Building Diversified Portfolios that Outperform Out-Of-Sample

Marcos López de Prado

Lawrence Berkeley National Laboratory
Computational Research Division

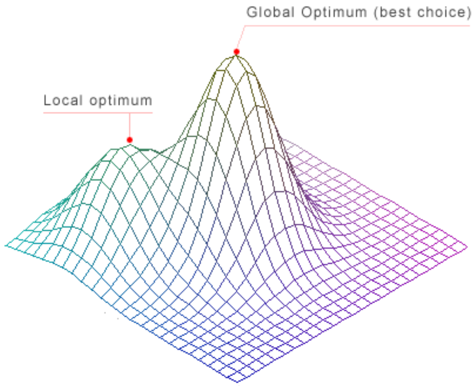
Key Points

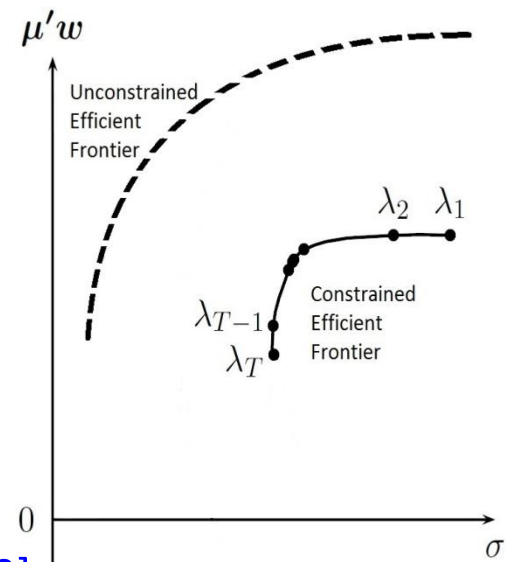
- The problem: Mean-Variance (MV) portfolios are optimal *in-sample*, however they tend to perform poorly *out-of-sample* (even worse than the 1/N naïve portfolio!)
- Two major causes:
  1. Returns can rarely be forecasted with sufficient accuracy
  2. Quadratic optimizers require the inversion of a positive-definite covariance matrix
- A partial solution: To deal with the first cause, some modern approaches drop returns forecasts, e.g. Risk Parity (RP)
- Still, matrix inversion is a major reason why MV and RP underperform out-of-sample (OOS)
- We introduce a new portfolio construction method that substantially improves the OOS performance of diversified portfolios

SECTION I
The Pitfalls of Quadratic Optimizers

Quadratic Optimization

- In 1956, Harry Markowitz developed the Critical Line Algorithm (CLA):
  - CLA is a quadratic optimization procedure specifically designed for inequality constraints portfolio optimization
  - It finds the exact solution after a known number of steps
  - It ingeniously circumvents the Karush-Kuhn-Tucker conditions through the notion of "turning point"
- A turning point occurs when a previously free weight hits a boundary
- The constrained efficient frontier between two neighbor turning points can be reformulated as an unconstrained problem
- CLA solves the optimization problem by finding *the sequence of turning points*

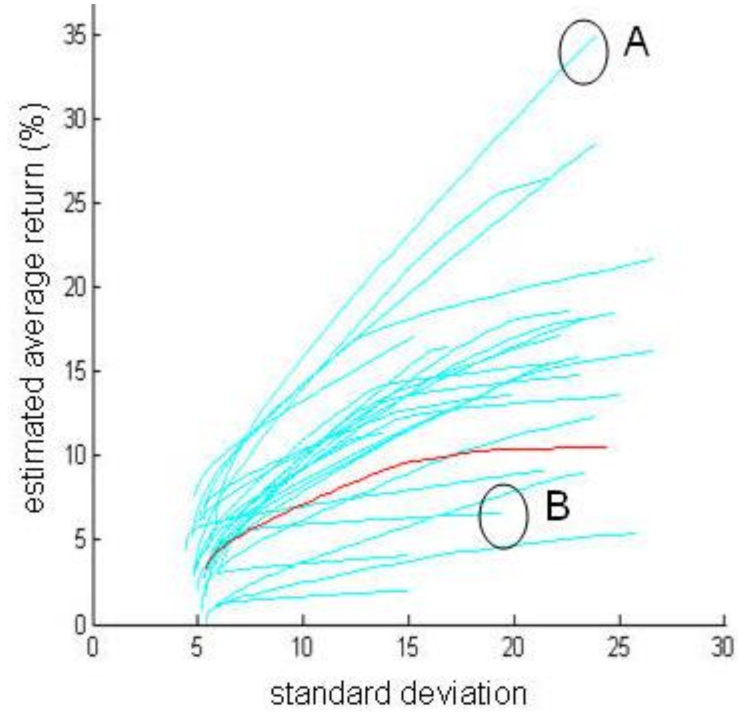




Open source CLA Python library: [Bailey and López de Prado [2013]]

Risk Parity

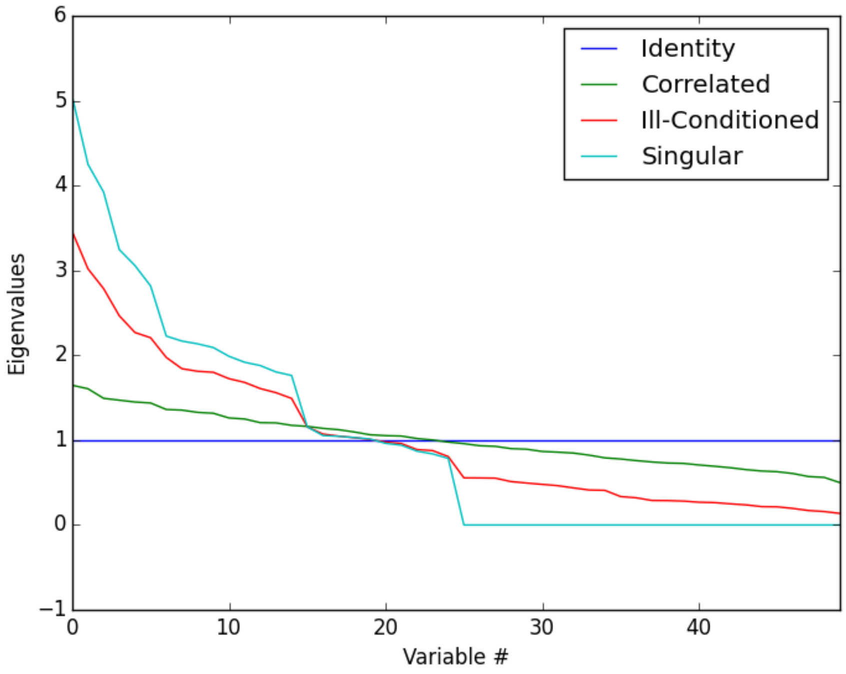
- Numerous studies show that quadratic optimizers in general produce unreliable solutions, e.g. Michaud [1998]
- One major reason for this is, returns can rarely be forecasted with sufficient confidence
- Small forecasting errors can lead to dramatically different Efficient Frontiers



As a consequence, many authors have dropped returns forecasting altogether, giving rise to risk-based asset allocation approaches. E.g.: Risk parity

Markowitz's curse

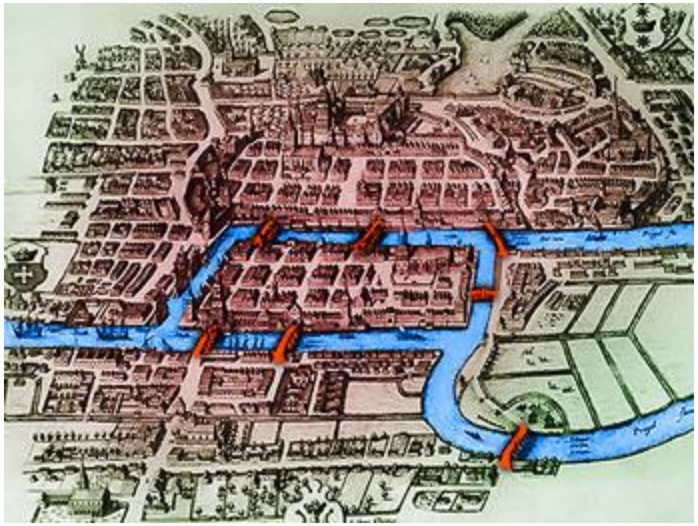
- De Miguel et al. [2009] show that many of the best known quadratic optimizers underperform the Naïve 1/N allocation OOS, even after dropping forecasted returns!
- The reason is, quadratic optimizers require the inversion of a positive-definite covariance matrix
- The *condition number* of a covariance matrix is the ratio between its highest and smallest (in moduli) eigenvalues
- The more correlated the assets, the higher the condition number, and the more unstable is the inverse matrix
- **Markowitz's curse: Quadratic optimization is likely to fail precisely when there is a greater need for finding a diversified portfolio**

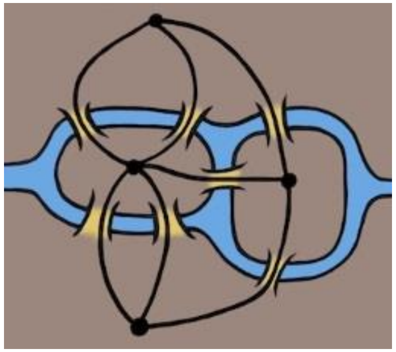


SECTION II
From Geometry To Topology

Topology

Is it possible to walk through the city of Königsberg crossing each bridge once and only once, ending at the starting point?

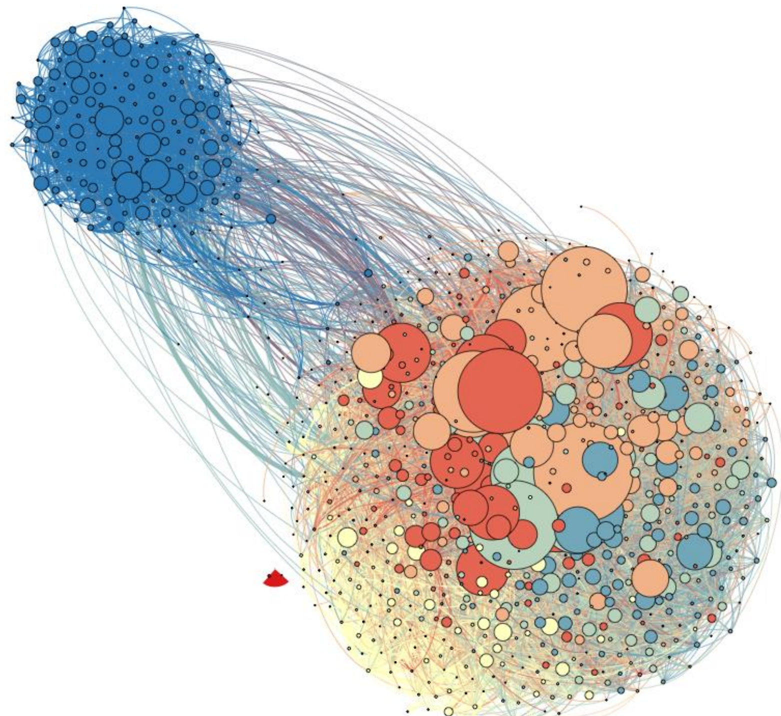




- Around 1735, Leonhard Euler asked this question
- Euler was one of the first to recognize that Geometry could not solve this problem
- **Hint: The relevant information is not the geometric location of the bridges, but their logical relations**

Graph Theory

- A graph can be understood as a **relational map** between pairs of items
- Once considered a branch of Topology, Graph Theory has grown to become a Mathematical subject in its own right
- Graph theory can answer questions regarding the logical structure, architecture and dynamics of a complex system



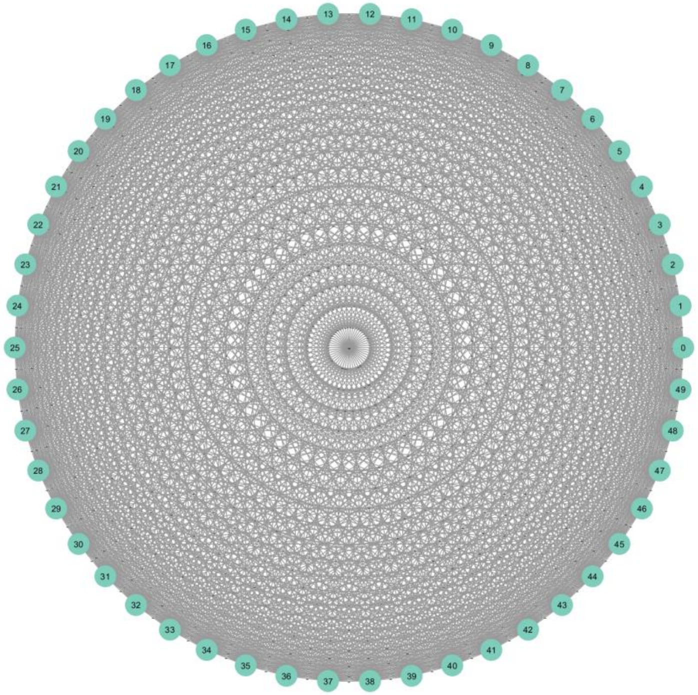
Graph Theory is applied by Google to rank hyperlinks, by GPS systems to find your shortest path home, by LinkedIn to suggest connections, by the NSA to track terrorists... In the example above, Graph Theory is used to derive the [political inclination of a community](), based on the speeches they re-tweet

Topology & Graph Theory at work

- Subway plans are not geographic maps, but topological representations of how lines and stations are interconnected
- Adding to that map geographic details would make it more difficult to solve problems such as how to find alternative routes, minimize waiting time, avoid congestion paths, etc.

What does it mean "inverting the matrix"?

- One reason for the instability of quadratic optimizers is that the vector space is modelled as a complete (fully connected) graph, where every node is a potential candidate to substitute another
- In algorithmic terms, inverting the matrix means evaluating the rates of substitution across the complete graph

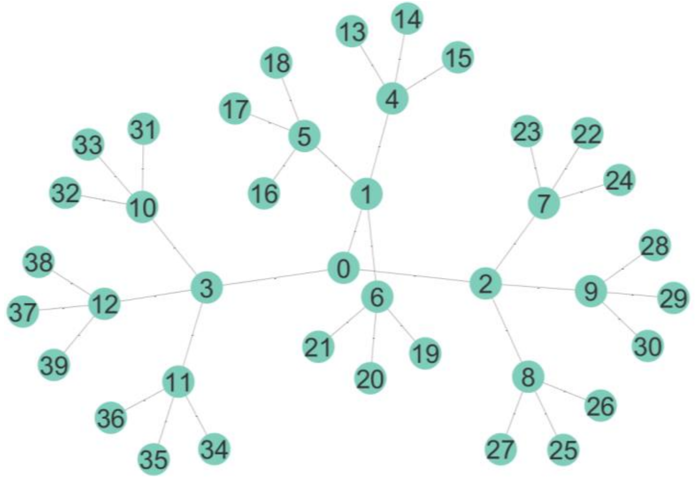


- For a numerically-ill conditioned covariance matrix, small estimation errors over several edges lead to grossly incorrect inversions
- Correlation matrices lack the notion of hierarchy, because all investments are potential substitutes to each other
- Intuitively it would be desirable to drop unnecessary edges

SECTION III
Hierarchical Risk Parity (HRP)

Adding a Hierarchical Structure

- A tree structure introduces two desirable features:
  - It has only *N-1* edges to connect *N* nodes, so the weights only rebalance among peers at various hierarchical levels
  - The weights are distributed top-down, consistent with how many asset managers build their portfolios, e.g. from asset class to sectors to individual securities



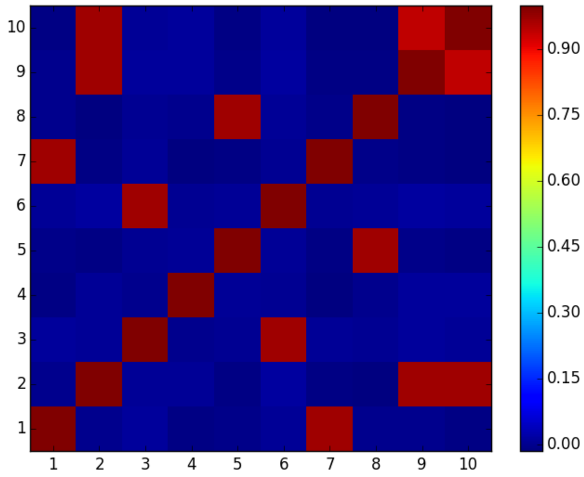
- The HRP algorithm works in three stages:
  1. **Tree Clustering**: Group similar investments into clusters, based on a proper distance metric
  2. **Quasi-diagonalization**: Reorganize the rows and columns of the covariance matrix, so that the largest values lie along the diagonal
  3. **Recursive bisection**: Split allocations through recursive bisection of the reordered covariance matrix

Stage 1: Tree Clustering (1/3)

- The only input needed is the correlation matrix, of size NxN

1. Define the distance measure $d: (X_i, X_j) \subset B \rightarrow \mathbb{R} \in [0,1]$, $d_{i,j} = d[X_i, X_j] = \sqrt{\frac{1}{2}(1-\rho_{i,j})}$, where *B* is the Cartesian product of items in $\{1, \ldots, i, \ldots, N\}$. This forms a proper metric space *D*
2. Compute the Euclidean distance on *D*, $\tilde{d}: (D_i, D_j) \subset B \rightarrow \mathbb{R} \in [0, \sqrt{N}] = \sqrt{\sum_{n=1}^{N}(d_{n,i} - d_{n,j})^2}$

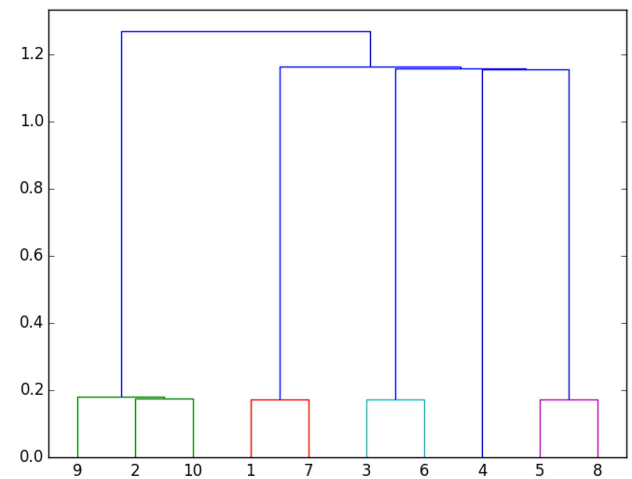
- Note the difference between distance metrics $d_{i,j}$ and $\tilde{d}_{i,j}$. Whereas $d_{i,j}$ is defined on column-vectors of *X*, $\tilde{d}_{i,j}$ is defined on column-vectors of *D* (a distance of distances)



Stage 1: Tree Clustering (2/3)

3. Cluster together the pair of columns (i^*, j^*) such that
$$(i^*, j^*) = \mathrm{argmin}_{\substack{(i,j) \\ i \neq j}} \{\tilde{d}_{i,j}\}$$
4. Update $\{\tilde{d}_{i,j}\}$ with the new cluster
5. Apply steps 3-4 recursively until all $N - 1$ clusters are formed

- Similar items are clustered together, in a tree structure where two leaves are bundled together at each iteration
- The dendogram's y-axis reports the distance between the two joining leaves

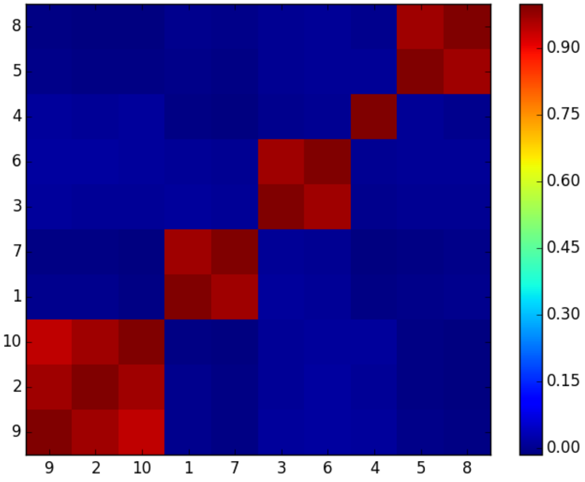


Stage 1: Tree Clustering (3/3)

```
#------------------------------------------------------------------------
import scipy.cluster.hierarchy as sch
import numpy as np
import pandas as pd
cov,corr=x.cov(),x.corr()
dist=((1-corr)/2.)**.5 # distance matrix
link=sch.linkage(dist,'single') # linkage matrix
```

Stage 2: Quasi-Diagonalization (1/2)

- This stage places correlated items together, and uncorrelated items far apart. This is accomplished by
  - replacing clusters with their components recursively, until no clusters remain
  - replacements preserve the order of the clustering
- Because the resulting covariance is quasi-diagonal, we define the variance of a continuous subset $L_i \in L$ as the quadratic form $\tilde{V}_i \equiv \tilde{w}_i{}' V_i \tilde{w}_i$, where L is the sorted list of all items and
  - V_i is the covariance matrix between the constituents of subset L_i
  - $\tilde{w}_i = \text{diag}[V_i]^{-1} \frac{1}{\text{tr}[\text{diag}[V_i]^{-1}]}$, where $\text{diag}[.]$ and $\text{tr}[.]$ are the diagonal and trace operators
- This definition of $\tilde{V}_i$ is motivated by the fact that the inverse-variance allocation is optimal for a diagonal covariance matrix



Stage 2: Quasi-Diagonalization (2/2)

```
sortIx=getQuasiDiag(link)
#------------------------------------------------------------------------------
def getQuasiDiag(link):
    # Sort clustered items by distance
    link=link.astype(int)
    sortIx=pd.Series([link[-1,0],link[-1,1]])
    numItems=link[-1,3] # number of original items
    while sortIx.max()>=numItems:
        sortIx.index=range(0,sortIx.shape[0]*2,2) # make space
        df0=sortIx[sortIx>=numItems] # find clusters
        i=df0.index;j=df0.values-numItems
        sortIx[i]=link[j,0] # item 1
        df0=pd.Series(link[j,1],index=i+1)
        sortIx=sortIx.append(df0) # item 2
        sortIx=sortIx.sort_index() # re-sort
        sortIx.index=range(sortIx.shape[0]) # re-index
    return sortIx.tolist()
```

Stage 3: Recursive Bisection (1/2)

- This stage carries out a top-down allocation

1. Assign a unit weight to all items: $w_n = 1, \forall n = 1, \ldots, N$
2. Recursively bisect a list L_i of items into two lists $L_i^{(1)} \cup L_i^{(2)}$
3. Compute the variance $\tilde{V}_i^{(j)}$ of $L_i^{(j)}, j = 1,2$
4. Compute the split factor: $\alpha_i = 1 - \frac{\tilde{V}_i^{(1)}}{\tilde{V}_i^{(1)} + \tilde{V}_i^{(2)}}$, so that $0 \leq \alpha_i \leq 1$
5. Re-scale allocations w_n by a factor of $\alpha_i, \forall n \in L_i^{(1)}$
6. Re-scale allocations w_n by a factor of $(1 - \alpha_i), \forall n \in L_i^{(2)}$
7. Stop once $|L_i| = 1, \forall L_i \in L$

- This algorithm takes advantage of the quasi-diagonalization bottom-up (step 3) and top-down (step 4)

Stage 3: Recursive Bisection (2/2)

```
hrp=getRecBipart(cov,sortIx)
#------------------------------------------------------------------------------
def getRecBipart(cov,sortIx):
    # Compute HRP alloc
    w=pd.Series(1,index=sortIx)
    cItems=[sortIx] # initialize all items in one section
    while len(cItems)>0:
        cItems=[i[j:k] for i in cItems for j,k in ((0,len(i)/2), \
            (len(i)/2,len(i))) if len(i)>1] # bi-section
        for i in xrange(0,len(cItems),2): # parse in pairs
            cItems0=cItems[i] # section 1
            cItems1=cItems[i+1] # section 2
            cVar0=getClusterVar(cov,cItems0)
            cVar1=getClusterVar(cov,cItems1)
            alpha=1-cVar0/(cVar0+cVar1)
            w[cItems0]*=alpha # weight 1
            w[cItems1]*=1-alpha # weight 2
    return w
```

A Numerical Example

- Simulate a matrix of observations *X*, of order $(10000x10)$
- Add random jumps and a random correlation structure
- Apply three alternative allocation methods:
  - Quadratic optimization, represented by CLA
  - Risk parity, represented by the Inverse Variance Portfolio (IVP)
  - Hierarchical allocation, represented by HRP

- CLA concentrates weights on a few investments, hence becoming exposed to idiosyncratic shocks
- IVP evenly spreads weights through all investments, ignoring the correlation structure. This makes it vulnerable to systemic shocks
- HRP diversifies across clusters & items

Weight #	CLA	HRP	IVP
1	14.44%	7.00%	10.36%
2	19.93%	7.59%	10.28%
3	19.73%	10.84%	10.36%
4	19.87%	19.03%	10.25%
5	18.68%	9.72%	10.31%
6	0.00%	10.19%	9.74%
7	5.86%	6.62%	9.80%
8	1.49%	9.10%	9.65%
9	0.00%	7.12%	9.64%
10	0.00%	12.79%	9.61%

SECTION IV
Out-Of-Sample Monte Carlo Experiments

Experiment Design

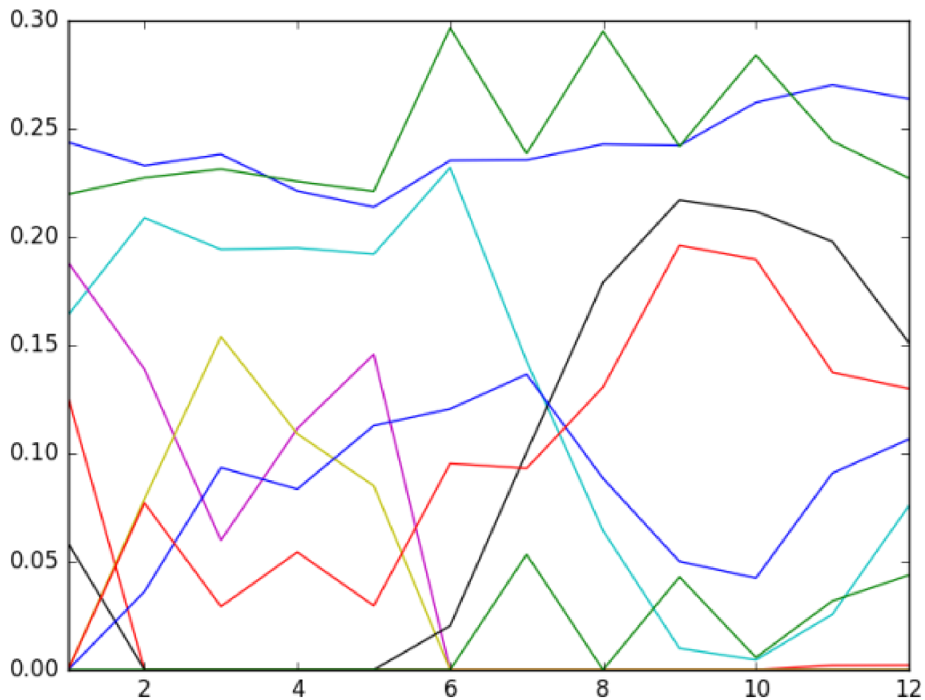
- By definition, CLA has the lowest variance *in-sample*
- **But what method delivers the lowest variance *out-of-sample*?**

1. Generate 10 series of random Gaussian returns (520 observations, equivalent to 2 years of daily history), with 0 mean and an arbitrary standard deviation of 10%
   - Add random shocks, idiosyncratic and common to correlated assets
   - Add a random correlation structure
2. Compute HRP, CLA and IVP portfolios by looking back at 260 observations (a year of daily history)
   - These portfolios are re-estimated and rebalanced every 22 observations (equivalent to a monthly frequency).
3. Compute the out-of-sample returns associated with the three portfolios: CLA, IVP, HRP
4. This procedure is repeated 10,000 times

CLA Allocations

- Variance of the out-of-sample portfolio returns: $\boldsymbol{\sigma^2_{CLA} = .1157}$
- Although **CLA**'s goal is to deliver the lowest variance (that is the objective of its optimization program), its performance happens to exhibit the highest variance *out-of-sample*, and 72.47% greater variance than HRP's

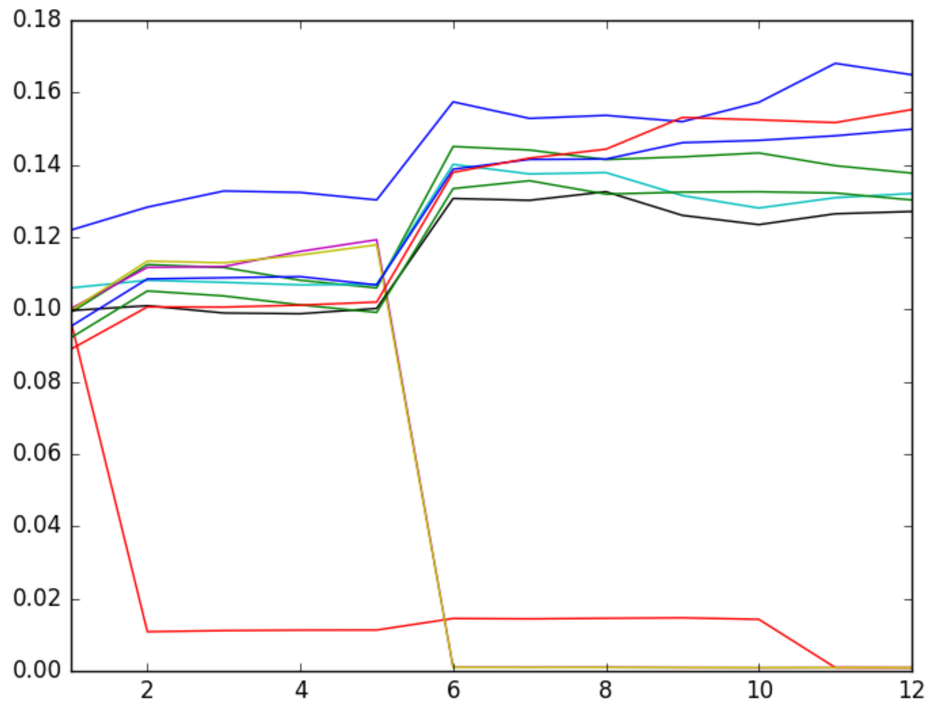
- Let's pick one of the 10,000 experiments, and see how CLA allocations changed between rebalances.
- CLA allocations respond erratically to idiosyncratic and common shocks



IVP Allocations

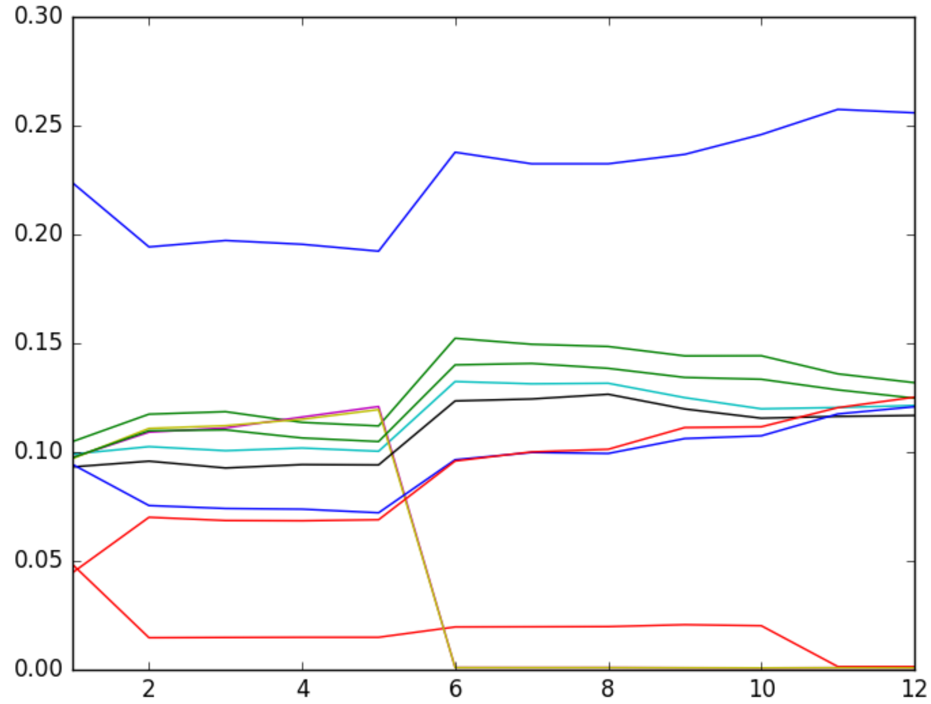
- Variance of the out-of-sample portfolio returns: $\boldsymbol{\sigma^2_{IVP} = .0928}$
- Assuming that the covariance matrix is diagonal brings some stability to the **IVP**, however its variance is still 38.24% greater than HRP's

- IVP's response to idiosyncratic and common shocks is to reduce the allocation to the affected investment, and spread that former exposure across all other investments
- Consequently, IVP allocations among the unaffected investments grow over time, regardless of their correlation



HRP Allocations

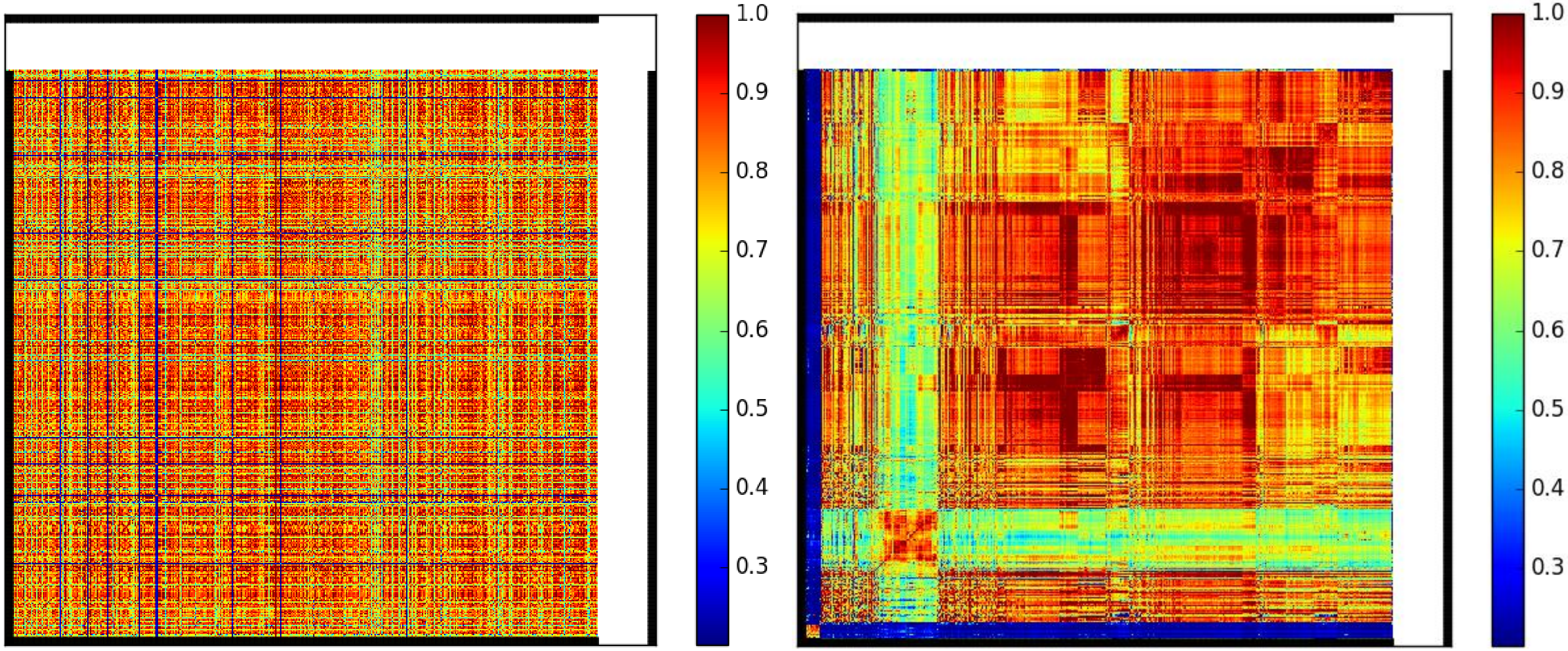
- Variance of the out-of-sample portfolio returns: $\sigma^2_{HPC} = .0671$
- HRP's response to the *idiosyncratic shock* is to reduce the allocation to the affected investment, and use that reduced amount to increase the allocation to a *correlated* investment that was unaffected
- As a response to the *common shock*, HRP reduces allocation to the affected investments, and increases allocation to *uncorrelated* ones (with lower variance)
- **Because Risk Parity funds are leveraged, this variance reduction is critical**



SECTION V
Further Research

Big Data is Hierarchical

- Small-data techniques developed decades and centuries ago (factor models, regression analysis, econometrics) fail to recognize the hierarchical nature of financial Big Data
- A PCA analysis on a large fixed income universe suffers the same drawbacks we described for CLA



Next Steps

- At its core, HRP is essentially a robust procedure to avoid matrix inversions
- It is relatively straightforward to incorporate forecasted returns and Black-Litterman-style views to this hierarchical approach
- The same ideas underlying HRP can be used to replace many econometric regression methods, notorious for their unstable outputs (like VAR or VECM)
- Traditional optimization or econometric methods fail to recognize the hierarchical structure of financial Big Data, where the **numerical instabilities defeat the benefits of the analysis, resulting in unreliable outcomes**

THANKS FOR YOUR ATTENTION!

Notice:

The full paper is available at:
http://ssrn.com/abstract=2708678

For information on other disruptive financial technologies:
http://www.ssrn.com/link/Big-Data-Innovative-Financial-Tech-RES.html

Bio

Marcos López de Prado is Senior Managing Director at *Guggenheim Partners*. He is also a Research Fellow at *Lawrence Berkeley National Laboratory*'s Computational Research Division (U.S. Department of Energy's Office of Science), where he conducts unclassified research in the mathematics of large-scale financial problems and supercomputing.

Before that, Marcos was Head of Quantitative Trading & Research at Hess Energy Trading Company (the trading arm of *Hess Corporation*, a Fortune 100 company) and Head of Global Quantitative Research at *Tudor Investment Corporation*. In addition to his 17 years of trading and investment management experience at some of the largest corporations, he has received several academic appointments, including Postdoctoral Research Fellow of *RCC at Harvard University* and Visiting Scholar at *Cornell University*. Marcos earned a Ph.D. in Financial Economics (2003), a second Ph.D. in Mathematical Finance (2011) from *Complutense University*, is a recipient of the National Award for Excellence in Academic Performance by the Government of Spain (National Valedictorian, 1998) among other awards, and was admitted into *American Mensa* with a perfect test score.

Marcos serves on the Editorial Board of the *Journal of Portfolio Management* (IIJ) and the *Journal of Investment Strategies* (Risk). He has collaborated with ~30 leading academics, resulting in some of the most read papers in Finance (SSRN), multiple international patent applications on Algorithmic Trading, three textbooks, numerous publications in the top Mathematical Finance journals, etc. Marcos has an Erdös #2 and an Einstein #4 according to the *American Mathematical Society*. www.QuantResearch.org

Disclaimer

- The views expressed in this document are the authors' and do not necessarily reflect those of the organizations he is affiliated with
- No investment decision or particular course of action is recommended by this presentation
-