Precision Matrix Estimation using Preconditioned Conjugate Gradient with Regularization

Alejandro Barrientos
C.C. 1017251865
abarrie6@eafit.edu.co

Advisor
Santiago Ortiz
sortiza2@eafit.edu.co

Master in Data Sciences and Analytics
School of Applied Sciences and Engineering
Universidad EAFIT, Medellín, Colombia

Abstract

This study investigates the estimation of precision matrices using the Preconditioned Conjugate Gradient (PCG). In high-dimensional or linearly dependent data scenarios, the sample covariance matrix becomes singular, posing challenges to conventional estimation approaches. This research proposes a methodology to enable precision matrix estimation by integrating regularization techniques and preconditioners. These transformations ensure the resulting matrix is symmetric, positive definite, and exhibits a lower condition number than the original. The efficacy of the PCG method was evaluated using simulated data, demonstrating its robust performance in high-dimensional settings. For all the simulation settings, PCG converged effectively when coupled with Ledoit & Wolf regularization, and the preconditioner derived from the Neumann series. Further research is suggested to find optimal regularization techniques to reduce the estimation error, and preconditioners to enhance converge rates in iterative methods.

Keywords: Precision Matrix Estimation, Preconditioned Conjugate Gradient, Regularization, High-Dimensional Data, Covariance Matrix, Statistical Learning.

1. Introduction

The inverse of the covariance matrix ($\mathbf{\Sigma}$), the precision matrix ($\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$), measures the conditional dependency between two variables, given the other variables (1). It is used in many statistical models such as linear regression, discrimination analysis, portfolio selection, complex data visualization and classification (2; 3; 4; 5). It is also used to compute the Mahalanobis distance, which is commonly used for multivariate outlier detection (6). Estimating the precision matrix is especially important in high-dimensional data applications, such as image classification, text recognition, bio-metric recognition and genome expression analysis (7; 8; 9), making it valuable in tasks like variable selection and model regularization, where identifying informative features and mitigating overfitting are crucial. In these scenarios, the estimation process involves the multiplication of large sparse matrices, making traditional methods impractical due to their inability to scale efficiently (10).

Estimated sample covariance matrices ($\hat{\mathbf{\Sigma}}$) often tend to be nearly singular. Therefore, computing its inverse can yield incorrect results due to numerical errors from round-off and truncation. In high-dimensional settings, or when multicollinearity between variables exists, $|\hat{\mathbf{\Sigma}}| = 0$, and $\hat{\mathbf{\Omega}}$ can no longer be computed. Therefore, some methods to estimate precision matrices directly have been proposed using different approaches: penalized maximum likelihood (11), column-by-column estimation (12) and Bayesian methods (13) are the most used in the literature. However, these methods usually assume a specific structure of $\hat{\mathbf{\Omega}}$, such as sparsity and conditional independence.

The Preconditioned Conjugate Gradient (PCG) method is an iterative technique for solving systems of linear equations (14). This method is suited for symmetric positive definite matrices, making it appropriate to compute $\hat{\mathbf{\Omega}}$. The implementation of this iterative method, coupled with an approximate solution from a direct method, can be used to refine the solution (15). Thus, by incorporating a preconditioner matrix, the PCG method can enhance the estimation accuracy of the precision matrix, altering the coefficient matrix but not the final result.

This research aims to design a methodology for $\hat{\mathbf{\Omega}} = \hat{\mathbf{\Sigma}}^{-1}$ computation based on PCG method. The study focuses on the conditions necessary for convergence, using simulated data and various transformations. This investigation intends to demonstrate the efficacy of the PCG method and its potential to improve precision matrix estimation. The manuscript is organized into the following sections: Section 2 introduces the definition and properties of the covariance and precision matrices. Section 3 examines the conditioning of the covariance matrix, including a review of regularization methods (3.1) and preconditioners (3.2). Section 4 outlines the simulation study, followed by the presentation and discussion of results in Section 5. Lastly, in Section 6 the conclusions with final remarks on the research are presented.

2. Covariance and Precision Matrices

For an $n \times p$ data matrix $\mathbf{X}$, where n is the number of observations and p the dimension, the unbiased sample covariance matrix is defined as

$$\hat{\mathbf{\Sigma}} = \frac{\mathbf{X^T P X}}{n-1}, \tag{1}$$

where $\mathbf{P} = \mathbf{I_n} - \frac{1}{n}\mathbf{1_n}$ is the centering matrix, $\mathbf{I_n}$ the $n \times n$ identity matrix, and $\mathbf{1_n}$ the $n \times n$ matrix of ones, respectively. The estimated precision matrix is then defined such that

$$\hat{\mathbf{\Sigma}}\hat{\mathbf{\Omega}} = \mathbf{I_p}, \tag{2}$$

It can be viewed as a set of systems of linear equations, where $\hat{\mathbf{\Sigma}}$ is the coefficient matrix, and each column of $\hat{\mathbf{\Omega}}$ is the solution to the system formed with each corresponding column of $\mathbf{I_p}$.

There are several methods to solve systems of linear equations, broadly categorized into direct and iterative methods (16). Direct methods include Gaussian elimination, LU decomposition, and determinant factoring methods. These approaches solve linear equations with a high level of precision, but tend to be slow when operating on large matrices (17). On the other hand, iterative methods, including stationary and Krylov subspace methods, produce approximate solutions to linear systems after a finite number of steps. These are particularly useful for large systems of equations, where trading off some precision for shorter run times is acceptable (18).

Since $\hat{\mathbf{\Sigma}}$ is symmetric, methods specifically designed for this type of matrix are appropriate. Among direct methods, Cholesky decomposition is commonly used due to its efficiency compared to LU decomposition (19). Similarly, iterative methods based on conjugate gradients are tailored for these scenarios (20). When dealing with high-dimensional settings, iterative methods are more suitable for computing $\hat{\mathbf{\Omega}}$. The performance of the above-mentioned methods is heavily impacted by the spectral properties of the coefficient matrix, which is also required to be positive definite to ensure convergence (21). Let $\lambda_p \geq \lambda_{p-1} \geq \cdots \geq \lambda_1$ be the eigenvalues of $\hat{\mathbf{\Sigma}}$. For the linear systems defined in Equation 2, the condition number $\kappa(\hat{\mathbf{\Sigma}})$ is defined as

$$\kappa(\hat{\mathbf{\Sigma}}) = \|\hat{\mathbf{\Sigma}}\|_2 \|\hat{\mathbf{\Omega}}\|_2 = \frac{|\lambda_p(\hat{\mathbf{\Sigma}})|}{|\lambda_1(\hat{\mathbf{\Sigma}})|} \geq 1 \tag{3}$$

for the spectral matrix norm $\|\cdot\|_2$. This number measures how inaccurate will the solution $\hat{\mathbf{\Omega}}$ be, that is, how much $\hat{\mathbf{\Omega}}$ changes when a relatively small change in $\hat{\mathbf{\Sigma}}$ occurs (22). The larger the condition number, the higher the possible change in the solution vector for the linear system. In some scenarios, the condition number is so high that perturbations or contamination in the data might lead to inconsistent results.

The eigenvalues can be used to provide a sense of both the positive definiteness and conditioning of $\hat{\mathbf{\Sigma}}$. If any of the eigenvalues is zero, then the condition number tends to

infinity and $\hat{\mathbf{\Sigma}}$ is semipositive definite. In such cases, $\hat{\mathbf{\Sigma}}$ is singular and therefore no longer invertible, even though $\mathbf{\Omega}$ exists. PCG incorporates a preconditioner matrix to decrease $\kappa(\hat{\mathbf{\Sigma}})$. Then, the linear systems in Equation 2 can be written as one of the two following options:

- Using a matrix $\mathbf{M^{-1}}$:

$$\mathbf{M}^{-1}\hat{\mathbf{\Sigma}}\hat{\mathbf{\Omega}} = \mathbf{M}^{-1}\mathbf{I} \tag{4}$$

- Using a matrix $\mathbf{M} = \mathbf{M_1}\mathbf{M_2}$ such that $\mathbf{M_1} = \mathbf{M_2^T}$:

$$(\mathbf{M}_1^{-1}\hat{\mathbf{\Sigma}}\mathbf{M}_2^{-1})\mathbf{M}_2\hat{\mathbf{\Omega}} = \mathbf{M}_1^{-1}\mathbf{I} \tag{5}$$

In general, the efficacy of the preconditioners depends on how well $\mathbf{M}$ approximates $\mathbf{\Sigma}$ or $\mathbf{\Omega}$, depending on how it is constructed. By applying this transformation, it is easy to show that the condition number of the coefficient matrix will change, and that the systems have the same solution as the original one. Nevertheless, two additional steps are involved in each case, to transform the variables at the beginning and at the end of the iteration process.

In both cases, it is required to ensure that $\mathbf{M}$ is also symmetric and positive definite for the method to converge. It is worth noting that the determinant of $\hat{\mathbf{\Sigma}}$ is also positive if all its eigenvalues are positive as well, meaning that now it is possible to invert it. By construction, the covariance matrix is semipositive definite, and some of its eigenvalues might be zero. This occurs when $p >> n$, or when two variables are not linearly independent. Then, to ensure convergence of PCG, and also improve its performance, it is necessary to apply transformation over $\hat{\mathbf{\Sigma}}$ so that the resulting matrix is positive definite with a low condition number.

3. Conditioning of the Covariance Matrix

3.1. Regularization

Regularization is a method to enforce conditions over a matrix (23). For covariance matrices, it is often applied to produce a positive definite matrix. This has a direct impact in the eigenvalues, which also make the condition number from being undefined to be a real value. For this study, Tikhonov regularization and Ledoit & Wolf shrinkage will be used, which are explained below.

Tikhonov Regularization

Tikhonov regularization (24) was developed for the ordinary least squares problem, which is modified to optimize over

$$\underset{\mathbf{X}}{\operatorname{argmin}} \|\mathbf{AX} - \mathbf{Y}\|^2 + \|\mathbf{\Gamma X}\|^2, \tag{6}$$

where $\mathbf{\Gamma}$ is a suitable Tikhonov matrix. It is common to choose this matrix as a scalar multiple of the identity matrix ($\rho\mathbf{I}$, with ρ the regularization coefficient). Extending the idea of this regularization, it is possible to rewrite the inversion problem as

$$(\hat{\mathbf{\Sigma}} + \rho\mathbf{I})\hat{\mathbf{\Omega}} = \mathbf{I}. \tag{7}$$

After this transformation, the eigenvalues of the resulting coefficient matrix are $\lambda_p + \rho, \lambda_{p-1} + \rho \geq, \ldots, \geq \lambda_1 + \rho$. It is clear now that the matrix has transformed to a positive definite matrix, the condition number is defined and the matrix is invertible. However, when ρ is close to zero, the resulting condition number might have a large magnitude. It is also possible to establish a new bound for the resulting condition number as (16)

$$\kappa(\hat{\mathbf{\Sigma}} + \rho\mathbf{I}) = \frac{\lambda_p + \rho}{\lambda_1 + \rho} \leq \frac{2\lambda_p}{\rho}, \quad 0 < \rho < \lambda_p. \tag{8}$$

As such, a larger ρ will reduce the condition number of the matrix, but will increase the estimation error.

Ledoit & Wolf Shrinkage

A popular regularization approach is the one developed by **(author?)** (25). In this, the matrix is said to be shrunk to a well-conditioned one. The problem is then to find the optimal linear combination $\mathbf{\Sigma}^* = (1 - \rho)\hat{\mathbf{\Sigma}} + \rho\mathbf{I}$ that solves the following optimization problem

$$\begin{aligned} \underset{\rho}{\operatorname{argmin}} \quad & E\left[\|\mathbf{\Sigma}^* - \hat{\mathbf{\Sigma}}\|^2\right] \\ \text{s.t.} \quad & \mathbf{\Sigma}^* = (1 - \rho)\hat{\mathbf{\Sigma}} + \rho\mathbf{I}. \end{aligned} \tag{9}$$

Likewise, the eigenvalues of the resulting values are $(1 - \rho)\lambda_p + \rho, (1 - \rho)\lambda_{p-1} + \rho \geq , \ldots, (1 - \rho)\lambda_1 + \rho$. Again, the matrix is now positive definite and with a lower condition number.

3.2. Precondtioners

There are several preconditioners that can be used that fall in one of the two categories mentioned before. Three preconditioner matrices are presented below, which are used in the iterative method.

Symmetric Successive Over-Relaxation (SSOR)

This preconditioner is derived from the first steps of the Successive Over-Relaxation (SOR) method (26). The original symmetric matrix is split into diagonal, lower and upper triangular as $\hat{\mathbf{\Sigma}} = \mathbf{D} + \mathbf{L} + \mathbf{L^T}$. The conditioning matrix is, parameterized by ω:

$$\mathbf{M} = \frac{\omega}{2-\omega}\left(\frac{1}{\omega}\mathbf{D} + \mathbf{L}\right)\mathbf{D}^{-1}\left(\frac{1}{\omega}\mathbf{D} + \mathbf{L}\right)^T \tag{10}$$

In practice, estimating an optimal value for ω is more expensive and does not provide a considerable improvement in the performance of the method (27). Therefore, it was assumed to be 1. Thus, it follows that $\mathbf{M_1}^{-1} = \mathbf{D}^{1/2}(\mathbf{D} + \mathbf{L})^{-1}$ and $\mathbf{M_2}^{-1} = ((\mathbf{D} + \mathbf{L})^T)^{-1}\mathbf{D}^{1/2}$. This definition ensures that the resulting matrix is symmetric.

Incomplete LU Factorization (ILU)

In the LU factorization, it can happen that the resulting matrix does not maintain the sparsity of the original (28). To account for this, the incomplete factorization keeps as zero the values in the positions where the original matrix are zero as well (29). This can be extended thanks to $\hat{\mathbf{\Sigma}}$ being symmetric positive definite after regularization to apply an Incomplete Cholesky Decomposition (30). Thus, the resulting matrix is $\hat{\mathbf{\Sigma}} \approx \mathbf{L}\mathbf{L^T}$, and then $\mathbf{M}^{-1} = \left(\mathbf{L}^T\right)^{-1}\mathbf{L}^{-1}$.

Neumann Polynomials

This preconditioner is based on the idea of providing an initial approximation to $\mathbf{\Omega}$. The Neumann polynomial is defined as (31)

$$\mathbf{\Omega} \approx \sum_{i=0}^{k}(\mathbf{I} - \mathbf{\Sigma})^i. \tag{11}$$

(32) developed a symmetric and positive definite preconditioner based on this series. It is defined as

$$\mathbf{M}_p^{-1} = \left(\sum_{i=0}^{k-1}(\mathbf{I} - \mathbf{M}^{-1}\hat{\mathbf{\Sigma}})^i\right)\mathbf{M}^{-1}, \quad \text{for} \quad k > 1. \tag{12}$$

The only missing step is to determine $\mathbf{M}^{-1}$ as a symmetric positive definite matrix. In the mentioned research, the matrix $\mathbf{M}^{-1} = \text{diag}(\hat{\mathbf{\Sigma}})^{-1}$ was used. The same implementation was applied in this study.

4. Numerical Experiments

The performance of the PCG method was evaluated with different regularization methods and preconditioners. To do this, several simulation scenarios were performed, with different sizes and contamination. This experiment is equivalent to the one used in (33). A p-dimensional random variable $\mathbf{X}$ following a contaminated multivariate normal distribution was considered. This distribution is given as a mixture of normal distributions of the form $(1-\alpha)\mathcal{N}(\mathbf{0}, \mathbf{\Sigma_1}) + \alpha\mathcal{N}(\delta\mathbf{e}, \lambda\mathbf{\Sigma_2})$, following a fully dependent contamination model (34), where α denotes the contamination percentage, δ the contamination mean, λ the contamination concentration, $\mathbf{e}$ the p-dimensional vector of ones, and $\mathbf{\Sigma_1}$ and $\mathbf{\Sigma_2}$ the population covariance matrices.

In a first scenario, *Normal Distribution*, both covariance matrices were set as $\mathbf{\Sigma_1} = \mathbf{\Sigma_2} = \mathbf{I}$. For the second one, *Correlated Normal Distribution*, the methodology from (35) was applied. As such, a covariance matrix was generated for each sample, for which the correlation matrix had a set condition number of 100, and the standard deviation of each variable was set to 1. The correlation between each pair of variables was generated randomly. A total of 100 p-dimensional samples of size n were generated for each combination of parameters. For each sample, $\hat{\mathbf{\Sigma}}$ was computed using Equation 1, for which every combination of regularization and preconditioning was applied. $\mathbf{I}$ was also used as a preconditioner as a point of comparison. Although more robust location and scatter estimators can be used, they weren't included to evaluate the robustness of the transformations here proposed.

For the PCG method, a maximum of 500 iterations and a tolerance of 1×10^{-6} were set. The residuals were computed using the euclidean norm after each iteration. To assess the performance of each combination of regularization method and preconditioner, the Mean Squared Error (MSE) was computed over the 100 samples. Other values such as κ, λ_p, λ_1, ρ, rank and iteration numbers were also obtained to perform comparisons. These were estimated between $\mathbf{\Omega}$ and $\hat{\mathbf{\Omega}}$.

5. Results and Discussion

For each regularization applied, all preconditioners exhibited the same MSE results across simulation settings when the method converged. This demonstrated the regularization's ability to modify the system for solvability, as shown in Table 1. For clean data, the behavior for independent and correlated data was similar, indicating that correlation does not introduce significant estimation error when using PCG. Moreover, MSE was close to zero for low dimensions, showing that PCG effectively estimated $\hat{\mathbf{\Omega}}$. Tikhonov regularization presented a lower error in this configuration due to its minimal alterations to the coefficient matrix. However, analysis revealed that $\kappa \approx 10^8$, making the preconditioning necessary. Further research is required to develop a method for computing an optimal regularization coefficient, based on minimizing the matrix norm between the sample covariance matrix

and the regularized one.

In cases where $n/p \leq 1$, PCG diverged for Tikhonov regularization, making Ledoit & Wolf more suitable for high-dimensional settings. Although some ILU values appeared lower, this was inconsistent among the generated samples, with residuals never converging to the established tolerance. With contamination, Ledoit & Wolf regularization maintained relative low error across all α, δ and λ values. This held true for increased dimensions, where MSE was higher, yet a solution was found. On the other hand, Tikhonov regularization did not converge when $n/p > 1$, making it unreliable in these scenarios. In general, MSE was lower with high α, δ and λ, as outliers were more easily separated from clean data.

Table 2 contains the number of iterations for each data and transformation combination. Like MSE, fewer iterations were needed for high values of α, δ and λ. For clean independent data, a solution for the Ledoit & Wolf regularization was found in only one iteration for all the preconditioners. As n/p decreased, iteration counts increased for this regularizer, which was still able to converge in less than 500 iterations. This trend was also observed for correlated data, where iterations increased but solutions were still found. SSOR and ILU reduced the number of iterations taken for convergence at large n/p values. However, as this ratio decreased, more iterations were required, and applying PCG without preconditioner became unreliable. Neumann exhibited the worst performance, being the slowest among preconditioners, including the identity matrix.

Tikhonov regularization required more iterations with lower error for clean data. Nevertheless, for $n/p \leq 1$, ILU and Neumann did not find any solutions. This contrasted with the MSE, where ILU showed low results but did not converge for any samples. Correlated data showed a substantial increase in iterations, with ILU performing the worst. Contaminated data imposed a natural increase in the number of iterations, causing some preconditioned systems to diverge. Ledoit & Wolf regularization had a good performance across the board, particularly when paired with no preconditioner or the Neumann polynomial, showing that performance improved with preconditioners based on estimating $\mathbf{\Omega}$. For SSOR and ILU, the initial solution improved at each step until residuals could not be reduced below 10^{-1}. Afterwards, the method failed to improve the solution further, reaching maximum iterations. Despite similar MSE to the methods that converged, these preconditioners were unreliable in contaminated settings. This pattern persisted across all contamination samples, where all the preconditioners with Ledoit & Wolf achieved the same MSE, but SSOR and ILU failed to converge in high-dimensional settings.

Overall, Tikhonov regularized systems required more iterations to converge, despite having a lower MSE. However, convergence was not achieved for $n/p \leq 1$. Unlike Ledoit & Wolf, ILU converged in under 500 iterations. It is noteworthy that applying a preconditioner with this regularization did not reduce iterations. Instead, in all the scenarios where this transformation converged, using $\mathbf{I}$ yielded a better performance, suggesting that system transformation post-regularization only affected convergence speed.

Table 1: MSE between $\hat{\mathbf{\Omega}}$ and $\mathbf{\Omega}$ of Iterations for Normal Distributed data.

Regularization	Preconditioner	800x100	200x100	200x200	200x400	800x100	200x100	200x200	200x400
Clean Data		Independent Data				Correlated Data			
Ledoit & Wolf	Identity	7.9×10^{1}	6.1×10^{1}	1.7×10^{2}	4.3×10^{2}	5.2×10^{1}	8.2×10^{1}	2.2×10^{2}	5.4×10^{2}
	SSOR	7.9×10^{1}	6.1×10^{1}	1.7×10^{2}	4.3×10^{2}	5.2×10^{1}	8.2×10^{1}	2.2×10^{2}	5.4×10^{2}
	ILU	7.9×10^{1}	6.1×10^{1}	1.7×10^{2}	4.3×10^{2}	5.2×10^{1}	8.2×10^{1}	2.2×10^{2}	5.4×10^{2}
	Neumann	7.9×10^{1}	6.1×10^{1}	1.7×10^{2}	4.3×10^{2}	5.2×10^{1}	8.2×10^{1}	2.2×10^{2}	5.4×10^{2}
Tikhonov	Identity	2.1×10^{-3}	5.3×10^{-2}	2.5×10^{11}	1.3×10^{13}	2.5×10^{-2}	8.4×10^{-1}	2.4×10^{11}	1.3×10^{13}
	SSOR	2.1×10^{-3}	5.3×10^{-2}	2.5×10^{11}	1.3×10^{13}	2.5×10^{-2}	8.4×10^{-1}	2.5×10^{11}	1.3×10^{13}
	ILU	2.1×10^{-3}	5.3×10^{-2}	1.4×10^{2}	3.0×10^{11}	2.5×10^{-2}	8.4×10^{-1}	8.5×10^{1}	3.2×10^{11}
	Neumann	2.1×10^{-3}	5.3×10^{-2}	1.8×10^{11}	1.3×10^{13}	2.5×10^{-2}	8.4×10^{-1}	9.2×10^{5}	1.2×10^{13}
Contaminated Data: $\alpha = 0.2.\ \delta = 50.\ \lambda = 0.1$									
Ledoit & Wolf	Identity	1.9×10^{2}	4.1×10^{1}	9.6×10^{1}	2.2×10^{2}	3.2×10^{2}	5.1×10^{1}	1.0×10^{2}	2.2×10^{2}
	SSOR	1.9×10^{2}	4.1×10^{1}	9.6×10^{1}	2.2×10^{2}	3.2×10^{2}	5.1×10^{1}	1.0×10^{2}	2.2×10^{2}
	ILU	1.9×10^{2}	4.1×10^{1}	9.6×10^{1}	2.2×10^{2}	3.2×10^{2}	5.1×10^{1}	1.0×10^{2}	2.2×10^{2}
	Neumann	1.9×10^{2}	4.1×10^{1}	9.6×10^{1}	2.2×10^{2}	3.2×10^{2}	5.1×10^{1}	1.0×10^{2}	2.2×10^{2}
Tikhonov	Identity	5.5×10^{-3}	1.6×10^{-1}	4.6×10^{6}	1.3×10^{13}	8.6×10^{-2}	4.6×10^{-1}	2.2×10^{6}	1.3×10^{13}
	SSOR	5.5×10^{-3}	1.6×10^{-1}	2.0×10^{4}	1.1×10^{13}	8.6×10^{-2}	4.6×10^{-1}	1.5×10^{4}	1.1×10^{13}
	ILU	5.5×10^{-3}	1.6×10^{-1}	2.3×10^{1}	3.0×10^{7}	8.6×10^{-2}	4.6×10^{-1}	2.2×10^{1}	9.4×10^{6}
	Neumann	5.5×10^{-3}	1.6×10^{-1}	1.8×10^{5}	1.3×10^{13}	8.6×10^{-2}	4.6×10^{-1}	1.4×10^{5}	1.3×10^{13}
Contaminated Data: $\alpha = 0.2.\ \delta = 100.\ \lambda = 0.1$									
Ledoit & Wolf	Identity	4.2×10^{1}	4.0	7.8	1.6×10^{1}	4.4×10^{1}	3.9	7.3	1.5×10^{1}
	SSOR	4.2×10^{1}	4.0	7.8	1.6×10^{1}	4.4×10^{1}	3.9	7.3	1.5×10^{1}
	ILU	4.1×10^{1}	4.1	7.7	1.6×10^{1}	4.5×10^{1}	3.8	7.4	1.5×10^{1}
	Neumann	4.2×10^{1}	4.0	7.8	1.6×10^{1}	4.4×10^{1}	3.9	7.3	1.5×10^{1}
Tikhonov	Identity	5.5×10^{-3}	1.5×10^{-1}	6.1×10^{6}	1.3×10^{13}	9.0×10^{-2}	4.9×10^{-1}	1.4×10^{6}	1.3×10^{13}
	SSOR	5.5×10^{-3}	1.5×10^{-1}	1.5×10^{4}	1.1×10^{13}	9.0×10^{-2}	4.9×10^{-1}	1.3×10^{4}	1.1×10^{13}
	ILU	5.5×10^{-3}	1.5×10^{-1}	1.7×10^{1}	3.3×10^{6}	9.0×10^{-2}	4.9×10^{-1}	1.6×10^{1}	1.6×10^{6}
	Neumann	5.5×10^{-3}	1.5×10^{-1}	1.7×10^{5}	1.3×10^{13}	9.0×10^{-2}	4.9×10^{-1}	1.1×10^{5}	1.3×10^{13}

Table 1: Continued

Regularization	Preconditioner	800x100	200x100	200x200	200x400	800x100	200x100	200x200	200x400
		Independent Data				Correlated Data			
Contaminated Data: $\alpha = 0.4.\ \delta = 50.\ \lambda = 0.1$									
Ledoit & Wolf	Identity	1.1×10^{2}	1.2×10^{1}	1.9×10^{1}	4.2×10^{1}	1.1×10^{2}	1.0×10^{1}	1.9×10^{1}	4.3×10^{1}
	SSOR	1.1×10^{2}	1.2×10^{1}	1.9×10^{1}	4.2×10^{1}	1.1×10^{2}	1.0×10^{1}	1.9×10^{1}	4.3×10^{1}
	ILU	1.1×10^{2}	1.2×10^{1}	2.0×10^{1}	4.2×10^{1}	1.2×10^{2}	1.1×10^{1}	1.9×10^{1}	4.2×10^{1}
	Neumann	1.1×10^{2}	1.2×10^{1}	1.9×10^{1}	4.2×10^{1}	1.1×10^{2}	1.0×10^{1}	1.9×10^{1}	4.3×10^{1}
Tikhonov	Identity	1.6×10^{-2}	5.4×10^{-1}	4.4×10^{6}	1.3×10^{13}	1.5×10^{-1}	9.4×10^{-1}	2.0×10^{6}	1.3×10^{13}
	SSOR	1.6×10^{-2}	5.4×10^{-1}	3.1×10^{4}	1.1×10^{13}	1.5×10^{-1}	9.4×10^{-1}	2.5×10^{4}	1.0×10^{13}
	ILU	1.6×10^{-2}	5.4×10^{-1}	3.1×10^{1}	2.5×10^{6}	1.5×10^{-1}	9.4×10^{-1}	3.2×10^{1}	1.5×10^{6}
	Neumann	1.6×10^{-2}	5.4×10^{-1}	2.5×10^{5}	1.3×10^{13}	1.5×10^{-1}	9.4×10^{-1}	1.8×10^{5}	1.3×10^{13}
Contaminated Data: $\alpha = 0.4.\ \delta = 100.\ \lambda = 0.1$									
Ledoit & Wolf	Identity	1.1×10^{1}	7.1×10^{-1}	1.4	2.8	1.2×10^{1}	8.8×10^{-1}	1.5	2.7
	SSOR	1.1×10^{1}	7.1×10^{-1}	1.4	2.8	1.2×10^{1}	8.8×10^{-1}	1.5	2.7
	ILU	1.1×10^{1}	7.3×10^{-1}	1.4	2.8	1.1×10^{1}	8.9×10^{-1}	1.5	2.7
	Neumann	1.1×10^{1}	7.1×10^{-1}	1.4	2.8	1.2×10^{1}	8.8×10^{-1}	1.5	2.7
Tikhonov	Identity	1.6×10^{-2}	5.5×10^{-1}	2.9×10^{6}	1.3×10^{13}	1.5×10^{-1}	9.3×10^{-1}	1.2×10^{6}	1.3×10^{13}
	SSOR	1.6×10^{-2}	5.5×10^{-1}	2.0×10^{4}	1.1×10^{13}	1.5×10^{-1}	9.3×10^{-1}	2.3×10^{4}	1.0×10^{13}
	ILU	1.6×10^{-2}	5.5×10^{-1}	2.4×10^{1}	1.7×10^{6}	1.5×10^{-1}	9.3×10^{-1}	2.3×10^{1}	9.7×10^{5}
	Neumann	1.6×10^{-2}	5.5×10^{-1}	1.8×10^{5}	1.3×10^{13}	1.5×10^{-1}	9.3×10^{-1}	1.5×10^{5}	1.3×10^{13}
Contaminated Data: $\alpha = 0.2.\ \delta = 50.\ \lambda = 1.0$									
Ledoit & Wolf	Identity	3.3×10^{1}	4.9	8.4	1.8×10^{1}	4.0×10^{1}	3.7	9.1	1.7×10^{1}
	SSOR	3.3×10^{1}	4.9	8.4	1.8×10^{1}	4.0×10^{1}	3.7	9.1	1.7×10^{1}
	ILU	3.3×10^{1}	5.0	8.5	1.8×10^{1}	3.9×10^{1}	3.7	9.2	1.7×10^{1}
	Neumann	3.3×10^{1}	4.9	8.4	1.8×10^{1}	4.0×10^{1}	3.7	9.1	1.7×10^{1}
Tikhonov	Identity	2.2×10^{-3}	5.3×10^{-2}	2.5×10^{11}	1.3×10^{13}	3.1×10^{-1}	3.4×10^{-1}	1.0×10^{7}	1.3×10^{13}
	SSOR	2.2×10^{-3}	5.3×10^{-2}	8.8×10^{3}	1.2×10^{13}	3.1×10^{-1}	3.4×10^{-1}	8.6×10^{3}	1.2×10^{13}
	ILU	2.2×10^{-3}	5.3×10^{-2}	2.3×10^{1}	1.3×10^{11}	3.1×10^{-1}	3.4×10^{-1}	1.8×10^{1}	2.0×10^{10}
	Neumann	2.2×10^{-3}	5.3×10^{-2}	2.2×10^{5}	1.3×10^{13}	3.1×10^{-1}	3.4×10^{-1}	1.1×10^{5}	1.3×10^{13}

Table 1: Continued

Regularization	Preconditioner	800x100	200x100	200x200	200x400	800x100	200x100	200x200	200x400
		Independent Data				Correlated Data			
Contaminated Data: $\alpha = 0.2$. $\delta = 100$. $\lambda = 1.0$									
Ledoit & Wolf	Identity	4.2	3.1×10^{-1}	6.3×10^{-1}	1.2	3.8	5.3×10^{-1}	6.1×10^{-1}	1.2
	SSOR	4.2	3.1×10^{-1}	6.3×10^{-1}	1.2	3.8	5.3×10^{-1}	6.1×10^{-1}	1.2
	ILU	4.6	3.1×10^{-1}	6.1×10^{-1}	1.2	3.8	5.4×10^{-1}	6.2×10^{-1}	1.1
	Neumann	4.2	3.1×10^{-1}	6.3×10^{-1}	1.2	3.8	5.3×10^{-1}	6.1×10^{-1}	1.2
Tikhonov	Identity	2.2×10^{-3}	5.3×10^{-2}	4.5×10^{10}	1.3×10^{13}	3.2×10^{-1}	3.4×10^{-1}	1.8×10^{7}	1.3×10^{13}
	SSOR	2.2×10^{-3}	5.3×10^{-2}	8.3×10^{3}	1.2×10^{13}	3.2×10^{-1}	3.4×10^{-1}	6.8×10^{3}	1.1×10^{13}
	ILU	2.2×10^{-3}	5.3×10^{-2}	1.5×10^{1}	7.3×10^{10}	3.2×10^{-1}	3.4×10^{-1}	1.2×10^{1}	2.6×10^{9}
	Neumann	2.2×10^{-3}	5.3×10^{-2}	2.2×10^{5}	1.3×10^{13}	3.2×10^{-1}	3.4×10^{-1}	1.4×10^{5}	1.3×10^{13}
Contaminated Data: $\alpha = 0.4$. $\delta = 50$. $\lambda = 1.0$									
Ledoit & Wolf	Identity	2.0×10^{1}	2.1	3.9	7.8	2.0×10^{1}	2.2	3.8	8.3
	SSOR	2.0×10^{1}	2.1	3.9	7.8	2.0×10^{1}	2.2	3.8	8.3
	ILU	1.9×10^{1}	2.2	3.8	7.9	2.0×10^{1}	2.3	3.9	8.5
	Neumann	2.0×10^{1}	2.1	3.9	7.8	2.0×10^{1}	2.2	3.8	8.3
Tikhonov	Identity	2.2×10^{-3}	5.2×10^{-2}	2.2×10^{11}	1.3×10^{13}	3.8×10^{-1}	4.0×10^{-1}	1.9×10^{7}	1.3×10^{13}
	SSOR	2.2×10^{-3}	5.2×10^{-2}	9.6×10^{3}	1.2×10^{13}	3.8×10^{-1}	4.0×10^{-1}	8.3×10^{3}	1.2×10^{13}
	ILU	2.2×10^{-3}	5.2×10^{-2}	2.0×10^{1}	1.3×10^{11}	3.8×10^{-1}	4.0×10^{-1}	1.7×10^{1}	2.4×10^{10}
	Neumann	2.2×10^{-3}	5.2×10^{-2}	5.8×10^{5}	1.3×10^{13}	3.8×10^{-1}	4.0×10^{-1}	1.4×10^{5}	1.3×10^{13}
Contaminated Data: $\alpha = 0.4$. $\delta = 100$. $\lambda = 1.0$									
Ledoit & Wolf	Identity	2.1	1.2×10^{-1}	2.7×10^{-1}	5.5×10^{-1}	2.1	4.3×10^{-1}	3.2×10^{-1}	5.2×10^{-1}
	SSOR	2.1	1.2×10^{-1}	2.7×10^{-1}	5.5×10^{-1}	2.1	4.3×10^{-1}	3.2×10^{-1}	5.2×10^{-1}
	ILU	2.2	1.2×10^{-1}	2.6×10^{-1}	5.5×10^{-1}	2.0	4.2×10^{-1}	3.2×10^{-1}	5.3×10^{-1}
	Neumann	2.1	1.2×10^{-1}	2.7×10^{-1}	5.5×10^{-1}	2.1	4.3×10^{-1}	3.2×10^{-1}	5.2×10^{-1}
Tikhonov	Identity	2.2×10^{-3}	5.3×10^{-2}	4.9×10^{9}	1.3×10^{13}	3.9×10^{-1}	3.9×10^{-1}	9.6×10^{6}	1.3×10^{13}
	SSOR	2.2×10^{-3}	5.3×10^{-2}	7.4×10^{3}	1.2×10^{13}	3.9×10^{-1}	3.9×10^{-1}	7.3×10^{3}	1.1×10^{13}
	ILU	2.2×10^{-3}	5.3×10^{-2}	1.4×10^{1}	2.5×10^{10}	3.9×10^{-1}	3.9×10^{-1}	1.2×10^{1}	3.3×10^{9}
	Neumann	2.2×10^{-3}	5.3×10^{-2}	3.2×10^{5}	1.3×10^{13}	3.9×10^{-1}	3.9×10^{-1}	1.4×10^{5}	1.3×10^{13}

Table 2: Number of Iterations for Normal Distributed data.

Regularization	Preconditioner	800x100	200x100	200x200	200x400	800x100	200x100	200x200	200x400
		Independent Data				Correlated Data			
Clean Data									
Ledoit & Wolf	Identity	1.00	12.53	23.55	37.82	9.14	20.30	30.52	46.00
	SSOR	1.00	6.78	14.80	29.90	4.87	12.77	22.56	41.03
	ILU	1.00	8.88	30.88	129.11	5.82	24.69	70.70	308.13
	Neumann	1.00	13.33	72.41	309.38	6.85	32.09	103.80	384.67
Tikhonov	Identity	13.61	35.59	356.18	80.20	54.04	82.81	498.00	99.64
	SSOR	6.00	14.91	153.75	248.79	23.68	36.10	213.55	322.09
	ILU	8.09	34.21	500.00	500.00	67.62	136.31	500.00	500.00
	Neumann	7.34	25.00	500.00	500.00	31.04	67.91	500.00	500.00
Contaminated Data: $\alpha = 0.2.\ \delta = 50.\ \lambda = 0.1$									
Ledoit & Wolf	Identity	22.91	20.44	28.88	42.37	31.35	24.42	32.65	47.01
	SSOR	123.17	136.08	360.53	500.00	143.06	149.18	392.29	500.00
	ILU	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00
	Neumann	27.52	26.76	43.65	63.96	32.45	31.37	48.57	70.58
Tikhonov	Identity	19.80	58.16	500.00	500.00	62.66	97.17	500.00	500.00
	SSOR	136.08	182.81	500.00	500.00	170.26	225.83	500.00	500.00
	ILU	25.73	111.37	500.00	500.00	113.74	257.74	500.00	500.00
	Neumann	29.33	86.59	500.00	500.00	93.17	148.84	500.00	500.00
Contaminated Data: $\alpha = 0.2.\ \delta = 100.\ \lambda = 0.1$									
Ledoit & Wolf	Identity	14.91	12.56	16.57	23.71	19.45	14.71	18.67	26.14
	SSOR	119.68	119.87	276.62	500.00	132.75	125.52	290.93	500.00
	ILU	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00
	Neumann	20.50	17.45	25.95	36.00	25.75	20.35	29.32	39.89
Tikhonov	Identity	20.58	59.47	500.00	500.00	65.10	102.55	500.00	500.00
	SSOR	142.11	191.65	500.00	500.00	179.10	239.52	500.00	500.00
	ILU	28.36	124.16	500.00	500.00	127.45	299.00	500.00	500.00
	Neumann	30.83	89.86	500.00	500.00	98.43	156.51	500.00	500.00

Table 2: Continued

Regularization	Preconditioner	800x100	200x100	200x200	200x400	800x100	200x100	200x200	200x400
		Independent Data				Correlated Data			
Contaminated Data: $\alpha = 0.4.\ \delta = 50.\ \lambda = 0.1$									
Ledoit & Wolf	Identity	15.42	13.39	17.70	26.16	19.37	15.02	19.88	28.84
	SSOR	121.19	122.14	284.53	500.00	133.14	126.25	300.47	500.00
	ILU	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00
	Neumann	20.93	18.62	27.80	40.12	25.85	20.90	31.25	44.11
Tikhonov	Identity	22.07	71.61	500.00	500.00	53.31	93.65	500.00	500.00
	SSOR	142.27	205.88	500.00	500.00	172.97	240.86	500.00	500.00
	ILU	30.11	169.28	500.00	500.00	97.61	273.75	500.00	500.00
	Neumann	33.30	108.13	500.00	500.00	80.82	142.05	500.00	500.00
Contaminated Data: $\alpha = 0.4.\ \delta = 100.\ \lambda = 0.1$									
Ledoit & Wolf	Identity	9.39	8.25	10.86	14.83	11.95	9.30	12.07	16.30
	SSOR	113.22	112.77	241.20	500.00	118.26	114.32	248.48	500.00
	ILU	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00
	Neumann	13.10	12.08	16.55	21.98	16.44	13.31	18.66	24.36
Tikhonov	Identity	23.09	75.10	500.00	500.00	55.68	96.87	500.00	500.00
	SSOR	148.07	216.67	500.00	500.00	181.36	253.28	500.00	500.00
	ILU	33.10	194.13	500.00	500.00	107.74	314.38	500.00	500.00
	Neumann	34.16	112.97	500.00	500.00	83.00	147.10	500.00	500.00
Contaminated Data: $\alpha = 0.2.\ \delta = 50.\ \lambda = 1.0$									
Ledoit & Wolf	Identity	17.02	14.51	18.59	26.91	21.82	15.83	21.43	28.90
	SSOR	120.21	123.73	289.18	500.00	133.37	127.42	310.16	500.00
	ILU	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00
	Neumann	22.40	20.04	29.35	41.38	28.22	21.73	33.62	44.57
Tikhonov	Identity	18.50	47.80	500.00	500.00	38.12	71.60	500.00	500.00
	SSOR	133.42	170.07	500.00	500.00	154.55	197.19	500.00	500.00
	ILU	23.25	81.58	500.00	500.00	56.77	154.53	500.00	500.00
	Neumann	26.73	71.02	500.00	500.00	56.01	107.96	500.00	500.00

Table 2: Continued

Regularization	Preconditioner	800x100	200x100	200x200	200x400	800x100	200x100	200x200	200x400
		Independent Data				Correlated Data			
Contaminated Data: $\alpha = 0.2$. $\delta = 100$. $\lambda = 1.0$									
Ledoit & Wolf	Identity	10.55	8.80	11.23	15.12	12.65	9.90	12.22	16.42
	SSOR	113.43	113.52	243.30	500.00	118.85	114.91	249.81	500.00
	ILU	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00
	Neumann	14.39	12.76	17.18	22.48	17.56	13.90	19.04	24.65
Tikhonov	Identity	18.93	48.91	500.00	500.00	38.83	72.34	500.00	500.00
	SSOR	138.63	178.30	500.00	500.00	161.67	206.21	500.00	500.00
	ILU	26.07	90.26	500.00	500.00	62.06	168.67	500.00	500.00
	Neumann	28.40	74.39	500.00	500.00	58.67	110.15	500.00	500.00
Contaminated Data: $\alpha = 0.4$. $\delta = 50$. $\lambda = 1.0$									
Ledoit & Wolf	Identity	14.52	12.41	16.04	22.37	17.84	13.82	17.35	24.33
	SSOR	118.13	118.95	271.12	500.00	126.64	123.07	281.71	500.00
	ILU	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00
	Neumann	19.80	17.12	24.95	33.66	23.53	19.17	27.55	37.15
Tikhonov	Identity	18.54	48.09	500.00	500.00	31.34	61.46	500.00	500.00
	SSOR	134.85	172.16	500.00	500.00	150.19	190.58	500.00	500.00
	ILU	23.89	84.06	500.00	500.00	43.63	123.99	500.00	500.00
	Neumann	27.50	71.22	500.00	500.00	46.25	91.58	500.00	500.00
Contaminated Data: $\alpha = 0.4$. $\delta = 100$. $\lambda = 1.0$									
Ledoit & Wolf	Identity	8.96	7.92	10.01	13.26	11.13	8.46	10.65	14.08
	SSOR	112.33	112.39	237.18	499.90	116.14	112.91	239.92	500.00
	ILU	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00
	Neumann	12.70	11.49	14.92	19.24	15.23	12.25	16.10	20.59
Tikhonov	Identity	19.29	49.54	500.00	500.00	32.65	63.88	500.00	500.00
	SSOR	140.17	180.39	500.00	500.00	157.60	199.60	500.00	500.00
	ILU	27.13	93.57	500.00	500.00	49.56	140.72	500.00	500.00
	Neumann	28.73	74.92	500.00	500.00	48.86	96.71	500.00	500.00

6. Conclusions

This study showed the potential of using PCG to estimate the precision matrix from an estimated sample covariance matrix. Regularization and preconditioning effectively transformed the initial coefficient matrix, enabling a solution to be found. Further research on transformations specifically tailored for the covariance matrix is recommended. The MSE highlighted the capability of PCG with regularization to produce accurate estimates of the precision matrix. As expected, error increased with dimensionality and contamination, while correlation did not pose a significant issue. When a solution was found, Tikhonov regularization produced a lower error than the Ledoit & Wolf transformation, as it introduced less bias to the estimation.

Tikhonov regularization appears promising for obtaining low-error precision matrix estimates, particularly in low-dimensional settings without requiring any preconditioner. However, developing a method to optimize the regularization coefficient is necessary to ensure PCG convergence and enhance computational efficiency. Setting restrictions on the condition number and estimation error is crucial to achieving this. Among the preconditioners, Neumann showed the best performance when combined with Ledoit & Wolf, despite requiring more iterations than using no preconditioner. This suggests that including more terms in the polynomial or developing a different expression might improve performance.

Future research should aim to establish a more robust minimization problem definition, integrating both the regularization step and the preconditioner into the conditioning minimization problem for the covariance matrix. This approach would involve determining the optimal regularization coefficient and the necessary number of terms for the Neumann polynomial prior to applying PCG.

References

[1] Q. Zhang, L. Li, and H. Yang, "Application of fused graphical lasso to statistical inference for multiple sparse precision matrices," *Plos one*, vol. 19, no. 5, p. e0304264, 2024.

[2] T. Sofer, L. Dicker, and X. Lin, "Variable selection for high dimensional multivariate outcomes," *Statistica Sinica*, vol. 24, no. 4, p. 1633, 2014.

[3] G. Tarr, S. Müller, and N. C. Weber, "Robust estimation of precision matrices under cellwise contamination," *Computational Statistics & Data Analysis*, vol. 93, pp. 404–420, 2016.

[4] G. Torri, R. Giacometti, and S. Paterlini, "Sparse precision matrices for minimum variance portfolios," *Computational Management Science*, vol. 16, no. 3, pp. 375–400, 2019.

[5] Y. Yang and S. Xia, "An integrated precision matrix estimation for multivariate regression problems," *Journal of Statistical Planning and Inference*, vol. 222, pp. 261–272, 2023.

[6] E. Cabana, R. E. Lillo, and H. Laniado, "Multivariate outlier detection based on a robust mahalanobis distance with shrinkage estimators," *Statistical papers*, vol. 62, pp. 1583–1609, 2021.

[7] T. T. Cai, W. Liu, and H. H. Zhou, "Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation," *The Annals of Statistics*, vol. 44, no. 2, pp. 455 – 488, 2016.

[8] M. O. Kuismin and M. J. Sillanpää, "Estimation of covariance and precision matrix, network structure, and a view toward systems biology," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 9, no. 6, p. e1415, 2017.

[9] J. Tong, J. Yang, J. Xi, Y. Yu, and P. O. Ogunbona, "Tuning the parameters for precision matrix estimation using regression analysis," *IEEE Access*, vol. 7, pp. 90585–90596, 2019.

[10] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct methods for sparse matrices*. Oxford University Press, 2017.

[11] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[12] T. Cai, W. Liu, and X. Luo, "A constrained $\ell 1$ minimization approach to sparse precision matrix estimation," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 594–607, 2011.

[13] S. Banerjee and S. Ghosal, "Posterior convergence rates for estimating large precision matrices using graphical models," *Electronic Journal of Statistics*, vol. 8, no. 2, pp. 2111 – 2137, 2014.

[14] O. Axelsson, *Iterative solution methods*. Cambridge university press, 1996.

[15] Z.-Z. Bai, "Motivations and realizations of krylov subspace methods for large sparse linear systems," *Journal of Computational and Applied Mathematics*, vol. 283, pp. 71–78, 2015.

[16] R. Kress, *Numerical analysis*, vol. 181. Springer Science & Business Media, 2012.

[17] S.-L. Zhang, K. Nakata, and M. Kojima, "Incomplete orthogonalization preconditioners for solving large and dense linear systems which arise from semidefinite programming," *Applied numerical mathematics*, vol. 41, no. 1, pp. 235–245, 2002.

[18] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.

[19] M. Agarwal and R. Mehra, "Review of matrix decomposition techniques for signal processing applications," *Int. Journal of Engineering Research and Applications*, vol. 4, no. 1, pp. 90–93, 2014.

[20] M. Fornasier, S. Peter, H. Rauhut, and S. Worm, "Conjugate gradient acceleration of iteratively re-weighted least squares methods," *Computational optimization and applications*, vol. 65, pp. 205–259, 2016.

[21] H. Zheng, *Large dimensional covariance matrix estimation with decomposition-based regularization*. PhD thesis, The University of Wisconsin-Madison, 2014.

[22] A. Paskaramoorthy and M. Woolway, "An empirical evaluation of sensitivity bounds for mean-variance portfolio optimisation," *Finance Research Letters*, vol. 44, p. 102065, 2022.

[23] F. Bauer, S. Pereverzev, and L. Rosasco, "On regularization algorithms in learning theory," *Journal of complexity*, vol. 23, no. 1, pp. 52–72, 2007.

[24] X. Liu and N. Zhang, "Sparse inverse covariance matrix estimation via the-norm with tikhonov regularization," *Inverse Problems*, vol. 35, no. 11, p. 115010, 2019.

[25] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *Journal of multivariate analysis*, vol. 88, no. 2, pp. 365–411, 2004.

[26] W. Turner and J. Flaherty, "An approximation to the ssor preconditioner," 1998.

[27] B. Carre, "The determination of the optimum accelerating factor for successive over-relaxation," *The computer journal*, vol. 4, no. 1, pp. 73–78, 1961.

[28] D. Anderson and M. Gu, "An efficient, sparsity-preserving, online algorithm for low-rank approximation," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 156–165, PMLR, 06–11 Aug 2017.

[29] M. Benzi, "Preconditioning techniques for large linear systems: a survey," *Journal of computational Physics*, vol. 182, no. 2, pp. 418–477, 2002.

[30] S. C. Eisenstat, "Efficient implementation of a class of preconditioned conjugate gradient methods," *SIAM Journal on Scientific and Statistical Computing*, vol. 2, no. 1, pp. 1–4, 1981.

[31] G. W. Stewart, *Matrix algorithms: volume 1: basic decompositions*. SIAM, 1998.

[32] P. F. Dubois, A. Greenbaum, and G. H. Rodrigue, "Approximating the inverse of a matrix for use in iterative algorithms on vector processors," *Computing*, vol. 22, p. 257–268, Sept. 1979.

[33] K. Boudt, P. J. Rousseeuw, S. Vanduffel, and T. Verdonck, "The minimum regularized

covariance determinant estimator," *Statistics and Computing*, vol. 30, no. 1, pp. 113–128, 2020.

[34] F. Alqallaf, S. Van Aelst, V. J. Yohai, and R. H. Zamar, "Propagation of Outliers in Multivariate Data," *The Annals of Statistics*, vol. 37, no. 1, pp. 311–331, 2009.

[35] C. Agostinelli, A. Leung, V. J. Yohai, and R. H. Zamar, "Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination," *Test*, vol. 24, pp. 441–461, 2015.