / Risk Parity . . . . . 26
6.1.3 Hierarchical Risk Parity (HRP) . . . . . 26
6.2 Monte Carlo Simulation Results . . . . . 27
6.3 Sharpe Ratio Implications . . . . . 28
6.4 Data Efficiency . . . . . 28
6.5 Computational Complexity . . . . . 29
6.6 Advantages of HRP . . . . . 29
6.7 Limitations and Open Questions . . . . . 30

7 **Part VII: The Bigger Picture** **31**
7.1 Machine Learning in Finance . . . . . 31
7.2 Ensemble Methods . . . . . 31
7.3 Philosophical Implications . . . . . 32
7.4 Future Directions . . . . . 32

8 **Conclusion** **34**
8.1 Key Insights . . . . . 34
8.2 From First Principles to Practice . . . . . 34
8.3 Final Thoughts . . . . . 34

# 1 Part I: Foundations - Building the Intuition

Before we can understand Hierarchical Risk Parity, we must first understand what problem it solves. This requires building our understanding from the ground up.

## 1.1 What is a Portfolio?

Definition

A **portfolio** is a collection of financial assets (stocks, bonds, commodities, etc.) held by an investor. Mathematically, a portfolio is characterized by a **weight vector** $\mathbf{w} = (w_1, w_2, \ldots, w_N)^T$ where:

- $N$ is the number of assets
- $w_i$ represents the proportion of total capital allocated to asset $i$
- $\sum_{i=1}^{N} w_i = 1$ (we invest all our capital)
- $w_i \geq 0$ (no short selling, in the simplest case)

Example

Suppose you have $100,000 to invest in three stocks: Apple (AAPL), Microsoft (MSFT), and Google (GOOG). If you invest $40,000 in AAPL, $30,000 in MSFT, and $30,000 in GOOG, your weight vector is:

$$\mathbf{w} = \begin{pmatrix} 0.4 \\ 0.3 \\ 0.3 \end{pmatrix}$$

## 1.2 Risk and Return: The Two Pillars

Every investment decision involves a trade-off between two fundamental quantities:

Definition

**Return** is the percentage change in value of an investment over a time period:

$$r_i(t) = \frac{P_i(t) - P_i(t-1)}{P_i(t-1)}$$

where $P_i(t)$ is the price of asset $i$ at time $t$.

Definition

**Expected Return** is the average return we anticipate:

$$\mu_i = \mathbb{E}[r_i] = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r_i(t)$$

In practice, we estimate this from historical data:

$$\hat{\mu}_i = \frac{1}{T} \sum_{t=1}^{T} r_i(t)$$

Definition

**Risk** (variance) measures the uncertainty or volatility of returns:

$$\sigma_i^2 = \mathbb{E}[(r_i - \mu_i)^2] = \mathrm{Var}(r_i)$$

The **standard deviation** $\sigma_i = \sqrt{\sigma_i^2}$ is often called **volatility**.

Intuition

Why do we use variance as a measure of risk? Consider two investments with the same expected return of 10%:

- Investment A: Returns are always exactly 10% (variance = 0)
- Investment B: Returns vary wildly between -20% and +40% (high variance)

Most investors prefer Investment A because the outcome is predictable. High variance means high uncertainty, which creates anxiety and potential for losses. Thus, variance captures the "riskiness" of an investment.

## 1.3 Portfolio Return and Risk

Now comes a crucial question: if we know the risk and return of individual assets, what is the risk and return of a portfolio combining them?

Theorem

[Portfolio Return] The return of a portfolio is the weighted average of individual asset returns:

$$r_p = \sum_{i=1}^{N} w_i r_i = \mathbf{w}^T \mathbf{r}$$

Therefore, the expected portfolio return is:

$$\mu_p = \mathbb{E}[r_p] = \sum_{i=1}^{N} w_i \mu_i = \mathbf{w}^T \boldsymbol{\mu}$$

*Proof:* This follows directly from the linearity of expectation:

$$\mu_p = \mathbb{E}[\mathbf{w}^T\mathbf{r}] = \mathbf{w}^T\mathbb{E}[\mathbf{r}] = \mathbf{w}^T\boldsymbol{\mu}$$

Portfolio return is straightforward: it's just the weighted average. But portfolio risk is where things get interesting.

## 1.4 Covariance: The Key to Understanding Portfolio Risk

Definition

The **covariance** between two assets $i$ and $j$ measures how they move together:

$$\sigma_{ij} = \text{Cov}(r_i, r_j) = \mathbb{E}[(r_i - \mu_i)(r_j - \mu_j)]$$

Intuition

Covariance captures the following:

- If $\sigma_{ij} > 0$: Assets tend to move in the same direction (both up or both down)
- If $\sigma_{ij} < 0$: Assets tend to move in opposite directions (one up when other is down)
- If $\sigma_{ij} = 0$: Assets move independently

Note that $\sigma_{ii} = \sigma_i^2$ (the variance of asset $i$).

Definition

The **covariance matrix** $\boldsymbol{\Sigma}$ is an $N \times N$ symmetric matrix containing all pairwise covariances:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{N1} & \sigma_{N2} & \cdots & \sigma_N^2 \end{pmatrix}$$

where $\sigma_{ij} = \sigma_{ji}$ (symmetry).

Theorem

[Portfolio Variance] The variance of a portfolio is:

$$\sigma_p^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij} = \mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}$$

*Proof:*

$$
\begin{aligned}
\sigma_p^2 = \text{Var}(r_p) = \text{Var}\left(\sum_{i=1}^{N} w_i r_i\right) \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{N} w_i r_i - \sum_{i=1}^{N} w_i \mu_i\right)^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{N} w_i (r_i - \mu_i)\right)^2\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j (r_i - \mu_i)(r_j - \mu_j)\right] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \mathbb{E}[(r_i - \mu_i)(r_j - \mu_j)] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij} = \mathbf{w}^T \mathbf{\Sigma} \mathbf{w}
\end{aligned}
$$

Key Takeaway

This formula is the foundation of all portfolio theory. It tells us that portfolio risk depends not just on individual asset risks, but crucially on how assets co-move (covariances). This is why diversification works: if assets don't move in perfect lockstep, the portfolio risk can be less than the average individual asset risk.

## 1.5 Correlation: A Normalized Measure of Co-movement

Covariance has a scaling problem: its magnitude depends on the units of measurement. This makes it hard to interpret.

Definition

The **correlation coefficient** between assets $i$ and $j$ is:

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

The **correlation matrix** is:

$$\mathbf{C} = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1N} \\ \rho_{21} & 1 & \cdots & \rho_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{N1} & \rho_{N2} & \cdots & 1 \end{pmatrix}$$

Theorem

[Properties of Correlation] The correlation coefficient satisfies:

1. $-1 \leq \rho_{ij} \leq 1$ for all $i, j$
2. $\rho_{ii} = 1$ (perfect correlation with itself)
3. $\rho_{ij} = 1$ implies perfect positive linear relationship
4. $\rho_{ij} = -1$ implies perfect negative linear relationship
5. $\rho_{ij} = 0$ implies no linear relationship (uncorrelated)

Intuition

Correlation normalizes covariance to a [-1, 1] scale, making it interpretable. If $\rho_{ij} = 0.8$, we know the assets are highly positively correlated. If $\rho_{ij} = 0.1$, they're weakly correlated. This normalization is crucial for comparing relationships across different assets.

## 1.6 The Fundamental Goal: Diversification

Definition

**Diversification** is the practice of combining multiple assets to reduce portfolio risk without necessarily sacrificing expected return.

Example

[The Power of Diversification] Consider two assets with:

- $\mu_1 = \mu_2 = 10\%$ (same expected return)
- $\sigma_1 = \sigma_2 = 20\%$ (same risk)
- $\rho_{12} = 0.3$ (moderately correlated)

If we invest 50% in each ($w_1 = w_2 = 0.5$), the portfolio has:

$$\begin{aligned}
\mu_p &= 0.5(10\%) + 0.5(10\%) = 10\% \quad \text{(same expected return)} \\
\sigma_p^2 &= 0.5^2(20\%)^2 + 0.5^2(20\%)^2 + 2(0.5)(0.5)(0.3)(20\%)(20\%) \\
&= 0.01 + 0.01 + 0.006 = 0.026 \\
\sigma_p &= \sqrt{0.026} = 16.1\% \quad \text{(lower risk than either asset!)}
\end{aligned}$$

We maintained the same expected return but reduced risk from 20% to 16.1%. This is the magic of diversification.

Key Takeaway

Diversification is not just about holding many assets. It's about holding assets that are not perfectly correlated. The lower the correlation, the greater the diversification benefit. This is why correlation structure is at the heart of portfolio construction.

# 2 Part II: Classical Portfolio Theory - Markowitz Optimization

Now that we understand the basics, we can formulate the classic portfolio optimization problem.

## 2.1 The Markowitz Mean-Variance Framework

In 1952, Harry Markowitz revolutionized finance by formalizing portfolio selection as an optimization problem.

Definition

[The Markowitz Problem] Given a universe of $N$ assets with expected returns $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, find the portfolio weights $\mathbf{w}$ that:

$$\begin{aligned} \text{minimize} \quad & \sigma_p^2 = \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^T \boldsymbol{\mu} = \mu_{\text{target}} \\ & \mathbf{w}^T \mathbf{1} = 1 \\ & \mathbf{w} \geq \mathbf{0} \end{aligned} \tag{1}$$

Alternatively, we can maximize the **Sharpe ratio**:

$$\text{Sharpe} = \frac{\mu_p - r_f}{\sigma_p}$$

where $r_f$ is the risk-free rate.

Intuition

Markowitz's insight was simple but profound: given a desired level of return, we should choose the portfolio with minimum risk. Or equivalently, given a desired level of risk, we should choose the portfolio with maximum return. This defines a frontier of optimal portfolios.

## 2.2 The Efficient Frontier

Definition

The **efficient frontier** is the set of portfolios that achieve the maximum possible expected return for each level of risk, or equivalently, the minimum risk for each level of expected return.

Mathematically, the efficient frontier is the curve traced out by solving the Markowitz problem for different values of $\mu_{\text{target}}$.

Intuition

Think of the efficient frontier as the “Pareto frontier” of the risk-return trade-off. Any portfolio below the frontier is inefficient: you could achieve either higher return for the same risk, or lower risk for the same return. Rational investors should only hold portfolios on the efficient frontier.

## 2.3 The Minimum Variance Portfolio

A particularly important point on the efficient frontier is the **minimum variance portfolio** (MVP):

Definition

The **minimum variance portfolio** has the lowest risk among all possible portfolios:

$$\begin{aligned} \mathbf{w}_{\text{MVP}} = \arg\min_{\mathbf{w}} \quad & \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^T \mathbf{1} = 1 \end{aligned} \tag{2}$$

Theorem

[Solution to Minimum Variance Problem] The minimum variance portfolio has weights:

$$\mathbf{w}_{\text{MVP}} = \frac{\boldsymbol{\Sigma}^{-1}\mathbf{1}}{\mathbf{1}^T\boldsymbol{\Sigma}^{-1}\mathbf{1}}$$

*Proof:* We use Lagrange multipliers. The Lagrangian is:

$$\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{1} - 1)$$

Taking derivatives and setting to zero:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} &= 2\boldsymbol{\Sigma}\mathbf{w} - \lambda\mathbf{1} = \mathbf{0} \\ \implies \mathbf{w} &= \frac{\lambda}{2}\boldsymbol{\Sigma}^{-1}\mathbf{1} \end{aligned}$$

Using the constraint $\mathbf{w}^T\mathbf{1} = 1$:

$$\frac{\lambda}{2}\mathbf{1}^T\boldsymbol{\Sigma}^{-1}\mathbf{1} = 1 \implies \frac{\lambda}{2} = \frac{1}{\mathbf{1}^T\boldsymbol{\Sigma}^{-1}\mathbf{1}}$$

Substituting back:

$$\mathbf{w}_{\text{MVP}} = \frac{\boldsymbol{\Sigma}^{-1}\mathbf{1}}{\mathbf{1}^T\boldsymbol{\Sigma}^{-1}\mathbf{1}}$$

Key Takeaway

Notice the appearance of $\boldsymbol{\Sigma}^{-1}$ in the solution. This matrix inversion is where all the problems begin. This seemingly innocent mathematical operation is the root cause of the instability that plagues Markowitz optimization.

## 2.4 Why Matrix Inversion?

Why does solving the Markowitz problem require matrix inversion?

Intuition

The Markowitz problem is a **quadratic programming** problem: we're minimizing a quadratic form $\mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}$ subject to linear constraints. The first-order optimality conditions (setting the gradient to zero) produce a system of linear equations involving $\boldsymbol{\Sigma}$. Solving this system requires computing $\boldsymbol{\Sigma}^{-1}$.

More intuitively: the covariance matrix encodes how assets interact. To find optimal weights, we need to "untangle" these interactions - essentially solving a system of $N$ equations in $N$ unknowns. This untangling is precisely what matrix inversion does.

But here's the problem: not all matrices can be inverted reliably. And even when they can be inverted, small errors in the input can lead to large errors in the output. This brings us to the core issues with Markowitz optimization.

# 3 Part III: The Problems with Classical Optimization

## 3.1 The Condition Number: A Measure of Stability

To understand why Markowitz optimization fails, we need to understand the concept of matrix conditioning.

Definition

[Eigenvalues and Eigenvectors] For a matrix $\boldsymbol{\Sigma}$, a scalar $\lambda$ is an **eigenvalue** and $\mathbf{v}$ is the corresponding **eigenvector** if:

$$\boldsymbol{\Sigma}\mathbf{v} = \lambda\mathbf{v}$$

Intuition

An eigenvector is a direction that the matrix stretches (or compresses) without rotation. The eigenvalue tells us the factor of stretching. For a covariance matrix, eigenvalues represent the variance along principal components, and eigenvectors represent the directions of these components.

For a covariance matrix $\boldsymbol{\Sigma}$ (which is symmetric and positive semi-definite), all eigenvalues are real and non-negative: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$.

Definition

[Condition Number] The **condition number** of a matrix $\boldsymbol{\Sigma}$ is:

$$\kappa(\boldsymbol{\Sigma}) = \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{\lambda_1}{\lambda_N}$$

where $\lambda_{\max}$ is the largest eigenvalue and $\lambda_{\min}$ is the smallest eigenvalue.

Theorem

[Condition Number and Inversion Stability] The condition number measures how sensitive the solution $\mathbf{x}$ of $\boldsymbol{\Sigma}\mathbf{x} = \mathbf{b}$ is to perturbations in $\mathbf{b}$. Specifically, if $\mathbf{b}$ is perturbed by $\Delta\mathbf{b}$, the solution changes by:

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \kappa(\boldsymbol{\Sigma})\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}$$

Intuition

A condition number of 100 means that a 1% error in input can lead to up to a 100% error in output. A condition number of 10,000 means that a 0.01% error in input can lead to a 100% error in output. This is catastrophic for numerical stability.

Key Takeaway

For covariance matrices of financial assets, condition numbers are often in the range of $10^3$ to $10^6$ or higher. This means the matrix inversion is numerically unstable, and small estimation errors in correlations and volatilities lead to wildly different portfolio weights.

## 3.2 Why Are Covariance Matrices Ill-Conditioned?

The condition number explodes when $\lambda_{\min} \to 0$, which happens when the covariance matrix is nearly singular (non-invertible).

Theorem

[Correlation and Condition Number] As the average pairwise correlation $\bar{\rho}$ between assets increases, the smallest eigenvalue $\lambda_{\min}$ decreases, and thus the condition number increases:

$$\kappa(\mathbf{\Sigma}) \to \infty \quad \text{as} \quad \bar{\rho} \to 1$$

Intuition

When assets are highly correlated, they carry redundant information. In the extreme case where all assets are perfectly correlated ($\rho_{ij} = 1$ for all $i, j$), the covariance matrix has rank 1 (all rows are multiples of each other), making it singular (non-invertible). In practice, high correlation leads to near-singularity, making inversion numerically unstable.

Definition

[Markowitz's Curse] The more correlated assets become, the more important diversification becomes (since assets move together). However, this is precisely when the Markowitz optimization becomes most unstable. This paradox is called **Markowitz's Curse**.

## 3.3 Two Sources of Instability

López de Prado identifies two fundamental sources of instability:

### 3.3.1 Noise-Induced Instability

Definition

[Signal-to-Noise Ratio in Finance] Financial return data has an extremely low signal-to-noise ratio. If we denote the true covariance matrix as $\mathbf{\Sigma}_{\text{true}}$ and our estimate as $\hat{\mathbf{\Sigma}}$, then:

$$\hat{\mathbf{\Sigma}} = \mathbf{\Sigma}_{\text{true}} + \mathbf{E}$$

where $\mathbf{E}$ is the estimation error. In financial data, $||\mathbf{E}||$ can be comparable to or even larger than $||\mathbf{\Sigma}_{\text{true}}||$.

Intuition

Covariance estimation requires observing joint movements of assets. But financial returns are noisy: most daily price movements are random fluctuations unrelated to fundamental relationships. To estimate covariances accurately, we need thousands of observations. But we typically have only hundreds of days of data, and using very old data introduces stale information (markets change). This leads to noisy estimates.

Theorem

[Sample Covariance Instability] For $N$ assets and $T$ observations, the sample covariance matrix $\hat{\mathbf{\Sigma}}$ is only reliable when $T \gg N$. A rule of thumb is $T \geq 10N$ for reasonable stability.

For a portfolio of 50 assets, this means we need at least 500 days (about 2 years) of data. But markets aren't stationary over such periods, creating a fundamental tension.

### 3.3.2 Signal-Induced Instability

Even with perfect information (no estimation error), high correlation causes instability.

Example

[Signal-Induced Instability] Consider three assets with perfect information:

- Assets 1 and 2: $\sigma_1 = \sigma_2 = 20\%$, $\rho_{12} = 0.99$
- Asset 3: $\sigma_3 = 20\%$, $\rho_{13} = \rho_{23} = 0.1$

Assets 1 and 2 are nearly identical (99% correlated). The optimizer will try to exploit tiny differences between them, leading to extreme long/short positions (e.g., 150% in asset 1, -130% in asset 2, 80% in asset 3). These extreme positions amplify any small error in the correlation estimate.
This is the signal-induced instability: the high correlation itself (not measurement error) causes numerical problems in optimization.

## 3.4 Practical Manifestations: The Three Failures

These theoretical problems manifest in three practical failures:

1. **Instability**: Small changes to inputs (e.g., changing the data window by a few days) lead to drastically different portfolios. Portfolios are not robust.
2. **Concentration**: The optimizer often places zero weight on most assets and extreme weights on a few, defeating the purpose of diversification. This is because the optimizer exploits small differences between highly correlated assets.
3. **Out-of-Sample Underperformance**: Most surprisingly, Markowitz portfolios often perform worse than naive equal-weighting ($1/N$) in out-of-sample tests. The in-sample "optimal" portfolio overfits to noise and performs poorly on new data.

Key Takeaway

The Markowitz framework is theoretically sound but practically flawed. The problem is not the economic objective (minimizing risk for given return), but the mathematical representation (fully-connected covariance matrix requiring inversion). This motivates us to seek a different mathematical framework.

# 4 Part IV: Mathematical Prerequisites for HRP

To understand HRP, we need to develop new mathematical tools beyond linear algebra. We need concepts from metric geometry and graph theory.

## 4.1 Distance Metrics and Metric Spaces

Definition

[Metric Space] A **metric space** is a set $X$ equipped with a **distance function** (or **metric**) $d : X \times X \to \mathbb{R}$ that satisfies for all $x, y, z \in X$:

1. **Non-negativity**: $d(x, y) \geq 0$
2. **Identity**: $d(x, y) = 0$ if and only if $x = y$
3. **Symmetry**: $d(x, y) = d(y, x)$
4. **Triangle inequality**: $d(x, z) \leq d(x, y) + d(y, z)$

Intuition

A metric space is simply a set where we can meaningfully talk about distances. The axioms ensure that distance behaves intuitively: it's always positive, symmetric, and obeys the triangle inequality (you can't shorten a path by adding a detour).

## 4.2 From Correlation to Distance

Correlation is not a distance metric. We need to convert it.

Definition

[Correlation-Based Distance] Given a correlation matrix $\mathbf{C}$ with entries $\rho_{ij}$, we define the distance between assets $i$ and $j$ as:

$$d_{ij} = \sqrt{\frac{1}{2}(1 - \rho_{ij})}$$

This is called the **correlation distance** or **angular distance**.

Theorem

[Correlation Distance is a Metric] The function $d_{ij} = \sqrt{\frac{1}{2}(1 - \rho_{ij})}$ satisfies all metric axioms.

*Proof sketch:*

1. Non-negativity: Since $-1 \leq \rho_{ij} \leq 1$, we have $0 \leq \frac{1}{2}(1 - \rho_{ij}) \leq 1$, so $d_{ij} \geq 0$.
2. Identity: $d_{ij} = 0 \iff \rho_{ij} = 1 \iff i = j$ (assuming distinct assets have $\rho < 1$).
3. Symmetry: $\rho_{ij} = \rho_{ji}$ implies $d_{ij} = d_{ji}$.

4. Triangle inequality: This requires more work but can be proven using properties of correlation matrices (positive semi-definiteness).

Intuition

Why this particular formula? The mapping $\rho \to \sqrt{\frac{1}{2}(1-\rho)}$ has nice properties:

- When $\rho = 1$ (perfect correlation): $d = 0$ (zero distance - assets are identical)
- When $\rho = 0$ (uncorrelated): $d = \frac{1}{\sqrt{2}} \approx 0.707$
- When $\rho = -1$ (perfect negative correlation): $d = 1$ (maximum distance)

The square root and factor of $\frac{1}{2}$ ensure the metric properties hold.

## 4.3 Graph Theory Basics

Definition

[Graph] A **graph** $G = (V, E)$ consists of:

- A set of **vertices** (or nodes) $V$
- A set of **edges** $E \subseteq V \times V$ connecting pairs of vertices

If there is an edge between every pair of vertices, the graph is **complete**.

Definition

[Weighted Graph] A **weighted graph** assigns a weight $w_{ij}$ to each edge $(i, j) \in E$. For our purposes, the weight represents distance: $w_{ij} = d_{ij}$.

Intuition

In traditional portfolio optimization, we implicitly work with a complete graph: every asset is directly compared to every other asset through the covariance matrix. This creates $\binom{N}{2} = \frac{N(N-1)}{2}$ pairwise relationships to estimate and manage. For 50 assets, that's 1,225 relationships. This complexity is a source of instability.

Definition

[Tree] A **tree** is a connected graph with no cycles. Equivalently:

- A tree with $N$ vertices has exactly $N - 1$ edges
- There is exactly one path between any two vertices
- Removing any edge disconnects the graph

Intuition

Trees are the simplest connected structures. Instead of $\frac{N(N-1)}{2}$ edges (complete graph), a tree has only $N - 1$ edges. For 50 assets, this reduces the complexity from 1,225 relationships to just 49. This dramatic simplification is the key to HRP's robustness.

## 4.4 Hierarchical Clustering

Now we come to the machine learning component: how do we construct a tree from a distance matrix?

Definition

[Hierarchical Clustering] **Hierarchical clustering** is an algorithm that builds a tree structure (called a **dendrogram**) from a set of objects based on their pairwise distances. It proceeds iteratively:

1. Start with each object as its own cluster (singleton)
2. Find the two closest clusters and merge them
3. Repeat until all objects are in a single cluster

But how do we define the distance between clusters (not just objects)?

Definition

[Linkage Methods] A **linkage method** defines the distance between two clusters $A$ and $B$:

- **Single linkage**: $d(A, B) = \min_{i \in A, j \in B} d_{ij}$ (nearest neighbors)
- **Complete linkage**: $d(A, B) = \max_{i \in A, j \in B} d_{ij}$ (farthest neighbors)
- **Average linkage**: $d(A, B) = \frac{1}{|A||B|} \sum_{i \in A} \sum_{j \in B} d_{ij}$ (average distance)
- **Ward linkage**: Minimize the increase in total within-cluster variance

López de Prado uses **single linkage** in the paper, though other choices are possible.

Example

[Hierarchical Clustering in Action] Suppose we have 4 assets with distance matrix:

$$\mathbf{D} = \begin{pmatrix} 0 & 0.2 & 0.8 & 0.9 \\ 0.2 & 0 & 0.7 & 0.85 \\ 0.8 & 0.7 & 0 & 0.3 \\ 0.9 & 0.85 & 0.3 & 0 \end{pmatrix}$$

Step 1: Find minimum distance: $d_{12} = 0.2$. Merge assets 1 and 2 into cluster $C_1 = \{1, 2\}$.
Step 2: Recompute distances. Using single linkage:

$$\begin{aligned} d(C_1, 3) &= \min(d_{13}, d_{23}) = \min(0.8, 0.7) = 0.7 \\ d(C_1, 4) &= \min(d_{14}, d_{24}) = \min(0.9, 0.85) = 0.85 \\ d(3, 4) &= 0.3 \end{aligned}$$

Step 3: Find minimum distance: $d_{34} = 0.3$. Merge assets 3 and 4 into cluster $C_2 = \{3, 4\}$.
Step 4: Merge $C_1$ and $C_2$: $d(C_1, C_2) = 0.7$.
Final dendrogram structure: $\{(\{1, 2\}, \{3, 4\})\}$

Definition

[Dendrogram] A **dendrogram** is a tree diagram showing the hierarchical clustering structure. The height at which two clusters merge represents their distance.

## 4.5 From Geometry to Topology: The Key Insight

Key Takeaway

Traditional portfolio optimization works in **geometry**: it uses the full covariance matrix, which encodes the precise numerical relationships between all assets. This requires matrix inversion, which is unstable.
HRP works in **topology**: it uses only the hierarchical structure of relationships (which assets are close, which are far), encoded in a tree. This doesn't require inversion and is much more robust.
This is analogous to using a subway map (topology: which stations connect) versus a geographic map (geometry: exact distances and angles). For navigation, topology is often more useful and robust to errors.

# 5 Part V: Hierarchical Risk Parity - The Complete Algorithm

We now have all the ingredients to understand HRP. The algorithm has three steps:

## 5.1 Overview and Intuition

Intuition

The HRP philosophy is:

1. Discover the natural hierarchical structure of assets (tree clustering)
2. Reorganize our view of the portfolio according to this structure (quasi-diagonalization)
3. Allocate capital hierarchically, treating clusters as units and distributing weight based on risk (recursive bisection)

This is fundamentally different from Markowitz: instead of solving a global optimization problem requiring matrix inversion, we make a series of simple local decisions based on the hierarchical structure.

## 5.2 Step 1: Tree Clustering

**Goal**: Discover the hierarchical structure of assets.

**Algorithm**:

1. Start with the correlation matrix $\mathbf{C}$ with entries $\rho_{ij}$
2. Convert to distance matrix: $d_{ij} = \sqrt{\frac{1}{2}(1 - \rho_{ij})}$
3. Apply hierarchical clustering with single linkage
4. Output: A dendrogram representing the hierarchical structure

Intuition

The dendrogram groups assets by similarity. Assets that are highly correlated (similar behavior) end up on nearby branches. Assets with different behavior end up on distant branches. This structure captures the essential information about which assets are substitutes (similar) and which are complements (different).

**Mathematical Details**:

Let $\mathcal{T}$ denote the resulting tree (dendrogram). Each internal node in $\mathcal{T}$ represents a cluster, and each leaf represents an individual asset. The tree defines a partial ordering on assets based on their similarity.

## 5.3 Step 2: Quasi-Diagonalization

**Goal**: Reorder the covariance matrix to reflect the hierarchical structure.

**Algorithm**:

1. Perform a depth-first search (or other tree traversal) of the dendrogram
2. Record the order in which leaves (assets) are visited
3. Let $\pi$ be the permutation of asset indices from this traversal
4. Reorder both rows and columns of $\mathbf{\Sigma}$ according to $\pi$

**Result**: A reordered covariance matrix $\tilde{\mathbf{\Sigma}}$ where similar assets are adjacent.

Intuition

Why is this called "quasi-diagonalization"? After reordering, the matrix has a block-diagonal structure: large values (high covariance) cluster along the diagonal in blocks corresponding to clusters. Off-diagonal blocks (representing covariances between distant clusters) have smaller values.

This reveals the hierarchical structure in the covariance matrix. Unlike principal component analysis (PCA), which changes the basis (creates synthetic portfolios), quasi-diagonalization merely reorders the original assets.

**Mathematical Perspective**:

If we denote the permutation matrix as $\mathbf{P}_\pi$, then:

$$\tilde{\mathbf{\Sigma}} = \mathbf{P}_\pi \mathbf{\Sigma} \mathbf{P}_\pi^T$$

This is a similarity transformation that preserves all eigenvalues and the overall structure of $\mathbf{\Sigma}$, but reorders it to reveal the hierarchical clustering.

## 5.4 Step 3: Recursive Bisection

**Goal**: Allocate portfolio weights hierarchically, flowing capital from the root of the tree down to individual assets.

This is the core innovation of HRP. We allocate weights recursively by splitting at each node in the dendrogram.

**Algorithm**:

1. Initialize: Allocate 100% weight to the root cluster (all assets)
2. At each internal node (cluster):
   (a) Identify the two sub-clusters (left and right children)
   (b) For each sub-cluster, compute its "cluster variance" using inverse-variance weighting within the cluster
   (c) Split the parent weight between the two sub-clusters in inverse proportion to their variances
3. Recurse down the tree until reaching leaf nodes (individual assets)

**Detailed Mathematics**:
Let's formalize step 2b and 2c.

Definition

[Cluster Variance] Consider a cluster $C$ containing assets $\{i_1, i_2, \ldots, i_k\}$. The cluster variance $V_C$ is computed as follows:
First, compute **inverse-variance weights** within the cluster:

$$w_j^{\text{IV}} = \frac{1/\sigma_j^2}{\sum_{j \in C} 1/\sigma_j^2} \quad \text{for } j \in C$$

Then, compute the cluster variance:

$$V_C = (\mathbf{w}_C^{\text{IV}})^T \mathbf{\Sigma}_C \mathbf{w}_C^{\text{IV}}$$

where $\mathbf{\Sigma}_C$ is the covariance sub-matrix for assets in cluster $C$.

Intuition

Inverse-variance weighting is a simple diversification strategy: allocate more to less volatile assets, less to more volatile assets. The weight is inversely proportional to variance, so $w_i \propto 1/\sigma_i^2$.
The cluster variance $V_C$ represents the risk of a portfolio invested in cluster $C$ with inverse-variance weights. This is a single number summarizing the cluster's risk.

Definition

[Weight Bisection] Suppose we have a parent cluster $P$ with weight $W_P$, which splits into left child $L$ and right child $R$ with variances $V_L$ and $V_R$. We allocate weights as:

$$W_L = W_P \cdot \frac{V_R}{V_L + V_R}, \quad W_R = W_P \cdot \frac{V_L}{V_L + V_R}$$

Note the inversion: the child with *lower* variance gets *higher* weight. This is the risk parity principle: equalize risk contributions, not nominal weights.

Key Takeaway

The recursive bisection proceeds from general to specific:

- First, split capital between major asset classes (e.g., equities vs. bonds)
- Then, split within each asset class (e.g., US equities vs. international equities)
- Then, split within sectors (e.g., technology vs. healthcare)
- Finally, split between individual assets

At each level, the split is based on risk: lower-risk clusters receive more capital. This ensures balanced risk exposure across the hierarchy.

## 5.5 Complete HRP Algorithm Summary

Hierarchical Risk Parity (HRP) Algorithm

**Input**: Correlation matrix $\mathbf{C}$ and covariance matrix $\boldsymbol{\Sigma}$ of $N$ assets
**Output**: Portfolio weight vector $\mathbf{w}$
**Step 1: Tree Clustering**

1. Compute distance matrix: $d_{ij} = \sqrt{\frac{1}{2}(1 - \rho_{ij})}$
2. Apply hierarchical clustering (single linkage) to produce dendrogram $\mathcal{T}$

**Step 2: Quasi-Diagonalization**

1. Traverse $\mathcal{T}$ (e.g., depth-first) to obtain asset ordering $\pi$
2. Reorder $\boldsymbol{\Sigma}$ according to $\pi$ to get $\tilde{\boldsymbol{\Sigma}}$

**Step 3: Recursive Bisection**

1. Initialize: $W_{\text{root}} = 1$ (100% weight at root)
2. For each internal node in $\mathcal{T}$ (top-down):
   (a) Identify children $L$ and $R$
   (b) Compute cluster variances $V_L$ and $V_R$ using inverse-variance weighting
   (c) Split weight: $W_L = W_P \cdot \frac{V_R}{V_L+V_R}$, $W_R = W_P \cdot \frac{V_L}{V_L+V_R}$
3. Continue until all leaf nodes (individual assets) have weights
4. Return: weight vector $\mathbf{w}$ with $w_i$ for each asset $i$

## 5.6 A Concrete Example

Let's work through a simple 4-asset example.

**Data**:

- Asset 1: $\mu_1 = 8\%$, $\sigma_1 = 15\%$
- Asset 2: $\mu_2 = 10\%$, $\sigma_2 = 20\%$
- Asset 3: $\mu_3 = 6\%$, $\sigma_3 = 10\%$
- Asset 4: $\mu_4 = 7\%$, $\sigma_4 = 12\%$

Correlation matrix:

$$\mathbf{C} = \begin{pmatrix} 1.0 & 0.8 & 0.1 & 0.2 \\ 0.8 & 1.0 & 0.15 & 0.25 \\ 0.1 & 0.15 & 1.0 & 0.7 \\ 0.2 & 0.25 & 0.7 & 1.0 \end{pmatrix}$$

**Interpretation**: Assets 1 and 2 are highly correlated (0.8) - perhaps both are stocks. Assets 3 and 4 are moderately correlated (0.7) - perhaps both are bonds. The correlations between stocks and bonds are low (0.1-0.25).

**Step 1: Tree Clustering**
Distance matrix:

$$d_{12} = \sqrt{\frac{1}{2}(1-0.8)} = \sqrt{0.1} = 0.316$$

$$d_{34} = \sqrt{\frac{1}{2}(1-0.7)} = \sqrt{0.15} = 0.387$$

$$d_{13} = \sqrt{\frac{1}{2}(1-0.1)} = \sqrt{0.45} = 0.671$$

(and so on)

Hierarchical clustering produces:

- First merge: Assets 1 and 2 (smallest distance 0.316) → cluster $C_{12}$
- Second merge: Assets 3 and 4 (distance 0.387) → cluster $C_{34}$
- Final merge: $C_{12}$ and $C_{34}$ → root

Dendrogram structure: $\{(\{1,2\},\{3,4\})\}$

**Step 2: Quasi-Diagonalization**
Traversal order: 1, 2, 3, 4 (assets already in this order, so no reordering needed).

**Step 3: Recursive Bisection**
*At root (weight = 1):*
Children: $L = \{1,2\}$ (stocks), $R = \{3,4\}$ (bonds)
Compute $V_L$ (variance of stock cluster):

$$w_1^{\text{IV}} = \frac{1/0.15^2}{1/0.15^2 + 1/0.20^2} = \frac{44.44}{44.44+25} = 0.64$$
$$w_2^{\text{IV}} = 0.36$$

Stock cluster covariance matrix (from $\mathbf{C}$ and $\boldsymbol{\sigma}$):

$$\boldsymbol{\Sigma}_{12} = \begin{pmatrix} 0.15^2 & 0.8\cdot 0.15\cdot 0.20 \\ 0.8\cdot 0.15\cdot 0.20 & 0.20^2 \end{pmatrix} = \begin{pmatrix} 0.0225 & 0.024 \\ 0.024 & 0.04 \end{pmatrix}$$

$$V_L = (0.64, 0.36)\begin{pmatrix} 0.0225 & 0.024 \\ 0.024 & 0.04 \end{pmatrix}\begin{pmatrix} 0.64 \\ 0.36 \end{pmatrix} = 0.0244$$

$$\sigma_L = \sqrt{0.0244} = 15.6\%$$

Similarly, compute $V_R$ (variance of bond cluster):

$$w_3^{\text{IV}} = \frac{1/0.10^2}{1/0.10^2 + 1/0.12^2} = \frac{100}{100+69.44} = 0.59$$
$$w_4^{\text{IV}} = 0.41$$

$$V_R = 0.0098 \implies \sigma_R = 9.9\%$$

Split weight:

$$W_L = 1\cdot\frac{V_R}{V_L+V_R} = \frac{0.0098}{0.0244+0.0098} = 0.29 \quad \text{(stocks)}$$

$$W_R = 1 \cdot \frac{V_L}{V_L + V_R} = \frac{0.0244}{0.0342} = 0.71 \quad \text{(bonds)}$$

Bonds get 71% because they have lower cluster risk.
*At cluster* $\{1, 2\}$ *(weight = 0.29):*
Split between assets 1 and 2 in inverse proportion to their variances:

$$w_1 = 0.29 \cdot \frac{0.20^2}{0.15^2 + 0.20^2} = 0.29 \cdot \frac{0.04}{0.0625} = 0.186$$

$$w_2 = 0.29 \cdot \frac{0.15^2}{0.0625} = 0.104$$

*At cluster* $\{3, 4\}$ *(weight = 0.71):*

$$w_3 = 0.71 \cdot \frac{0.12^2}{0.10^2 + 0.12^2} = 0.71 \cdot \frac{0.0144}{0.0244} = 0.419$$

$$w_4 = 0.71 \cdot \frac{0.10^2}{0.0244} = 0.291$$

**Final HRP weights**:

$$\mathbf{w}_{\text{HRP}} = (0.186, 0.104, 0.419, 0.291)^T$$

**Interpretation**:

- 29% in stocks (18.6% in asset 1, 10.4% in asset 2)
- 71% in bonds (41.9% in asset 3, 29.1% in asset 4)
- Allocation favors lower-risk assets and lower-risk clusters
- All assets receive positive weight (diversified)

# 6 Part VI: Analysis and Performance

## 6.1 Comparison with Other Methods

How does HRP compare to traditional approaches?

### 6.1.1 Critical Line Algorithm (CLA) / Minimum Variance

**Method**: Solve the quadratic optimization problem:

$$\mathbf{w}_{\text{CLA}} = \arg\min_{\mathbf{w}} \mathbf{w}^T \mathbf{\Sigma} \mathbf{w} \quad \text{s.t.} \quad \mathbf{w}^T \mathbf{1} = 1, \, \mathbf{w} \geq \mathbf{0}$$

**Characteristics**:

- Requires matrix inversion
- Highly concentrated: often assigns zero weight to most assets
- Optimal in-sample but unstable out-of-sample
- Vulnerable to estimation errors

### 6.1.2 Inverse Variance Portfolio (IVP) / Risk Parity

**Method**: Allocate weights inversely proportional to variance:

$$w_i = \frac{1/\sigma_i^2}{\sum_{j=1}^N 1/\sigma_j^2}$$

**Characteristics**:

- No matrix operations required
- Fully diversified: all assets get positive weight
- Ignores correlation structure completely
- Vulnerable to systematic (correlated) risk

### 6.1.3 Hierarchical Risk Parity (HRP)

**Method**: Recursive bisection based on hierarchical clustering (as described)

**Characteristics**:

- No matrix inversion required
- Diversified: typically gives positive weight to all assets
- Incorporates correlation structure through tree hierarchy
- Robust to estimation errors
- Balances idiosyncratic and systematic risk

Key Takeaway

HRP represents a middle ground:

- CLA is too aggressive (exploits tiny differences, concentrates)
- IVP is too naive (ignores correlations)
- HRP uses correlation structure but in a robust, hierarchical way

## 6.2 Monte Carlo Simulation Results

López de Prado tests these methods using 10,000 Monte Carlo simulations. The setup:

1. Generate a random covariance matrix for $N = 50$ assets
2. Use this as the "true" population covariance
3. Sample $T = 260$ observations (1 year of daily returns) from this distribution
4. Estimate covariance from the sample (with noise)
5. Construct portfolios using CLA, IVP, and HRP
6. Evaluate out-of-sample performance using the true covariance

**Key Metric**: Out-of-sample variance $\sigma^2_{\text{OOS}} = \mathbf{w}^T \boldsymbol{\Sigma}_{\text{true}} \mathbf{w}$
**Results** (averaged over 10,000 simulations):

$$\begin{aligned} \sigma^2_{\text{CLA}} &= 0.1157 \quad \text{(Worst - despite being ``optimal'' in-sample)} \\ \sigma^2_{\text{IVP}} &= 0.0928 \\ \sigma^2_{\text{HRP}} &= 0.0671 \quad \text{(Best)} \end{aligned}$$

**Performance Improvements**:

- HRP vs. CLA: $\frac{0.1157-0.0671}{0.1157} = 42\%$ lower variance
- HRP vs. IVP: $\frac{0.0928-0.0671}{0.0928} = 28\%$ lower variance

Intuition

Why does the "optimal" CLA perform worst out-of-sample?
CLA is optimal for the *estimated* covariance matrix. But estimation errors cause it to overfit: it exploits noise as if it were signal. When applied to new data (out-of-sample), these overfitted positions perform poorly.
This is a classic bias-variance tradeoff: CLA has low bias (uses all available information) but high variance (sensitive to estimation errors). HRP has slightly higher bias (uses only tree structure, not full covariance) but much lower variance (robust to errors).

## 6.3 Sharpe Ratio Implications

Lower variance means higher Sharpe ratio (assuming similar returns):

$$\text{Sharpe} = \frac{\mu_p - r_f}{\sigma_p}$$

If HRP reduces variance by 42% compared to CLA:

$$\frac{\text{Sharpe}_{\text{HRP}}}{\text{Sharpe}_{\text{CLA}}} = \sqrt{\frac{\sigma^2_{\text{CLA}}}{\sigma^2_{\text{HRP}}}} = \sqrt{\frac{0.1157}{0.0671}} \approx 1.31$$

HRP can deliver a **31% higher Sharpe ratio** than Markowitz optimization out-of-sample.

## 6.4 Data Efficiency

Another remarkable property of HRP is its data efficiency.

Theorem

[Data Requirements]

- Traditional Markowitz optimization requires $T \gg N$ observations for stable covariance estimation. A rule of thumb is $T \geq 10N$.
- HRP can produce robust portfolios with as few as $T \approx N$ observations.

Example

For a 50-asset portfolio:

- Markowitz needs $\geq 500$ days (about 2 years of data)
- HRP can work with 50-100 days (2-4 months of data)

Intuition

Why is HRP more data-efficient?
Estimating a full $N \times N$ covariance matrix requires $\frac{N(N+1)}{2}$ parameters (due to symmetry). For 50 assets, that's 1,275 parameters.
HRP's hierarchical tree has only $N - 1 = 49$ splitting decisions. While we still use the full covariance matrix for variance calculations at each split, the hierarchical structure effectively regularizes the problem, reducing the effective degrees of freedom.
This is similar to how decision trees in machine learning can work with less data than fully-connected neural networks.

## 6.5 Computational Complexity

Theorem

[Complexity of HRP] The HRP algorithm has time complexity $O(N^2 \log_2 N)$:

- Step 1 (Clustering): $O(N^2 \log_2 N)$ using efficient linkage algorithms
- Step 2 (Quasi-diagonalization): $O(N)$ for tree traversal
- Step 3 (Recursive bisection): $O(N^2)$ for variance calculations at each node

Total: $O(N^2 \log_2 N)$

**Comparison with Markowitz optimization:**

- Computing $\mathbf{\Sigma}^{-1}$: $O(N^3)$ using Gaussian elimination
- Quadratic programming solver: $O(N^3)$ in general

For large portfolios ($N > 1000$), HRP is significantly faster than traditional optimization.

## 6.6 Advantages of HRP

Let's consolidate the advantages:

1. **Robustness**: No matrix inversion means no numerical instability from ill-conditioned covariance matrices.
2. **Stability**: Small changes to input data lead to small changes in portfolio weights (Lipschitz continuity).
3. **Interpretability**: The tree structure is intuitive and mirrors how portfolio managers think (asset class → sector → security).
4. **Flexibility**: Easy to incorporate constraints, investor views, or custom clustering algorithms.
5. **Scalability**: $O(N^2 \log_2 N)$ complexity allows application to large universes.
6. **Data Efficiency**: Requires much less historical data than Markowitz.
7. **Diversification**: Typically produces well-diversified portfolios with all assets receiving positive weight.
8. **Out-of-Sample Performance**: Empirically outperforms both naive diversification and mean-variance optimization.

## 6.7 Limitations and Open Questions

No method is perfect. HRP has limitations:

1. **Hyperparameter Sensitivity**: Results depend on choice of:
    - Distance metric (angular distance, Euclidean, etc.)
    - Linkage method (single, complete, average, Ward)
    - These choices are somewhat arbitrary
2. **Purely Risk-Based**: The base version ignores expected returns (only uses covariance). Extensions to incorporate return forecasts are possible but not fully developed.
3. **Tree Structure Assumption**: Not all asset relationships are hierarchical. Some relationships might be better represented by other graph structures.
4. **Regime Changes**: The paper doesn't explicitly test performance during major market regime changes or crises. How does HRP perform during 2008-style correlations-go-to-one events?
5. **Transaction Costs**: More diversified portfolios might incur higher rebalancing costs. The paper doesn't address turnover.

# 7 Part VII: The Bigger Picture

## 7.1 Machine Learning in Finance

López de Prado emphasizes that HRP is just one example of how machine learning can solve problems in finance.

Key Takeaway

The most valuable applications of ML in finance are often *not* predicting prices, but solving other problems:

- **Portfolio Construction**: As demonstrated by HRP
- **Position Sizing**: Dynamically adjusting bet sizes based on confidence (like Kelly criterion)
- **Regime Detection**: Identifying when market dynamics have changed
- **Meta-Learning**: Learning which strategies work in which environments
- **Backtest Overfitting Detection**: Identifying when a strategy is too fitted to historical data

Intuition

Why is price prediction the ”least interesting” application?
Financial prices have extremely low signal-to-noise ratios. Most price movements are random, and the predictable component is tiny. ML models trained to predict prices often overfit to noise.
In contrast, problems like portfolio construction have more signal: we're leveraging statistical properties (correlations, variances) that are more stable and measurable than price changes. ML can add real value here.

## 7.2 Ensemble Methods

Definition

[Ensemble Methods] Rather than relying on a single portfolio construction method, an **ensemble approach** combines multiple methods, potentially gaining the benefits of each while reducing the weaknesses.

Example

A meta-portfolio could:

- Allocate 40% to HRP
- Allocate 30% to risk parity (IVP)
- Allocate 20% to minimum variance (CLA)
- Allocate 10% to equal weight (1/N)

If the methods have different failure modes, the ensemble is more robust than any single method.

## 7.3 Philosophical Implications

The success of HRP teaches us a deep lesson:

Key Takeaway

[The Representation Problem] In quantitative finance, the choice of mathematical representation is often more important than the choice of optimization algorithm. Markowitz and HRP solve the *same economic problem* (diversify efficiently), but they represent the problem differently:

- Markowitz: Represent as a quadratic program in Euclidean space (geometry)
- HRP: Represent as hierarchical allocation in metric space (topology)

The second representation is more robust because it matches the structure of the data better. Financial assets naturally cluster (sectors, asset classes), and a tree representation captures this better than a fully-connected graph.

This mirrors broader themes in ML and AI: inductive biases matter. The model that best captures the underlying structure of the data will generalize best.

## 7.4 Future Directions

Several extensions of HRP are possible:

1. **Incorporating Returns**: Develop principled ways to tilt HRP portfolios toward higher expected returns while preserving robustness.
2. **Dynamic HRP**: Allow the tree structure to change over time as market regimes shift.
3. **Non-Hierarchical Structures**: Explore other graph representations (e.g., Minimum Spanning Trees, community detection).
4. **Higher Moments**: Extend beyond variance to account for skewness and kurtosis.
5. **Factor Models**: Combine HRP with factor-based views (e.g., value, momentum).

6. **Multi-Period Optimization**: Extend to sequential decision-making with rebalancing.

# 8 Conclusion

Hierarchical Risk Parity represents a paradigm shift in portfolio construction. By moving from geometric optimization (matrix inversion) to topological allocation (hierarchical trees), HRP solves the 60-year-old problem of Markowitz optimization's instability.

## 8.1 Key Insights

1. **The Problem**: Markowitz optimization fails due to ill-conditioned covariance matrices arising from high correlation and noisy estimation.
2. **The Solution**: Represent assets as a hierarchical tree (dendrogram) and allocate capital recursively based on cluster risk.
3. **The Result**: HRP delivers superior out-of-sample performance, greater stability, better diversification, and requires less data than traditional methods.
4. **The Lesson**: The choice of mathematical representation (topology vs. geometry) can be more important than the optimization algorithm.

## 8.2 From First Principles to Practice

We built our understanding from the ground up:

- Started with basic probability (risk, return, covariance)
- Developed portfolio theory (Markowitz optimization)
- Understood the failure modes (condition number, instability)
- Introduced new mathematics (metric spaces, graph theory, clustering)
- Constructed the HRP algorithm (tree clustering, quasi-diagonalization, recursive bisection)
- Analyzed performance (Monte Carlo simulations, comparisons)
- Placed it in context (ML in finance, ensemble methods, philosophy)

This journey from foundations to frontier illustrates how solving real problems often requires stepping outside established frameworks and thinking creatively about representation and structure.

## 8.3 Final Thoughts

HRP is more than just an algorithm; it's a case study in how thoughtful application of machine learning and graph theory can solve long-standing problems in finance. It reminds us that:

- Theoretical optimality (Markowitz's "efficient frontier") doesn't guarantee practical success
- Simplicity and robustness often beat complexity and fragility

- Interdisciplinary thinking (combining finance, ML, and graph theory) can yield breakthrough solutions
- The structure of our representations shapes the quality of our solutions

As López de Prado emphasizes, we are still in the early days of applying ML to finance. HRP is one success story, but many more innovations await those who approach financial problems with mathematical rigor, computational sophistication, and an open mind.

*”The reasonable man adapts himself to the world; the unreasonable one persists in trying to adapt the world to himself. Therefore all progress depends on the unreasonable man.”*
— George Bernard Shaw

**END OF DOCUMENT**