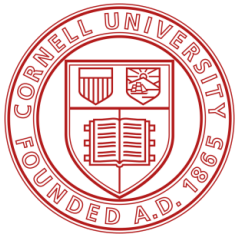


Advances in Financial Machine Learning: Lecture 7/10

Marcos López de Prado, Ph.D.

Advances in Financial Machine Learning

ORIE 5256

Key Points

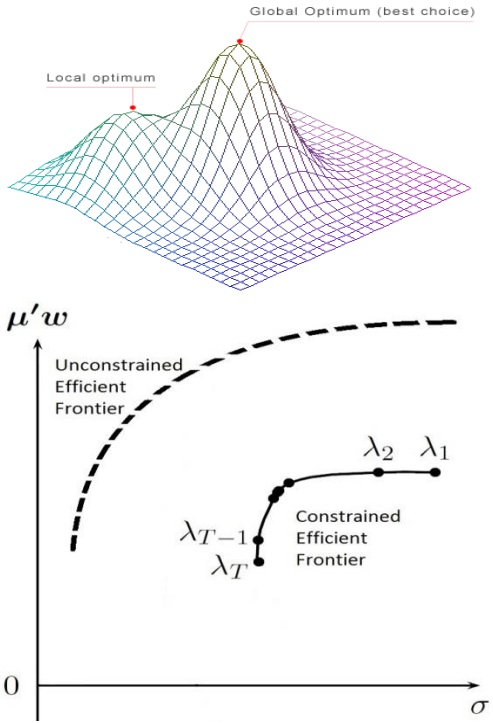
- The problem: Mean-Variance (MV) portfolios are optimal *in-sample*, however they tend to perform poorly *out-of-sample* (even worse than the 1/N naïve portfolio!)
- Two major causes:
  1. Returns can rarely be forecasted with sufficient accuracy
  2. Quadratic optimizers require the inversion of a 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)
- Today, we will learn a new portfolio construction method that substantially improves the OOS performance of diversified portfolios

SECTION I
The Pitfalls of Quadratic Optimizers

Quadratic Optimization

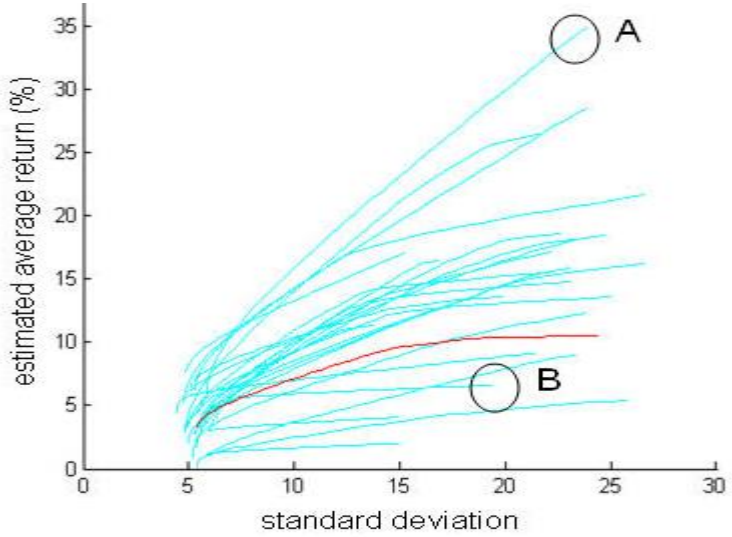
- In 1956, Harry Markowitz developed the Critical Line Algorithm (CLA):
  - CLA is a quadratic optimization procedure specifically designed for inequality constraints portfolio optimization
  - It finds the exact solution after a known number of steps
  - It ingeniously circumvents the Karush-Kuhn-Tucker 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

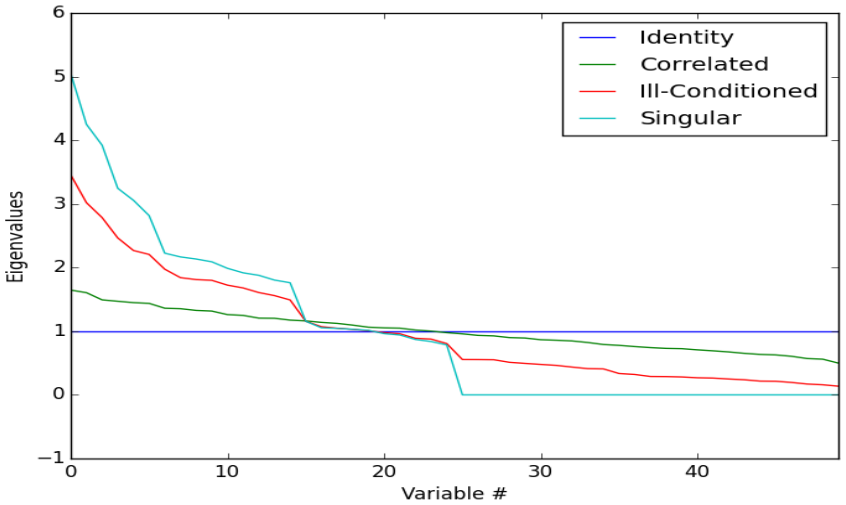


As a consequence, many authors have dropped returns forecasting altogether, giving rise to risk-based asset allocation approaches. E.g.: Risk parity

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**



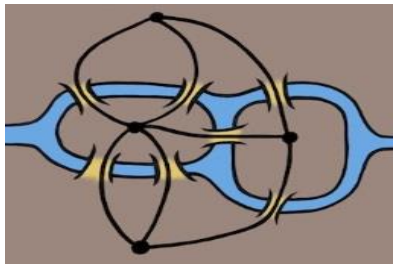
SECTION II
From Geometry To Topology

Topology

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

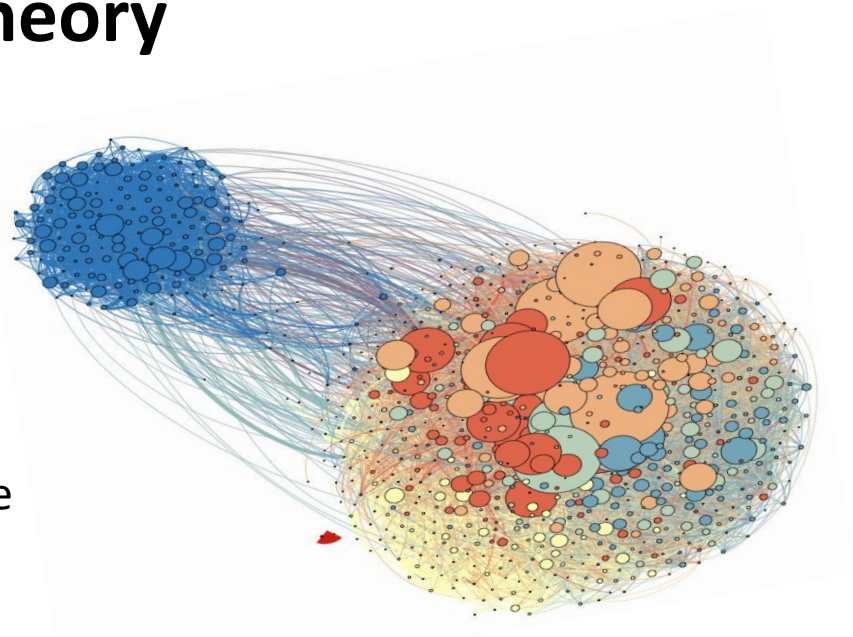
- Around 1735, Leonhard Euler asked this question
- Euler was one of the first to recognize that Geometry could not solve this problem

- **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



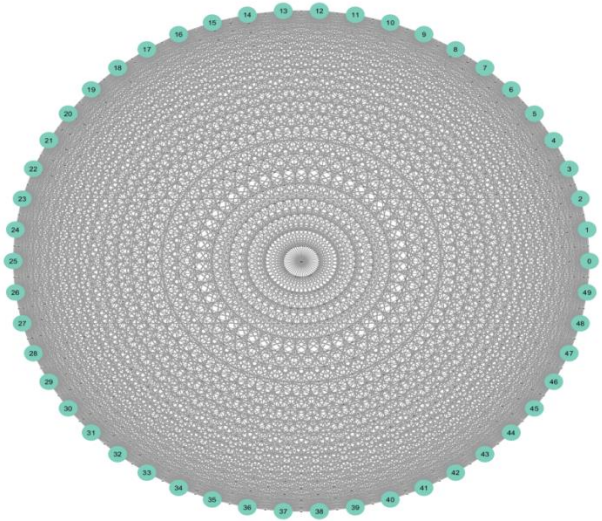
Graph Theory is applied by Google to rank hyperlinks, by GPS systems to find your shortest path home, by LinkedIn to suggest connections, by the NSA to track terrorists... In the example above, Graph Theory is used to derive the political inclination of a community, based on the speeches they re-tweet

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

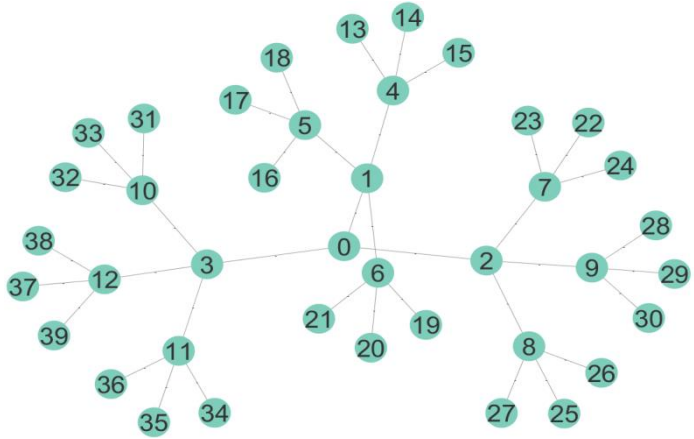


- For a numerically-ill conditioned covariance matrix, small estimation errors over several edges lead to grossly incorrect inversions
- Correlation matrices lack the notion of hierarchy, because all investments are potential substitutes to each other
- Intuitively it would be desirable to drop unnecessary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



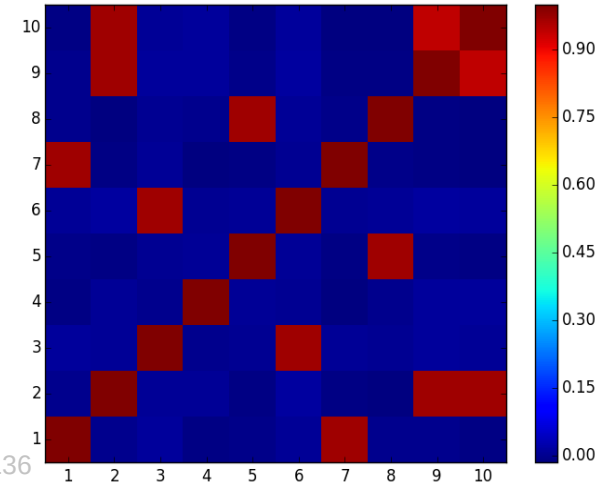
- The HRP algorithm works in three stages:
  1. **Tree Clustering**: Group similar investments into clusters, based on a proper distance metric
  2. **Quasi-diagonalization**: Reorganize the rows and columns of the covariance matrix, so that the largest values lie along the diagonal
  3. **Recursive 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*, $\tilde{d}$: $(D_i, D_j) \subset B \rightarrow \mathbb{R} \in [0, \sqrt{N}] = \sqrt{\sum_{n=1}^{N}(d_{n,i} - d_{n,j})^2}$

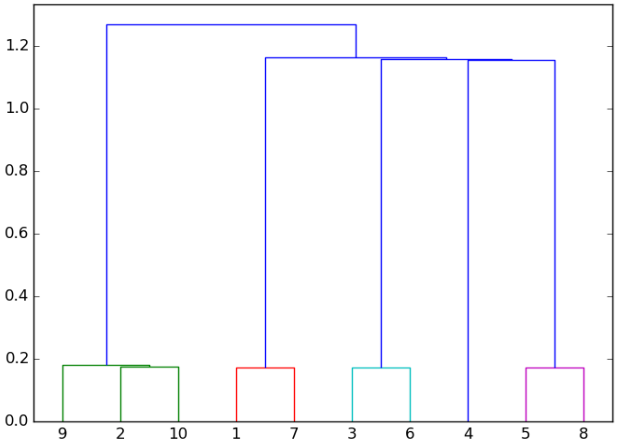
- Note the difference between distance metrics $d_{i,j}$ and $\tilde{d}_{i,j}$. Whereas $d_{i,j}$ is defined on column-vectors of *X*, $\tilde{d}_{i,j}$ is defined on column-vectors of *D* (a distance of distances)



Stage 1: Tree Clustering (2/3)

3. Cluster together the pair of columns (i^*, j^*) such that
$$(i^*, j^*) = \underset{\substack{(i,j) \\ i \neq j}}{\operatorname{argmin}} \{\tilde{d}_{i,j}\}$$
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

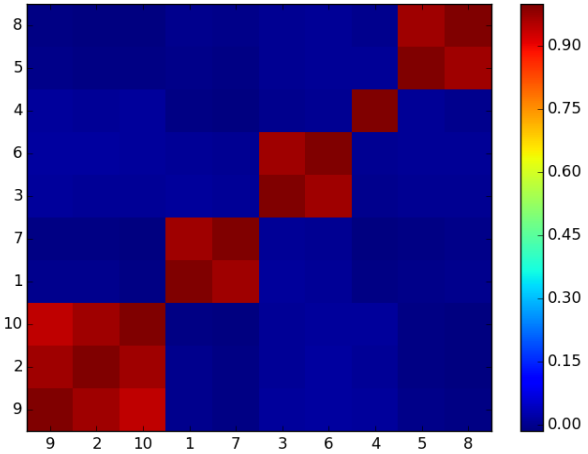


Stage 1: Tree Clustering (3/3)

```
#-----------------------------------------------------------------------------
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: $w_n = 1, \forall n = 1, \dots, N$
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 α_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

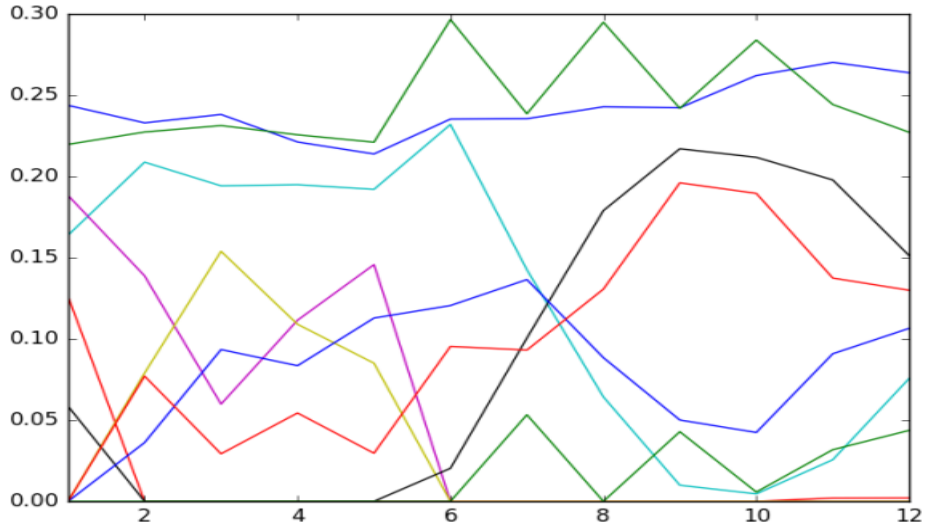
- By definition, CLA has the lowest variance *in-sample*
- **But what method delivers the lowest variance *out-of-sample*?**

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: $\sigma^2_{CLA} = .1157$
- 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

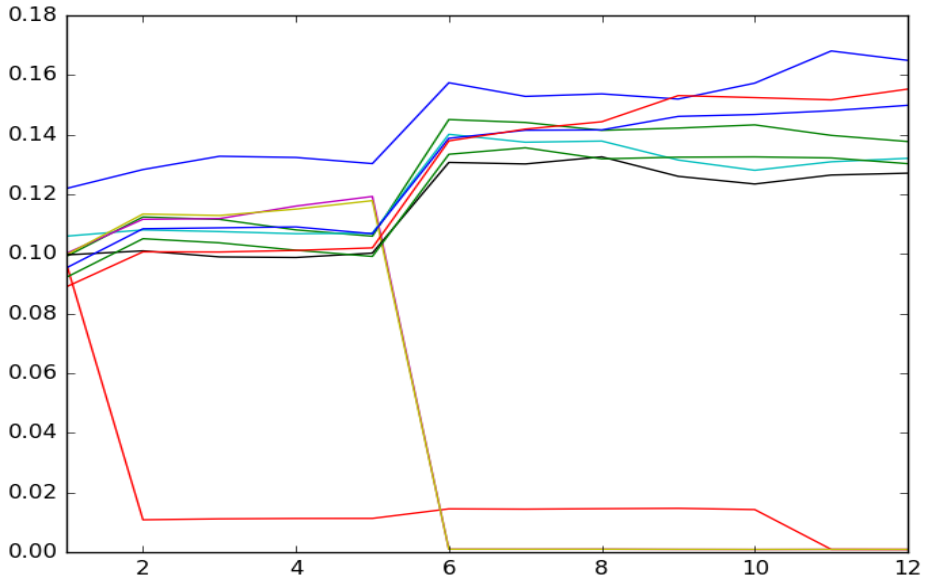
- Let's pick one of the 10,000 experiments, and see how CLA allocations changed between rebalances.
- CLA allocations respond erratically to idiosyncratic and common shocks



IVP Allocations

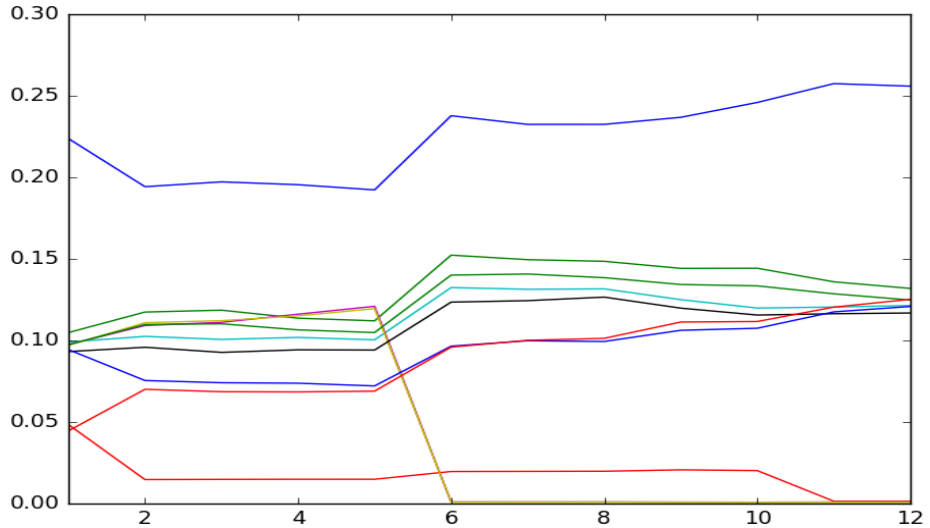
- Variance of the out-of-sample portfolio returns: $\sigma^2_{IVP} = .0928$
- Assuming that the covariance matrix is diagonal brings some stability to the **IVP**, however its variance is still 38.24% greater than HRP's

- 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

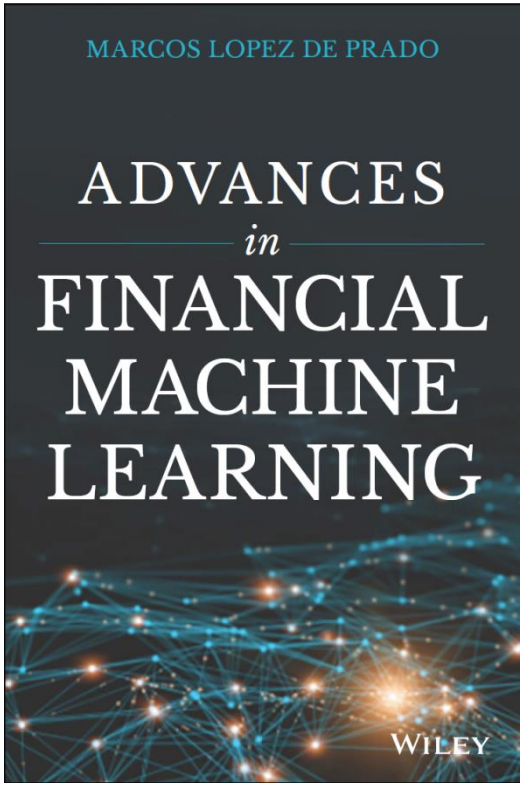


HRP Allocations

- Variance of the out-of-sample portfolio returns: $\sigma^2_{HRP} = .0671$
- 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**



For Additional Details



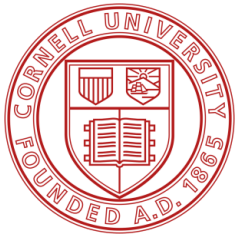
The first wave of quantitative innovation in finance was led by Markowitz optimization. Machine Learning is the second wave and it will touch every aspect of finance. López de Prado's Advances in Financial Machine Learning is essential for readers who want to be ahead of the technology rather than being replaced by it.
— Prof. **Campbell Harvey**, Duke University. Former President of the American Finance Association.

Financial problems require very distinct machine learning solutions. Dr. López de Prado's book is the first one to characterize what makes standard machine learning tools fail when applied to the field of finance, and the first one to provide practical solutions to unique challenges faced by asset managers. Everyone who wants to understand the future of finance should read this book.
— Prof. **Frank Fabozzi**, EDHEC Business School. Editor of The Journal of Portfolio Management.

THANKS FOR YOUR ATTENTION!

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.
-



Machine Learning Asset Allocation

Prof. Marcos López de Prado

Advances in Financial Machine Learning

ORIE 5256

Key Points

- Convex optimization solutions tend to be unstable, to the point of entirely offsetting the benefits of optimization.
  - For example, in the context of financial applications, it is known that portfolios optimized in-sample often underperform the naïve (equal weights) allocation out-of-sample.
- This instability can be traced back to two sources:
  - noise in the input variables
  - signal structure that magnifies the estimation errors in the input variables.
- There is abundant literature discussing noise-induced instability.
- In contrast, signal-induced instability is often ignored or misunderstood.
- **We introduce a new optimization method that is robust to signal-induced instability.**
- For additional details, see the full paper at: https://ssrn.com/abstract=3469961

SECTION I
Problem Statement

The Problem

- Consider a system with N random variables, where the expected value of draws from these variables is represented by an array μ, and the variance of these draws is represented by the covariance matrix V.
- We would like to minimize the variance of the system, measured as $\omega' V \omega$, subject to achieving a target $\omega' a$, where a characterizes the optimal solution.
- The problem can be stated as

$$\min_{\omega} \frac{1}{2} \omega' V \omega$$
$$\text{s.t.: } \omega' a = 1$$

The Solution (1/2)

- This problem can be expressed in lagrangian form as

$$L[\omega, \lambda] = \frac{1}{2}\omega' V \omega - \lambda(\omega' a - 1)$$

with first order conditions

$$\frac{\partial L[\omega, \lambda]}{\partial \omega} = V\omega - \lambda a$$
$$\frac{\partial L[\omega, \lambda]}{\partial \lambda} = \omega' a - 1$$

- Setting the first order (necessary) conditions to zero, we obtain that $V\omega - \lambda a = 0 \Longrightarrow \omega = \lambda V^{-1} a$, and $\omega' a = a'\omega = 1 \Longrightarrow \lambda a' V^{-1} a = 1 \Longrightarrow \lambda = \frac{1}{a' V^{-1} a}$, thus

$$\omega^* = \frac{V^{-1} a}{a' V^{-1} a}$$

The Solution (2/2)

- The second order (sufficient) condition confirms that this solution is the minimum of the lagrangian,

$$\begin{vmatrix} \frac{\partial L^2[\omega,\lambda]}{\partial \omega^2} & \frac{\partial L^2[\omega,\lambda]}{\partial \omega \partial \lambda} \\ \frac{\partial L^2[\omega,\lambda]}{\partial \lambda \partial \omega} & \frac{\partial L^2[\omega,\lambda]}{\partial \lambda^2} \end{vmatrix} = \begin{vmatrix} V' & -a' \\ a & 0 \end{vmatrix} = a'a \geq 0$$

- The issue is, this solution is mathematically correct, but impractical.

Numerical Instability

- The common approach to estimating ω^* is to compute

$$\widehat{\omega}^* = \frac{\hat{V}^{-1}\hat{a}}{\hat{a}'\hat{V}^{-1}\hat{a}}$$

where $\hat{V}$ is the estimated V, and $\hat{a}$ is the estimated a.

- In general, replacing each variable with its estimate will lead to unstable solutions, that is, solutions where a small change in the inputs will cause extreme changes in $\widehat{\omega}^*$.
- This is problematic, because in many practical applications there are material costs associated with the re-allocation from one solution to another.

SECTION II
Noise-induced Instability

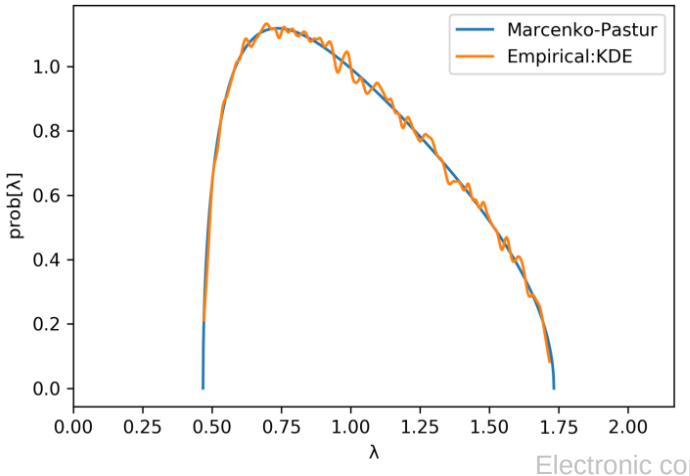
The Marcenko-Pastur Distribution (1/2)

- Consider a matrix of independent and identically distributed random observations X, of size TxN, where the underlying process generating the observations has zero mean and variance σ^2.
- The matrix $C = T^{-1}X'X$ has eigenvalues λ that asymptotically converge (as $N \to +\infty$ and $T \to +\infty$ with $1 < {}^{T}/_{N} < +\infty$) to the Marcenko-Pastur probability density function (PDF),

$$f[\lambda] = \begin{cases} \frac{T}{N}\frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi\lambda\sigma^2} & \text{if } \lambda \in [\lambda_-, \lambda_+] \\ 0 & \text{if } \lambda \notin [\lambda_-, \lambda_+] \end{cases}$$

The Marcenko-Pastur Distribution (2/2)

… where the maximum expected eigenvalue is $\lambda_+ = \sigma^2 \left(1 + \sqrt{\frac{N}{T}}\right)^2$, and the minimum expected eigenvalue is $\lambda_- = \sigma^2 \left(1 - \sqrt{\frac{N}{T}}\right)^2$. When $\sigma^2 = 1$, then C is the correlation matrix associated with X.



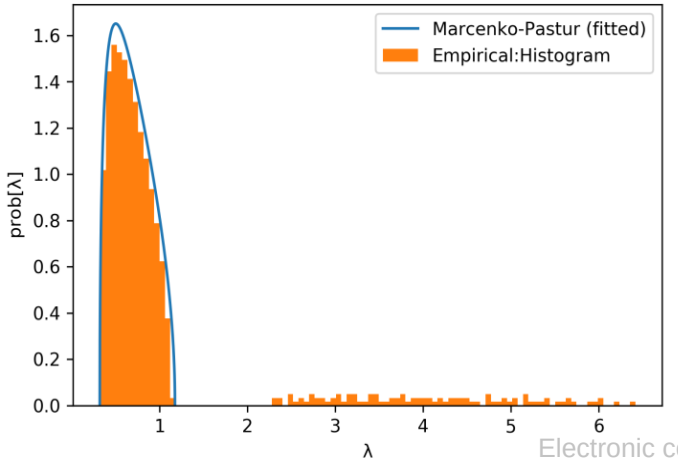
Eigenvalues $\lambda \in [\lambda_-, \lambda_+]$ are consistent with random behavior, and eigenvalues $\lambda \notin [\lambda_-, \lambda_+]$ are consistent with non-random behavior. Specifically, we associate eigenvalues $\lambda \in [0, \lambda_+]$ with noise.

Problem: In empirical covariance matrices, most of the eigenvalues fall under the Marcenko-Pastur distribution, and are insignificant.

The implication is that neither C^{-1} nor V^{-1} can be computed robustly. Solutions are only optimal in-sample, not out-of-sample.

Fitting the Marcenko-Pastur PDF

- Laloux et al. [2005] argue that, since only part of the variance is caused by random eigenvectors, we can adjust σ^2 accordingly in the above equations.
  - For instance, if we assume that the eigenvector associated with the highest eigenvalue is *not* random, then we should replace σ^2 with $\sigma^2\left(1-\frac{\lambda_+}{N}\right)$ in the above equations.
- In fact, we can fit the function $f[\lambda]$ to the empirical distribution of the eigenvalues to derive the implied σ^2.



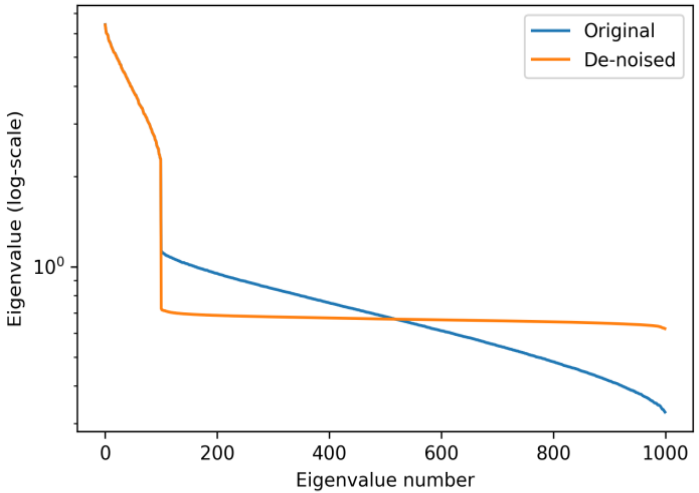
That will give us the variance that is explained by the random eigenvectors present in the correlation matrix, and it will determine the cut-off level λ_+, adjusted for the presence of non-random eigenvectors.

Key point: Because we know what eigenvalues are associated with noise, we can shrink only those, without diluting the signal!

SECTION III
De-Noising and De-Toning

The Constant Residual Eigenvalue Method

- Let $\{\lambda_n\}_{n=1,\ldots,N}$ be the set of all eigenvalues, ordered descending, and i be the position of the eigenvalue such that $\lambda_i > \lambda_+$ and $\lambda_{i+1} \leq \lambda_+$.
- Then we set $\lambda_j = \frac{1}{N-i}\sum_{k=i+1}^{N}\lambda_k, j = i+1, \ldots, N$, hence preserving the trace of the correlation matrix.



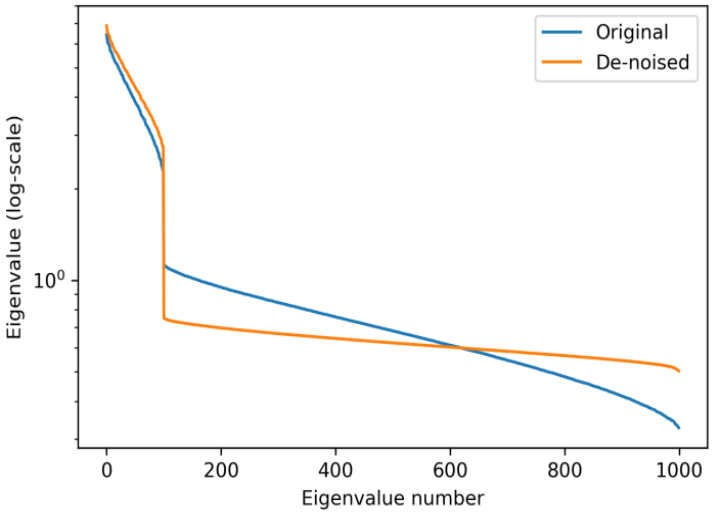
Given the eigenvector decomposition $VW = W\Lambda$, we form the de-noised correlation matrix C_1 as

$$\tilde{C}_1 = W\tilde{\Lambda}W'$$
$$C_1 = \left(\text{diag}[\tilde{C}_1]\right)^{-1/2}\tilde{C}_1\left(\text{diag}[\tilde{C}_1]\right)^{-1/2}$$

where $\tilde{\Lambda}$ is the diagonal matrix holding the corrected eigenvalues. The reason for the second transformation is to re-scale the matrix $\tilde{C}_1$, so that the main diagonal of C_1 is an array of 1s.

The Targeted Shrinkage Method

- The numerical method described earlier is preferable to shrinkage, because it removes the noise while preserving the signal.
- Alternatively, we could target the application of the shrinkage strictly to the random eigenvectors. Consider the correlation matrix C_1



$$C_1 = W_L \Lambda_L W_L' + \alpha W_R \Lambda_R W_R' + (1-\alpha)\text{diag}[W_R \Lambda_R W_R']$$

where W_R and Λ_R are the eigenvectors and eigenvalues associated with $\{n|\lambda_n \leq \lambda_+\}$, W_L and Λ_L are the eigenvectors and eigenvalues associated with $\{n|\lambda_n > \lambda_+\}$, and α regulates the amount of shrinkage among the eigenvectors and eigenvalues associated with noise ($\alpha \to 0$ for total shrinkage).

De-Toning (1/3)

- Financial correlation matrices usually incorporate a market component.
- The market component is characterized by the first eigenvector, with loadings $W_{n,1} \approx N^{-\frac{1}{2}}, n = 1, \ldots, N$.
- Accordingly, a market component affects every item of the covariance matrix.
- By removing the market component, we allow a greater portion of the correlation to be explained by components that affect specific subsets of the securities.
- Intuition: De-toning is similar to removing a loud tone that prevents us from hearing other sounds.

De-Toning (2/3)

- We can remove the market component from the de-noised correlation matrix, C_1, to form the de-toned correlation matrix,

$$\tilde{C}_2 = C_1 - W_M \Lambda_M W_M' = W_D \Lambda_D W_D'$$
$$C_2 = \left(\text{diag}\left[\tilde{C}_2\right]\right)^{-1/2} \tilde{C}_2 \left(\text{diag}\left[\tilde{C}_2\right]\right)^{-1/2}$$

where W_M and Λ_M are the eigenvectors and eigenvalues associated with market components (usually only one, but possibly more), and W_D and Λ_D are the eigenvectors and eigenvalues associated with non-market components.

De-Toning (3/3)

- The de-toned correlation matrix is singular, as a result of eliminating (at least) one eigenvector.
  - This is not a problem for clustering applications, as most approaches do not require the invertibility of the correlation matrix.
- Still, a de-toned correlation matrix C_2 cannot be used directly for mean-variance portfolio optimization.
- Instead, we can optimize a portfolio on the selected (non-zero) principal components, and map the optimal allocations f^* back to the original basis.
- The optimal allocations in the original basis are

$$\omega^* = W_+ f^*$$

where W_+ contains only the eigenvectors that survived the de-toning process (i.e., with a non-null eigenvalue), and f^* is the vector of optimal allocations to those same components.

Experimental Results (1/2)

- We generate a vector of means and a covariance matrix out of 10 blocks of size 50 each, where off-diagonal elements within each block have a correlation of 0.5.
  - This covariance matrix is a stylized representation of a "true" (non-empirical) de-toned correlation matrix of the S&P 500, where each block is associated with an economic sector.
  - Without loss of generality, the variances are drawn from a uniform distribution bounded between 5% and 20%, and the vector of means is drawn from a Normal distribution with mean and standard deviation equal to the standard deviation from the covariance matrix.
  - This is consistent with the notion that in an efficient market all securities have the same expected Sharpe ratio.
- We use this means vector and covariance matrix to draw 1,000 random matrices X of size $TxN = 1000x500$, compute the associated empirical covariance matrices and vectors of means, and evaluate the (empirical) optimal portfolios.
- We compute the root-mean-square error (RMSE) between the "empirical" and "true" optimal portfolios.

Experimental Results (2/2)

	Not De-Noised	De-Noised
Not Shrunk	4.95E-03	1.99E-03
Shrunk	3.45E-03	1.70E-03

Minimum Variance Portfolio

De-noising is much more effective than shrinkage: the de-noised minimum variance portfolio incurs only 40.15% of the RMSE incurred by the minimum variance portfolio without de-noising. That is a 59.85% reduction in RMSE from de-noising alone, compared to a 30.22% reduction using Ledoit-Wolf shrinkage. Shrinkage adds little benefit beyond what de-noising contributes. The reduction in RMSE from combining de-noising with shrinkage is 65.63%, which is not much better than the result from using de-noising only.

	Not De-Noised	De-Noised
Not Shrunk	9.48E-01	5.27E-02
Shrunk	2.77E-01	5.17E-02

Maximum Sharpe Ratio Portfolio

The de-noised maximum Sharpe ratio portfolio incurs only 0.04% of the RMSE incurred by the maximum Sharpe ratio portfolio without de-noising. That is a 94.44% reduction in RMSE from de-noising alone, compared to a 70.77% reduction using Ledoit-Wolf shrinkage. While shrinkage is somewhat helpful in absence of de-noising, it adds no benefit in combination with de-noising. This is because shrinkage dilutes the noise at the expense of diluting some of the signal as well.

SECTION IV
Signal-induced Instability

The Condition Number (1/2)

- Certain covariance structures can make the mean-variance optimization solution unstable.
- Consider a correlation matrix between two securities,

$$C = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

where ρ is the correlation between their returns.

- Matrix C can be diagonalized as $CW = W\Lambda$ as follows, where

$$W = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, \Lambda = \begin{bmatrix} 1+\rho & 0 \\ 0 & 1-\rho \end{bmatrix}$$

The Condition Number (2/2)

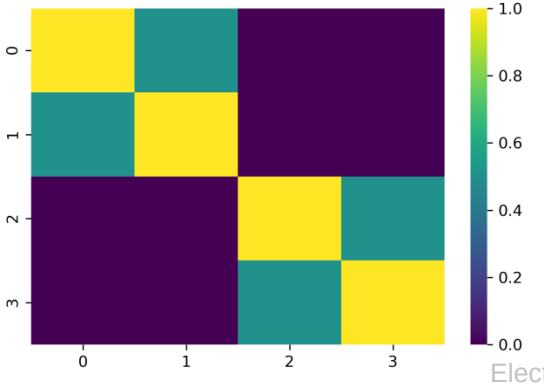
- The trace of C is $tr(C) = \Lambda_{1,1} + \Lambda_{2,2} = 2$, so ρ sets how big one eigenvalue gets at the expense of the other.
- The determinant of C is given by $|C| = \Lambda_{1,1}\Lambda_{2,2} = (1+\rho)(1-\rho) = 1-\rho^2$.
  - The determinant reaches its maximum at $\Lambda_{1,1} = \Lambda_{2,2} = 1$, which corresponds to the uncorrelated case, $\rho = 0$.
  - The determinant reaches its minimum at $\Lambda_{1,1} = 0$ or $\Lambda_{2,2} = 0$, which corresponds to the perfectly correlated case, $|\rho| = 1$.
- The inverse of C is $C^{-1} = W\Lambda^{-1}W' = \frac{1}{|C|}\begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix}$.
- The implication is that, the more ρ deviates from zero, the bigger one eigenvalue becomes relative to the other, causing $|C|$ to approach zero, which makes the values of C^{-1} explode. ***This happens regardless of the N/T ratio***.

Markowitz's Curse

- Matrix C is just a standardized version of V, and the conclusions we drew on C^{-1} apply to the V^{-1} used to compute ω^*.
- When securities within a portfolio are highly correlated ($-1 < \rho \ll 0$ or $0 \ll \rho < 1$), C has a high condition number, and the values of V^{-1} explode.
- This is problematic in the context of portfolio optimization, because ω^* depends on V^{-1}, and unless $\rho \approx 0$, we must expect an unstable solution to the convex optimization program.
- In other words, Markowitz's solution is guaranteed to be numerically stable only if $\rho \approx 0$, which is precisely the case when we don't need it!
- The reason we needed Markowitz was to handle the $\rho \not\approx 0$ case, but the more we need Markowitz, the more numerically unstable is its estimation of ω^*.

Signal-induced Instability in Finance

- When a subset of securities exhibits greater correlation among themselves than to the rest of the investment universe, that subset forms a cluster within the correlation matrix.
- Clusters appear naturally, as a consequence of hierarchical relationships.
- When K securities form a cluster, they are more heavily exposed to a common eigenvector, which implies that the associated eigenvalue explains a greater amount of variance.



But because the trace of the correlation matrix is exactly N, that means that an eigenvalue can only increase at the expense of the other $N - K$ eigenvalues, resulting in a condition number greater than 1.

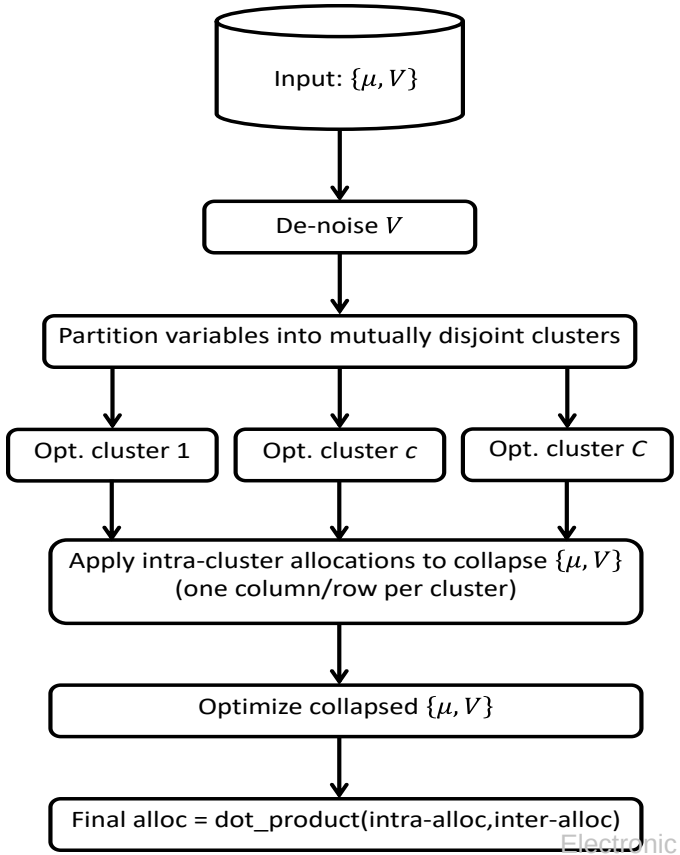
Accordingly, the greater the intra-cluster correlation is, the higher the condition number becomes.

SECTION V
The NCO Algorithm

NCO's Structure (1/2)

- The Nested Clustered Optimization (NCO) algorithm is composed of five steps:
  1. **Correlation Clustering**: Find the optimal number of clusters.
     - One possibility is to apply the ONC algorithm, however NCO is agnostic as to what particular algorithm is used for determining the number of clusters.
     - For large matrices, where T/N is relatively low, it is advisable to de-noise the correlation matrix prior to clustering, following the method described in Section III.
  2. **Intra-Cluster Weights**: Compute optimal intra-cluster allocations, one optimal allocation per cluster, using the de-noised covariance matrix. This operation can be parallelized.
  3. **System Reduction**: Use the intra-cluster weights to reduce the system (one row/column per cluster).
  4. **Inter-Cluster Weights**: Compute optimal inter-cluster allocations, using the reduced covariance matrix.
  5. **Dot Product**: The final allocation per security results from multiplying intra-cluster weights with the inter-cluster weights.

NCO's Structure (2/2)



Why does NCO beat Markowitz?

By construction, the reduced covariance matrix is close to a diagonal matrix, and the optimization problem is close to the ideal Markowitz case.

In other words, the clustering and intra-cluster optimization steps have allowed us to transform a "Markowitz-cursed" problem ($|\rho| \gg 0$) into a well-behaved problem ($\rho \approx 0$).

Experimental Results (1/2)

- We generate a vector of means and a covariance matrix out of 10 blocks of size 5 each, where off-diagonal elements within each block have a correlation of 0.5.
  - This covariance matrix is a stylized representation of a "true" (non-empirical) de-toned correlation matrix of the S&P 500, where each block is associated with an economic sector.
  - Without loss of generality, the variances are drawn from a uniform distribution bounded between 5% and 20%, and the vector of means is drawn from a Normal distribution with mean and standard deviation equal to the standard deviation from the covariance matrix.
  - This is consistent with the notion that in an efficient market all securities have the same expected Sharpe ratio.
- We use this means vector and covariance matrix to draw 1,000 random matrices X of size $TxN = 1000x50$, compute the associated empirical covariance matrices and vectors of means, and evaluate the (empirical) optimal portfolios.
- We compute the root-mean-square error (RMSE) between the "empirical" and "true" optimal portfolios.

Experimental Results (2/2)

	Markowitz	NCO
Raw	7.95E-03	4.21E-03
Shrunk	8.89E-03	6.74E-03

Minimum Variance Portfolio

NCO computes the minimum variance portfolio with 52.98% of Markowitz's RMSE, i.e. a 47.02% reduction in RMSE. While Ledoit-Wolf shrinkage helps reduce the RMSE, that reduction is relatively small, around 11.81%. Combining shrinkage and NCO yields a 15.30% reduction in RMSE, which is better than shrinkage but worse than NCO alone. The implication is that NCO delivers substantially lower RMSE than Markowitz's solution, even for a small portfolio of only 50 securities, and that shrinkage adds no value.

	Markowitz	NCO
Raw	7.02E-02	3.17E-02
Shrunk	6.54E-02	5.72E-02

Maximum Sharpe Ratio Portfolio

NCO computes the maximum Sharpe ratio portfolio with 45.17% of Markowitz's RMSE, i.e. a 54.83% reduction in RMSE. The combination of shrinkage and NCO yields a 18.52% reduction in the RMSE of the maximum Sharpe ratio portfolio, which is better than shrinkage but worse than NCO. Once again, NCO delivers substantially lower RMSE than Markowitz's solution, and shrinkage adds no value. It is easy to test that NCO's advantage widens for larger portfolios.

SECTION VI
Robustness Analysis via Monte Carlo

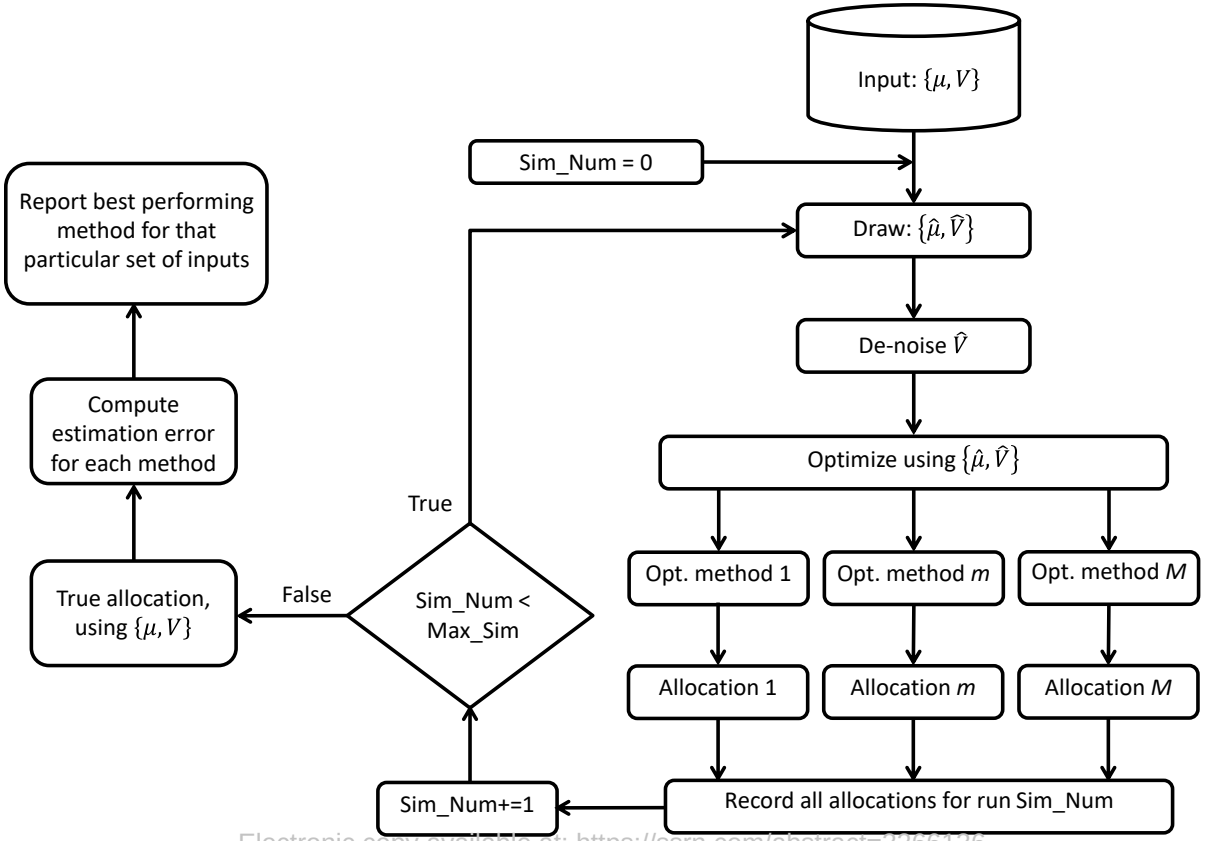
Monte Carlo Optimization Selection (MCOS)

- There is no reason to believe that one particular optimization method is the most robust under all conditions.
- The interactions between noise and signal-induced instabilities make it hard to determine *a priori* what is the most robust optimization approach for a particular problem.
- Thus, rather than relying always on one particular approach, researchers should apply opportunistically whatever optimization method is best suited to a particular setting.
- We introduce a Monte Carlo approach that derives the estimation error produced by various optimization methods on a particular set of input variables.
- The goal is to determine what method is most robust to a particular case.

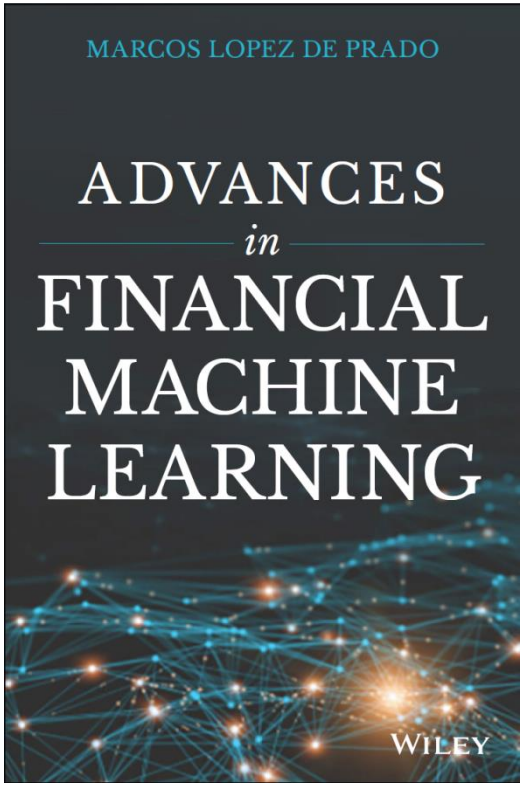
MCOS' Structure (1/2)

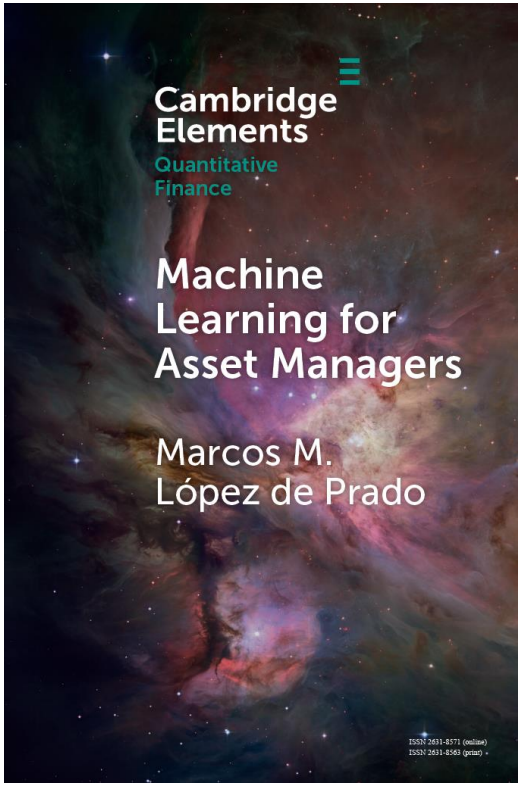
1. **Compute input set**: Get the input variables that characterize the problem, $\{\mu, V\}$.
2. **Random draw**: Draws T observations from the data-generating process characterized by $\{\mu, V\}$, and derive from those observations the pair $\{\hat{\mu}, \hat{V}\}$.
3. **Optimal allocations**: Compute optimal allocations for $\{\hat{\mu}, \hat{V}\}$, applying M alternative methods, resulting in M alternative allocations.
4. **Storage**: Record the M allocations associated with this run of the MCOS method.
5. **Loop**: Repeat steps 1-4 a user-defined number of times.
6. **Benchmark**: Compute the true allocation derived from $\{\mu, V\}$.
7. **Estimation error**: Compute the estimation error associated with each of the M alternative methods.
8. **Report**: Report the method that yields the most robust allocation for the particular set of inputs $\{\mu, V\}$.

MCOS' Structure (2/2)



For Additional Details





The first wave of quantitative innovation in finance was led by Markowitz optimization. Machine Learning is the second wave and it will touch every aspect of finance. López de Prado's Advances in Financial Machine Learning is essential for readers who want to be ahead of the technology rather than being replaced by it.
— Prof. **Campbell Harvey**, Duke University. Former President of the American Finance Association.

Financial problems require very distinct machine learning solutions. Dr. López de Prado's book is the first one to characterize what makes standard machine learning tools fail when applied to the field of finance, and the first one to provide practical solutions to unique challenges faced by asset managers. Everyone who wants to understand the future of finance should read this book.
— Prof. **Frank Fabozzi**, EDHEC Business School. Editor of The Journal of Portfolio Management.

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.
-

www.QuantResearch.org