18.338 Final Project report: Random Matrix filtering

Hussein Fellahi

December 2021

1 Introduction

Classical estimators such as sample mean, variance and covariance show canoncial statistical properties such as consistency, convergence and unbiasedness that prove crucial in practice. Yet,these properties usually show up asymptotically, i.e. with a significant amount of *i.i.d.* realizations of the same random variable. Numerous cases appear in practice where this is hardly the case. In the context of large scale investing, suitable prediction techniques would require thousands of entries (around 50 years of daily data) to provide with relevant portfolio optimization strategies. Yet, Amazon, Google or Tesla did not even exist 30 years ago, thus finding relevant data becomes impossible. Lowering the frequency of sampling is also not a solution, as the high frequency setting presents a much different correlation pattern, which is not generalizable to lower frequencies.

In the project, we will tackle this issue of estimation in a non-asymptotic setting from the lense of filtering: we will use Random Matrix Theory to bridge the gap between the asymptotic properties of our estimators and the amount of information we have from the finite (rather small) sample. To illustrate the added value of RMT in this setting, we will study two use cases:

- Covariance matrix filtering with RMT in the context of Large scale Portfolio Optimization
- Weight matrix filtering with RMT to avoid overfitting in the context of Neural Networks training

2 Covariance matrix filtering in Large Scale Portfolio Optimization

2.1 The classical Portfolio Optimization model

In a context of Portfolio Optimization, the paradigm aims at minimizing the risk while maximizing the return of the portfolio. To do so, the dominant framework,

both in literature and in practice, in the so-called Mean-Variance optimization problem derived by Harry Markowitz (1957). Under some assumptions on the distribution of the stock prices and returns, the model aims at minimzing the risk, modeled by the variance, while maximizing the returns, modeled by the expectation, of the return of a given portfolio. Formally this gives:

$$\min_w w^T \Sigma w$$

s.t.

$$\mu^T w \geq r$$
$$\mathbf{1}^T w = 1$$

where Σ is the covariance matrix of the returns, μ is the vector of expected returns, $\mathbf{1} = [1, ..., 1]^T$ and r is the target minimum return on the portfolio.

Under certain condition, this problem has a closed form. In particular, we might be interested in considering the tangency portfolio,that maximizes the ratio:$\frac{\mu^T w}{w^T \Sigma w}$. Its closed form solution is: $w^* = \frac{\Sigma^{-1}\mu}{\mathbf{1}^T \Sigma^{-1}\mu}$.
In practice and with naive estimators (namely sample covariance matrix) this model has numerous drawbacks. Among them, we find:

- Degeneracy of the solution with the returns matrix is not full rank, i.e. Σ is not invertible.
- Sensitivity to small changes in the data, coming from the fact that the solution is proportional to the inverse of the eigenvalues of Σ.
- Overfitting on the data and poor performances out of sample.
- Instability of the weights of the optimal portfolio.

2.2 RMT-inspired correlation matrix filters

2.2.1 Mathematical intuition

The first filter for covariance matrices considered relies on comparing the eigenvalues of the sample covariance matrix to what the Marchenko-Pastur distribution predicts.

Definition 1. *The Marchenko-Pastur distribution: Let (X_n) be a sequence of $m \times n, m \geq n$ matrices, such that:*

- *All x_{ij} (elements of X_n) are independent*
- $\mathbf{E}(x_{ij}) = 0$ *and* $\mathbf{Var}(x_{ij}) = 1$
- $\forall n, \mathbf{E}(|x_{ij}|^k) \leq B$ *for some B independent of n*
- *m depends on n:* $\lim_{n\to\infty} \frac{n}{m} = r \leq 1$

Then, the distribution of the eigenvalues of the matrix $\frac{1}{m}X_n^T X_n$ *approaches the Marchenko-Pastur law as* $n \to \infty$*:*

$$f(x) = \frac{\sqrt{(x-(1-\sqrt{r})^2)((1+\sqrt{r})^2-x)}}{2\pi x r}$$

The sample covariance matrix is precisely of the form $\frac{1}{m}X_n^T X_n$, assuming that X_n has been de-meaned, which allows us to use the Marchenko-Pastur distribution to analyze the distribution of the eigenvalues of our matrix. Given that the spectrum of the sample covariance matrix is larger that the one of the true (population) covariance matrix (see below for more details about this observation), the assumption becomes that the spectrum of the sample covariance matrix is comprised of:

- Eigenvalues corresponding to true information
- Noise

The structure of the Marchenko-Pastur distribution allows us to extract a bound on "noise" eigenvalues, which is the upper bound of the distribution: $(1+\sqrt{r})^2$. Thus, the eigenvalues below this bound will be deemed result of pure noise, while the ones above will be considered as bearing true information.

At this stage, multiple filters can be used in order to clean the eigenvalues of the covariance matrix. A popular one is as follows (*Eigenvalue clipping*:

- Consider the eigenvalue decomposition of the sample covariance matrix
- Keep the eigenvalues above the significance threshold as wel as the eigenvectors untouched
- Replace all the "noise" eigenvalues by their sample mean

Intuitively, this filter aims at shrinking the noise eigenvalues towards a single value, while keeping the trace untouched. Another advantage of this procedure is that it makes the covariance matrix non-singular (which might not have been the case before - see following section), which numerically greatly speeds up the resolution of the problem to optimize as it has a closed form.

A second class of filters that we will be considering are the ones based on shrinkage estimators. The general form of the estimator is as follows:

$$\Sigma^{shrink.} = \alpha_s \hat{\Sigma} + (1-\alpha_s) I_n$$

with $\hat{\Sigma}$ the sample covariance matrix and α_s a parameter to estimate. Multiple ways exist to estimate this parameter, and we will present two of them.

A first consistent estimator is given by:

$$\alpha_s = 1 - \frac{\beta}{\gamma}$$

with

$$\beta = \frac{1}{N} Tr[(\hat{\Sigma} - I_N)(\hat{\Sigma} - I_N)^T]$$

$$\gamma = \max(\beta, \frac{1}{T}^2 \sum_{k=1}^{T} \frac{1}{N} Tr[(y_k y_k^T - \hat{\Sigma}) y_k y_k^T - \hat{\Sigma})^T])$$

Another RMT-based estimator of α_s is given using the Inverse-Marchenko-Pastur distribution.

Definition 2. *The Inverse-Marchenko-Pastur distribution: Starting from the Marchenko-Pastur distribution, we perform the following steps:*

- *Change of variables:* $u = \frac{1}{x(1-r)}$
- $\kappa = \frac{1}{2}(\frac{1}{r} - 1)$
- $u_{\pm} = \frac{1}{\kappa}(\kappa + 1 \pm \sqrt{2\kappa + 1})$

The density of the Inverse-Marchenko-Pastur distribution is given by ($u \in [u_-, u_+]$)*:*

$$f_{IMP}(u) = \frac{\kappa}{\pi u^2} \sqrt{(u_+ - u)(u - u_-)}$$

The estimation uses the assumption that in the linear shrinkage, the underlying covariance matrix has eigenvalues following an Inverse-Marchenko-Pastur distribution. This gives that

$$\alpha_s = \frac{1}{1 + 2r\kappa}$$

. Using the relationship between the Stieltjes transforms of the sample covariance matrix and the true covariance matrix, we find:

$$\kappa = \frac{1}{2}((1 - q)\frac{Tr(\hat{\Sigma}^{-1})}{N} - 1)$$

The main issue with this approach is the assumption that $\hat{\Sigma}$ is invertible, which might not hold in several cases in practice.

A final filtering method that we did not have the time to implement was the "Eigenvalue substitution" method: building on the eigenvalue clipping method, the idea is to "fit" the "noise" eigenvalues using the Marchenko-Pastur distribution. To do so, we define the following equation:

Definition 3. *The Marchenko-Pastur equation: for* $z \in \mathbf{C}$

$$zg_{\hat{\Sigma}}(z) = Z(z)g_{\Sigma}(Z(z))$$

with $g_A(z) = \int_I \frac{\rho_A(t)}{z-t}dt, z \in \mathbf{C} - I$ *the Stieltjes transform of the density of the eigenvalues of* A *and* $Z(z) = \frac{z}{1-r+rzg_{\hat{\Sigma}}(z)}$.

The idea is therefore is to assume some prior on the distribution $\hat{f}_{\Sigma}$ (i.e. parametrize the distribution) and use the equation to fit these parameters using Maximum Likelihood on the sample distribution of the eigenvalues of $f_{\hat{\Sigma}}$. Then, obtaining $\hat{f}_{\Sigma}$ (thus the ML estimator of the density of the eigenvalues of the population covariance matrix), we know that this density corresponds to a Marchenko-Pastur distribution (which contrasts with the density of the eigenvalues of the sample covariance matrix that follows a Wishart distribution).
Then, the substitution aims at using the (known) quantiles of the Marchenko-Pastur distribution:

$$\lambda_i^{noise} = \mu_i^{\hat{f}} with \frac{i}{N} = \int_{\mu_i^{\hat{f}}}^{\infty} \hat{f}_{\Sigma}(x)dx$$

2.3 Numerical results

See attached Julia notebook for details on the implentations of the different methods. We summarize the results in the following table (IS: In-sample, OOS: out-of-sample):

Model	IS return	IS variance	OOS return	OOS variance
Naive Markowitz	0.28	0	-0.12	0.28
Eigenvalue clipping	0.24	0.46	0.046	0.1
Linear Shrinkage	0.24	10^{-5}	-0.07	0.1

Two main observations on this table:

- As expected the Naive Markowitz as well as the RMT-Linear Shrinkage estimation both perform badly, as they try to invert the sample covariance matrix which creates instability.
- The eigenvalue clipping filter appears to perform better, while providing a crucial element of performance, consistency, as we can see it in the rather similar ratio $\frac{return}{variance}$ from IS to OOS.

3 Filtering Deep Learning weight matrices with RMT

In this second section, we will have a similar reasoning to what has been done before, this time applying it to Deep Learning. In the context of Neural Networks fitting, a known phenomenon is overfitting on the training set, which results in

very dense and noisy weight matrices for given layers. A popular solution to this issue is using dropout layers, which deactivate randomly chosen neurons by assigning them a weight of 0.
This solution has two main advantages:

- It prevents overfitting: intuitively, decreasing the number of parameters to fit decreases the number of degrees of freedom of the model which helps preventing fitting te noise in the training data.
- It increases the sparsity of the weight matrix, which helps accelerate the fitting of the model.

The idea of this section is to generalize the idea of the dropout layers: rather than randomly choosing the neurons to deactivate and entailing the risk of setting to zero a weight that bore information, we will analyze the eigenvalue distribution of the weight matrix of our network and compare it to the distribution predicted by Random Matrix Theory. This will allow us to separate between "noise" and "information" eigenvalues, and therefore give us a filtering of the weight matrix.

3.1 Mathematical intuition

3.1.1 Random matrix theory prediction

The weight matrix considered is a $n \times n$ real-valued matrix. Other than that, there are no specific structure requirements (in [?] symmetric weight matrices are explored, but we will not pursue this direction which is too technically constraining in practice).
The distribution of the eigenvalues predicted for random square matrices is the Circular distribution, that we define next.

Definition 4. *The Circular law: Let (X_n) be a sequence of $n \times n$ matrices, such that:*

- *All x_{ij} (elements of X_n) are independent*
- $\mathbf{E}(x_{ij}) = 0$ *and* $\mathbf{Var}(x_{ij}) = 1$

Then the distribution of eigenvalues of $\frac{1}{\sqrt{n}} X_n$ converge towards a uniform distribution over the unit disk in the complex plane.

3.1.2 Filtering procedure

Therefore, similarly to the previous section, we build the following procedure to filter weight matrices:

1. Sample m weight matrices of the same layer by training m independent neural networks on the same dataset.

2. For each neural network, we stop before convergence as the dropout layers typically occur at the earlier iterations of the training. Moreover, we want to keep a significant variability in the sampled weight matrices.

   We now have a set (X^k) of weight matrices.

3. Standardize each element X_{ij} of the matrices:

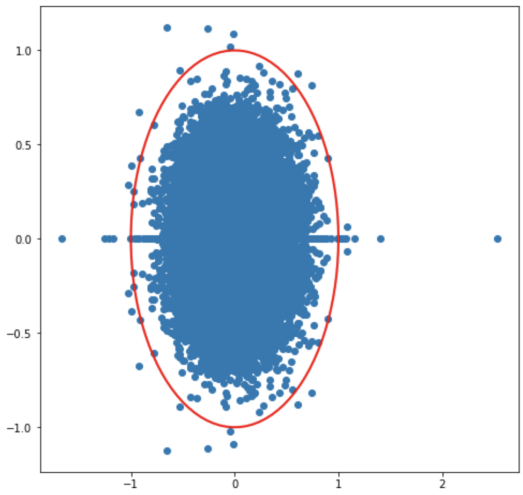
$$\forall k \in \{1, ..., m\}, X_{ij}^k = \frac{X_{ij}^k - \bar{X}_{ij}}{\frac{1}{m}\sum_{p=1}^{m}(X_{ij}^p - \bar{X}_{ij})^2}$$

4. Compute the eigenvalue decomposition of these standardized weight matrices

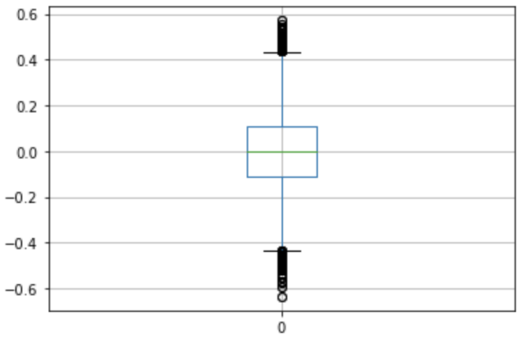
5. Filter the eigenvalues that are inside the unit circle: they are the "noise" eigenvalues

6. Reconstruct the original (non-stardard) weight matrices with the filtered eigenvalues

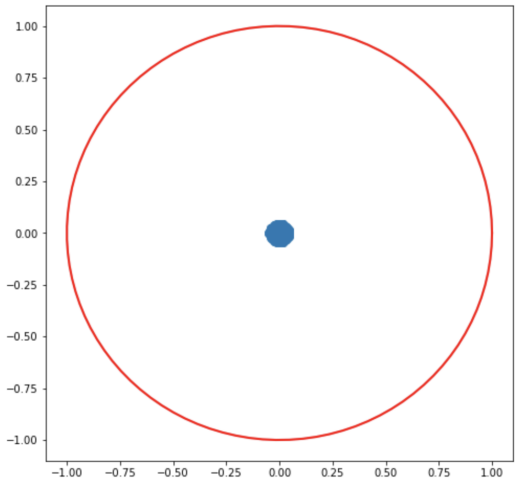
3.2 Numerical application



Eigenvalues of sampled weight matrices of first layer



Distribution of correlations between coefficients of the weight matrices



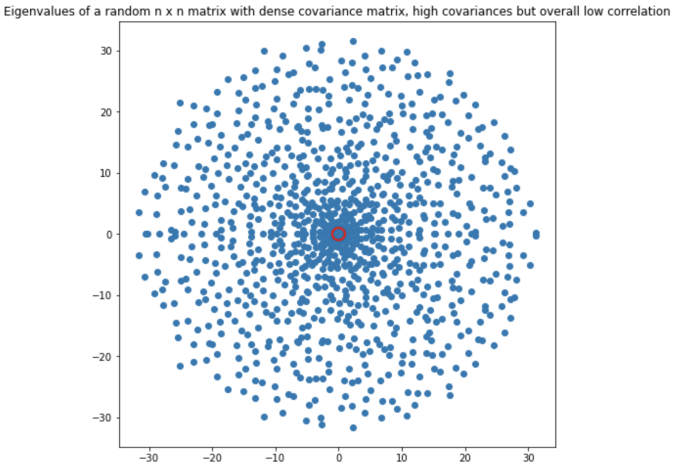
Eigenvalues of sampled weight matrices of second layer

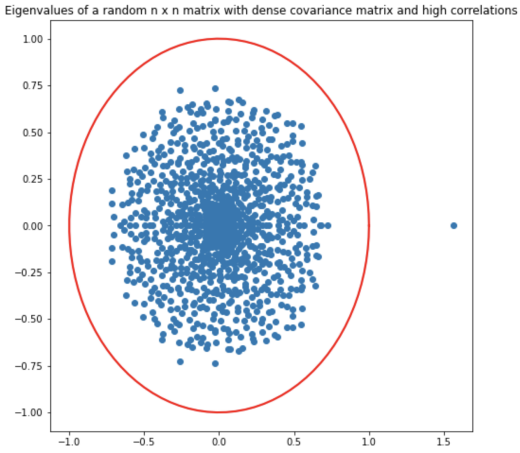
We can therefore observe three main phenomena here:

- There appears to be some dimensionality reduction mecanism at work, as we see a characteristic pattern of seemingly random eigenvalues projected on a smaller space. We attribute part of this phenomenon to correlation, as the correlation matrix of the weights seems to be dense.
- Confirming the intuition developed in the previous section, there are some eigenvalues that significantly deviate from the unit circle prediction, which might allow for some filtering to be effective.
- When looking at the spectrum of deeper layers of the neural network, we see that the dimensionality reduction is much stronger. We explain that by the "rank collapse phenomenon", common in deep learning, where the rank of weight matrices drastically decreases the deeper we go into the network.

We now try to investigate further the first pattern of eigenvalues by simulation. The idea is to find the right covariance structure that led to this plot. Due to the very large size of the weight matrices (280^2 random variables), we are unable to compute their covariance matrix and simulate data with the exact same structure. We therefore test 2 possibilities:

- Dense covariance matrix with high values but low correlations
- Dense covariance matrix with high correlations





As we see, it seems that the previous hypothesis was correct and the phenomenon is primarly due to high correlation between the weights.

4 Conclusion and further work

Random matrix theory seems to be a promising concept, as it gives a solid theoretical grounding to a number of empirical practices in finance. The concept of

filtering itself showed encouraging results as it allows to bridge the gap between the asymptotic properties and the reality of the curse of dimensionality.
Regarding further directions for this project, we can think of multiple possibilities:

- For the covariance matrix part, the idea would be to dive deeper into more advances filters, that would for instance exploit the Wishart structure of a sample covariance matrix.
- For the Neural Network part, the idea would be to actually perform the filtering: du to time constraint, we were not able to significantly test the filtering method to come up with a meaningful result for its impact on Neural networks training.
- Yet, we were able to uncover an interesting phenomenon that arises in Neural network structure. Another direction would be to further the testing of this phenomenon, and see under what conditions / what structures it arises. We note here that when tried on another dataset with random parameters for the network, we did not have such a structure. Thus, we might explore on what type of data does it happen, or if the network should be well parametrized (or at least parametrized in a certain way) to see this high correlation phenomenon.
- A final point would be exploiting this high correlation property of the weights. Some papers (such as [2]) explore imposing specific geometries on the weight matrices to decrease the number of parameters to fit.

5 References

[1] Cleaning large Correlation Matrices: tools from Random Matrix Theory, J.Bun, J.P. Bouchaud, M. Potters

Exploring Weight Symmetry in Deep Neural Networks, X. Shell Hu, S. Zagoruyko, N. Komodakis, 2019