Random Matrix Filtering: use cases in Portfolio Optimization and Neural Networks

Hussein Fellahi

December 10, 2021

General Principle of Random Matrix Filtering

Correct the estimation error in a non-asymptotic setting using RMT.
General Procedure:

- Sample estimated matrix from data and compute SVD / EVD
- Compare the eigenvalues of the estimator to the ones of a random matrix of similar structure
- Eigenvalues significantly deviating from the RMT prediction are considered bearing information, others are considered as noise
- Filter the noise eigenvalues
- Reconstruct new estimator with same eigenvectors

Portfolio Optimization in 2 slides - Part 1

Central problem of Portfolio Optimization (Markowitz, 1952):

$$\min_{w} w^T \Sigma w$$

s.t.

$$\mu^T w \geq r$$

$$1^T w = 1$$

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

Portfolio Optimization in 2 slides - Part 2

Specific interest in the tangency portofolio, that maximizes the ratio: $\frac{\mu^T w}{w^T \Sigma w}$, closed form solution under some conditions: $w^* = \frac{\Sigma^{-1}\mu}{1^T \Sigma^{-1}\mu}$
Practical issues:

- Σ can be singular (e.g. returns matrix is not full rank)
- $\hat{\Sigma}$ has bad non-asymptotic properties, especially in large scale setting
- Overfitting in-sample

Mathematical basis of Covariance matrix filtering

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
- $\mathrm{E}(x_{ij} = 0$ and $\mathrm{Var}(x_{ij}) = 1$
- $\forall n, \mathrm{E}(|x_{ij}|^k) \leq B$ 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 $n \to \infty$:

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

RMT filters of Covariance matrices

Three popular filters based on RMT:

- Eigenvalue clipping: replace "noise" eigenvalues by their sample mean
- Linear Shrinkage: $\alpha_s\hat{\Sigma} + (1-\alpha_s)I_d$ with the optimal value of α_s derived using the Cauchy transforms of Σ and $\hat{\Sigma}$
- Eigenvalue substitution: parametrize the distribution of the noise eigenvalues (e.g. power law with parameter α), fit the parameters by MLE using the relationship between the Cauchy tranforms of the density of eigenvalues of the sample covariance matrix and the true covariance then replace the sample eigenvalues by the fitted ones

Numerical experiments - tangency portfolio

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

Weight matrices filtering in Deep Learning

Applying the same filtering procedure to square weight matrices in Deep Learning.

<u>The Circular law:</u> Let (X_n) be a sequence of $n \times n$ matrices, such that:

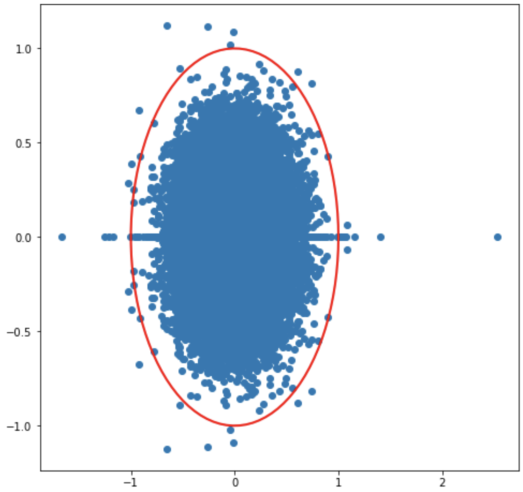
- All x_{ij} (elements of X_n) are independent
- $\mathrm{E}(x_{ij}) = 0$ and $\mathrm{Var}(x_{ij}) = 1$

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.

Sampling and filtering procedure

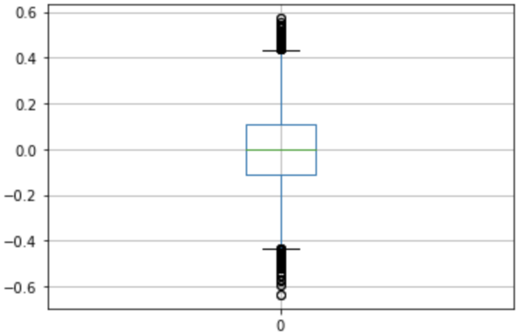
1. Sample m weight matrices of the same layer by training m independent neural networks on the same dataset.
2. For each neural network, we stop before convergence as the dropout layers typically occur at the earlier iterations of the training. Moreover, we want to keep a significant variability in the 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

Numerical experiments



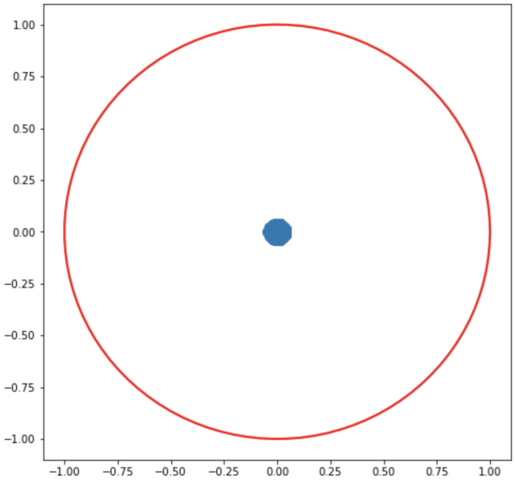
Eigenvalues of sample weight matrices

Numerical experiments



Distribution of correlations of weight matrices

Numerical experiments



Eigenvalues of sample weight matrices - 2nd layer