High-dimensional portfolio optimization using wavelet thresholding

September 22, 2011

Mean-variance portfolio optimization framework

Mean-variance portfolio optimization framework

- We consider a vector of returns R of dimension $(N \times 1)$ with $R \sim MN(\mu, \Omega)$
- The portfolio return is then defined as $x_1 R_1 + x_2 R_2 + \ldots + x_N R_N$
- Thus, the portfolio return follows a $N(x^t \mu, x^t \Omega x)$

Optimal weight vector

- The optimal weight vector x^* minimizes the variance of the portfolio return for a given value of the portfolio mean return
- x^* is found on the basis of a quadratic programming problem [Markowitz, 1952]

$$\begin{gathered} min \;\; (x^t \Omega x) \\ x^t \mu = r \\ x^t e = 1 \end{gathered}$$

- Use the Lagrange multiplier method to obtain the optimal weights vector [Devolder, 1996]

$$x^* = \frac{c - rb}{ac - b^2} \Omega^{-1} e + \frac{ar - b}{ac - b^2} \Omega^{-1} \mu$$

where

$$\begin{gathered} a = e^t \Omega^{-1} e \\ b = e^t \Omega^{-1} \mu \\ c = \mu^t \Omega^{-1} \mu \end{gathered}$$

Mean-variance portfolio optimization framework

Input needs to be estimated

- Ω and μ are unknown and hence need to be replaced by estimates $\left(\hat{\Omega}, \hat{\mu}\right)$
- Use $\hat{\Omega}$ and $\hat{\mu}$ as inputs in the optimization framework
- This leads to an estimate $\hat{x}^*$ of the optimal weights vector

High-dimensional settings and overfitting

- Natural choices for $\hat{\mu}$ and $\hat{\Sigma}$:

  sample mean return vector and sample covariance matrix

- In high-dimensional settings the portfolio size N is large relative to the sample size T

  $\Rightarrow$ Sparsity of the data points
  $\Rightarrow$ Samples are not representative
  $\Rightarrow$ Sample covariance matrix is not a good estimate

- Leads to a $\hat{x}^*$ with poor out-of-sample performance [El Karoui, N. 2011]

Regularization methods

- Build a regularized version of the sample covariance matrix

  $\Rightarrow$ Avoid the problem of overfitting
  $\Rightarrow$ Obtain $\hat{x}^*$ with a good out-of-sample performance

[Johannes, J., Van Bellegem, S. and Vanhems, A., 2010]

Regularization using wavelet thresholding

Regularization using wavelet thresholding

- The input data of the optimization framework is projected on a matrix ψ containing the $N-1$ detail vectors of an unbalanced Haar basis
- The quadratic programming problem is written as follows

$$\min\left(\hat{c}^t\left(\psi^t\hat{\Omega}\psi\right)\hat{c}\right)$$
$$\hat{c}^t\left(\psi^t\hat{\mu}\right)=r-\left(\tfrac{1}{N}\sum_{i=1}^{N}\hat{\mu}_i\right)$$

- Obtain $\hat{x}^*$ using

$$\hat{x}^*=\frac{1}{N}+\psi\hat{c}^*$$

Detail vectors of an unbalanced Haar basis

- The jth detail vector is defined as follows [Fryzlewicz, 2007]

$$\psi_j(l)=\left\{\frac{1}{BP_j-SP_j+1}-\frac{1}{EP_j-SP_j+1}\right\}I\left[SP_j\leq l\leq BP_j\right]-\left\{\frac{1}{EP_j-BP_j}-\frac{1}{EP_j-SP_j+1}\right\}I\left[BP_j+1\leq l\leq EP_j\right]$$
$$j=1,\ldots,N-1.$$

- SP is the starting point, BP is the breaking point and EP is the end point of the detail vector

$$[SP,EP]=\text{Support of the detail vector}$$
$$BP\in[SP,EP-1]$$

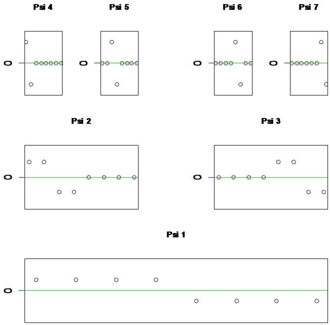
Regularization of the projected sample covariance matrix

- Rewrite $\psi^t\hat{\Omega}\psi$ as $\left(\psi^t\hat{D}\psi\right)\left(\psi^t\hat{\Sigma}\psi\right)\left(\psi^t\hat{D}\psi\right)$

  where $\hat{\Sigma}$ is the sample correlation matrix and $\hat{D}$ is a diagonal matrix containing the sample standard deviations
- We will regularize $\psi^t\hat{\Omega}\psi$ by producing a regularized version of $\psi^t\hat{\Sigma}\psi$

Example: $N = 8$

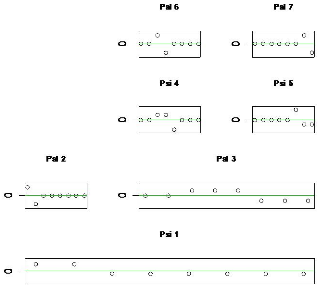
ψ matrix: One possibility



l	$\psi_1(l)$	$\psi_2(l)$	$\psi_3(l)$	$\psi_4(l)$	$\psi_5(l)$	$\psi_6(l)$	$\psi_7(l)$
1	0.35	0.5	0	0.71	0	0	0
2	0.35	0.5	0	-0.71	0	0	0
3	0.35	-0.5	0	0	0.71	0	0
4	0.35	-0.5	0	0	-0.71	0	0
5	-0.35	0	0.5	0	0	0.71	0
6	-0.35	0	0.5	0	0	-0.71	0
7	-0.35	0	-0.5	0	0	0	0.71
8	-0.35	0	-0.5	0	0	0	-0.71

Example: $N = 8$

ψ matrix: Another possibility



l	$\psi_1(l)$	$\psi_2(l)$	$\psi_3(l)$	$\psi_4(l)$	$\psi 5(l)$	$\psi_6(l)$	$\psi_7(l)$
1	0.61	0.71	0	0	0	0	0
2	0.61	-0.71	0	0	0	0	0
3	-0.20	0	0.41	0.41	0	0.71	0
4	-0.20	0	0.41	0.41	0	-0.71	0
5	-0.20	0	0.41	-0.82	0	0	0
6	-0.20	0	-0.41	0	0.82	0	0
7	-0.20	0	-0.41	0	-0.41	0	0.71
8	-0.20	0	-0.41	0	-0.41	0	-0.71

Regularization using wavelet thresholding

Basic idea of non-linear hard thresholding

- The relevant information within $\psi^t \hat{\Sigma} \psi$ is concentrated in a few coefficients
- The large majority of the wavelet coefficients contain nothing but pure noise
- Detect these non-informative coefficients and set them to 0

Objective

- Reduce the number of estimated parameters within $\psi \hat{\Sigma} \psi$
- Obtain an estimate of c^* with a good out-of-sample performance

Drawbacks

- Defines thresholds on coefficients
- Deals with each coefficient individually
- Does not take into account the structure which exists among the wavelet coefficients [Freyermuth, J-M., Ombao, H. and Von Sachs, R., 2010]
- Regularized covariance matrix may not be semi-definite positive

A priori information and constrained thresholding

A priori information and constrained thresholding

- Overcome these drawbacks by giving up full non-linearity
- Add a priori information to the model setting
- The thresholding algorithm is then submitted to constraints derived from this a priori information

A priori information

- There exists a hierarchical structure between the wavelet coefficients
- Wavelet coefficients in $\psi^t \hat{\Sigma} \psi$ form a network of $(N-1)$ fully-rooted trees
- Fully rooted-tree: A tree which only contains complete branches from the terminal nodes up to the root (no holes)

  [Freyermuth, J-M., Ombao, H. and Von Sachs, R., 2010]
  [Autin, F., Freyermuth, J-M. and Von Sachs, R., 2011]

Constraint

- The thresholding algorithm must produce a network of $(N-1)$ fully-rooted

Advantages

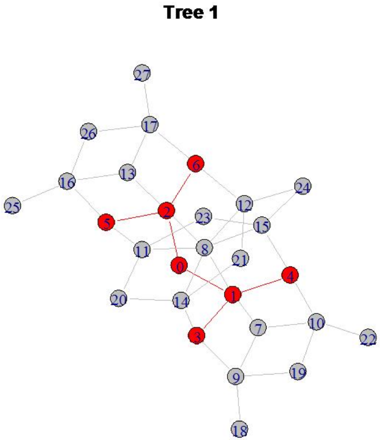
- Coherent form of thresholding, links between the wavelet coefficients are taken into account
- $\psi^t \hat{\Sigma} \psi$ is always semi-definite positive

Example $N = 8$

Tree 1

Vertices of Tree 1:

$$\psi_1^t \hat{\Sigma} \psi_j$$
$$j = 1, \ldots, 7$$

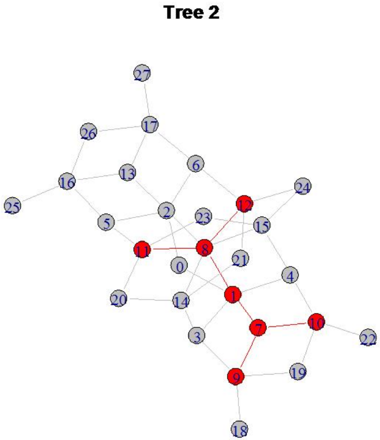


Example $N = 8$

Tree 2

Vertices of Tree 2:

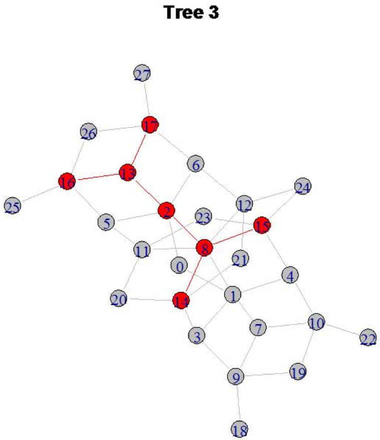
$$\psi_2^t \hat{\Sigma} \psi_j$$
$$j = 1, \ldots, 7$$



Example $N = 8$
Tree 3

Vertices of Tree 3:

$$\psi_3^t \hat{\Sigma} \psi_j$$
$$j = 1, \ldots, 7$$

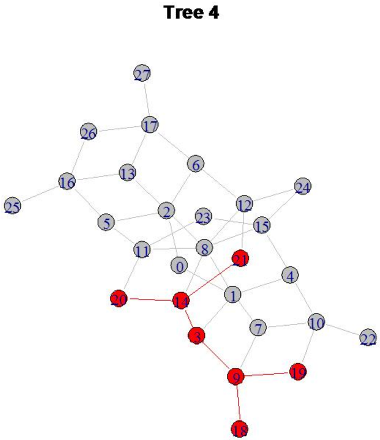


Example $N = 8$

Tree 4

Vertices of Tree 4:

$$\psi_4^t \hat{\Sigma} \psi_j$$
$$j = 1, \ldots, 7$$

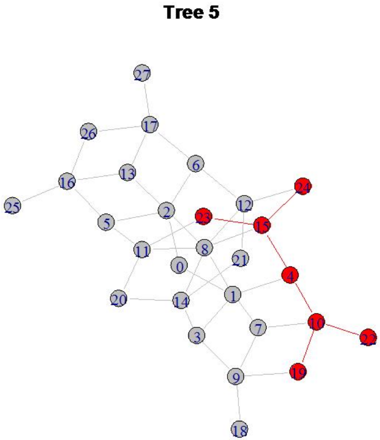


Example $N = 8$

Tree 5

Vertices of Tree 5:

$$\psi_5^t \hat{\Sigma} \psi_j$$
$$j = 1, \ldots, 7$$

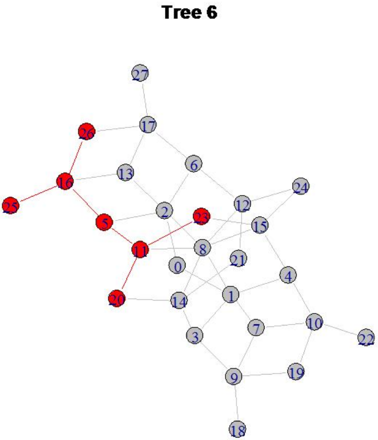


Example $N = 8$

Tree 6

Vertices of Tree 6:

$$\psi_6^t \hat{\Sigma} \psi_j$$
$$j = 1, \ldots, 7$$

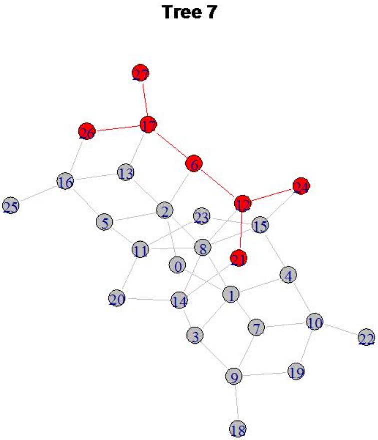


Example $N = 8$

Tree 7

Vertices of Tree 7:

$$\psi_7^t \hat{\Sigma} \psi_j$$
$$j = 1, \ldots, 7$$



Constrained thresholding algorithm

Constrained thresholding algorithm

1. Define $(N-1)$ nested networks of fully-rooted trees

$$
\begin{array}{ccccccccc}
 & & & & \mathit{Network}1 & & & & \\
 & & \swarrow & & & & \searrow & & \\
 & & \mathit{Network}2 & & & & \mathit{Network}3 & & \\
 & \swarrow & & \searrow & & \swarrow & & \searrow & \\
 & \mathit{Network}4 & & \mathit{Network}5 & & \mathit{Network}6 & & \mathit{Network}7 & \\
\swarrow & \downarrow & \swarrow & \downarrow & & \downarrow & \searrow & \downarrow & \searrow \\
\ldots & \ldots & \ldots & \ldots & & \ldots & \ldots & \ldots & \ldots
\end{array}
$$

Network 1
$\psi_i^t \hat{\Sigma} \psi_j$
$i =$ Wavelet 1 and daughter wavelets
$j = 1, \ldots, N-1$

Network 2
$\psi_i^t \hat{\Sigma} \psi_j$
$i =$ Wavelet 2 and daughter wavelets
$j = 1, \ldots, N-1$

Network k
$\psi_i^t \hat{\Sigma} \psi_j$
$i =$ Wavelet k and daughter wavelets
$j = 1, \ldots, N-1$

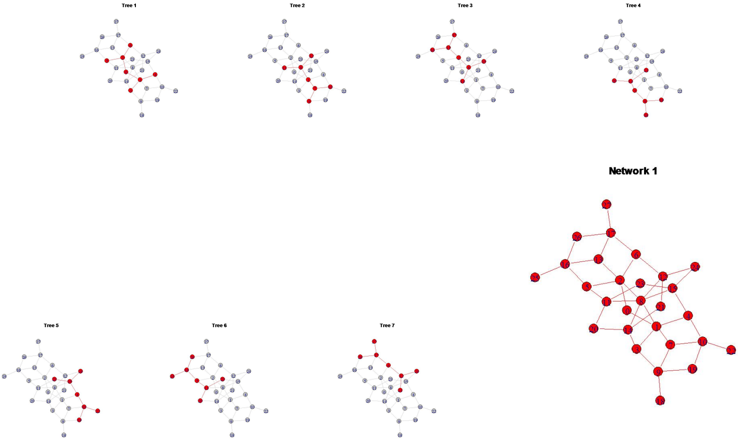
Network N-1
$\psi_i^t \hat{\Sigma} \psi_j$
$i =$ Wavelet N-1
$j = 1, \ldots, N-1$

- Design a thresholding rule for each one of these $(N-1)$ networks
- Thresholding is done iteratively

Iteration 1 $\rightarrow$ Network 1
Iteration 2 $\rightarrow$ Network 2
...
Iteration N-1 $\rightarrow$ Network N-1

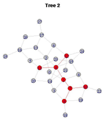
Example $N = 8$

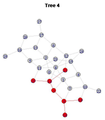
Network 1

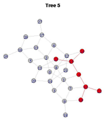


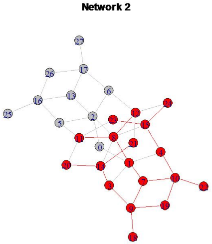
Example $N = 8$

Network 2



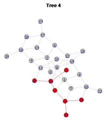


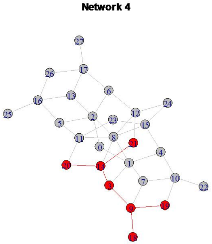




Example $N = 8$

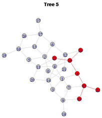
Network 4

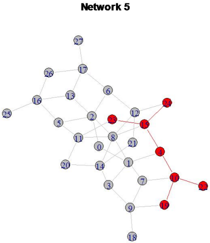




Example $N = 8$

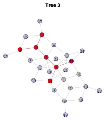
Network 5

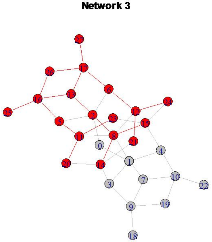


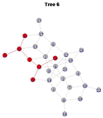


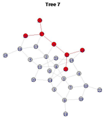
Example $N = 8$

Network 3



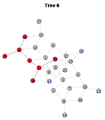


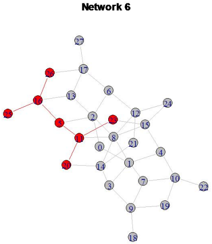




Example $N = 8$

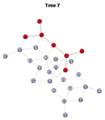
Network 6

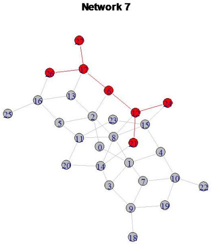




Example $N = 8$

Network 7





Constrained thresholding algorithm

2. Thresholding rule for Network k

- In an iterative scheme the thresholding rule contains of three elements
  1. Decision rule to detect non-informative wavelet coefficients
  2. Action on the wavelet coefficients according to the decision rule
  3. Impact on further iteration steps according to the decision rule

- Group-structured thresholding approach
  1. Decision rule:

$$D \begin{array}{l} \nearrow D_A: \text{ coefficients are all non-informative} \\ \searrow D_B: \text{ coefficients are not all non-informative} \end{array}$$

  2. Action:

$$D \begin{array}{l} \nearrow D_A \Rightarrow \text{put the coefficients to } 0 \\ \searrow D_B \Rightarrow \text{keep the values of the coefficients} \end{array}$$

  3. Further iteration steps:

$$D \begin{array}{l} \nearrow D_A \Rightarrow \text{stop the iterative scheme for the networks nested in Network } k \\ \searrow D_B \Rightarrow \text{pursue the iterative scheme on the networks nested in Network } k \end{array}$$

Constrained thresholding algorithm

3. Incorporate the decision rule into a statistical testing framework

- The group-structured thresholding approach allows to define the decision rule as part of a statistical testing framework
- Empirical coefficient is non-informative $\Leftrightarrow$ Theoretical counterpart is equal to 0

Hypotheses

$$\begin{gathered} H_0 : \psi_i^t \Sigma \psi_j = 0 \\ H_A : \exists (i,j) | \psi_i^t \Sigma \psi_j \neq 0 \\ i = \text{Wavelet } k \text{ and daughter wavelets} \\ j = 1, \dots, N-1 \\ j \neq i \end{gathered}$$

Decision rule

$$D \begin{array}{l} \nearrow \bar{R}H_0 : \text{ coefficients are all equal to 0} \\ \searrow RH_0 : \text{ coefficients are not all equal to 0} \end{array}$$

Properties of the algorithm

- If a specific vertex of the j-th tree belongs to Network k, all the vertices which belong to a lower level of the j-th tree also belong to Network k ($\forall j, k$)
- At each iteration we either set all the coefficients to 0 or maintain their initial values

$\Rightarrow$ The algorithm always produces a network of $(N-1)$ fully-rooted trees

Two-sided Kolmogorov-Smirnov testing for Network k

Two-sided Kolmogorov-Smirnov testing

Reexpress the hypotheses of the testing framework

- Let us assume that the support of the k-th detail vector is $r, \ldots, r+(s-1)$
- We isolate the set of s neighboring rows $\hat{V}_r, \ldots, \hat{V}_{r+(s-1)}$ from $\hat{\Sigma}$
- $\hat{D}_{lp}$ is defined as the Euclidean norm between the vectors $\hat{V}_l$ and $\hat{V}_p$

$$\hat{D}_{lp} = \sqrt{\sum_{k=1}^{N} \left(\hat{V}_{kl} - \hat{V}_{kp}\right)^2}, \ k \neq l, p$$

- Note that

$$\begin{aligned} &\psi_i^t \Sigma \psi_j = 0 \\ &i = \text{Wavelet k and daughter wavelets} \\ &j = 1, \ldots, N-1 \\ &j \neq i \\ \\ &\Leftrightarrow D_{lp} = 0 \\ &l = r, \ldots, r+(s-1) \\ &p > l \\ \\ &\Leftrightarrow F_{\hat{D}_{lp}} \text{ are identical} \\ &l = r, \ldots, r+(s-1) \\ &p > l \end{aligned}$$

Two-sided Kolmogorov-Smirnov testing

- Thus, we can reexpress the hypotheses as

$$\begin{aligned} &H_0 : F_{\hat{D}_{lp}} \text{ are identical} \\ &H_A : F_{\hat{D}_{lp}} \text{ are not identical} \\ &l = r, \ldots, r+(s-1) \\ &p > l \end{aligned}$$

<u>Testing procedure</u>

- Divide $\hat{V}_r, \ldots, \hat{V}_{r+(s-1)}$ into two subsets

$$\begin{gathered} \hat{V}_r, \ldots, \hat{V}_{r+(q-1)} \\ \hat{V}_{r+q}, \ldots, \hat{V}_{r+(s-1)} \end{gathered}$$

- We then define the Euclidean distances

$$\hat{D}^{(A)}_{lp} = \sqrt{\Sigma_{k=1}^{N}\left(\hat{V}_{kl} - \hat{V}_{kp}\right)^2}, \; k \neq l, p \quad \substack{l = r, \ldots, r+(q-1) \\ p > l} \qquad \hat{D}^{(B)}_{lp} = \sqrt{\Sigma_{k=1}^{N}\left(\hat{V}_{kl} - \hat{V}_{kp}\right)^2}, \; k \neq l, p \quad \substack{l = r+q, \ldots, r+(s-1) \\ p > l} \qquad \hat{D}^{(C)}_{lp} = \sqrt{\Sigma_{k=1}^{N}\left(\hat{V}_{kl} - \hat{V}_{kp}\right)^2}, \; k \neq l, p \quad \substack{l = r, \ldots, r+(q-1) \\ p = r+q, \ldots, r+(s-1)}$$

- Create the two vectors $\hat{D}^{(1)}$ and $\hat{D}^{(2)}$ as follows

$$\hat{D}^{(1)} = \left(\hat{D}^{(A)}, \hat{D}^{(C)}\right); \; \hat{D}^{(2)} = \hat{D}^{(C)}$$

- Reexpress the hypotheses

$$\begin{aligned} &H_0 : F_{\hat{D}^{(1)}} = F_{\hat{D}^{(2)}} \\ &H_A : F_{\hat{D}^{(1)}} \neq F_{\hat{D}^{(2)}} \end{aligned}$$

- Use $\hat{D}^{(1)}_{obs}$ and $\hat{D}^{(2)}_{obs}$ to get $\hat{F}^{\hat{D}^{(1)}}$ and $\hat{F}_{\hat{D}^{(2)}}$ and apply a two-sided Kolmogorov-testing scheme

Bootstrap approach for Network k

Bootstrap approach

Reexpress the hypotheses of the testing framework

- We isolate the set of s neighboring rows $\hat{V}_r, \ldots, \hat{V}_{r+(s-1)}$ from $\hat{\Sigma}$
- $\hat{D}$ is then defined as the vector of Euclidean distances of the path $r, \ldots, r+(s-1)$ between $\hat{V}_r, \ldots, \hat{V}_{r+(s-1)}$
- Note that

$$\begin{aligned} &\psi_i^t \Sigma \psi_j = 0 \\ &i = \text{Wavelet k and daughter wavelets} \\ &j = 1, \ldots, N-1 \\ &j \neq i \\ \\ &\Leftrightarrow D = (0, \ldots, 0) \\ &\Leftrightarrow D_l = 0 \end{aligned}$$

where D_l is the largest component of D

- Thus, we can reexpress the hypotheses as

$$\begin{aligned} H_0 &: D_l = 0 \\ H_A &: D_l \neq 0 \end{aligned}$$

Testing procedure

- We denote as $F_{\hat{D}_l}^{(H_0)}$ the distribution of $\hat{D}_l$ under H_0
- $F_{\hat{D}_l}^{(H_0)}$ is then approximated by $\tilde{F}_{\hat{D}_l}^{(H_0)}$ using bootstrap
- The decision rule takes the form

$$\begin{aligned} \hat{D}_l^{(obs)} \leq \tilde{F}_{\hat{D}_l}^{(H_0)^{-1}}(p) \quad &\Rightarrow \quad \bar{R}H_0 \\ \hat{D}_l^{(obs)} > \tilde{F}_{\hat{D}_l}^{(H_0)^{-1}}(p) \quad &\Rightarrow \quad RH_0 \end{aligned}$$

Bootstrap approach

Fast algorithm

- Define $\hat{V}_1^{(obs)},\ldots,\hat{V}_N^{(obs)}$ as the set of all the observed neighboring rows of $\hat{\Sigma}$
- $\hat{D}^{(obs)}$ is then defined as the vector of Euclidean distances of the path $1,\ldots,N$
- Regroup the neighboring vectors 1 by 1 on the basis of the smallest components of $\hat{D}^{(obs)}$
- Let us assume that vector i and j have been regrouped
- We compute

$$\hat{V}_k^{(ij)} = \frac{\hat{V}_{ki}^{(obs)}+\hat{V}_{kj}^{(obs)}}{2}$$
$$k=1,\ldots,N$$
$$k \neq i,j$$

- Put

$$\tilde{V}_i = \hat{V}^{(ij)}$$
$$\tilde{V}_j = \hat{V}^{(ij)}$$

- We do this for all regrouped vectors
- We then denote as $\tilde{\Sigma}$ the matrix which contains the $\tilde{V}$ vectors as rows
- Simulate K scenarios from $\tilde{\Sigma}$ and compute the K realizations of the Euclidean distances between $\tilde{V}_i$ and $\tilde{V}_j$
- These K realizations are then used to obtain $\tilde{F}_{\hat{D}_{ij}}^{(H_0)}$
- If $l=i$ or $l=j$ we set $\tilde{F}_{\hat{D}_{ij}}^{(H_0)}\tilde{F}_{\hat{D}_l}^{(H_0)} = \tilde{F}_{\hat{D}_{ij}}^{(H_0)}$

Improve the performance of the regularization method

Improve the performance of the regularization method

- Note that the regularization of the sample correlation matrix is only possible if for some iterations $\bar{R}H_0$
- Recall that

$$\begin{array}{l} \text{KS testing} \rightarrow H_0 : D_{ij} = 0;\ i = r, \ldots, r + (l-1);\ j > i \\ \text{Bootstrap} \rightarrow H_0 : D = (0, \ldots, 0) \end{array}$$

- The performance of the regularization method depends on the existence of groups of identical rows in Σ
- Two-step procedure to make these groups appear

1. Reorder the assets

- Objective: Find a permutation vector P which is used to reorder the assets in the optimization framework
- Define the set of neighboring vectors

$$\hat{V}_1^{(obs)}, \ldots, \hat{V}_N^{(obs)}$$

- Denote as $\hat{D}^{(obs)}$ the vector containing the Euclidean distances between all pairs of vectors
- Step 1
  1. Find the smallest component $\hat{D}_{ij}^{(obs)}$ of $\hat{D}^{(obs)}$
  2. Put $P_1 = i$ and $P_2 = j$
  3. Put $\hat{D}_{ij}^{(obs)} = A$
- Step 2
  1. Find the smallest component $\hat{D}_{ij}^{(obs)}$ of $\hat{D}^{(obs)}$ with $i = P_2$
  2. Put $P_3 = j$
  3. Put $\hat{D}_{ij}^{(obs)} = A$

Improve the performance of the regularization method

2. Define the breaking points of the detail vectors

- Objective: Define the vector BP containing the breaking points of the $(N-1)$ detail vectors of ψ
- Define $\hat{D}^{(obs)}$ as the vector of Euclidean distances between the neighboring vectors of the reordered Σ
- Step 1
  1. Find the largest component $\hat{D}_l^{(obs)}$ of $\hat{D}^{(obs)}$
  2. Put $BP_1 = \hat{D}_l^{(obs)}$
  3. Put $\hat{D}_l^{(obs)} = 0$
- Step 1
  1. Find the largest component $\hat{D}_l^{(obs)}$ of $\hat{D}^{(obs)}$
  2. Put $BP_1 = \hat{D}_l^{(obs)}$
  3. Put $\hat{D}_l^{(obs)} = 0$

Optimization algorithm

Optimization algorithm

The optimization algorithm can be divided into 6 steps

1. Reorder the assets
2. Define the breaking points of the ψ vectors
3. Project the input data on ψ
$$\psi^t\hat{\mu},\ \psi^t\hat{\Sigma}\psi$$
4. Use iterative thresholding to obtain a regularized version of $\psi^t\hat{\Sigma}\psi$
5. Find $\hat{c}^*$ using the Lagrange multiplier method
6. Obtain $\hat{x}^*$ through
$$\hat{x}^* = \frac{1}{N} + \psi\hat{c}^*$$

Simulation studies

Simulation Studies

Criterion

- Regularization methods tackle the problem of overfitting
- Hence, we need to use an out-of-sample type criterion to asses the performance of our regularization method
- Out-of-sample Sharpe ratio [Carrasco, M. and Noumon, N., 2010]

$$\hat{SR} = \frac{{\hat{x}^*}^t \mu}{\sqrt{{\hat{x}^*}^t \Sigma {\hat{x}^*}^t}}$$

- $\hat{SR}$ is an estimate of the real Sharpe ratio defined as

$$SR = \frac{{x^*}^t \mu}{\sqrt{{x^*}^t \Sigma {x^*}^t}}$$

Simulation procedure

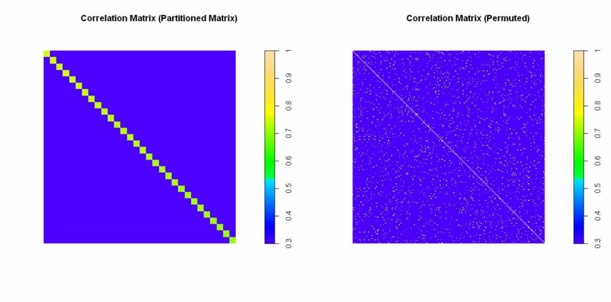
- Get K realizations for $\hat{SR}$
- Compute the bias and the standard deviation of these K realizations

Simulation settings

- $K = 500$
- Benchmark: Tikhonov regularization
- Define 3 different scenarios
- Fix 2 values of $\frac{N}{T}$ for each one of these 3 scenarios

Simulation Studies

Scenario 1



Simulation studies
Scenario 1

Table: Scenario 1: $\frac{N}{T} = \frac{120}{100} = 1.2$

Mean Return	Sharpe Ratio	Iterative Thresholding	Tikhonov
1%	1.6478	−0.0083 (0.0146)	−0.2310 (0.0713)
2%	3.2945	−0.0186 (0.0168)	−0.4666 (0.1031)
3%	4.9328	−0.0291 (0.0202)	−0.7005 (0.1393)
4%	6.5554	−0.0398 (0.0240)	−0.9306 (0.1756)
5%	8.1553	−0.0510 (0.0278)	−1.1553 (0.2107)
6%	9.7263	−0.0627 (0.0316)	−1.3728 (0.2438)

Simulation studies

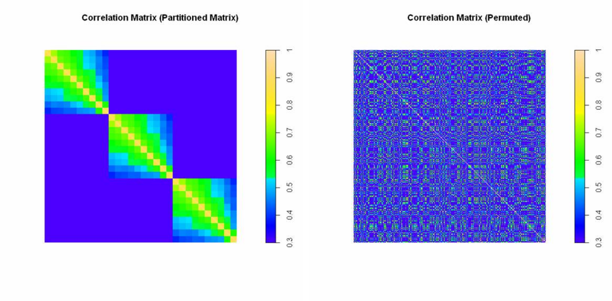
Scenario 1

Table: Scenario 1: $\frac{N}{T} = \frac{300}{100} = 3$

Mean Return	Sharpe Ratio	Iterative Thresholding	Tikhonov
1%	1.7125	-0.0224 (0.0085)	-0.3184 (0.0762)
2%	3.4215	-0.0421 (0.0097)	-0.6290 (0.1288)
3%	5.1238	-0.0621 (0.0122)	-0.9375 (0.1839)
4%	6.8159	-0.0824 (0.0152)	-1.2432 (0.2389)
5%	8.4947	-0.1032 (0.0187)	-1.5453 (0.2930)
6%	10.1572	-0.1246 (0.0226)	-1.8431 (0.3456)

Simulation Studies

Scenario 2



Simulation Studies

Scenario 2

Table: Scenario 2: $\frac{N}{T} = \frac{120}{100} = 1.2$

Mean Return	Sharpe Ratio	Iterative Thresholding	Tikhonov
1%	1.3826	-0.0511 (0.0122)	-0.1692 (0.0970)
2%	2.7634	-0.1668 (0.0189)	-0.3546 (0.1155)
3%	4.0982	-0.2787 (0.0346)	-0.5428 (0.1368)
4%	5.3482	-0.3828 (0.0525)	-0.7338 (0.1569)
5%	6.4857	-0.4763 (0.0709)	-0.9254 (0.1751)
6%	7.4959	-0.5577 (0.0886)	-1.1137 (0.1918)

Simulation Studies

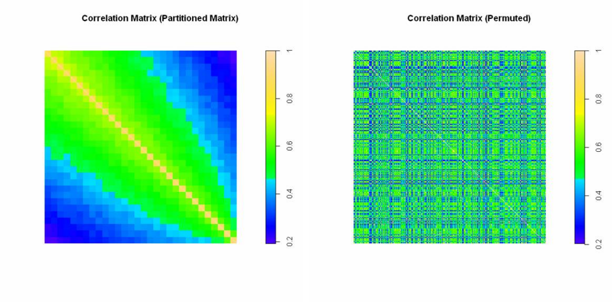
Scenario 2

Table: Scenario 2: $\frac{N}{T} = \frac{300}{100} = 3$

Mean Return	Sharpe Ratio	Iterative Thresholding	Tikhonov
1%	1.5345	-0.0484 (0.0088)	-0.2767 (0.0945)
2%	3.0446	-0.2005 (0.0271)	-0.6444 (0.1252)
3%	4.4731	-0.3424 (0.0441)	-1.0096 (0.1551)
4%	5.7764	-0.4704 (0.0566)	-1.3681 (0.1863)
5%	6.9294	-0.5826 (0.0661)	-1.7124 (0.2193)
6%	7.9248	-0.6790 (0.0761)	-2.0351 (0.2525)

Simulation Studies

Scenario 3



Simulation Studies

Scenario 3

Table: Scenario 3: $\frac{N}{T} = \frac{120}{100} = 1.2$

Mean Return	Sharpe Ratio	Iterative Thresholding	Tikhonov
1%	1.3826	-0.0511 (0.0122)	-0.1692 (0.0970)
2%	2.7634	-0.1668 (0.0189)	-0.3546 (0.1155)
3%	4.0982	-0.2787 (0.0346)	-0.5428 (0.1368)
4%	5.3482	-0.3828 (0.0525)	-0.7338 (0.1569)
5%	6.4857	-0.4763 (0.0709)	-0.9254 (0.1751)
6%	7.4959	-0.5577 (0.0886)	-1.1137 (0.1918)

Simulation Studies

Scenario 3

Table: Scenario 3: $\frac{N}{T} = \frac{300}{100} = 3$

Mean Return	Sharpe Ratio	Iterative Thresholding	Tikhonov
1%	1.4372	-0.1122 (0.0105)	-0.2687 (0.1003)
2%	2.8642	-0.2323 (0.0172)	-0.5483 (0.1341)
3%	4.2683	-0.3546 (0.0299)	-0.8241 (0.1719)
4%	5.6379	-0.4798 (0.0477)	-1.0948 (0.2090)
5%	6.9629	-0.6085 (0.0692)	-1.3591 (0.2436)
6%	8.2348	-0.7409 (0.0937)	-1.6159 (0.2750)

Bibliography

Markowitz, H.M.
Mean-variance analysis in portfolio choice and capital markets
Journal of Finance, 1952

Devolder, P.
Finance Stochastique
Editions de l'Universite de Bruxelles, 1996

Fryzlewicz, P.
Unbalanced Haar technique for nonparametric function estimation
Journal of the American Statistical Association, 2007

Brodie, J., Daubechies, I., De Mol, C., Giannone, D. and Loris, I.
Sparse and stable Markowitz portfolios
Proceedings of the National Academy of Sciences, 2009

Freyermuth, J-M., Ombao, H. and Von Sachs, R.
Tree-structured wavelet estimation in a mixed effects model for spectra of replicated time series
Journal of the American Statistical Association, 2010

Bibliography

Autin, F., Freyermuth, J-M. and Von Sachs

Ideal denoising within a family of tree-structured wavelet estimators

Electronic Journal of Statistics, 2011

El Karoui, N.

High-dimensionality effects in the Markowitz problem and other quadratic programs with linear equality constraints: risk underestimation

Annals of Statistics, 2011 (to appear)

Lillo, F. and Mantegna, R. N.

Spectral density of the correlation of factor models: A random matrix theory approach

Physical Review, 2005

Laloux, L., Cizeau, P., Potters, M. and Bouchaud, J-P.

Random matrix theory and financial correlations

Physical Review, 1999

Carrasco, M. and Noumon, N.

Optimal Portfolio Selection using Regularization

Preliminary and Incomplete

Johannes, J., Van Bellegem, S. and A. Vanhems

Convergence rate for ill-posed inverse problems with an unknown operator

Forthcoming in Econometric Theory