Financial Big Data Project: Covariance Matrix estimation techniques for large financial dataset

Nicolo' Taroni, Girolamo Vurro
Financial Engineering, EPFL, Switzerland

***Abstract*—The aim of this project is to show empirically the importance of covariance filtering in the prediction of the risk of a portfolio. To this extent, we compare the out-of-sample risk of the Global Minimum Variance Portfolio. We first compute it naively and then with each of the following methods: *eigen values clipping*, *Rotationally Invariant Estimators* (RIE), *Average Oracle* (AO) and *Bootstrap Average linkage Hierarchical Clustering* (BAHC). We analyze also the Optimal Mean-Variance portfolio performance for each of the mentioned techniques.**

I. Introduction

This project aims to investigate the performance of different techniques for the estimation of high dimensional Covariance Matrices in financial time series. In particular, we implement and compare the performance of the following techniques: Naive Covariance estimation, Eigenvalue Clipping, optimal Non-Linear Shrinkage (NLS), Average Oracle (AO) and Bootstrapped Average Hierarchical Clustering (BAHC). Given the non-stationarity of a market environment, we perform our comparison using Rolling Windows. As a measure of the performance of the different methods, we use the out-of-sample risk obtained by the Global Minimum Variance portfolio and the Frobenius Norm of the difference between the realized covariance matrix and the predicted one. In the second part of the project, we measure also the different performances in terms of returns obtained with mean-variance optimum portfolio for a given expected return G.

In the following section II we briefly summarize the dataset which we adopted for the implementation of the covariance filtering methods. In section III we first present a review of the theoretical concepts that are fundamental for the description and implementation of the covariance matrix filtration. Furthermore, we provide a motivation for the covariance matrix filtration, by summarizing the theory proposed by Marčenko - Pastur relative to the eigenvalues of the covariance matrix and the crucial role that they might have in the estimation process. Then, we describe the different covariance filtering methods that we use in our implementation. Subsequently, in section IV we describe and discuss the implementation of the rolling window that we used in order to compare the different portfolio strategies. In section V we present the results that we obtained, and we report our conclusions in sectionVI.

II. The dataset

The dataset that we use for the analysis consists of daily close-to-close return observation of the US equity markets data for 42 years: from 01-01-1980 to 31-03-2022. We select the 1135 stocks with the largest market capitalization using WRDS. Therefore, in total, we have 5598 rows and 1135 columns.

III. Methodologies

In this section, we recall the fundamental theory of the different methods that are used in the project to estimate the Covariance Matrices.

A. Dependence in Finance and notation

Given N assets and T time steps, we call $r_{i,t}$ the return of the i-th asset at time t. We use the notation g_i to indicate the average return for asset i and σ_i^2 to indicate the variance. The simplest and naive method to characterize the correlation between the stocks is to estimate the covariance matrix as:

$$\Sigma_{i,j} = \frac{\sum_{t=1}^{T}(x_{i,t} - g_i)(x_{j,t} - g_j)}{T}$$

.

Under the assumptions of stationarity and Gaussian distributed returns, for a really large number of observations T compared to N, we should expect to obtain an estimation close to the real value of Σ. However, in Finance we can assume neither stationarity nor an infinite number of observations. In fact, the ratio between the number of variables N and the timesteps T turns out to tend to a finite number q and not to 0 in most of the practical cases, with $q = O(1)$. For these reasons is necessary to examine different methods with the aim to estimate properly the covariance matrix and to capture results that can be established in this special asymptotic limit, where the empirical density of eigenvalues (the spectrum) is strongly distorted when compared to the 'true' density (corresponding to q $\to$ 0) [1].

B. Spectral Decomposition Theorem and Random Matrix Theory

We recall the *Spectral Decomposition Theorem* (used for Principal Component Analysis) which will be useful to understand the concepts that we will describe in the next sections, where we will use direct applications of these results. We consider a given covariance matrix Σ:

$$\Sigma = \begin{bmatrix} \sigma_1 & \sigma_{1,2} & \dots & \sigma_{1,N} \\ \sigma_{2,1} & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \sigma_{(N-1),N} \\ \sigma_{N,1} & \dots & \sigma_{N,(N-1)} & \sigma_N \end{bmatrix}$$

with $\sigma_i = 1$ for $i \in \{1, \dots, N\}$ which implies that $\sigma_{i,j} = \rho_{i,j}\sigma_i\sigma_j = \rho_{i,j}$, $\forall i \neq j$ and therefore $C = \Sigma$, where C is the correlation matrix.

Thus, being C the correlation matrix we know that it is symmetric non-negative definite and that it we can rewrite it like:

$$C = V'\Lambda V$$

where V is the *orthonormal* matrix of the eigenvectors of C. It holds that:

$$VV' = V'V = I_N.$$

Moreover, Λ is the diagonal matrix of the eigenvectors of C:

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{bmatrix}$$

with the eigenvalues on the diagonal in decreasing order as follows:

$$\lambda_1 \geq \ldots \geq \lambda_N \geq 0.$$

Now we can introduce the density of the eigenvalues, i.e. the *spectrum* $P(\lambda)$. We consider a sample of data X, a matrix of dimensions $N \times T$. N is the number of random variables (in the case that we treat, the random variables are the returns for each stock included in a portfolio) and T is the number of observations (realization of each random variable). Therefore, X contains the realizations of a multivariate random variable. From this sample, one could estimate the C correlation matrix with a method which provides an *unbiased* estimation. However, given that X contains realizations of random variables, this implies that also the correlation matrix that is estimated from X is random, and therefore also the related eigenvalues are random. Moreover, this also implies that the distribution of the eigenvalues $P(\lambda)$ is random. However, we would like to quantify how much of the correlation matrix C is random and therefore how much of $P(\lambda)$ is random.

In order to do this, we refer to the *Random Matrix Theory* that was developed by Marčenko - Pastur [1].

C. Random matrix theory: Correlation matrix spectrum

Marčenko - Pastur (1967) [2] formulated their theory under the null hypothesis that we consider a matrix of Gaussian returns, which are uncorrelated. In particular, under this hypothesis, the X matrix would have entries that are the realization of a standard normal random variable:

$$(X)_{i,t} = x_{i,t} \sim \mathcal{N}(0, 1)$$

for $i \in \{1, \dots, N\}$ and $t \in \{1, \dots, T\}$.

The correlation matrix would be estimated as follows:

$$\hat{C} = \frac{1}{T}X'X$$

And we know that this estimate is consistent as:

$$\lim_{T\to\infty} \hat{C_{ij}} = C_{ij}$$

When $N, T \to \infty$, given that $q = \frac{N}{T}$, we would have that $P(\lambda) \to P_0(\lambda)$ where:

$$P_0(\lambda) = \frac{1/q}{2\pi}\frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{\lambda}$$

is the probability density of λ under the null hypothesis of Gaussian uncorrelated returns, i.e $C = I$. In other words, it is the distribution of a randomly drawn eigenvalue of the correlation matrix estimated from a sample of uncorrelated Gaussian variables. Heavy-tailed variables are able to modify the Marčenko - Pastur distribution.

Marčenko - Pastur showed that the eigenvalue boundaries are $\lambda_\pm = 1 + q \pm 2\sqrt{q} = (1 \pm \sqrt{q})^2$ and eigenvalues varies as follows:

$$\lambda \in [\lambda_-, \lambda_+]$$

This means that even if you have no structure at all (the data were generated by independent samples of standard normal distribution), there is a signature in the distribution of the eigenvalues of the correlation matrix.

In particular when $N/T \to 0$, $\lambda_\pm \to 1$.

Moreover, we have that the expected values and standard deviation of this distribution are the following:

- $\mathbf{E}(\lambda) = 1$
- $std(\lambda) \propto \sqrt{q}$

Therefore, the Marčenko - Pastur distribution represents the density of random eigenvalues of a correlation matrix constituted by totally random price returns, which means that it does not have structure at all. However, the noise produces a structure that we can observe in this distribution. Moreover, the latter distribution only depends on the ratio q and therefore we can intuitively interpret that the *course of dimensionality* (meaning that in higher dimensions we need a very high amount of data points to get an approximate estimate) spreads the eigenvalues: indeed, we saw, on the contrary, that for $q \to 0 \Rightarrow \lambda = 1$. When $q > 0$ we have the distribution that we described above and that we show on the plot below in orange. The larger the value of q, the more spread will be the distribution; we can deduce this also from the formulas of the bounds $\lambda_\pm$. Therefore, the noise comes from the fact that if we do not have enough data points, then we will never be able to reach the correct limit of the covariance matrix estimator.

In the plot below we reproduced the Marčenko - Pastur distribution after we choose $q = \frac{1}{2}$ (so we fixed $N = 1000$ and $T = 2000$). The orange distribution represents the probability density $P_0(\lambda)$. Moreover, the histogram represents the empirical distribution of the eigenvalues computed with the following procedure:

- We generated returns $r_{t,i} \sim \mathcal{N}(0, \sigma^2)$ with $t \in \{1, \dots, T\}$ and $i \in \{1, \dots, N\}$ which were the $x_{t,i}$ entries of the matrix X.

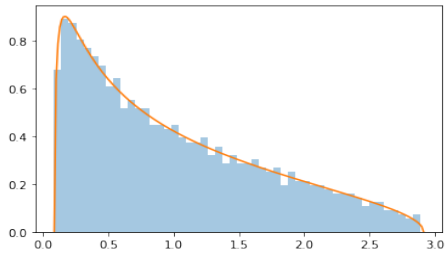

Fig. 1: Marčenko - Pastur distribution for $N = 1000$, $T = 2000$

- We computed the estimated correlation matrix.
- Finally, we computed the eigenvalues of the estimated correlation matrix and we plotted them in the previous histogram.

As we can observe, the empirical distribution stays approximately within the bounds of the Marčenko - Pastur distribution and they also roughly have the same shape. This brief experiment shows what was previously described, i.e. uncorrelated time series lead to a covariance matrix which produces eigenvalues that asymptotically follow the distribution $P_0(\lambda)$.

D. Empirical result of the analysed dataset

For our dataset, we also compute the correlation matrix and obtain the eigenvalues. We plot the histogram of the eigenvalues as a representation of their empirical distribution and compare it with the Marčenko - Pastur distribution, as we show in the figure below.

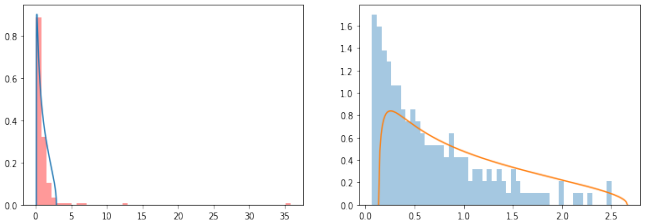

Fig. 2: Empirical average eigenvalues spectrum of the correlation matrix, compared to the Marcenko-Pastur prediction. On the right we removed the several eigenvalues that leak out of the Marcenko-Pastur band

As we can observe, the vast majority of eigenvalues take values within the bulk delimited by the $\lambda_{\pm}$ boundaries. However, we can also observe that a minority of eigenvalues has a bigger value than the upper bound λ_+. This result follows what is obtained by Laloux et al. (1999), Plerou et al. (2002) [3], [4]. Indeed, if financial data would produce an empirical eigenvalue distribution similar to 1, then this would have meant that there is no structure at all. However, it is interesting to observe that there are some deviations (3.5% of the eigenvalues), which are far larger than the theoretical upper limit and this means that we could use them to extract useful information. In particular, while the 96.5% of the information conveyed from the eigenvalues of the correlation matrix cannot be distinguished from randomness (and it also shows how covariance matrices in Finance are mostly random), we could keep and exploit the information of the remaining eigenvalues that are far out of the boundaries.

E. Correlation Eigenvalue clipping

Therefore, following the hypothesis that the eigenvalues within the interval that delimits the bulk are noise, we can replace them with a constant.
Thus, we adopt the following procedure, in order to apply the eigenvalue clipping:

- First we compute the empirical eigenvalues $\lambda_i^{(e)}$, for $i \in \{1, \dots, 200\}$ from the data matrix containing the time series of the stocks, and we compute also the upper bound λ_+
- We replace all the eigenvalues smaller than the upper bound with δ: $\lambda_i^{(e)} \leq \lambda_+ \rightarrow \delta$ where

$$\delta = \frac{1}{N_{bulk}} \sum_{i:\lambda_i^{(e)} \leq \lambda_+} \lambda_i^{(e)}$$

with N_{bulk} that is the number of eigenvalues within the bulk.
It is important to notice that when we have an empirical eigenvalue distribution which is very close to the Marčenko - Pastur distribution, like the one that we observe in Figure 1, we also know that we should replace these eigenvalues with one. This means that the true distribution should peak at 1 because we know that by definition $C = \mathbf{I}_N$.
In our case, where we observe what is plotted in figure 2, we proceed as explained above because we want to replace the eigenvalues in the bulk with a constant (as this represents the noisy part) and at the same time we also need to keep in consideration the fact that the sum of the eigenvalues should be N. Formally this means that:

$$\sum_{i=1}^{N} \lambda_i(e) \mathbb{1}_{\{\lambda_i^{(e)} - \lambda_+ > 0\}} + \delta N_{bulk} = N$$

Therefore, we cannot replace the eigenvalues in the bulk by one, as also their average is smaller than 1, and thus we replace them with their average. The rest of the eigenvalues should be kept as they are. In this way we are able to completely filter out the noise part of the eigenvalues.
To sum up, we have:

$$\lambda_i^{(clip)} = \begin{cases} \delta & \text{if } \lambda_i^{(e)} \leq \lambda_+ \\ \lambda_i^{(e)} & \text{otherwise} \end{cases}$$

Once we compute the eigenvalues clipping, we can define the diagonal matrix that contains the flipped eigenvalues like:

$$\Lambda^{(clip)} = diag(\lambda_i^{(clip)}), \text{ for } i \in \{1, \dots, N\}$$

Then, we can compute the filtered matrix with the eigenvalues clipping method as follows:

1) We compute the filtered correlation matrix through the following formula
$$\hat{C}^{(clip)} = V' \Lambda^{(clip)} V$$
2) We check if the filtered correlation matrix obtained with the previous formula does not have ones on the diagonal. We know that this is a requirement that a correlation matrix should satisfy by definition. Therefore, we set:
$$\hat{C}_{ii}^{(clip)} = 1$$

Notice that we keep the *unitary* matrix which contains the eigenvectors of the initial (non-filtered) estimate correlation matrix V.

F. Optimal RIE (NLS)

The class of Rotationally Invariant Estimators it's a generalization of the Eigenvalue Clipping method described before. The RIE method starts from the assumption that no structure of the correlation matrix is available. It estimates C keeping the eigenvectors v_i from V and changing the eigenvalues λ_i through a transformation $\xi(\lambda_i)$. Doing so we obtain the estimator
$$\Xi = \sum_i \xi(\lambda_i) v_i^T v_i.$$

The optimal RIE is therefore the estimator with the transformation $\xi(\lambda_i^{in})$ as close as possible to λ_i^{out} (see [5] for the details of the derivation). However, the numerical estimation of ξ is hard for small eigenvalues [6].

G. Average Oracle

The Average Oracle is another technique that allows us to perform a covariance matrix filtration. In particular, given a data matrix $R \in \mathbf{R}^{N \times T}$ where N is the dimension of the multivariate random variable and T is the number of its realizations, the Average Oracle algorithm consists in K cross validations over the data matrix which can be described by the following steps:

1) Pick a random fraction ϕ of the total number of realizations T and use this fraction of the dataset as *virtual in - sample* training set, i.e. we select random rows of the initial data matrix which are not necessarily consecutive and we used them as a training set.
2) Assign the remaining $1 - \phi$ fraction of the dataset (namely the remaining rows) to the out of *virtual out - of - sample* training set.
3) Therefore, for each step of cross-validation $k \in K$:
   - estimate the correlation matrix based on the out-of-sample training set $C^{out,k}$;
   - Compute the spectral decomposition of the correlation matrix estimated by using the in sample train set, and keep the *unitary* matrix of the eigenvectors of the correlation matrix estimate $V^{in,k}$.
4) Using the fact that $C = V' \Lambda V \Rightarrow VCV'$, compute the eigenvalues for the $k-th$ cross validation as follows:
$$\Lambda_{CV}^{(k)} = diag(V^{in,k} C^{out,k} V^{in,k,\dagger})$$
5) Compute the average of the eigenvalues computed in each step of the cross-validation:
$$\Lambda_{CV} = \frac{1}{K} \sum_{k=1}^{K} \Lambda_{CV}^{(k)}$$

However, this implementation really rests on stationarity which is a requirement that is not typically met by financial markets. Therefore, in order to take this into account, for our implementation we follow the approach proposed by Bongiorno and Challet (2022), [7] which consists of calculating:
$\Lambda^{\text{Oracle}} = \text{diag}\left(V^{in,\dagger} C^{\text{out}} V^{\text{in}}\right)$
and finally:
$\text{avg}\left(\Lambda^{\text{Oracle}}\right) = \text{avg}\left[\text{diag}\left(V^{\text{prev},\dagger} C^{\text{next}} V^{\text{prev}}\right)\right]$

H. BAHC

The usage of *Rotationally Invariant estimators* assumes that no a priori structure of the correlation matrix is available. However, as shown by [8] price return correlation matrices are approximately hierarchical, exhibiting squares in squares along the diagonal. We can capture this and the persistence structure through the Bootstrapped linkage Hierarchical Clustering method. The method consists of:

- computing M bootstraps of the return matrix R;
- computing the HCAL-filtered matrix for each bootstrap;
- finally computing the average of the M HCAL-filtered matrices

We refer to [8] for the details of the implementation.

I. Exploratory Analysis of the methods

Before implementing the rolling windows, we take a sample of data of length $T = 1000$ and stocks $N = 200$ and we implement each of the described methods to estimate the covariance matrix between the stocks of the window. Then we use the resulting Covariance matrices to compute the weights of the optimum mean variance portfolio (of which we provide details in the appendix section VII) for a range of expected returns G, going from 0 to 50%. Then we calculate the realized risk of these portfolios on the out of sample window of length $T_{out} = 250$. We plot the relation between G and the risk (in and out of sample) for each method, summarized in the figures below (figures 1, 2).
Indeed, we have a clear evidence that when we naively estimate the covariance matrix we significantly *underestimate* the actual risk. This is due to the fact that more than 90% of

the portfolio is noise and therefore 90% of the optimization of the portfolio is on the noise itself! The out-of-sample volatility of the portfolio is therefore much higher than the one predicted. Thus, if we use the methods to filter the covariance matrices, we would be able to avoid optimizing over the noise and therefore we would have better results.
In figure 3 we can observe that with the naive covariance matrix estimation, the estimated risk for the specific example of the mean variance portfolio (predicted) is quite far from the realized risk per level of return, due to the fact that we are implicitly including the noise in our estimate.
A clear evidence of the goodness of filtering methods can be visualized in the plots represented in figure 4 where we can observe that, after the filtration, we get a consistent improvement of the quality of our prediction, compared to the realized level of risk.
Indeed, from the plots in figure 2 it is evident that the two curves are closer to each other. This indicates that for each level of return, the predicted level of risk is much closer to the realized risk, if compared to the previous plot in figure 3 which represented the naive estimation.

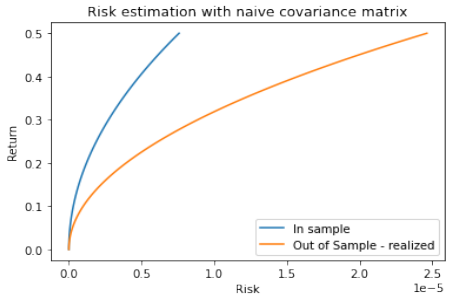

Fig. 3: Plot of the predicted and realized risk-expected return relation in the case of the naive estimation

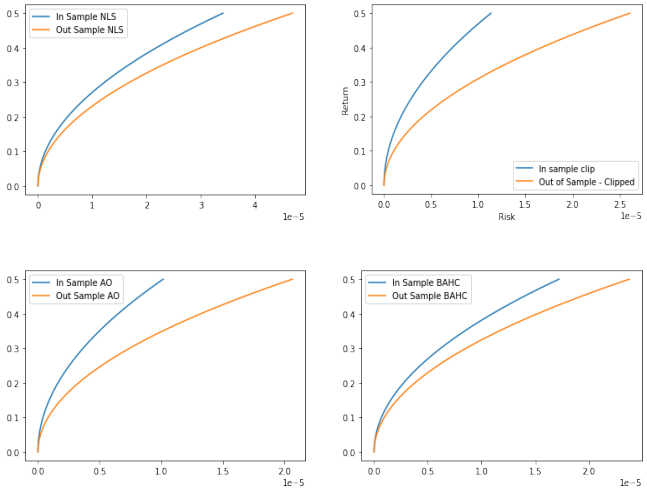

Fig. 4: Plots of the predicted and realized risk-expected return relation with filtered covariance estimation: NLS, Eig. clipping, AO, BAHC

IV. Implementations of the Rolling window

After an exploratory analysis of our data, we observe that keeping only the stocks without missing values for the entire dataset would have led to only 88 complete stocks. Therefore we decide to split the dataset into two parts: we keep the observation from 01/01/1980 to 31/12/1999 to calibrate the Average Oracle method. In this window, we delete the stocks with missing observations, keeping only 210 stocks and we apply the Average Oracle calibration algorithm. In the second half of the dataset (containing observations from 01/01/2000 to 31/03/2022), we delete the columns of stocks with missing observations, obtaining a sub-dataset of 5598 rows and 228 columns. We test all the covariance estimation methods for two different N (number of stocks to consider in the estimation), namely 200 and 100, in order to observe the eventual differences. For each value of N, the columns are selected randomly from the dataset. Then, we choose T (the length of the rolling window) to obtain a T/N ratio bigger than 1. Our choice of T falls on 252 (number of days corresponding to 1 year of market observations) and 500 days (roughly 2 years). To summarize, we implement four rolling windows with the four combinations between the values of N and T.
In each rolling window of length T we estimate the covariance matrix with each one of the methods described before, and then we calculate the respective Global Minimum Variance Portfolio. The same is done for the mean-variance portfolio with constrained weights (of which we provide details in the appendix section VII) and expected objective return $G = 3\%$, obtaining again one vector of weights for each method. To compare the results obtained with the different methods, we implement at the same time a second *out of sample* rolling window of reduced length $T_{out} = 252$ when $T = 500$ and $T_{out} = 125$ when $T = 252$. This rolling window moves in parallel with the first one and starts from the first observation not included in the first rolling window. We use this second window to calculate:

- The *Frobenius Norm* of the difference between the predicted Covariance Matrix and the realized one;
- The *out-of-sample risk* obtained with the weights of the Global Minimum Variance Portfolio;
- the *realized returns* obtained from the previously constructed Optimal Mean-Variance Portfolios with expected return G;
- the *ratio between the realized return and the realized volatility* for the Optimal Mean-Variance Portfolios with expected return G.

We store these values in several lists, one for each combination between the covariance estimation method and the quantities above described. In this way, we obtain different time series describing the evolution of the quantities for each method.

V. Results

In this section, we present the results of the implementation that we have described and we provide some intuitive

comments on the plots, in order to highlight the impact of the covariance filtering on the quality of risk prediction and on the performance of the different types of portfolios that we have constructed.
Moreover, we divide this section into two subsections: in subsection V-B we keep fixed the number of stocks of our portfolio, $N = 200$ and we compare the performances of the estimators and of the portfolios for different dimensions of the rolling window ($T = 500$ and $T = 252$) in order to highlight the importance of the number of data points in the estimation of the covariance matrix and its impact on the different portfolios' performances; in subsection V-B we present the same comparison as in the previous subsection, but for a different number of stocks in the portfolio, namely $N = 100$.

A. Comparison between rolling windows of size $T = 500$ and $T = 252$ and fixed size $N = 200$

To visualize the results obtained with the metrics that we use to evaluate the goodness of each Covariance Matrix estimation method, we first plot the time series of:

1) the Frobenius Norm of the difference between the predicted Covariance Matrix and the realized one (figure 5);
2) the ratio of the realized risk obtained with the Naive Covariance estimation and the realized risk obtained with each method (for the Global Minimum Variance portfolio, figure 6);
3) the ratio between realized (out of sample) and predicted (in sample) risk for each method (figure 7);
4) the realized returns obtained from each method with Optimal Mean-Variance Portfolios (expected return G, figure 8);
5) the ratio between the realized return and the realized volatility for each method with the Optimal Mean-Variance Portfolios (figure 9).

To interpret better the results, we also calculate the fraction of times in which each method outperforms the naive covariance estimation time as reported in table I for each criterion. We can observe from the Frobenius Norm distance that BAHC, Eigenvalue Clipping and Average Oracle obtain better prediction in at least 53% of the cases that the Naive method for $T = 500$. When the length of the rolling window decreases, the performances of Eigenvalue Clipping and BAHC increase again, outperforming naive ones respectively 63 % and 70% of times. This trend is confirmed also by the out-of-sample risk criterion (and even more evident): Average Oracle and BAHC outperform out-of-sample naive risk 100% of times for the reduced window length $T = 252$, and Eigenvalue Clipping is really close to them. It is interesting to observe how BAHC method, in the $T = 500$ case, outperforms instead the naive method "only" in 65 % of cases. This could be interpreted as related to the *course of dimensionality*: in the cases where the dimension of the problem is high (large N value), more data points are needed to get an accurate estimate of the covariance matrix. Therefore, with a reduced window size combined with a high number of stocks, the naive method seems to suffer compared to other techniques. Regarding the return of the mean variance optimum portfolio, Eigenvalue Clipping, BAHC and AO slightly outperform the naive method for $T = 500$, while surprisingly their performance worsens in the $T = 252$ case. The NLS method in both cases does better than the naive only around 43 % of the time. Analyzing the return-risk ratio, we can observe that the performances of BAHC, AO and Clipping seem to improve compared to the *return scenario*. We expected this, given the lower out-of-sample risk of the three models.

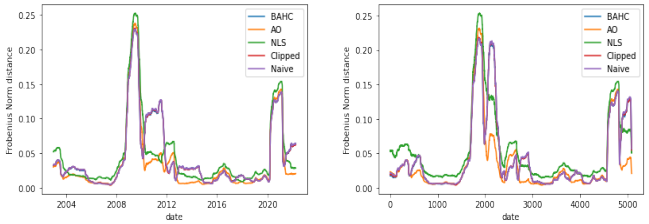

Fig. 5: Frobenius Norm for $N = 200$ stocks. $T = 500$ on the left, $T = 252$ on the right

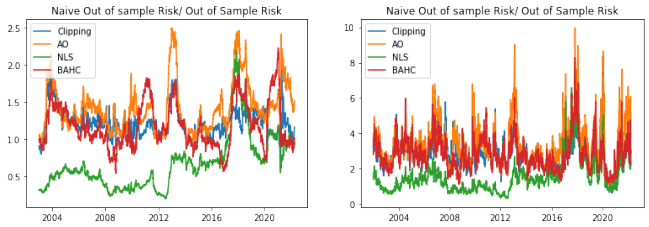

Fig. 6: Out out for $N = 200$ stocks. $T = 500$ on the left, $T = 252$ on the right

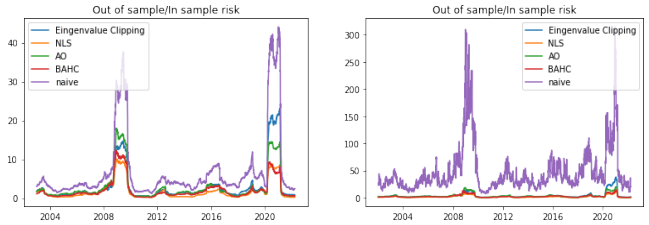

Fig. 7: Out in for $N = 200$ stocks. $T = 500$ on the left, $T = 252$ on the right

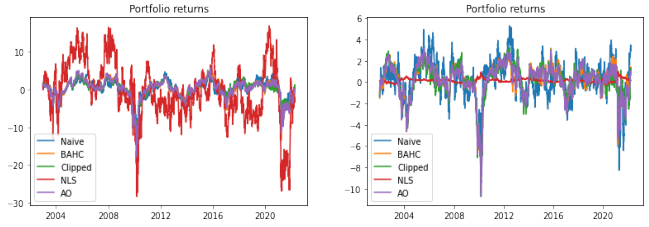

Fig. 8: Realized return for $N = 200$ stocks. $T = 500$ on the left, $T = 252$ on the right

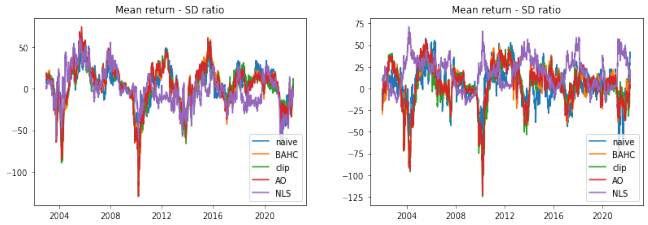

Fig. 9: Mean return - SD ratio for $N = 200$ stocks. $T = 500$ on the left, $T = 252$ on the right

Frobenius Norm		
Cov. f. method	$T = 500$	$T = 252$
Eig. Clipping	0.5674	0.6357
BAHC	0.5348	0.7056
NLS	0.3540	0.1610
Average Oracle	0.5529	0.5277
Out - out ratio		
Cov. f. method	$T = 500$	$T = 252$
Eig. Clipping	0.9183	0.9727
BAHC	0.6514	1.0
NLS	0.1389	0.6789
Average Oracle	0.9723	1.0
Portfolio Returns		
Cov. f. method	$T = 500$	$T = 252$
Eig. Clipping	0.5178	0.4839
BAHC	0.5376	0.4867
NLS	0.4489	0.4266
Average Oracle	0.5170	0.4863
Mean return - SD ratio		
Cov. f. method	$T = 500$	$T = 252$
Eig. Clipping	0.5352	0.5374
BAHC	0.5622	0.5405
NLS	0.4332	0.6067
Average Oracle	0.5285	0.5356

TABLE I: Fraction of times in which each method outperforms the naive method for the measures: Frobenius Norm, Out - out ratio and Mean return - SD ratio, ($N = 200$)

B. Comparison between rolling windows of size $T = 500$ and $T = 252$ and fixed size $N = 100$

Analyzing the plots (figures 10, 11, 12, 13, 14) of the time series and the results of table II, we observe similar results to the ones described in the previous subsection (V-B): Eigenvalue Clipping, Average Oracle and BAHC seem to be the best models while the performances of NLS are disappointing compared to naive ones. BAHC, AO and Clipping outperform the naive method more than 55 % of times for the Frobenius Norm criterion. Regarding out-of-sample risk, in the T = 252 rolling window we obtain satisfying results, in particular with Average Oracle. However, we can observe that the percentage are not as extreme as the ones of the previous subsectionV-B and they decrease significantly for Eigenvalue Clipping and BAHC in the $T = 500$ scenario. In term of returns, the number of times in which the method outperform the naive one is close to 50 % and it increases only slightly in the case where we divide returns by portfolio volatility.

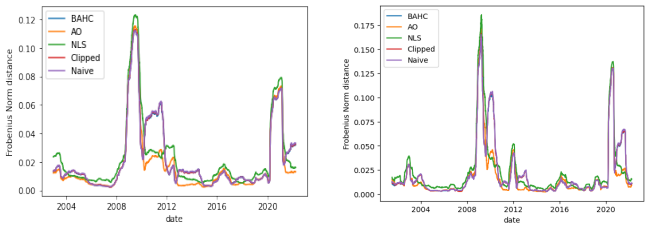

Fig. 10: Frobenius Norm for $N = 100$ stocks. $T = 500$ on the left, $T = 252$ on the right

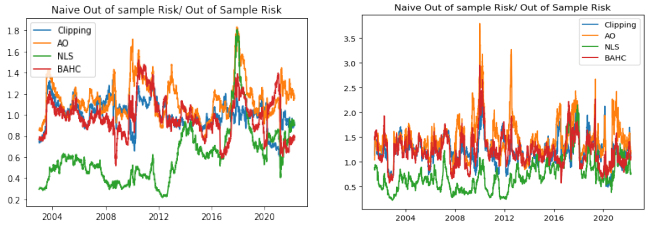

Fig. 11: Out out for $N = 100$ stocks. $T = 500$ on the left, $T = 252$ on the right

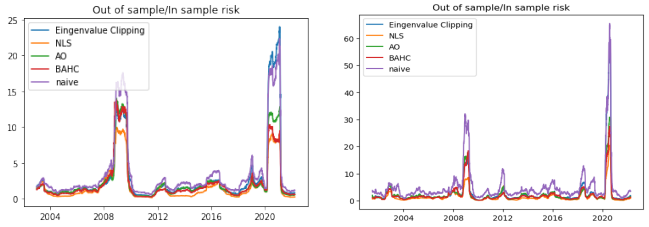

Fig. 12: Out in for $N = 100$ stocks. $T = 500$ on the left, $T = 252$ on the right

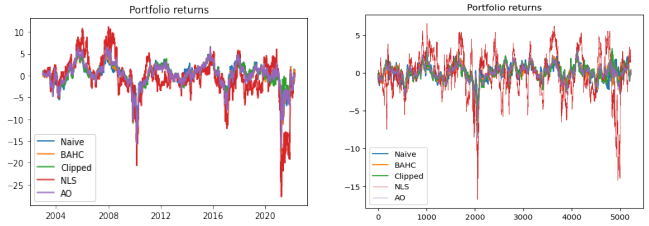

Fig. 13: Realized return for $N = 100$ stocks. $T = 500$ on the left, $T = 252$ on the right

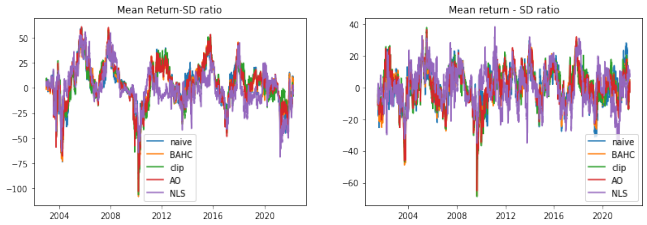

Fig. 14: Mean return - SD ratio for $N = 100$ stocks. $T = 500$ on the left, $T = 252$ on the right

Frobenius Norm		
Cov. f. method	$T = 500$	$T = 252$
Eig. Clipping	0.5686	0.6434
BAHC	0.5589	0.6478
NLS	0.3556	0.2454
Average Oracle	0.5770	0.5647
Out - out ratio		
Cov. f. method	$T = 500$	$T = 252$
Eig. Clipping	0.5331	0.8293
BAHC	0.497	0.7919
NLS	0.0621	0.1582
Average Oracle	0.8208	0.9080
Portfolio Returns		
Cov. f. method	$T = 500$	$T = 252$
Eig. Clipping	0.5001	0.4982
BAHC	0.4961	0.4517
NLS	0.4189	0.4977
Average Oracle	0.5139	0.4632
Mean return - SD ratio		
Cov. f. method	$T = 500$	$T = 252$
Eig. Clipping	0.5003	0.5024
BAHC	0.4998	0.4697
NLS	0.4	0.4969
Average Oracle	0.5116	0.4781

TABLE II: Fraction of times in which each method outperforms the naive method for the measures: Frobenius Norm, Out - out ratio and Mean return - SD ratio, ($N = 100$)

VI. Conclusion

In this project, we have investigated empirically the importance of covariance filtering in the prediction of the risk of a portfolio. To this extent, comparing the out-of-sample risk of the Global Minimum Variance Portfolio through rolling windows, we have observed as Average Oracle, BAHC and Eigenvalue clipping could lead to significant improvement with respect to the naive method, especially when the $\frac{N}{T}$ value increases. Improvements are also obtained in terms of the Frobenius norm criterion, which measures Covariance Matrix similarity. However, we have not observed particular improvements in terms of the return of the Mean-Variance portfolio. Therefore one possible improvement is to try different portfolio implementations and strategies involving the use of the Covariance Matrix. Moreover, a further improvement could be to try larger values of N and T and more combinations between these two parameters. This could be investigated through the use of high frequency data.

VII. Appendix

In our applications we build the following optimal portfolios: Global minimum variance portfolio, Mean Variance optimal portfolio and Mean Variance portfolio with constraint on the weights. In the following section we briefly describe the optimization problems that each of those portfolio solves.

A. Global Minimum Variance

In the first application, we aim to build a portfolio of N stocks with the lowest attainable variance, with the constraint that we can only invest all the available capital that we have.

Therefore, in this optimization problem $\Sigma \in \mathbf{R}^{N\times N}$ represents the covariance matrix relative to the 200 stocks which are constituting our portfolio. Each of the proportions of our wealth that are allocated to the $i-th$ stock will be represented by w_i which is the $i-th$ entry of the weight vector $\mathbf{w} \in \mathbf{R}^N$. Consequently, the total variance of our portfolio will be:

$$\sigma_p^2 = \mathbf{w}'\Sigma\mathbf{w}$$

Thus, the optimization problem can be formulated as follows ($\mathbf{e} \in \mathbf{R}^N$ and has 1 in all its entries):

$$\min_w \mathrm{w}'\Sigma\mathrm{w} \text{ s.t. } \sum_i w_i = 1$$

$$\Leftrightarrow \min_w \frac{1}{2}\mathbf{w}'\Sigma\mathbf{w} + \lambda(1-\mathbf{w}'\mathbf{e})$$

Therefore, we solve the penalized minimization objective (cost function):

$$\Sigma\mathbf{w} = \lambda\mathbf{e}$$

$$\Rightarrow \mathbf{w} = \lambda\Sigma^{-1}\mathbf{e}$$

And from the constraint we have that:

$$\lambda\mathbf{e}'\Sigma^{-1}\mathbf{e} = 1$$

$$\lambda = \frac{1}{\mathbf{e}'\Sigma^{-1}\mathbf{e}}$$

And so we have that the vector of the weights of the *Global Minimum Variance Portfolio* are:

$$\mathbf{w}_{GMV} = \frac{\Sigma^{-1}\mathbf{e}}{\mathbf{e}'\Sigma^{-1}\mathbf{e}}$$

B. Mean Variance portfolio

This portfolio has the minimum variance, given a certain level of desired return that we want to achieve. Such approach is the *Mean Variance Portfolio* and it can be formally described as follows.

We have to introduce additional notation with respect to the previous strategy:

- $\mathbf{E}[r_i] = g_i$, that is the expected return of the $i-th$ stock, so the vector of expecter stock returns will be $\mathbf{g} \in \mathbf{R}^{200}$;
- The portfolio return $r_p = \sum_i w_i r_i$;
- The desired portfolio return G.

Therefore, the new optimization problem will be described by the following cost function and constraint:

$$\min_w \mathrm{w}'\Sigma\mathrm{w} \text{ s.t. } \sum_i w_i g_i = G$$

$$\Leftrightarrow \min_w \frac{1}{2}\mathbf{w}'\Sigma\mathbf{w} + \lambda(G-\mathbf{w}'\mathbf{g})$$

And solving the penalized minimization objective we obtain:

$$\Sigma\mathbf{w} = \lambda\mathbf{g}$$

$$\Rightarrow \mathbf{w} = \lambda\Sigma^{-1}\mathbf{g}$$

And from the constraint we have that:

$$\lambda\mathbf{g}'\Sigma^{-1}\mathbf{g} = G$$

$$\lambda = \frac{G}{\mathbf{g}'\Sigma^{-1}\mathbf{g}}$$

And so we have that the vector of the weights of the *Mean Variance Portfolio* are:

$$\mathbf{w}_{MVP} = G\frac{\Sigma^{-1}\mathbf{g}}{\mathbf{g}'\Sigma^{-1}\mathbf{g}}$$

$$\begin{aligned} R_{in}^2 &= \mathbf{w}_{in}'\Sigma_{in}\mathbf{w}_{in} \\ &= \frac{G^2}{(\mathbf{g}'\Sigma_{in}^{-1}\mathbf{g})^2}\mathbf{g}'\Sigma_{in}^{-1}\Sigma_{in}\Sigma_{in}^{-1}\mathbf{g} \\ &= \frac{G^2}{\mathbf{g}'\Sigma_{in}^{-1}\mathbf{g}} \\ R_{out}^2 &= \mathbf{w}_{in}'\Sigma_{out}\mathbf{w}_{in} \\ &= \frac{G^2}{(\mathbf{g}'\Sigma_{in}^{-1}\mathbf{g})^2}\mathbf{g}'\Sigma_{in}^{-1}\Sigma_{out}\Sigma_{in}^{-1}\mathbf{g} \end{aligned}$$

C. Mean Variance portfolio with constrains on the weights

This portfolio was adopted in order to make the comparison of the performances of portfolios constructed by using different covariance filtration methods. In particular, the *Mean Variance portfolio* with constraint on the weights solves the following problem optimization:

$$\min_w \mathrm{w}'\Sigma\mathrm{w}$$

$$\text{s.t. } \sum_i w_i g_i = G, \sum_i w_i = 1, |w_i| \le 1, \text{for } i = \{1,\dots,N\}$$

In order to compute the weights that solve this problem we use `cvxpy` which is an open source Python-embedded modelling language for convex optimization. Therefore, this does not yield a close-form solution as in the previous cases [9]. However, it might happen that the solver does not converge to a solution. Thus, when this happens, we replace the weight vector with the normalized version of $\mathbf{w}_{MVP}$, i.e. the Mean Variance portfolio but with rescaled weights such that they sum up to one.

REFERENCES

[1] J.-P. Bouchaud and M. Potters, "Financial applications of random matrix theory: a short review," *arXiv preprint arXiv:0910.1205*, 2009.

[2] V. A. Marchenko and L. A. Pastur, "Distribution of eigenvalues for some sets of random matrices," *Matematicheskii Sbornik*, vol. 114, no. 4, pp. 507–536, 1967.

[3] L. Laloux, P. Cizeau, J.-P. Bouchaud, and M. Potters, "Noise dressing of financial correlation matrices," *Physical review letters*, vol. 83, no. 7, p. 1467, 1999.

[4] V. Plerou, P. Gopikrishnan, B. Rosenow, L. A. N. Amaral, T. Guhr, and H. E. Stanley, "Random matrix approach to cross correlations in financial data," *Physical Review E*, vol. 65, no. 6, p. 066126, 2002.

[5] J. Bun, R. Allez, J.-P. Bouchaud, and M. Potters, "Rotational invariant estimator for general noisy matrices," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 7475–7490, 2016.

[6] J. Bun, J.-P. Bouchaud, and M. Potters, "Cleaning large correlation matrices: Tools from random matrix theory," *Physics Reports*, vol. 666, pp. 1–109, jan 2017. [Online]. Available: https://doi.org/10.10162Fj.physrep.2016.10.005

[7] C. Bongiorno, D. Challet, and G. Loeper, "Cleaning the covariance matrix of strongly nonstationary systems with time-independent eigenvalues," 2021. [Online]. Available: https://arxiv.org/abs/2111.13109

[8] C. Bongiorno and D. Challet, "Covariance matrix filtering with bootstrapped hierarchies," 2021.

[9] "Welcome to cvxpy 1.3¶." [Online]. Available: https://www.cvxpy.org/