RIE Singular Value Cleaning of Lagged Covariance Matrices

Bryan Wei Xuan Cheong

A final year project submitted in partial fulfilment

of the requirements for the degree of

BSc Computer Science

of

University College London.

Department of Computer Science

University College London

February 7, 2025

Supervisor: Dr. Paolo Barucca

Abstract

This study introduces a novel algorithm for estimating the lagged covariance matrix of high-dimensional time-series data, applying principles from research on optimal cleaning for singular values of cross-covariance matrices [1] to the domain of lagged covariances. This approach is tested on Random Gaussian and AR(1) models, measuring its efficacy by comparing cleaned outputs with theoretical true outputs derived from autocovariance formulas.

The algorithm significantly improved the estimation of singular values for large-dimensional data. In the Random Gaussian model, it effectively removed noise, resulting in negligible error. In the AR(1) model, it maintained the signal-to-noise ratio while effectively reducing noise.

Its successful application to multiple models highlights the algorithm's versatility, making it a valuable tool for precise covariance matrix estimations across various fields, including finance and signal processing.

Acknowledgements

I am deeply grateful to my parents for their unwavering support throughout my educational journey. Their generous financial aid, boundless emotional warmth, and constant encouragement have been the cornerstones of my achievements. Without them, I would not have the opportunity to write this paper.

I owe a profound debt of gratitude to my supervisor, Dr. Paolo Barucca, whose steadfast support and mentorship have been invaluable. His ability to inspire learning, whether in the quiet of his office or amidst the lively discussions of a seminar, has enriched my academic experience immeasurably. Our discussions have not only guided my research but have also profoundly shaped my intellectual growth in unfamiliar topics.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Context

1.1 Introduction

High-dimensional statistical analysis has become increasingly relevant with the arrival of large-scale datasets. Traditional methods for estimating covariance matrices are less effective due to the curse of dimensionality [2], where empirical estimates lose their reliability as dimensions increase. Specifically, the eigenvalues of empirical covariance matrices in high dimensions tend to spread out over an interval, as predicted by the Marchenko-Pastur law [3], which can obscure the true underlying relationships between variables.

One promising approach to mitigate these issues is cleaning singular values in covariance matrices [4, 5]. This technique has proven effective for cross-covariance matrices [1], providing more accurate and stable estimates by adjusting singular values to counteract the effects of random noise.

Building on this groundwork, this paper proposes an optimised method for cleaning singular values specifically in lagged covariance matrices. These matrices are crucial for understanding the temporal dynamics in time series data, allowing us to capture time-dependent relationships that are otherwise lost in static analyses.

The need for this research is clear when considering its broad applicability. From financial markets [6, 7, 8], where accurate models of time series data can inform investment strategies, to climatology [9, 10], where lagged relationships are key to predicting weather patterns, the implications are vast [11]. This work aims to provide a robust statistical tool that enhances our capacity to make informed decisions based on time series analyses.

1.2 Context

1.2.1 Autoregressive Models

Autoregressive (AR) models are a class of linear models where the future points in a time series are predicted using a linear combination of past observations. This method of forecasting, where the variable of interest is regressed against its own previous values, is known as autoregression [12]. The self-referential nature of these models makes them especially effective for analysing and predicting time series data.

The simplest autoregressive model is the AR(0), which assumes that observations are independent and identically distributed (i.i.d.). This model is the same as a random Gaussian white noise process [13], where each value in the series is random and uncorrelated with past values:

$$X_t = \varepsilon_t, \tag{1.1}$$

where ε_t is white noise with a mean of zero and a constant variance.

The AR(1) model extends this by introducing a dependency on the immediate previous value, encapsulating a memory of one time step [12]:

$$X_t = \phi_1 X_{t-1} + \varepsilon_t, \tag{1.2}$$

where ϕ_1 is the coefficient that measures the impact of the immediate past value on the current value.

Generalising further, the AR(p) model incorporates dependencies up to p time steps back, providing a richer representation of time series dynamics [12]:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t. \tag{1.3}$$

In this equation, $\phi_1, \phi_2, \ldots, \phi_p$ are parameters quantifying the influence of past p observations on the current value.

These models are foundational in time-series analysis and particularly valuable for their ability to capture and utilise serial dependencies, characterising many temporal processes.

1.2.2 Stationarity

Stationarity refers to a condition where the statistical properties of the series—such as mean, variance, and autocorrelation—do not depend on the time at which the series is observed [14]. This trait is essential because tools like the autocovariance (Section 1.2.3) and Marchenko-Pastur law (Section 1.2.6), assume or require stationarity to function correctly. Without stationarity, trends and seasonal patterns can distort these statistical properties, leading to misleading results and interpretations [15].

This study utilises stationary autoregressive models due to their inherent ability to maintain consistent statistical properties across time:

1. **AR(0):** Inherently stationary as it consists of i.i.d. random variables.

2. **AR(1):** Stationary provided that $|\phi| < 1$, preventing the series from exhibiting explosive behaviour which could vary mean and variance unpredictably [12].

3. **AR(p):** Stationary when all roots of its characteristic polynomial lie outside the unit circle such that the model's parameters $\phi_1, \phi_2, \ldots, \phi_p$ stabilise the process [16].

To check if a time series is stationary, analyse summary statistics or perform statistical tests like the Augmented Dickey-Fuller test [15].

On the other hand, non-stationary time series require transformation to meet the stationarity conditions for analysis. Techniques employed to make them stationary include:

1. **Differencing:** This involves subtracting an observation from its predecessor, typically used as a component to remove trends or seasonality [14].

2. **Detrending:** This involves fitting a linear model to remove a trend from the data, leaving the residuals to analyse [17].

3. **Deseasonalising:** This technique adjusts for seasonal variations by model fitting or differencing, allowing for a more stable analysis of the underlying patterns [18].

Implementing a combination of these preprocessing steps ensures that the assumptions of many time series models are met, enabling more accurate and reliable analysis.

1.2.3 Autocovariance

Autocovariance measures the linear dependency of a time series with itself at different points in time, separated by a lag τ. It is an essential tool for determining the time-dependent structure within a series, crucial for models like AR where the influence of past values dictates current behaviour. Mathematically, the autocovariance for a stationary time series X_t at lag τ is defined as [19]:

$$\gamma(\tau) = \mathbb{E}[(X_t - \mu)(X_{t-\tau} - \mu)] \tag{1.4}$$

This expression, $\mathbb{E}$ represents the expected value operator. For stationary processes, such as AR(0) and AR(1) with $|\phi| < 1$, the mean μ is invariant over time. Simplifying the autocovariance function to be equivalently expressed as:

$$\gamma(\tau) = \mathbb{E}[X_t X_{t-\tau}] - \mu^2 \tag{1.5}$$

A pronounced autocovariance at a specific lag indicates a significant relationship between values separated by that time interval. In Chapter 2, new forms of autocovariance functions for AR(0) and AR(1) models will be examined.

1.2.4 Cross-Covariance Matrix

Cross-covariance matrices are crucial in assessing the relationship between two time-dependent random variables. For a series of observations over time indexed by $t = 1, \ldots, T$, the empirical cross-covariance matrix $\mathbf{C_{XY}}$ between the vectors $\mathbf{X}_t \in \mathbb{R}^n$ and $\mathbf{Y}_t \in \mathbb{R}^p$ is calculated as follows [19]:

$$\mathbf{C_{XY}} = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{X}_t - \overline{\mathbf{X}})(\mathbf{Y}_t' - \overline{\mathbf{Y}}) \quad \text{where} \quad \mathbf{Y}_t' = \text{transpose of } \mathbf{Y}_t \tag{1.6}$$

where $\overline{X}$ and $\overline{Y}$ are the means of X and Y respectively.

The primary use of a cross-covariance matrix is quantifying how changes in variables of one vector (like $\mathbf{X}$) are associated with changes in variables of another vector (like $\mathbf{Y}$). For example, consider two vectors where $\mathbf{X}$ represents n distinct features in the USD/EUR exchange rate, and $\mathbf{Y}$ represents p distinct features in the USD/GBP exchange rate over the same period. These features could contain various market indicators, like the daily exchange rate, opening price, and trading volume. The cross-covariance matrix computed between $\mathbf{X}$ and $\mathbf{Y}$ would provide insight into the interdependencies between these features.

For datasets with a large number of variables (n and p), cross-covariance matrices become essential tools for understanding the inter-variable relationships at a granular, quantifiable level. They help identify patterns and dependencies that might not be evident through simpler analyses.

1.2.5 Lagged Covariance Matrix

Lagged covariance matrices extend the concept of cross-covariance to include temporal dynamics, crucial for capturing how past values influence future observations in time series data. Unlike cross-covariance matrices that measure simultaneous interactions between different datasets, lagged covariance measures how past values of a series influence its future values. The empirical method of calculating lagged covariance between a time series $\mathbf{X}_t \in \mathbb{R}^n$ and its own past values $\mathbf{X}_{t-\tau} \in \mathbb{R}^n$ over time $t = 1, \ldots, T$ is defined as:

$$\mathbf{C}_{\mathbf{X}\tau} = \frac{1}{T-\tau} \sum_{t=1}^{T-\tau} (\mathbf{X}_t - \overline{\mathbf{X}})(\mathbf{X}_{t-\tau} - \overline{\mathbf{X}})' \tag{1.7}$$

Here, $\mathbf{X}_{t-\tau}$ represents the state of the time series τ periods in the past, highlighting the delayed effects within the series.

These matrices are fundamental for understanding temporal patterns like cycles, trends and seasonality and can significantly enhance predictive accuracy. By calculating the lagged covariance for different values of τ, analysts can understand how much historical data can inform future expectations and identify the optimal lags to be included in predictive models.

1.2.6 Marchenko-Pastur Law

The Marchenko-Pastur Law predicts the distribution of eigenvalues for large sample covariance matrices when the number of variables n and the number of observations T are both large. When the parameter n/T, termed the aspect ratio, approaches a finite limit, the eigenvalues spread out over a region instead of clustering at the true eigenvalue.

For example, consider the empirical covariance matrix derived from a sample of T independent observations of an n-dimensional Gaussian signal, where the **true** underlying covariance is the identity matrix I_n. In low-dimensional settings, where $T >> n$, the singular values of the empirical covariance matrix tend to cluster around the value of 1, reflecting the true covariance structure accurately. However, as the dimensionality n increases and becomes similar to the number of observations T, the behaviour of the singular values diverges significantly from this ideal scenario. Instead of clustering around 1, the singular values spread out and follow a distinct distribution known as the Marchenko-Pastur law.

The aspect ratio n/T serves as a crucial element in determining the spread of eigenvalues as described by the Marchenko-Pastur Law. An aspect ratio close to zero suggests that the number of observations far exceeds the number of variables, leading to a tight clustering of eigenvalues. Conversely, an aspect ratio approaching or exceeding one indicates that the dimensions are on par with the number of observations, leading to the aforementioned spread and necessitating techniques such as singular value cleaning to mitigate its effects on covariance estimation [3].

1.2.7 Singular Values and Singular-Value-Decomposition (SVD)

Singular Value Decomposition (SVD) is a fundamental matrix factorisation technique used in many areas of statistics and machine learning for dimensional reduction and data simplification. SVD decomposes a matrix A into three other matrices [20]:

$$A = U\Sigma V^*, \tag{1.8}$$

where U and V are orthogonal matrices representing the left and right singular vectors of A, and Σ is a diagonal matrix containing the singular values of A. Singular values in Σ quantify the contribution of each corresponding singular vector to the overall structure of the data.

In the context of covariance matrix analysis, especially for lagged covariance matrices, SVD plays a crucial role. It helps identify and retain significant patterns in the data while reducing noise. Adjusting the singular values during the cleaning process, as in the technique proposed in this paper, optimises the representation of these patterns, ensuring that the essential temporal dynamics are preserved while minimising the influence of noise and redundancies [1].

This enhanced representation through SVD is pivotal for applications that rely on accurate covariance estimations, such as predictive modelling and time series forecasting in high-dimensional spaces.

1.3 Literature Review

1.3.1 Benaych-Georges et al. Paper

The pioneering work by Benaych-Georges et al. [1] introduced an optimal cleaning technique for the singular values of cross-covariance matrices, providing a robust framework for handling high dimensional datasets. This paper introduces a new algorithm that cleans singular values of cross-covariance optimally for the Frobenius norm among RIEs. The method effectively approximates the oracle functions from its data, outperforming traditional estimators in high-dimensional, complex scenarios.

1.3.2 Oracle Function and Oracle Estimators

Oracle functions [1] represent idealised functions that provide the best possible outcomes, assuming complete knowledge of the underlying processes. These functions often remain theoretical, as they depend on unknown parameters, making them impractical for direct use.

Conversely, oracle estimators [5] are practical implementations designed to approximate the performance of oracle functions. They leverage estimable data quantities to simulate the decision-making of an oracle function, providing researchers with a robust framework for handling complex datasets.

Additionally, oracle functions can be used as benchmarks in statistical analysis, allowing researchers to measure the efficacy of practical estimators against theoretical results. This evaluation involves models specifically designed to access oracle functions, providing an idealised baseline to compare with empirical outcomes.

1.3.3 Isotropic Vectors

Before discussing Rotationally Invariant Estimators (RIEs), it's crucial to understand the concept of isotropic vectors. In the context of covariance matrix estimation, isotropic vectors refer to vectors whose distribution is uniform and symmetric across all directions in space. This isotropy means that the properties of these vectors remain unchanged regardless of the dataset's rotation or transformation. Such a characteristic is vital for RIEs, as these estimators rely on the assumption that the principal directions (singular vectors) of the data should not influence the estimation process, thus ensuring a fair and unbiased representation of the dataset's intrinsic characteristics.

1.3.4 Rotationally Invariant Estimators

Enhancing covariance matrix estimation in high-dimensional data often requires techniques beyond traditional approaches, due to the unique challenges posed by large datasets. Nowadays, there are several methods used to improve estimation accuracy:

- **Regularization** adapts to structured covariance matrices, integrating assumed prior structural knowledge into the estimation process. Band matrix techniques are discussed in [21, 22], while hard thresholding is explored in [10].

- **Shrinkage** combines empirical covariance estimates with a structured prior (e.g. a scaled identity matrix), adjusting overestimations and underestimations towards a more reliable middle ground. This method enhances stability and accuracy by balancing raw data with predefined assumptions [23, 24, 25].

- **Clipping** adjusts all but the largest eigenvalues to a constant value to preserve certain matrix properties, such as the trace.

- **Cleaning** optimises singular values to align the estimated matrix more closely with the true underlying matrix—this is our main focus in this paper.

Among these, Rotationally Invariant Estimator (RIE) cleaning is notable for preserving the singular vectors while adjusting singular values. This preservation is essential in high-dimensional statistics to ensure directional data remains unaltered while enhancing the clarity of dimensional reduction.

It works by ensuring estimators are invariant under orthogonal transformations, thus pre-

serving the isotropic nature of singular vectors (unlike regularization). This isotropic property makes RIEs ideal for unbiased data representation, particularly in SVD, it is said to enhance the signal-to-noise ratio, ensuring a clearer depiction of principal components.

Overall, RIEs play a crucial role in balancing the preservation of geometric data structure with statistical accuracy, particularly beneficial in complex analytical scenarios where underlying data structures are key but may be obscured by noise or dimensionality.

1.3.5 Transition to Lagged Covariance Matrix

This study builds upon the foundational insights provided by Benaych-Georges et al. (2019), investigating lagged covariance matrices instead of cross-covariance matrices. In the context of lagged covariance matrices, the relationship between a time series is denoted as $\mathbf{X}_t$ and its lagged version $\mathbf{X}_{t-\tau}$. This is conceptually similar to cross-covariance matrices, where C_{XY} captures the relationship between different sets of data; however, for lagged covariance, $\mathbf{Y}_t$ is effectively $\mathbf{X}_{t-\tau}$.

The similarity between cross-covariance and lagged covariance matrices lays the groundwork for extending optimal cleaning techniques, initially developed for the former, to the realm of the latter. This extension is grounded in the mathematical principles common to both cross-covariance and lagged covariance matrices. By applying established optimal cleaning techniques, originally designed for cross-covariance matrices, to lagged covariance matrices, with the aim to significantly improve the estimation accuracy of these matrices in high-dimensional time-series data.

1.3.6 Impact of Lags on Dimensionality

To understand the impact of lags on the dimensionality of covariance matrices in time series analysis, consider a basic scenario with n variables without any lags. In this instance, the covariance matrix dimensions are $n \times n$, reflecting the pairwise covariances between each pair of variables at the same time point, also known as autocovariance.

Then, introduce L lags for each variable, which effectively augment each variable to be represented by $L+1$ distinct data points—the original variable plus L lagged versions. This results in a more comprehensive and granular covariance matrix that captures the temporal dynamics. The dimensions of this lagged covariance matrix are $\mathbf{n}(\mathbf{L}+\mathbf{1}) \times \mathbf{n}(\mathbf{L}+\mathbf{1})$ instead.

Therefore, as the number of lags increases, the increase in dimensionality grows linearly per variable but results in a quadratic increase in the matrix's overall size.

1.4 Thesis Contribution

This thesis contributes to the field by developing, validating, and evaluating an optimised method for cleaning singular values of lagged covariance matrices. The approach leverages RIEs, a breakthrough discovery in matrix analysis, to enhance the robustness and accuracy of time-series data analysis. The focus on lagged covariance matrices addresses a critical gap in understanding temporal dynamics, particularly in financial time series, where lagged relationships can signify predictive patterns. By improving the estimation of these matrices, the research aims to provide more reliable tools for forecasting and understanding dynamic systems.

Chapter 2

Methodology

2.1 Model Analysis

This section analyses the subtle differences in algorithm 1 in [1] that arise when transitioning to lagged covariance matrices. This will result in a modified algorithm tailored for singular value cleaning of the lagged covariance matrix.

Differences

1. Instead of random vectors $(X,Y) \in \mathbb{R}^n \times \mathbb{R}^p$, they are instead $(X_t, X_{t-\tau}) \in \mathbb{R}^n \times \mathbb{R}^n$ where τ is the lag.

2. The $b_{[n+1:p]}$ term [1] is always omitted as both singular vectors have the same dimension, which means $p = n$ so $b_{[n+1:p]} = b_{[n+1:n]} = 0$. Therefore, Equation 2.4 is simplified from:

$$\frac{1}{T}\left(\sum_{\ell=1}^{n} \frac{b_\ell}{z^2 - s_\ell^2} + z^{-2} b_{[n+1:p]}\right) \quad \text{to} \quad \frac{1}{T}\left(\sum_{\ell=1}^{n} \frac{b_\ell}{z^2 - s_\ell^2}\right)$$

   as the $b_{[n+1:p]}$ term is omitted.

3. Apply isotonic regression only when modelling dependent variables with a non-decreasing relationship with the independent variables. In the lagged case, an example would be modelling cumulative metrics over time where later time points are not expected to have lower values than earlier ones.

2.2 Algorithm

The algorithm for optimally cleaning lagged covariance matrices aims to adjust the singular values obtained from the SVD of the lagged covariance matrix $C_{X_{t-\tau}X}$. The process ensures that the cleaned singular values better represent the true underlying structure of the time series data.

Summary: RIE Cleaning for Singular Values of Lagged Covariance Matrices

Input: $X \in \mathbb{R}^{n\times T}$, lag τ.

Output: RIE cleaned singular values $s_{1,\text{cleaned}}, \ldots, s_{n,\text{cleaned}}$.

1: **Compute** $\mathbf{C}_{\mathbf{X}_{\mathbf{t}-\tau}\mathbf{X}} = \frac{1}{T} X_{t-\tau} X'$, $\mathbf{C}_{\mathbf{X}} = \frac{1}{T} XX'$, $\mathbf{C}_{\mathbf{X}_{\mathbf{t}-\tau}} = \frac{1}{T} X_{t-\tau} X'_{t-\tau}$

2: **Compute** the SVD (see Equation 1.8) of $C_{X_{t-\tau}X}$

3: **Compute** the vectors $(a_\ell)_{\ell=1,\ldots,n}$ and $(b_\ell)_{\ell=1,\ldots,n}$:

$$a_\ell = u'_\ell C_X u_\ell, \quad b_\ell = v'_\ell C_{X_{t-\tau}} v_\ell \tag{2.1}$$

where u_ℓ and v_ℓ are the left and right singular vectors of $C_{X_\tau X}$ respectively.

4: For each $k \in \{1, \ldots, n\}$:

- Set $z = s_k + i(npT)^{-1/12}$ for s_k the k-th singular value of $C_{X_{t-\tau}X}$.
- Compute H, A, B:

$$H(z) = \frac{1}{T} \sum_{\ell=1}^{n} \frac{s_\ell^2}{z^2 - s_\ell^2} \tag{2.2}$$

$$A(z) = \frac{1}{T} \sum_{\ell=1}^{n} \frac{a_\ell}{z^2 - s_\ell^2} \tag{2.3}$$

$$B(z) = \frac{1}{T} \sum_{\ell=1}^{n} \frac{b_\ell}{z^2 - s_\ell^2} \tag{2.4}$$

- Compute $\Theta = \frac{z^2 AB}{1+H}$ and $L = 1 - \frac{1}{1+H-\Theta}$
- Compute $s_{k,\text{cleaned, algo}} = s_k \times \left(max\{0, \frac{\Im(L)}{\Im(H)}\} \right)$

5: Optionally, apply the isotonic regression algorithm to the $s_{k,\text{cleaned, algo}}$.

Then, the following sections will be the models used to generate time series data to validate the efficacy of our algorithm.

2.3 Random Gaussian Model or AR(0)

2.3.1 Purpose

The random Gaussian (RG) model, also referred to as AR(0), where $X_t = \varepsilon_t$ (see Equation 1.1), serves as a fundamental baseline for comparative analysis. The model serves two purposes in our analysis.

Firstly, it acts as a validation tool, providing a scenario with no autocorrelation, allowing for a clear assessment of the cleaning method's effectiveness. Secondly, it sets a benchmark for evaluating the cleaning process's performance against more complex autoregressive structures that exhibit inherent autocorrelation. Moreover, its true lagged covariance is easy to calculate to compare with empirical and cleaned methods.

2.3.2 Process

Five multivariate RG time series will be generated with parameters varying by dimensions n and keeping the total time steps T constant, lagged by τ time steps. Their lagged covariance matrix will be calculated using Equation 1.7, it will also be cleaned using Algorithm 1, and both empirical and cleaned matrices will be compared to the true matrix. The following table summarises the parameters used in the analysis:

n	T (Total Time Steps)	τ (Lag)	$Q = T/n$ (Ratio)
250	2,500	1	10
500	2,500	1	5
1,250	2,500	1	2
2,500	2,500	1	1

Table 2.1: Parameters used in the generation of Gaussian time series

Then, the time series will be generated, and the empirical and true covariance matrix will be calculated:

For each row of parameters:

1. Generate an n-dimensional time series representing a white noise process:

```
X = np.matrix(np.random.randn(n, T))
```

2. Create a lagged series $X_{t-\tau}$, where $\tau = 1$, and match the dimensions of the original matrix by slicing the matrix X:

```
XT = X[:, 1:]
XT_L1 = X[:, :-1]
```

3. Stack the original and lagged matrices to form a combined matrix Z:

$$Z = \begin{bmatrix} \texttt{XT} \\ \texttt{XT\_L1} \end{bmatrix},$$

```
Z = np.vstack((XT, XT_L1))
```

4. Apply mean centering to Z by subtracting the mean of each row from itself:

```
Z = Z - np.mean(Z, axis=1)
```

5. Calculate the empirical lagged covariance matrix of Z:

$$Z = \begin{bmatrix} C_{X_t X_t} & C_{X_{t-\tau} X_t} \\ C_{X_t X_{t-\tau}} & C_{X_{t-\tau} X_{t-\tau}} \end{bmatrix}$$

```
lagged_cov_matrix = (Z @ Z.T) / (T - 1)
```

6. Return the true singular values which is an n-dimensional vector of zeros and the empirical lagged covariance matrix:

```
return ([0] * n, lagged_cov_matrix)
```

7. Proceed to the main algorithm in Section 2.2 and split the block matrix `lagged_cov_matrix` into its submatrices and clean the singular values.

8. Analyse the results, which will be elaborated in Section 2.3.3 and Chapter 3.

2.3.3 Expected Results

The anticipated outcomes for this model analysis involve two categories of submatrices within the lagged covariance matrix derived from the stacked matrix Z.

1. The top-left and bottom-right submatrices, $C_{X_tX_t}$ and $C_{X_{t-\tau}X_{t-\tau}}$, are expected to be diagonal matrices with ones along the diagonal, reflecting the unit variance of the Gaussian process. The zeros off-diagonal indicate no autocorrelation within each independent time series. This setup validates the lack of internal correlation in a RG or AR(0) process.

2. The top-right and bottom-left submatrices, $C_{X_{t-\tau}X_t}$ and $C_{X_tX_{t-\tau}}$, should ideally be zero matrices, showing no correlation between the current values X_t and its lagged version $X_{t-\tau}$. Empirically, as the dimension n increases, the noise levels in these matrices may also increase due to the higher dimensionality (see Section 1.2.6). After applying the cleaning, it is expected to see a less noisy version of the empirical matrix.

The fact that the autocovariance function for a white noise process is [26, 27]:

$$\gamma(\tau) = \mathbb{E}[X_tX_{t-\tau}] = \sigma^2\delta_{\tau,0} \equiv \begin{cases} \sigma^2 & \text{for } \tau = 0 \\ 0 & \text{for } \tau > 0 \end{cases} \tag{2.5}$$

further supports the expected outcome of the two types of submatrices.

The effectiveness of this model should be the most noticeable because there are no inherent signals or correlations by design. Thus, any significant reduction in noise or improvement in the clarity of the empirical matrices can be attributed directly to the method's efficacy. Furthermore, the model ensures that the cleaning process does not introduce any unwarranted signals, which is vital for applications involving more complex time series datasets where maintaining data integrity is crucial.

2.4 AR(1) Model

2.4.1 Purpose

The AR(1) model (see Equation 1.2), plays a critical role in advancing our understanding of cleaning lagged covariance matrices from autoregressive processes within time series. This model introduces a simple yet significant autocorrelation structure, where each value in the series is linearly dependent on its immediate predecessor, accompanied by a Gaussian noise component ε_t.

Validation of Cleaning Methods The AR(1) model serves as an essential tool for validating the effectiveness of our proposed cleaning methods under the simplest conditions of inherent temporal correlation. Unlike the AR(0) model, which provides a baseline with no autocorrelation, the AR(1) model allows us to test the robustness of our cleaning techniques in the presence of known, systematic correlations. This is pivotal for ensuring the cleaning process accurately preserves meaningful temporal dynamics while reducing noise.

Benchmark for Complex Models Furthermore, the AR(1) model sets a foundational benchmark for comparing the performance of our cleaning algorithm against more complex autoregressive models. Starting with this first-order correlation allows for methodically assessing how well the cleaning process manages to extract and clarify the true signal from the noise, which is crucial for models where higher-order lags and more intricate dynamics are considered, like VAR and AR(p) models, and real-world data.

Parameterisation and Comparative Analysis Use the parameters outlined in Table 2.1 to compare the empirical and cleaned matrices with the true matrix. The difference is that the true matrix is not as simple as a zero matrix—the method for calculating the true matrix is outlined in this paragraph in Section 2.4.2.

2.4.2 Process

Our first step is generating n number of random ϕ values, where each ϕ represents the autoregressive coefficient for an individual AR(1) process within our model. This coefficient is crucial as it determines the degree of influence that the previous time point's value will have on the current value in the time series. Outlined below is the process of generating the time series and calculating the empirical and true covariance matrices:

For each row of parameters in Table 2.1:

1. Initialise two empty matrices to hold the original and lagged time series data:

```
XT_matrix = np.zeros((n, T - 1))
XT_L1_matrix = np.zeros((n, T - 1))
```

2. Generate the AR(1) process coefficients ϕ, note that $|\phi| < 1$ ensures stationarity (Section 1.2.2):

```
phi_values = np.random.uniform(0.0, 0.9, size=n)
```

3. Generate the true singular values:

```
sigma_epsilon_squared = 1
true_s = [(phi * sigma_epsilon_squared) / (1 - phi^2)
          for phi in phi_values]
```

4. For each variable *i*, simulate an AR(1) process and store the original and lagged series:

```
for i, phi in enumerate(phi_values):
    ar1 = np.array([1, -phi])
    ma1 = np.array([1])
    AR_object = ArmaProcess(ar1, ma1)
    X = AR_object.generate_sample(nsample=T)

    XT_matrix[i, :] = X[1:]
    XT_L1_matrix[i, :] = X[:-1]
```

5. Stack the original and lagged matrices to form the combined matrix Z:

```
Z = np.vstack([XT_matrix, XT_L1_matrix])
```

6. Apply mean centering to the combined matrix Z:

```
Z = Z - Z.mean(axis=1)[:, np.newaxis]
```

7. Calculate the empirical lagged covariance matrix from the combined matrix Z and ensure it is in `numpy.matrix` type for the compatibility of the algorithm:

```
lagged_cov_matrix = np.asmatrix((Z @ Z.T) / (T - 1))
```

8. Return the theoretical list of singular values derived from the autocovariance formula and the empirical lagged covariance matrix:

   ```
   return (true_s, lagged_cov_matrix)
   ```

9. Proceed to the main algorithm in Section 2.2 and split the block matrix `lagged_cov_matrix` into its submatrices and clean the singular values.

10. Analyse the results, which will be elaborated in Section 2.4.3 and Chapter 3.

Theoretical True Matrix and Singular Values The autocovariance function for an AR(1) model has a direct relationship with the parameter ϕ, proven here [28], the autocovariance at lag τ is given by:

$$\gamma(\tau) = \sigma_\varepsilon^2 \frac{\phi^{|\tau|}}{1-\phi^2} \quad \text{for } |\phi| \leq 1, \tag{2.6}$$

where $\mu = 0$ and ϕ is the autoregressive coefficient.

This function remains valid for negative lags (i.e. $X_{t-\tau}$), due to the symmetry of the covariance function $\gamma(\tau) = \gamma(-\tau)$ in a stationary time series and X must be real [19, 27]. Therefore, it is important to note that Equation 2.6 uses the absolute value of τ to reflect this symmetry explicitly.

This forms the basis for constructing the theoretical true lagged covariance matrix for n independent AR(1) processes, denoted as $\mathbf{C}^{\text{true}}_{X_{t-\tau}X_t}$, which is constructed by placing these autocovariance values calculated for each ϕ value (see Equation 2.6) along the diagonal elements of the matrix:

$$\mathbf{C}^{\text{true}}_{X_{t-\tau}X_t} = \begin{bmatrix} \gamma(\tau)_1 & 0 & \cdots & 0 \\ 0 & \gamma(\tau)_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma(\tau)_n \end{bmatrix}, \tag{2.7}$$

where $\gamma(\tau)_n = \sigma_\varepsilon^2 \cdot \phi_n/(1-\phi_n^2)$.

Given that the off-diagonal elements are zero, indicating no cross-correlations between different time series and the autocovariance values on the diagonal are positive because ϕ is ensured to be positive by design. The singular values of $\mathbf{C}^{\text{true}}_{X_t - \tau X_t}$ are equivalent to its diagonal entries [20].

These true singular values can be used as a benchmark for evaluating empirical and cleaned singular values derived from AR(1) models. When sorted from largest to smallest, these true singular values serve as a prior for assessing the effectiveness of cleaning methods, allowing for a clear comparison of performance.

To compare lagged covariance matrices, these three matrices will be used:

1. **Theoretical True Matrix:** Matrix 2.7.

2. **Empirical Matrix:** Derived from the empirical formula 1.7.

3. **Cleaned Matrix:** Reconstructed using the cleaned singular values, along with the appropriate right singular vectors (U) and left singular vectors (V).

This reconstruction allows comparison directly between the true, empirical, and cleaned matrices.

This structured approach directly ties autocovariance to singular values in the true lagged covariance matrix, which is crucial for validating singular value cleaning algorithms in time series analysis. By establishing a clear and accurate benchmark for evaluating cleaning methods, it becomes possible to effectively gauge their impact and determine their efficacy in enhancing empirical data.

2.4.3 Expected Results

The primary expectation from applying the singular value cleaning to the AR(1) model is an enhancement in signal clarity along the diagonal and noise reduction, particularly in the off-diagonal elements of the lagged covariance matrices. These improvements are anticipated to align the cleaned matrices more closely with the theoretical true lagged covariance matrix, which has autocovariance values precisely positioned on its diagonal and zeros elsewhere.

Quantitatively, this enhancement should be evident through metrics such as the Frobenius norm, which is expected to show a marked decrease between the empirical and cleaned matrices compared to the theoretical values. This reduction indicates an effective noise reduction, especially crucial as the dimensionality n of the model increases. Visual tools, including heatmaps and singular value plots, will offer intuitive proof of reduced noise and enhanced structural definition, affirming the method's utility in improving the accuracy and reliability of time-series analysis in high-dimensional settings.

Chapter 3

Results

3.1 Random Gaussian (RG) Model Results

n	True vs Emp	True vs Clean	True vs Iso Clean
250	4.9677	0.0247	0.0146
500	10.030	0.1160	0.0890
1,250	24.985	0.1564	0.1063
2,500	50.004	0.3459	0.2231

Table 3.1: RG's Table of Frobenius Norm Differences with $T = 2,500$

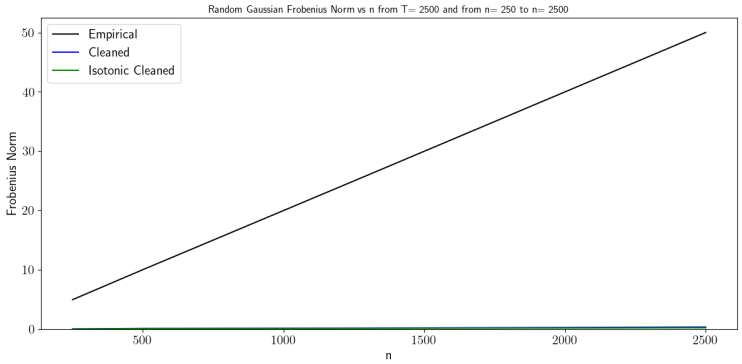

Figure 3.1: RG's Frobenius Norm against Dimensions from n = 250 to n = 2,500

The data in Table 3.1 and Figure 3.1 show that, as the dimensionality n increases, the cleaned matrices display progressively better noise suppression compared to the empirical matrices, highlighting the effectiveness of the cleaning approach for high-dimensional data. The empirical matrices exhibit a consistent increase in Frobenius norm, suggesting a linear relationship with n that is likely to persist as dimensionality further increases.

3.1.1 Histogram Analysis

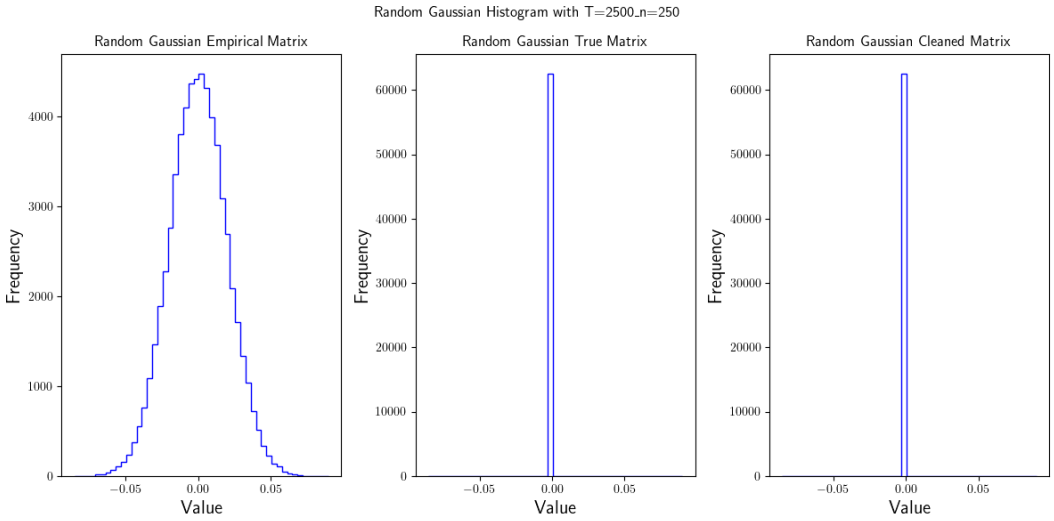

Figure 3.2: RG's Histogram with n = 250 and T = 2,500 (Left to Right: Empirical, True, Cleaned)

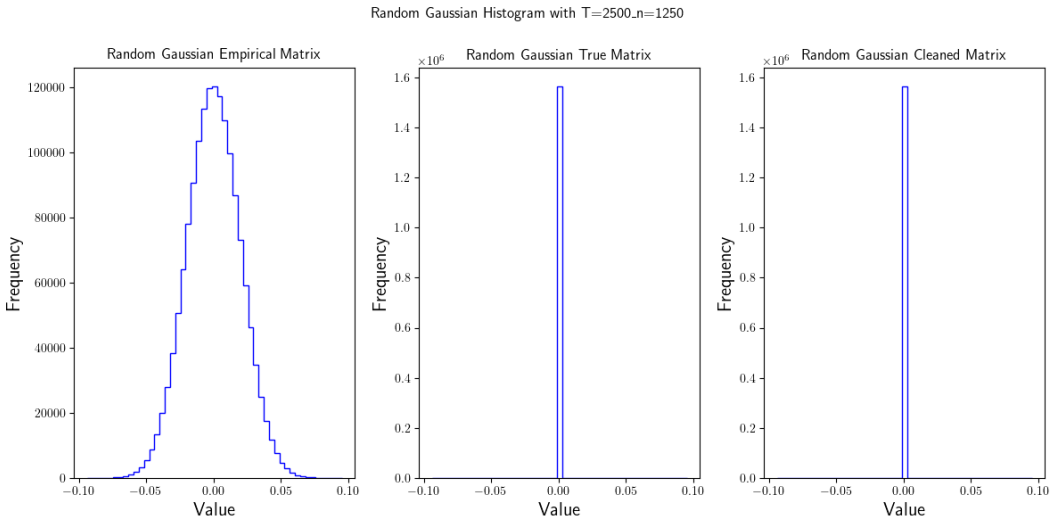

Figure 3.3: RG's Histogram with n = 1,250 and T = 2,500 (Left to Right: Empirical, True, Cleaned)

Figures 3.2 and 3.3 show the histograms of the matrix elements. The cleaning process shrunk the values in the matrix into zeros, from what was a Gaussian distribution. This aligns much more closely with the true matrix values, hence the low Frobenius norms in Table 3.1.

3.1.2 Matrix Heatmap Analysis

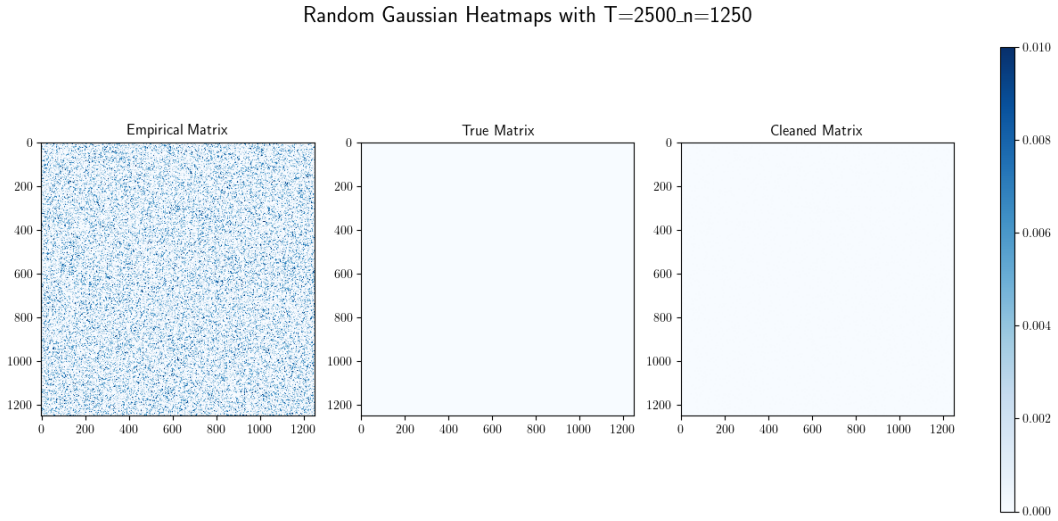

Figure 3.4: RG Heatmap with n = 1,250 and T = 2,500 (Left to Right: Empirical, True, Cleaned)

Figure 3.4 shows a heatmap for each method, representing a substantial decrease in noise after cleaning the singular values and reconstructing the matrix, aligning more closely to the true matrix.

3.2 AR(1) Model Results

n	True vs Emp	True vs Clean	True vs Iso Clean
250	12.561	11.683	11.676
500	23.594	20.624	20.613
1,250	62.527	48.241	48.130
2,500	124.504	85.751	85.306

Table 3.2: Frobenius Norm of the Differences for AR(1) Model with $T = 2,500$

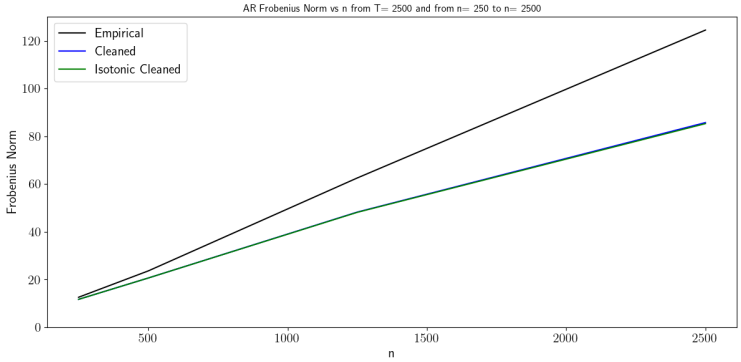

Figure 3.5: AR(1) Model's Frobenius Norm against Dimensions from n = 250 to n = 2,500

Table 3.2 and Figure 3.5 show consistently lower Frobenius norms for the cleaned matrices than the empirical matrices. It also has a decreasing gradient, indicating that the cleaning is more effective as the dimensions n increase. The empirical matrix continues to have a constant, increasing gradient in its Frobenius norm.

3.2.1 Singular Values Line Graph Analysis

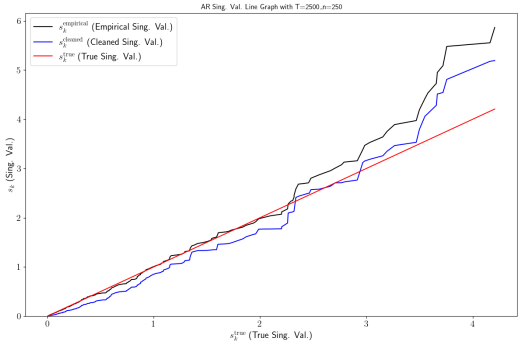

Figure 3.6: AR(1) Singular Values Line Graph when n = 250

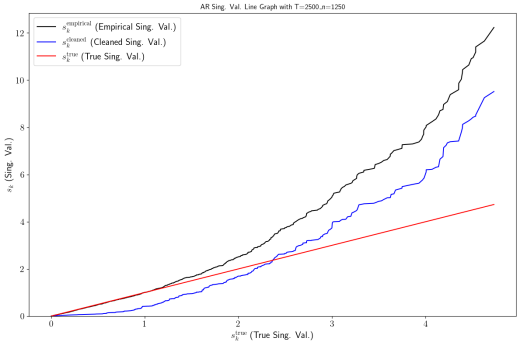

Figure 3.7: AR(1) Singular Values Line Graph when n = 1,250

Figures 3.6 and 3.7 present line graphs comparing the singular values from empirical and cleaned matrices against the true singular values for AR(1) models at dimensions $n = 250$

and $n = 1,250$ respectively. The x-axis reflects the true singular values, creating a reference line at $y = x$, indicated in red, which serves as a benchmark for comparison.

The black line (empirical) deviates from the red line more than the blue line (cleaned), and this deviation increases at higher dimensionality $n = 1,250$. This closer adherence to the true singular values by the cleaned data shows the effectiveness of the cleaning process, particularly in higher-dimensional contexts.

3.2.2 Matrix Heatmaps and Difference Heatmaps

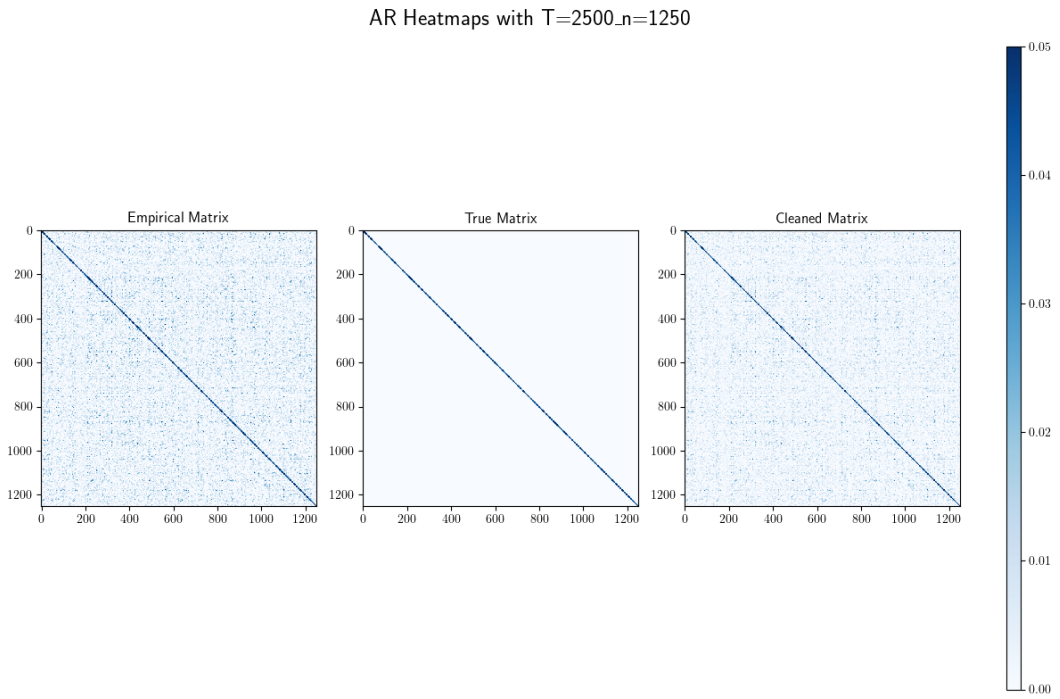

Figure 3.8: AR(1) Heatmap with n = 1,250 and T = 2,500 (Left to Right: Empirical, True, Cleaned)

The heatmaps in Figure 3.8 for an AR process with $n = 1,250$ reveal the characteristic diagonal line across empirical, true, and cleaned matrices, signifying preserved structural features corresponding to the theoretical model in Matrix 2.7. When zoomed in, the diagonal elements across methods show similar colours, implying similar values when referenced against the colour bar. Most notably, the cleaned matrix exhibits marginally reduced noise compared to the empirical matrix, which suggests improvements in noise reduction through the cleaning process.

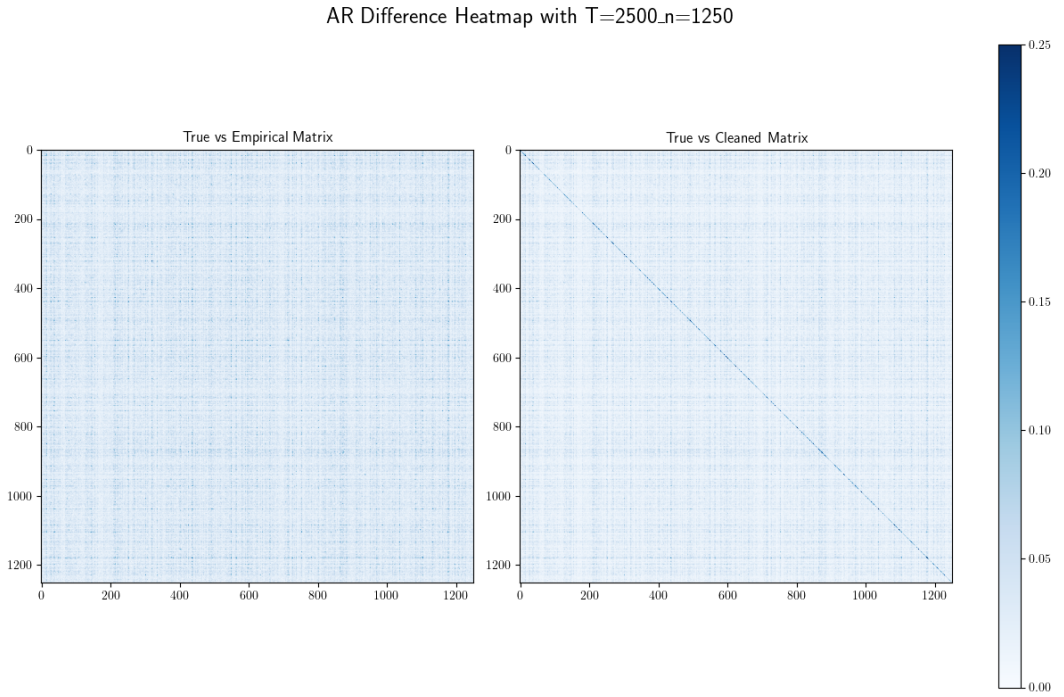

Figure 3.9: AR(1) Heatmap Absolute Difference with n = 1,250 and T = 2,500 (True vs Empirical and True vs Cleaned)

Figure 3.9 displays heatmaps of the absolute differences between the true and empirical matrices, and between the true and cleaned matrices, for the dimension $n = 1,250$. The cleaned matrix shows subtly lower noise levels than the empirical one, as indicated by the lighter shade in the overall matrix.

However, a diagonal line in the True vs Cleaned matrix suggests disparities in the signal's representation post-cleaning. The significance and implications of this discrepancy will be further analysed and elaborated in Section 3.2.3.

3.2.3 Investigating the Diagonal Values

n	Empirical Diagonal	Cleaned Diagonal
250	0.0564	0.1446
500	0.0535	0.2039
1,250	0.0596	0.2952
2,500	0.0580	0.3596

Table 3.3: Mean Absolute Error of Diagonal Elements for AR(1) Model with $T = 2,500$

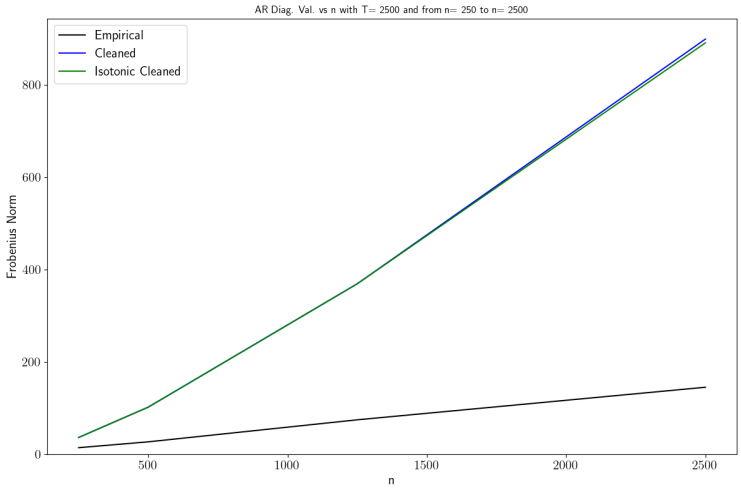

Figure 3.10: AR(1) Diagonal Value Against Dimensions from n = 250 to n = 2,500

Upon examining the true, empirical, and cleaned matrices' diagonal elements, Figure 3.10 shows an increased discrepancy in the cleaned matrix's diagonal values compared to the empirical matrix as the dimension n grows. This discrepancy, evidenced by the larger mean absolute error in Table 3.3, is consistent with the visible diagonal line observed in the heatmap of Figure 3.9, suggesting an alteration in the signal's representation post-cleaning.

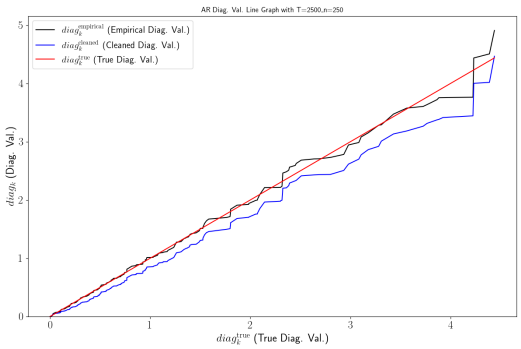

Figure 3.11: AR(1) Diagonal Values Line Graph when n = 250

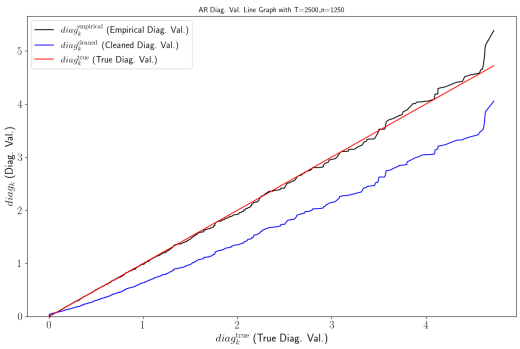

Figure 3.12: AR(1) Diagonal Values Line Graph when n = 1,250

The line graphs in Figures 3.11 and 3.12 further affirm this trend; while the cleaning process reduces noise, it also appears to consistently underestimate the true matrix's diagonal values with increasing n. The observed pattern of underestimation of diagonal values in the cleaned matrices, as dimensions increase, raises a critical point about the interpretation of 'optimal' within the cleaning process, which is further analysed and elaborated in Chapter 4.

3.2.4 Signal-to-Noise Ratio (SNR)

n	Empirical SNR	Cleaned SNR	Isotonic Cleaned SNR
250	1.4463	1.4530	1.4566
500	0.8277	0.8304	0.8315
1,250	0.2963	0.2911	0.2926
2,500	0.1494	0.1470	0.1490

Table 3.4: Signal-to-Noise Ratios for AR Models with varying dimensions n and fixed time $T = 2500$

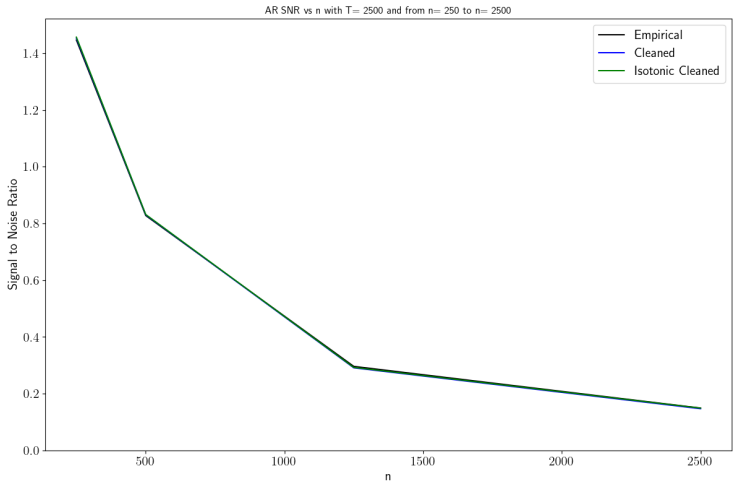

Figure 3.13: AR(1) Signal-to-Noise Ratio Against Dimensions from n = 250 to n = 2,500

The table 3.4 and figure 3.13 display the signal-to-noise ratios (SNR) for AR models at varying dimensions, calculated as follows [29]:

$$\text{SNR} = \frac{\sigma^2_{\text{signal}}}{\sigma^2_{\text{noise}}} \tag{3.1}$$

- σ^2_{signal} is the variance of the diagonal elements of the matrix, signifying the signal in the AR(1) model.
- σ^2_{noise} is the variance of the off-diagonal elements of the matrix, signifying the noise.

The consistent SNR across different matrices indicates that the smaller signals remain distinct and discernable relative to the surrounding noise. This finding suggests that the clarity of the signal remained similar relative to the noise, but it did not make it more clear.

Chapter 4

Discussion

4.1 RG Expected Results vs Actual Results

In the analysis of the RG model, the actual outcomes in Section 3.1 demonstrate a consistent enhancement in matrix cleaning with increasing dimensionality, conforming to expectations detailed in Section 2.3.3.

The observed improvements in the RG model primarily stem from the negligible impact of singular vectors in matrices with no inherent signals. As dimensionality increases, the RIE cleaning method effectively minimises the accompanying rise in noise by adjusting singular values without altering singular vectors. This results in a cleaned matrix that closely mirrors the true matrix, particularly in high-dimensional settings where noise reduction is essential.

Validated against a RG model with only white noise, the method's efficacy highlights its robustness in reducing noise. However, its performance in scenarios with actual signals requires further exploration to fully understand its potential and limitations.

4.2 AR(1) Expected Results vs Actual Results

The AR(1) model presents a more nuanced scenario. Unlike the RG case, the AR(1) model contains intrinsic temporal correlations, requiring a more careful cleaning process. Here is a summary of the results in Section 3.2:

Metric	Implication
Frobenius Norm	Improved performance in cleaned matrices
Singular Values Line Graph	Improved singular values
Heatmaps	Decreased noise, enhanced matrix clarity
Absolute Difference Heatmaps	Signal potentially affected by cleaning
Diagonal Values	Lower than expected, possible overcorrection
Signal-to-Noise Ratio	Signals remain discernible among the reduced noise

Table 4.1: Summary of AR(1) Model Results and Implications

While the Benaych-Georges et al. paper [1] highlights the optimisation of the Frobenius norm among RIEs, the application of this method reveals that achieving optimality in the Frobenius norm does not guarantee a perfect replication of the true covariance matrix. This distinction is crucial, particularly when considering the method's practical implications in fields requiring high fidelity in covariance matrix estimation.

The notion of "optimality" within the framework established by Benaych-Georges et al. is specific to the cleaning of singular values, with an explicit decision to preserve the empirical singular vectors. Although the method effectively minimises the Frobenius norm distance to the true covariance matrix, it acknowledges the inherent noise in the singular vectors by not altering them. This approach ensures reduced noise in the matrix but does not eradicate discrepancies arising from noisy singular vectors.

4.2.1 Implications for Matrix Reconstruction

Empirical Singular Vectors: The retention of noisy singular vectors while adjusting singular values implies that the cleaning process, although effective in reducing noise, does not address inaccuracies in the vector components themselves. As a result, the cleaned matrix, though less noisy, may not accurately mirror the true underlying covariance structure.

Limitations and Practical Application: This insight into the method's optimality is vital for interpreting its effectiveness, particularly in applications like financial modeling or signal processing, where precise matrix reconstruction is critical. The limitations highlighted by the analysis suggest that while the method is robust in scenarios with no inherent signals, its effectiveness may diminish in more complex datasets where signal fidelity is paramount.

4.2.2 Examples of Practical Applications

The review by Bun et al. [8] illustrates the efficacy of RIEs in cleaning large covariance matrices in real-world financial markets. These estimators adapt effectively without requiring prior knowledge of signals because it cleans noise without altering the signals, proving versatile across varied applications. Their empirically validated efficacy highlights RIEs as robust tools for practical use in financial analysis and similar domains.

In contrast, others point out notable limitations within the framework. For instance, studies such as Bongiorno et al. [30, 31, 32] highlight that Frobenius-optimised singular values may not be ideally suited for portfolio optimisation. This is primarily because they do not effectively filter out noise in the singular vectors nor address issues of non-stationarity. As datasets grow increasingly complex, the challenges posed by noise in singular vectors and non-stationary data become more pronounced. Consequently, exploring multiple methodologies to refine covariance matrix estimation becomes imperative when dealing with real-world data.

The paper [33] underscores both the strengths and limitations of RIEs in practical applications. Hierarchical Clustering Estimators (HCEs) sometimes outperform RIEs when filtering sample cross-covariance matrices under various loss functions (Stein's loss and Kullback Leilbler divergence), indicating their efficacy in contexts where RIEs falter. Moreover, the integration of HCEs with RIEs in a two-step estimation process can yield even better results, optimising the balance between noise reduction and maintaining the signal of the covariance matrix. This adaptability highlights RIEs' utility and the benefit of combining methodologies to enhance accuracy in complex statistical models, suggesting a more complete approach to the application of these estimators, especially in finance and complex systems analysis where precision is crucial.

Chapter 5

Conclusion

5.1 Conclusion of Results

The primary goal of this paper was to enhance the accuracy of covariance matrix estimation in high-dimensional time series data using RIE cleaning techniques. The design achieved the goal in the RG model and mostly achieved the goal in the AR(1) model with some caveats. The approach denoised the matrix, but the signal's strength was also decreased, maintaining a similar signal-to-noise ratio.

A significant advancement made in this research was the adaptation of Random Matrix Theory to lagged covariance matrices, which introduced a nuanced approach to handle temporal correlations. This innovation, along with the ability to compute the true lagged covariance matrix for the AR(1) model, provided a robust framework for testing and validating the effectiveness of our cleaning techniques.

The RIE technique's versatility and efficacy in noise reduction were evident, yet this versatility means it is optimised primarily for reducing noise, potentially at the expense of signal strength. This highlights the need for developing methods that can improve signal strength as well. Ideally, these methods could be combined with RIE to harness the best of each algorithm, balancing noise reduction with signal enhancement.

A key limitation of this technique is its dependence on the assumption of stationarity in time series data. This assumption often does not hold in complex, real-world settings, particularly in financial markets where data elements frequently show non-stationary behaviour.

In conclusion, the exploration of these results clarifies the strengths and limitations of the RIE-based cleaning method. It shows that it is essential to apply the method judiciously, ensuring that its use is tailored to the specific requirements of the analytical context since it optimally addresses one part of improving matrix estimation—reducing the noise of covariance matrices in high-dimension.

5.2 Challenges

Undertaking this research presented several unique challenges, both technical and conceptual. Understanding the complex statistical terminology and methodologies used by CFM—a research-focused French multinational company in quantitative asset management—was initially daunting. The depth and breadth of their state-of-the-art scientific investment strategies required a steep learning curve.

Moreover, the project reinforced a crucial lesson about the nature of data analysis: results can often be misleading if not rigorously tested. For instance, there was an unnoticed bug in my code when reconstructing the matrix, but due to the number of noise being much greater than the number of signals, the results still displayed a significant improvement in the Frobenius Norm despite the loss of signal. After detailed analysis and debugging, the bug was fixed, but it resulted in lost time.

Additionally, the findings emphasised that different methods for computing errors can lead to varying results, which accentuated the importance of testing thoroughly to gain the full picture.

Random Matrix Theory was not a personal area of expertise, which added another layer of complexity. Engaging deeply with this aspect of the study through research using textbooks and research papers was profoundly educational, significantly enhancing my analytical skills and understanding of this advanced field.

Each challenge encountered during this research was an opportunity for valuable learning and experience in writing a paper, providing invaluable insights into the sophisticated world of statistical analysis in high-dimensional data environments.

5.3 Future Work

The promising results from this study open several avenues for further research to enhance the algorithm's effectiveness and applicability:

- **Expansion to More Complex Models:** Future work should incorporate more complex time-series models such as AR(p) and VAR(1). These models introduce a greater number of lags, significantly increasing the dimensionality of the data. Additionally, exploring how autocovariance in VAR models relates to phi values will be more challenging and add depth to our understanding. Additionally, implementing spiked matrix models [34] could also be beneficial, as these models are characterised by a few large, isolated eigenvalues that stand out from the bulk of the eigenvalue spectrum, leading to potential inconsistencies in top eigenvalue estimations.

- **Exploration of Alternative Methods:** It would be valuable to compare other methods like non-linear shrinkage [25] that aim to improve matrix estimation to see if they yield similar enhancements. This comparison could help identify the most effective techniques or combinations thereof for specific types of data and analysis scenarios.

- **Application to Non-Stationary Data:** Applying the RIE cleaning technique to non-stationary time series data would test its robustness and effectiveness in more dynamic and less predictable environments. This extension is crucial for fields such as financial markets, where non-stationarity is a common characteristic of the data.

- **Experiment with Different Loss Functions:** Additionally, experimenting and optimising with different loss functions, such as the Kullback-Leibler Divergence [35], could provide insights into optimising the balance between noise reduction and signal preservation.

Appendix A

Code

My GitHub Repository:

Personal GitHub Link

GitHub Repository and Function Names from Source [1]:

Source GitHub Link

```
def check_matrix(M, n, p): ...
```

```
def get_submatrices_of_lagged_cov_mat(n, p, CZZ): ...
```

```
def Coeffs(n, p, U, V, CXXemp, CYYemp): ...
```

```
def approx_L_or_imLoimH(z,
                        n,
                        p,
                        T,
                        Coeff_A=None,
                        Coeff_B=None,
```

```
                        Coeff_B_n_to_p=None,
                        CXXemp=None,
                        CYYemp=None,
                        CXYemp=None,
                        U=None,
                        s=None,
                        stwo=None,
                        V=None,
                        algo_used=1,
                        return_L=False): ...


def RIE_Cross_Covariance(
        CZZemp,
        T,
        n,
        p,
        Return_Sing_Values_only=False,
        Return_Ancient_SV=False,
        Return_New_SV=False,
        Return_Sing_Vectors=False,
        adjust=False,
        return_all=False,
        isotonic=False,
        exponent_eta=0.5,
        c_eta=1,
        algo_used=1): ...
```

Appendix B

Colophon

This document was set in the Times Roman typeface using LaTeX and BibTeX, composed with an Overleaf text editor.

Bibliography

[1] F. Benaych-Georges, J. P. Bouchaud, and M. Potters. Optimal cleaning for singular values of cross-covariance matrices. *arXiv preprint arXiv:1901.05543*, pages 38 pages, 8 figures, 2 tables, Jan 2019.

[2] R. Bellman, Rand Corporation, and Karreman Mathematics Research Collection. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957.

[3] L. A. Pastur V. A. Marchenko. Distribution of eigenvalues for some sets of random matrices. *Math. USSR-Sb*, page 28 pages, 1967.

[4] O. Ledoit and S. Péché. Eigenvectors of some large sample covariance matrix ensembles, 2009.

[5] J. Bun, R. Allez, J. P. Bouchaud, and M. Potters. Rotational invariant estimator for general noisy matrices. *IEEE Transactions on Information Theory*, 62(12):7475–7490, December 2016.

[6] N. Firoozye, V. Tan, and S. Zohren. Canonical portfolios: Optimal asset and signal combination, 2023.

[7] J. P. Bouchaud and M. Potters. Financial applications of random matrix theory: a short review, 2009.

[8] J. Bun, J. P. Bouchaud, and M. Potters. Cleaning large correlation matrices: Tools from random matrix theory. *Physics Reports*, 666:1–109, January 2017.

[9] G. Jain and B. Mallick. A review on weather forecasting techniques. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(12):177–180, 2016.

[10] P. J. Bickel and E. Levina. Covariance regularization by thresholding. *The Annals of Statistics*, 36(6), December 2008.

[11] J. Browell, C. Gilbert, and M. Fasiolo. Covariance structures for high-dimensional energy forecasting. *Electric Power Systems Research*, 211:108446, 2022.

[12] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*, chapter 9.3. OTexts, Melbourne, Australia, 3rd edition, 2021.

[13] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*, chapter 2.9. OTexts, Melbourne, Australia, 3rd edition, 2021.

[14] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*, chapter 9.1. OTexts, Melbourne, Australia, 3rd edition, 2021.

[15] J. Brownlee. *Introduction to Time Series Forecasting With Python: How to Prepare Data and Develop Models to Predict the Future*, chapter 15, pages 131–142. Machine Learning Mastery, 2017.

[16] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time Series Analysis: Forecasting and Control*, chapter 3.2.1, pages 54–55. Wiley Series in Probability and Statistics. Wiley, 5th edition, 2015.

[17] J. Brownlee. *Introduction to Time Series Forecasting With Python: How to Prepare Data and Develop Models to Predict the Future*, chapter 13, pages 112–118. Machine Learning Mastery, 2017.

[18] J. Brownlee. *Introduction to Time Series Forecasting With Python: How to Prepare Data and Develop Models to Predict the Future*, chapter 14, pages 119–130. Machine Learning Mastery, 2017.

[19] K. I. Park. *Fundamentals of Probability and Stochastic Processes with Applications to Communications*, chapter 6.9, pages 175–184. Springer Cham, 1st edition, 2018.

[20] G. Strang. *Linear Algebra and Its Applications*, chapter 6.3, pages 367–373. Brooks Cole, 4th edition, 2006.

[21] P. J. Bickel and E. Levina. Regularized estimation of large covariance matrices. *The Annals of Statistics*, 36(1), February 2008.

[22] P. J. Bickel and Y. R. Gel. Banded regularization of autocovariance matrices in application to parameter estimation and forecasting of time series. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(5), August 2011.

[23] O. Ledoit and M. Wolf. Honey, i shrunk the sample covariance matrix. *The Journal of Portfolio Management*, 30, 07 2003.

[24] O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2), 2004.

[25] O. Ledoit and M. Wolf. Nonlinear shrinkage estimation of large-dimensional covariance matrices. *The Annals of Statistics*, 40(2):1024–1060, 2012.

[26] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time Series Analysis: Forecasting and Control*, chapter 2.1, pages 21–34. Wiley Series in Probability and Statistics. Wiley, 5th edition, 2015.

[27] R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*, chapter 1, pages 16–23. Springer Texts in Statistics. Springer International Publishing, 2017.

[28] R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*, chapter 3.1, pages 77–79. Springer Texts in Statistics. Springer International Publishing, 2017.

[29] D. K. Manolakis J. G. Proakis. *Digital Signal Processing: Principles, Algorithms and Applications*. Prentice Hall, 3rd edition, 1995.

[30] C. Bongiorno and D. Challet. Non-linear shrinkage of the price return covariance matrix is far from optimal for portfolio optimisation, 2022.

[31] C. Bongiorno and D. Challet. The Oracle estimator is suboptimal for global minimum variance portfolio optimisation. *Finance Research Letters*, 52:103383, March 2023.

[32] C. Bongiorno and L. Lamrani. Quantifying the information lost in optimal covariance matrix cleaning, 2023.

[33] A. García-Medina, S. Miccichè, and R. N. Mantegna. Two-step estimators of high dimensional correlation matrices, 2023.

[34] B. Aubin, B. Loureiro, A. Maillard, F. Krzakala, and L. Zdeborová. The spiked matrix model with generative priors, 2019.

[35] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.