symmetry

MDPI

Article

Robust Sparse Reduced-Rank Regression with Response Dependency

Wenchen Liu [1], Guanfu Liu [2,*] and Yincai Tang [3]

[1] School of Statistics and Mathematics, Interdisciplinary Research Institute of Data Science, Shanghai Lixin University of Accounting and Finance, Shanghai 201209, China
[2] School of Statistics and Information, Shanghai University of International Business and Economics, Shanghai 201620, China
[3] KLATASDS-MOE, School of Statistics, East China Normal University, Shanghai 200241, China
* Correspondence: gfliu@suibe.edu.cn

Abstract: In multiple response regression, the reduced rank regression model is an effective method to reduce the number of model parameters and it takes advantage of interrelation among the response variables. To improve the prediction performance of the multiple response regression, a method for the sparse robust reduced rank regression with covariance estimation(Cov-SR4) is proposed, which can carry out variable selection, outlier detection, and covariance estimation simultaneously. The random error term of this model follows a multivariate normal distribution which is a symmetric distribution and the covariance matrix or precision matrix must be a symmetric matrix that reduces the number of parameters. Both the element-wise penalty function and row-wise penalty function can be used to handle different types of outliers. A numerical algorithm with a covariance estimation method is proposed to solve the robust sparse reduced rank regression. We compare our method with three recent reduced rank regression methods in a simulation study and real data analysis. Our method exhibits competitive performance both in prediction error and variable selection accuracy.

Keywords: reduced rank regression; robust; sparsity; precision matrix

Citation: Liu, W.; Liu, G.; Tang, Y. Robust Sparse Reduced-Rank Regression with Response Dependency. *Symmetry* **2022**, *14*, 1617. https://doi.org/10.3390/sym14081617

Academic Editors: Piao Chen and Lorentz JÄNTSCHI

Received: 22 June 2022
Accepted: 4 August 2022
Published: 6 August 2022

1. Introduction

In traditional statistical models, the response variables are usually studied in the one-dimensional case, for example, references [1–4]. Recently, there has been an increasing need to predict several response variables by a common set of covariates. For example, in bioinformatics, multiple phenotypes of a patient are to be predicted based on measures of genetic variation. In economics, the returns of multiple stocks can be predicted based on a common set of economic and market variables. In neuroscience, the responses of different regions of the brain can be predicted based on the input stimulus. Such data analysis problems are widely available in multiple regression.

For multiple response variable regression, one can naively model linear regression for each response variable individually by ignoring the inherent correlation structure between the response variables. In fact, performing a separate ordinary least squares regression for each response variable to estimate the coefficient matrix is consistent with performing maximum likelihood estimation. Another practical problem is that the number of parameters in the regression coefficient matrix can be huge even for a modest number of response variables.

It is natural to think of ways to improve this simple approach. Here two directions can be used to reduce the degrees of freedom parameters of the model and to improve its predictive power and interpretability. One of these directions is to reduce the dimensionality of the coefficient matrix. To take advantage of correlations between the response variables, the rank of the coefficient matrix is restricted, which gives rise to the well-known reduced-order regression (RRR), as in references [5,6]. RRR allows the response variables to borrow information from each other through a common set of latent factors to improve the

prediction accuracy. However, each latent factor is a linear combination of all predictors. When there are a large number of predictor variables, some of them may not be useful for prediction. Therefore, an alternative approach is to reduce the number of free parameters by variable selection to exclude redundant predictors in forming latent factors. The issue of rank reduction estimation and variable selection in the RRR framework has been extensively studied in the recent literature [7–12]. However, in all these works, the components of the error terms are assumed to be independently identically distributed. Ref. [13] added variable selection and covariance estimation to the RRR model to improve the predictive performance and facilitate interpretation.

Although RRR can substantially reduce the number of free parameters in multivariate problems, Ref. [14] found that it is extremely sensitive to outliers. In real-world data analysis, outliers and leverage points are bound to occur, and thus, the genuine reduced-rank structure could easily be masked or distorted. Ref. [15] proposed a method to detect outliers in a continuous distribution based on the cumulative distribution function. Ref. [16] proposes the use of order statistics to detect extreme values in continuously distributed samples. Ten case studies of outlier detection are given in reference [16], the method of order statistics demonstrates its excellence in a comparative study. However, this method of outlier detection is without covariates and the effect of covariates needs to be taken into account in our paper.

The closely related research of this paper includes sparse reduced-rank regression with covariance estimation (Cov-SRRR) in literature [13] and robust reduced-rank regression (R4) in literature [14]. However, the Cov-SRRR method does not consider the robustness and the R4 method does not consider the correlations among error terms and the sparsity of the coefficient matrix. They only considered a subset of our objectives. The main contributions of this paper are highlighted as follows.

- In this work, the mean-shift method is used to get a robust estimation that can explicitly detect outliers, account for general correlations among error terms, and improve estimation accuracy by applying shrinkage on the precision matrix. In addition, we also incorporate variable selection in the model estimation, which is done through a penalty function on the coefficient.
- A block coordinate descent algorithm is proposed to obtain parameter estimates of the objective function. The monotonicity of the algorithm is also given.

The rest of the paper is organized as follows: in Section 2, we formally introduce the framework of reduced-rank regression with a sparse mean-shift outlier component and an unknown error covariance matrix. The penalized estimation method is used to get a sparse estimation. A block coordinate descent algorithm is developed in Section 3 to get the estimation of the RRR model proposed in Section 2. The simulation studies are given in Section 4, which compared the proposed model to relevant competitors. A real data application is shown in Section 5. Finally, a summary and brief discussion are given in Section 6.

2. Robust Reduced-Rank Regression Model with Unknown Covariance

Firstly, we write the multivariate regression model in matrix notation as

$$\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{\mathcal{E}} \ \text{ subject to } \ \mathrm{r}(\boldsymbol{B}) = r, \tag{1}$$

resulting in the famous RRR model, where $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)^T$, $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$, $\boldsymbol{y}_i \in \mathbb{R}^m$, $\boldsymbol{x}_i \in \mathbb{R}^p$, $r \leq \min(p, m)$, $\mathrm{r}(\cdot)$ denote rank and $\boldsymbol{B}$ is a $p \times m$ coefficient matrix, and $\boldsymbol{\mathcal{E}} = (\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_n)^T$ is a random $n \times m$ matrix and $\boldsymbol{\varepsilon}_i$ from the $N(\boldsymbol{0}, \boldsymbol{\Sigma}_\varepsilon)$ distribution.

Ref. [17] investigates the outlier detection from the perspective of adding a mean shift parameter for each data point in a one-dimensional response variable. Ref. [14] introduces a multivariate mean-shift regression model for detecting outlier terms. We introduce a

mean-shift regression model that includes multivariate response variable dependence. The mean-shift multivariate response variable regression model can be written as

$$\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{C} + \boldsymbol{\mathcal{E}}, \tag{2}$$

where $\boldsymbol{Y}$ is the $n \times m$ response matrix, $\boldsymbol{X}$ is the $n \times p$ predictor matrix, $\boldsymbol{B}$ is the $p \times m$ coefficient matrix, $\boldsymbol{C}$ is the $n \times m$ matrix which describes the outlying effects on $\boldsymbol{Y}$ and $\boldsymbol{\mathcal{E}}$ is the $n \times m$ error matrix whose rows are independent draws from $N(\boldsymbol{0}, \boldsymbol{\Sigma}_\varepsilon)$ which is a symmetric distribution. The unknown parameters in the model can be collectively written as $(\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{\Sigma}_\varepsilon)$ or $(\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{\Omega})$, where $\boldsymbol{\Omega} = \boldsymbol{\Sigma}_\varepsilon^{-1}$ denotes the precision matrix. The assumption that errors are correlated is necessary for many applications, for example, when the predictors are not sufficient to eliminate all the correlations between the dependent variables. Obviously, this leads to an over-parameterized model, so we must regularize the unknown matrices appropriately. We assume that $\boldsymbol{B}$ has a low-rank structure and $\boldsymbol{C}$ is a sparse matrix with only a few nonzero terms due to their inconsistency with the majority of the data. So the following sparse robust reduced-rank regression problem is proposed

$$\begin{aligned} \min_{\boldsymbol{B},\boldsymbol{C},\boldsymbol{\Omega}} \frac{1}{n}\mathrm{tr}\{(\boldsymbol{Y} - \boldsymbol{XB} - \boldsymbol{C})\boldsymbol{\Omega}(\boldsymbol{Y} - \boldsymbol{XB} - \boldsymbol{C})^T\} - \log|\boldsymbol{\Omega}| + \lambda_1 \sum_{j \neq j'} |\omega_{j'j}| \\ + P(\boldsymbol{B}; \lambda_2) + P(\boldsymbol{C}; \lambda_3) \ \text{ subject to } \ r(\boldsymbol{B}) \leq r, \end{aligned} \tag{3}$$

where $P(\boldsymbol{B}; \lambda_2)$ and $P(\boldsymbol{C}; \lambda_3)$ are the penalty functions for $\boldsymbol{B}$ and $\boldsymbol{C}$. The penalty parameters are λ_2 and λ_3, respectively. The term $\omega_{j'j}$ is the (j', j) entry in $\boldsymbol{\Omega}$ and the lasso penalty on the off-diagonal entries of $\boldsymbol{\Omega}$ has two reasons. First, it ensures the optimal solution of $\boldsymbol{\Omega}$ has a finite objective function value when there are more responses than samples; second, the penalty has the effect of getting a sparse estimation. The penalty of $P(\boldsymbol{B}; \lambda_2)$ encourages sparsity estimation of coefficient matrix $\boldsymbol{B}$. The term of $P(\boldsymbol{C}; \lambda_3)$ is used to get a sparse estimation of $\boldsymbol{C}$. The λ_1, λ_2 and λ_3 are the tuning parameters. Based on the symmetry of multivariate normal distribution, the covariance matrix or precision matrix must be a symmetric matrix, which reduces the number of parameters and improves the efficiency of parameter estimation. Following reference [18], we estimate the precision matrix using the Gaussian likelihood together with a sparsity-inducing L_1-penalty on all of its entries

$$\min_{\boldsymbol{\Omega}} \mathrm{tr}(\boldsymbol{S}_e\boldsymbol{\Omega}) - \log|\boldsymbol{\Omega}| + \lambda_1 \sum_{j \neq j'} |\omega_{j'j}| \tag{4}$$

where tr is the trace of the matrix and $\boldsymbol{S}_e = \frac{1}{n}(\boldsymbol{Y} - \boldsymbol{XB} - \boldsymbol{C})^T(\boldsymbol{Y} - \boldsymbol{XB} - \boldsymbol{C})$.

As shown in the above formula $\mathrm{rank}(\boldsymbol{B}) = r$ and $r \leq \min(p, m)$ resulting in the reduced-rank regression (RRR) model given in book [19]. The reduced-rank restriction means a number of linear constraints on regression coefficients which can reduce the number of parameters and improve the estimation efficiency. According to the rank constraint $\boldsymbol{B}$ can be expressed as a product of two rank r matrices as follows

$$\boldsymbol{B} = \boldsymbol{SV}^T. \tag{5}$$

Inspired by the penalized regression with a grouped lasso penalty in literature [12,20] consider the following optimization problem

$$\begin{aligned} \min_{\boldsymbol{S},\boldsymbol{V}} \frac{1}{n}\mathrm{tr}(\boldsymbol{Y} - \boldsymbol{XSV}^T)^T(\boldsymbol{Y} - \boldsymbol{XSV}^T) + \sum_{j=1}^{p} \lambda_j ||\boldsymbol{S}^j||_2 \\ \text{subject to } \ \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_r. \end{aligned} \tag{6}$$

where $\boldsymbol{S}^j$ is the jth row of $\boldsymbol{S}$, λ_j is the corresponding tuning parameter and $\boldsymbol{I}_r$ is a unit matrix of order r. The sparsity of the rows means that the unselected variables are not associated with any response variable. The element-wise sparsity penalty can also be applied to $\boldsymbol{S}$.

These two types of sparsity can be blended, firstly, with row sparsity employed in a screening step to reduce p to some dimension d ($d < p$) desired, and then, the element-wise sparsity pursuit in each individual loading vector. In this paper, similar to sparse reduced rank regression (SRRR) in literature [12], only the row-wise sparsity is considered for $\boldsymbol{S}$.

The problem (3) can be expressed as

$$\min_{\boldsymbol{S},\boldsymbol{V},\boldsymbol{C},\boldsymbol{\Omega}} f = \frac{1}{n}\mathrm{tr}\{(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{S}\boldsymbol{V}^T-\boldsymbol{C})\boldsymbol{\Omega}(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{S}\boldsymbol{V}^T-\boldsymbol{C})^T\} - \log|\boldsymbol{\Omega}| + \lambda_1\sum_{j\neq j'}|\omega_{j'j}| + P(\boldsymbol{S};\lambda_2) + P(\boldsymbol{C};\lambda_3) \text{ subject to } \boldsymbol{V}^T\boldsymbol{\Omega}\boldsymbol{V} = \boldsymbol{I}_r. \quad (7)$$

As mentioned above, the assumptions of the model are very mild and, therefore, it can be applied to a wide range of applications. In addition, the model parameters are appropriately regularized, so the model fitting should be robust and the result has well interpretations. As is given in Theorem 2 by paper [14], when the covariance matrix is not considered, the mean-shift model is equivalent to the robust M-estimation problem for $\boldsymbol{B}$. There is no doubt that both simulation studies and the form of model expressions can show the robustness of the method.

3. Computational Algorithm

This section presents an algorithm for solving the optimization problem (7), iteratively optimizing with respect to $\boldsymbol{S}$, $\boldsymbol{V}$, $\boldsymbol{C}$ and $\boldsymbol{\Omega}$.

For given ($\boldsymbol{S}^{[t]}$, $\boldsymbol{V}^{[t]}$, $\boldsymbol{C}^{[t]}$), the problem of solving for the precision matrix $\boldsymbol{\Omega}$ can be expressed as

$$\min_{\boldsymbol{\Omega}} \mathrm{tr}(\boldsymbol{S}_R\boldsymbol{\Omega}) - \log|\boldsymbol{\Omega}| + \lambda_1\sum_{j\neq j'}|\omega_{j'j}|, \quad (8)$$

where $\boldsymbol{S}_R = \frac{1}{n}(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{S}^{[t]}\boldsymbol{V}^{[t]^T}-\boldsymbol{C}^{[t]})^T(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{S}^{[t]}\boldsymbol{V}^{[t]^T}-\boldsymbol{C}^{[t]})$. Reference [21] describes that this problem can be solved by applying the DP-GLASSO algorithm in the R package dpglasso. The DP-GLASSO algorithm optimizes the primal objective with the actual precision matrix and the GLASSO algorithm given in reference [22] can solve the dual of the graphical lasso penalized likelihood with the covariance matrix. However, reference [21] shows that the DP-GLASSO algorithm performs better than the GLASSO algorithm.

For fixed $\boldsymbol{\Omega}^{[t+1]}$ and $\boldsymbol{C}^{[t]}$, the objective of minimization over $(\boldsymbol{S},\boldsymbol{V})$ is

$$\frac{1}{n}\mathrm{tr}\{(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{S}\boldsymbol{V}^T-\boldsymbol{C}^{[t]})\boldsymbol{\Omega}^{[t+1]}(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{S}\boldsymbol{V}^T-\boldsymbol{C}^{[t]})^T\} + P(\boldsymbol{S};\lambda_2), \quad (9)$$

subject to the constraint $\boldsymbol{V}^T\boldsymbol{\Omega}^{[t+1]}\boldsymbol{V} = \boldsymbol{I}_r$. Applying the transformation $\tilde{\boldsymbol{V}} = (\boldsymbol{\Omega}^{[t+1]})^{1/2}\boldsymbol{V}$, we have $\tilde{\boldsymbol{V}}^T\tilde{\boldsymbol{V}} = \boldsymbol{I}_r$. The first term in the objective function (9) can be written as

$$\begin{aligned} & \frac{1}{n}\mathrm{tr}\{(\boldsymbol{Y}(\boldsymbol{\Omega}^{[t+1]})^{1/2}-\boldsymbol{C}^{[t]}(\boldsymbol{\Omega}^{[t+1]})^{1/2}-\boldsymbol{X}\boldsymbol{S}\tilde{\boldsymbol{V}}^T)(\boldsymbol{Y}(\boldsymbol{\Omega}^{[t+1]})^{1/2}-\boldsymbol{C}^{[t]}(\boldsymbol{\Omega}^{[t+1]})^{1/2}-\boldsymbol{X}\boldsymbol{S}\tilde{\boldsymbol{V}}^T)^T\} \\ = & \frac{1}{n}||\tilde{\boldsymbol{Y}}-\boldsymbol{X}\boldsymbol{S}\tilde{\boldsymbol{V}}^T||^2 \\ = & H(\boldsymbol{S},\boldsymbol{V}), \end{aligned} \quad (10)$$

where $\tilde{\boldsymbol{Y}} = \boldsymbol{Y}(\boldsymbol{\Omega}^{[t+1]})^{1/2}-\boldsymbol{C}^{[t]}(\boldsymbol{\Omega}^{[t+1]})^{1/2}$. When $P(\boldsymbol{S};\lambda_2) = \sum_{i=1}^{p}\lambda_2||\boldsymbol{s}_i||_2$, then the problem (9) becomes a SRRR problem given in literature [12], which can be solved iteratively. For the sparsity-inducing penalty functions $P(\boldsymbol{S};\lambda_2)$, there are a lot of choices. The ℓ_1 penalty in reference [23] is most popular among the sparse penalty literature but suffers from inconsistency and biased estimates, especially when predictors are correlated. To alleviate those problems, some non-convex penalties such as the $\ell_p(0 < p < 1)$ penalty, SCAD and the capped ℓ_1 penalty are proposed to approximate the ideal non-convex $\ell_0(||\boldsymbol{S}||_0 = \sum_{i,j}1_{s_{ij}\neq 0})$ penalty.

To solve the S-optimization problem for a general penalty function $P(t;\lambda)$, we propose to use a threshold rule based $\boldsymbol{\Theta}$-estimator as multiple penalty functions usually correspond to a threshold rule. The threshold-based iterative selection procedure (TISP) in reference [24] can be used to solve the penalty problem for any P associated with a threshold rule (an odd, unbounded monotone shrinkage function) given in reference [17].

According to reference [25], $\boldsymbol{\Theta}$-estimator is linked to a general penalty function $P(t;\lambda)$ by

$$p(t;\lambda) - p(0;\lambda) = \int_0^{|t|} (\sup\{s : \boldsymbol{\Theta}(s,\lambda) \leq u\} - u)\mathrm{d}u + q(t;\lambda), \tag{11}$$

for some nonnegative $q(\cdot;\lambda)$ such that $q(\boldsymbol{\Theta}(s,\lambda);\lambda) = 0$ for all s. Any thresholding rule can be solved by this method. We can apply this technique to solve all popular penalties including but not limited to the aforementioned ℓ_1, SCAD, ℓ_0 and $\ell_p (0 < p < 1)$. This means this method is universally valid. The partial derivative of the function $H(\boldsymbol{S},\boldsymbol{V})$ at $\boldsymbol{S}$ can be calculated:

$$\nabla_S H(\boldsymbol{S},\boldsymbol{V}) = \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{S}(\boldsymbol{V})^T - \tilde{\boldsymbol{Y}})\boldsymbol{V}. \tag{12}$$

The closed-form solution for $\boldsymbol{S}$ given $\boldsymbol{V}^{[t]}$ is

$$\boldsymbol{S}^{[t+1]} = \boldsymbol{\Theta}(\boldsymbol{S}^{[t]} - \nabla_S H(\boldsymbol{S}^{[t]}, \boldsymbol{V}^{[t]})/\rho; q_g), \tag{13}$$

where $\rho > ||\boldsymbol{X}^T\boldsymbol{X}||_2^2$ and $\boldsymbol{\Theta}$ is the thresholding function corresponding to penalty $P(\boldsymbol{S};\lambda_2)$. The further details are given in the literature [25].

When $\boldsymbol{S}^{[t+1]}$ is given, the optimization problem of $\boldsymbol{V}$ becomes

$$\min_{\boldsymbol{V}} ||\tilde{\boldsymbol{Y}} - \boldsymbol{X}\boldsymbol{S}^{[t+1]}\tilde{\boldsymbol{V}}^T||_F^2 \text{ subject to } \tilde{\boldsymbol{V}}^T\tilde{\boldsymbol{V}} = \boldsymbol{I}_{r\times r}. \tag{14}$$

This can be identified as a Procrustes rotation problem, realizable through computing the singular value decomposition of $\tilde{\boldsymbol{Y}}^T\boldsymbol{X}\boldsymbol{S} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{Q}^T$. The optimal $\boldsymbol{V} = (\boldsymbol{\Omega}^{[t+1]})^{-1/2}\boldsymbol{P}\boldsymbol{Q}^T$. The full algorithm is presented in Algorithm 1.

For fixed $\boldsymbol{\Omega}^{[t+1]}$, $\boldsymbol{S}^{[t+1]}$ and $\boldsymbol{V}^{[t+1]}$, the problem of solving $\boldsymbol{C}$ reduces to the minimization of

$$F(\boldsymbol{C}) = \frac{1}{n}\mathrm{tr}\{(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{S}^{[t+1]}\boldsymbol{V}^{[t+1]^T} - \boldsymbol{C})\boldsymbol{\Omega}^{[t+1]}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{S}^{[t+1]}\boldsymbol{V}^{[t+1]^T} - \boldsymbol{C}\} + P(\boldsymbol{C};\lambda_3). \tag{15}$$

Suppose $\ell(\boldsymbol{C}) = \frac{1}{n}\mathrm{tr}\{(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{S}^{[t+1]}\boldsymbol{V}^{[t+1]^T} - \boldsymbol{C})\boldsymbol{\Omega}^{[t+1]}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{S}^{[t+1]}\boldsymbol{V}^{[t+1]^T} - \boldsymbol{C})^T\}$. To solve problem (15), construct a surrogate function

$$g(\boldsymbol{C};\boldsymbol{C}^{[t]}) = \ell(\boldsymbol{C}^{[t]}) + \langle\nabla\ell(\boldsymbol{C}^{[t]}), \boldsymbol{C} - \boldsymbol{C}^{[t]}\rangle + \frac{\rho}{2}||\boldsymbol{C} - \boldsymbol{C}^{[t]}||_F^2 + P(\boldsymbol{C};\lambda_3), \tag{16}$$

where $\nabla\ell(\boldsymbol{C}^{[t]})$ is the gradient of ℓ with respect to $\boldsymbol{C}$. Now define the $(t+1)$th iterate as

$$\boldsymbol{C}^{[t+1]} = \mathrm{argmin}_{\boldsymbol{C}} g(\boldsymbol{C};\boldsymbol{C}^{[t]}). \tag{17}$$

The problem of minimizing $\boldsymbol{C}$ boils down to the g-optimization in (17), which is simpler than direct minimizing F. We rewrite the problem in the form of

$$\min_{\boldsymbol{C}} \frac{\rho}{2}||\boldsymbol{C} - \boldsymbol{C}^{[t]} - \frac{1}{\rho}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{S}^{[t+1]}\boldsymbol{V}^{[t+1]^T} - \boldsymbol{C}^{[t]})\boldsymbol{\Omega}^{[t+1]}||_F^2 + P(\boldsymbol{C};\lambda_3). \tag{18}$$

To solve this problem (18), for a general penalty function $P(\boldsymbol{C};\lambda_3)$, we still propose to use the thresholding rule based Θ-estimators. The ℓ_0 penalty has the appealing properties of promoting sparsity and can directly restrict the cardinality of nonzero elements/rows in the estimation matrix. Moreover, it is easy to see that tuning for the constraint form $\sum_{i,j} 1_{\boldsymbol{C}_{ij}\neq 0}/(pm) \leq q_e$ is intuitive and easy compared to the penalty form $\lambda||\boldsymbol{C}||_0$. Here q_e is

used as an upper bound for the percentage of nonzero terms. In this paper, to guarantee the sparsity of $\boldsymbol{C}$, we advise using ℓ_0 constraint in place of the ℓ_0 penalty $\lambda||\boldsymbol{C}||_0$ for tuning ease. By employing a quantile thresholding rule $\boldsymbol{\Theta}^\sharp$, this problem can be solved through TISP easily. The quantile thresholding rule is a special case of hard thresholding which correspond ℓ_0 penalty. Let $\boldsymbol{\Theta}^\sharp(\boldsymbol{C}; q_e)$ be the element-wise quantile threshold function, then

$$\boldsymbol{\Theta}^\sharp(\boldsymbol{C}; q_e) = \begin{cases} 0, & |c_{ij}| \leq \lambda_e \\ c_{ij}, & |c_{ij}| > \lambda_e, \end{cases} \tag{19}$$

where λ_e is the $(1-q_e)^{th}$ quantile of the $|c_{ij}|$. The tuning parameter q_e has a defined range $(0 < q_e < 1)$ as an upper bound for the nonzero percentage, which is easier to interpret.

The row sparsity regularization can be used for variable selection in our multiple response problem and is used in our main algorithm. In particular, for the row sparsity regularization ℓ_0 constraint $||\boldsymbol{C}||_{2,0}/p \leq q_g$, it calls for the multivariate version of quantile thresholding $\overrightarrow{\boldsymbol{\Theta}}^\sharp(\boldsymbol{c}; q_g)$:

$$\overrightarrow{\boldsymbol{\Theta}}^\sharp(\boldsymbol{c}; q_g) = \begin{cases} 0, & ||\boldsymbol{c}||_2 \leq \lambda_g \\ \boldsymbol{c}, & ||\boldsymbol{c}||_2 > \lambda_g, \end{cases} \tag{20}$$

where λ_g is the $(1-q_g)^{th}$ quantile of the norm of the vector $\boldsymbol{c}$. The row-wise constraint version can be realized by simply changing $\boldsymbol{\Theta}^\sharp$ to $\overrightarrow{\boldsymbol{\Theta}}^\sharp$.

Both the row-wise sparsity and element-wise sparsity are considered for $P(\boldsymbol{C}; \lambda_3)$ to find the outlier rows and outlier elements, respectively.

Assuming that $P(\boldsymbol{C}; \lambda_3)$ is constructed by (11), our algorithm iteratively applies the DP-GLASSO, the SRRR, and simple multivariate thresholding operations until the function (7) has converged. The complete numerical algorithm is summarized in Algorithm 1.

Algorithm 1 Algorithm to solve Cov-SR4

Require: $\boldsymbol{Y} \in \mathbb{R}^{n\times m}$, $\boldsymbol{X} \in \mathbb{R}^{n\times p}$, r: the desired rank; M: the maximum iteration number; ε: error tolerance; and the initial estimates $\boldsymbol{S}^{[0]} \in \mathbb{R}^{p\times r}$, $\boldsymbol{V}^{[0]} \in \mathbb{R}^{m\times r}$, $\boldsymbol{C}^{[0]} \in \mathbb{R}^{n\times m}$.

1: $t \leftarrow 0$;
2: **repeat**
3: for fixed $(\boldsymbol{S}^{[t]}, \boldsymbol{V}^{[t]}, \boldsymbol{C}^{[t]})$, solve $\boldsymbol{\Omega}$ with objective function (8) using the DP-GLASSO algorithm;
4: for fixed $(\boldsymbol{\Omega}^{[t+1]}, \boldsymbol{C}^{[t]})$, solve $(\boldsymbol{S}, \tilde{\boldsymbol{V}})$ defined by the SRRR problem (11).
5: for fixed $(\boldsymbol{S}^{[t+1]}, \boldsymbol{V}^{[t+1]}, \boldsymbol{\Omega}^{[t+1]})$, the multivariate thresholding operations can be used to solve problem (18) to get $\boldsymbol{C}^{[t+1]}$;
6: $t \leftarrow t+1$;
7: Calculate $f^{[t+1]}$ and $f^{[t]}$ according to Equation (7);
8: **until** $t \geq M$ or $|f^{[t+1]} - f^{[t]}| \leq \varepsilon$
9: **return** $\boldsymbol{S}^{[t+1]}$, $\boldsymbol{V}^{[t+1]}$, $\boldsymbol{C}^{[t+1]}$ and $\boldsymbol{\Omega}^{[t+1]}$.

Theorem 1. *Assume the largest eigenvalue of $\boldsymbol{\Omega}^{[t]}$ equals to L. Then, as long as $\rho \geq L$, the sequence of iterates defined by (1) satisfies*

$$f(\boldsymbol{S}^{[t+1]}, \boldsymbol{V}^{[t+1]}, \boldsymbol{C}^{[t+1]}, \boldsymbol{\Omega}^{[t+1]}) \leq f(\boldsymbol{S}^{[t]}, \boldsymbol{V}^{[t]}, \boldsymbol{C}^{[t]}, \boldsymbol{\Omega}^{[t]}). \tag{21}$$

That is, the objective function values are non-increasing during the iteration.

Proof. Firstly, for fixed $\boldsymbol{S}^{[t]}$, $\boldsymbol{V}^{[t]}$ and $\boldsymbol{C}^{[t]}$, the optimization for $\boldsymbol{\Omega}$ with objective function (8) using the DP-GLASSO algorithm is a block coordinate descent (BCD) which is given in reference [21].

Secondly, for fixed $\boldsymbol{\Omega}^{[t+1]}$, $\boldsymbol{V}^{[t]}$ and $\boldsymbol{C}^{[t]}$, the optimization for $\boldsymbol{S}$ with closed-form solution (13) is equivalent to the minimization of following equation

$$h(\boldsymbol{S};\boldsymbol{S}^{[t]}) = H(\boldsymbol{S}^{[t]},\boldsymbol{V}^{[t]}) + \langle \nabla_{\boldsymbol{S}} H(\boldsymbol{S}^{[t]},\boldsymbol{V}^{[t]}), \boldsymbol{S}-\boldsymbol{S}^{[t]}\rangle + \frac{\rho}{2}||\boldsymbol{S}-\boldsymbol{S}^{[t]}||_F^2 + P(\boldsymbol{S};\lambda_2). \quad (22)$$

Let $G(\boldsymbol{S}) = H(\boldsymbol{S},\boldsymbol{V}^{[t]}) + P(\boldsymbol{S};\lambda_2)$. Then

$$G(\boldsymbol{S}^{[t+1]}) \le h(\boldsymbol{S}^{[t+1]};\boldsymbol{S}^{[t]}) \le h(\boldsymbol{S}^{[t]};\boldsymbol{S}^{[t]}) \le G(\boldsymbol{S}^{[t]}), \quad (23)$$

where the first inequality comes from the Taylor expansion and $\rho > ||\boldsymbol{X}^T\boldsymbol{X}||_2^2$, and the second inequality holds because $\boldsymbol{S}^{[t+1]}$ is a minimum point of $h(\boldsymbol{S};\boldsymbol{S}^{[t]})$. The monotonicity of V-optimality is due to the fact that it is a closed form minimization.

The monotonicity of C-optimality can be deduced from the following inequality

$$F(\boldsymbol{C}^{[t+1]}) \le g(\boldsymbol{C}^{[t+1]};\boldsymbol{C}^{[t]}) \le g(\boldsymbol{C}^{[t]};\boldsymbol{C}^{[t]}) \le F(\boldsymbol{C}^{[t]}), \quad (24)$$

where the first inequality comes from the Taylor expansion and $\rho \ge L$, and the second inequality holds because $\boldsymbol{C}^{[t+1]}$ is a minimum point of $g(\boldsymbol{C};\boldsymbol{C}^{[t]})$. □

4. Simulation

4.1. Related Methods

In this section, the simulation method is used to illustrate the proposed method (Cov-SR4) and compare it with some related methods. In particular, we compare Cov-SR4 with SRRR, Cov-SRRR, and R4. For the methods of SRRR and R4, the R package rrpack is available. Ref. [13] gives two Cov-SRRR methods according to two cross-validation criteria for tuning. The first one (Cov-SRRR1) is based on likelihood and the second one (Cov-SRRR2) is based on the prediction error. Based on the similar results in literature [13], for convenience, only the Cov-SRRR2 is considered in our paper. To ensure comparability of model effects and algorithm times, all algorithms were rewritten in R and the R packages CVXR and glasso were used in the optimization algorithm. In our simulation, SRRR and R4 assume uncorrelated and homogeneous errors, which means the unweighted residual sum of squares is used to measure the goodness-of-fit. For Algorithm 1, the stopping criterion is based on the difference between the current and previous objective functions. That is, the algorithm used stops if

$$|f(\boldsymbol{S}^{[t]},\boldsymbol{V}^{[t]},\boldsymbol{C}^{[t]},\boldsymbol{\Omega}^{[t]}) - f(\boldsymbol{S}^{[t-1]},\boldsymbol{V}^{[t-1]},\boldsymbol{C}^{[t-1]},\boldsymbol{\Omega}^{[t-1]})| \le \delta, \quad (25)$$

where $\delta = 10^{-4}$ is used for our simulation study.

4.2. Simulation Setups

In the simulation study, the training data are generated using the model $\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{C} + \boldsymbol{\mathcal{E}}$ and the prediction data are generated using the model $\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{\mathcal{E}}$. The design matrix $\boldsymbol{X}$ is generated by sampling its n rows from multivariate normal distribution $N(\boldsymbol{0},\boldsymbol{\Sigma}_x)$, where $\boldsymbol{\Sigma}_x(i,j) = 0.5^{|i-j|}$. The rows of the $n \times m$ random noise matrix $\boldsymbol{\mathcal{E}}$ are generated from $N(\boldsymbol{0},\sigma^2\boldsymbol{\Sigma}_\varepsilon)$. The error structure of $\boldsymbol{\Sigma}_\varepsilon$ is $\boldsymbol{\Sigma}_\varepsilon(i,j) = \rho_\varepsilon^{|i-j|}$, denoted by ar($\rho_\varepsilon$). The corresponding precision matrix $\boldsymbol{\Omega}$ is a tri-diagonal matrix and therefore sparse. We set $\rho_\varepsilon = 0$, 0.5 or 0.9, representing independent errors, moderate correlation or strongly correlated errors. In each simulation, σ^2 is calculated to control the signal-to-noise ratio. In our paper, σ^2 is chosen to let trace$(\boldsymbol{C}^T\boldsymbol{\Sigma}_x\boldsymbol{C})$/trace$(\boldsymbol{E}^T\boldsymbol{E})$ equal 1.

According to reference [10], the coefficient matrix $\boldsymbol{B}$ has the low-rank structure $\boldsymbol{B} = \boldsymbol{SV}^T$ and row-wise sparsity. For the $p \times r$ component matrix $\boldsymbol{S}$, the elements of its first p_0 rows are generated from $N(0,1)$ and the rest $p - p_0$ rows are set to be zero. The elements of $m \times r$ matrix $\boldsymbol{V}$ are generated from $N(0,1)$.

All the elements of the first q row of $\boldsymbol{C}$ are added by 10. This yields some outliers with high leverage values. Overall, the signal is contaminated by both random errors and gross outliers. Under each setting, the entire data generation process described above is replicated 50 times. The rank for all the methods in our simulation is set to the true value. To choose the optimal combination of parameters for the Cov-SR4 method, cross-validation (CV) would seem to be an option. However, it lacks theoretical support in the robust low-rank setting, and for three tuning parameters, cross-validation can be quite expensive. As used in reference [14], the predictive information criterion is used for tuning. The five-fold CV is used for tuning the parameters for the other methods.

4.3. Performance Evaluation

In our simulation, the performance of prediction error and variable selection accuracy is considered. We compared the predictive error of these methods in terms of the mean squared prediction error (MSE), defined as

$$\text{MSE} = \text{tr}||(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{B}})||^2/(mn), \tag{26}$$

where $\hat{\boldsymbol{B}} = \boldsymbol{S}^{[t]}\boldsymbol{V}^{[t]^T}$ and the algorithm stop at step t. As used in reference [13], two other aspects of the row-wise variable selection for the coefficient matrix $\boldsymbol{B}$ and the element-wise variable selection for the precision matrix $\boldsymbol{\Omega}$ are also considered. The first one is sensitivity, the proportion of non-zero elements which are correctly selected and the second one is specificity, the proportion of zero elements that are correctly excluded.

4.4. Simulation Results

The data size of the prediction data set is the same as the training data set, which is used for the evaluation of the prediction error. The five-fold cross-validation is used to choose tuning parameters for all methods based on the prediction error.

We show the prediction errors in Table 1. Firstly, we consider the result without outliers. When $\rho_\varepsilon = 0$, these four methods shows the similar prediction error results. However, when $\rho_\varepsilon = 0.5$ or 0.9, the Cov-SR4 is similar to Cov-SRRR2 and these two method shows better performance than the other two methods because they do not consider the dependency between $\boldsymbol{Y}$ except the predictor matrix $\boldsymbol{X}$. When outliers are considered, Cov-SR4 and R4 show better results than the other two methods because highly leveraged outliers have a significant influence on the other two methods. The mean shift method can stably capture outliers. The Cov-SR4 and R4 show robust estimation results and the performance of Cov-SR4 is slightly better than R4.

Table 1. Comparison of four methods in terms of prediction errors with rank $r = 3$. The means and SEs (in parentheses) are given based on 50 simulation runs.

Parameters						**Predictions**			
n	p	p_0	m	q	Σ_e	**Cov-SR4**	**SRRR**	**R4**	**Cov-SRRR2**
50	10	5	5	0	ar(0)	0.448(0.272)	0.548(0.212)	0.370(0.228)	0.452(0.274)
					ar(0.5)	0.361(0.324)	0.470(0.247)	0.355(0.259)	0.369(0.254)
					ar(0.9)	0.312(0.270)	0.478(0.242)	0.329(0.250)	0.315(0.268)
50	10	5	5	1	ar(0)	0.484(0.240)	0.636(0.203)	0.542(0.211)	1.056(0.552)
					ar(0.5)	0.394(0.234)	0.571(0.205)	0.488(0.207)	0.802(0.465)
					ar(0.9)	0.343(0.181)	0.524(0.188)	0.428(0.184)	0.851(0.665)
100	20	10	5	0	ar(0)	0.275(0.035)	0.402(0.047)	0.267(0.029)	0.276(0.036)
					ar(0.5)	0.330(0.091)	0.451(0.056)	0.344(0.082)	0.332(0.092)
					ar(0.9)	0.507(0.098)	0.626(0.086)	0.408(0.112)	0.512(0.095)
100	20	10	5	1	ar(0)	0.395(0.120)	0.584(0.133)	0.425(0.170)	1.292(0.311)
					ar(0.5)	0.417(0.143)	0.578(0.179)	0.416(0.151)	1.835(1.445)
					ar(0.9)	0.368(0.151)	0.561(0.153)	0.403(0.177)	1.312(0.773)

The results in terms of variable selection of coefficient matrix and precision matrix are given in Table 2. Here, sen is sensitivity and spe is specificity. Their formulas can be found in reference [26]. Firstly, the variable selection of coefficient matrix are considered. The method of R4 in reference [14] does not consider the sparsity of coefficient matrix and the function r4 in R package rrpack to calculate R4 estimation does not give a sparsity coefficient matrix estimation. So we do not consider the variable selection of coefficient matrix accuracy of R4. When outliers are not considered, all the other three methods give a good performance in terms of the sensitivity of the coefficient matrix. When $\rho_\varepsilon \neq 0$, the Cov-SR4 and Cov-SRRR2 show similar results about specificity, which are slightly better than SRRR. When outliers are considered, the Cov-SR4 also shows a good performance in terms of sensitivity, while Cov-SRRR2 and SRRR influenced by the outliers show a bad performance in sensitivity. However, in this case, the results regarding specificity are similar. From the simulation study, we see that, on the whole, the Cov-SR4 method always shows satisfactory results. Secondly, the variable selection of the precision matrix is considered. When outliers are considered and $\rho_\varepsilon \neq 0$, the Cov-SR4 shows a better performance than Cov-SRRR2 both in sensitivity and specificity.

Table 2. Comparison of four methods in terms of variable selection accuracy with rank $r = 3$.

Parameters						**Variable Selection**							
n	p	p_0	m	q	Σ_e	**Cov-SR4**		**SRRR**		**R4**		**Cov-SRRR2**	
						Sen	**Spe**	**Sen**	**Spe**	**Sen**	**Spe**	**Sen**	**Spe**
50	10	5	5	0	ar(0)	1	0.96	1	0.98	-	-	1	0.96
					ar(0.5)	0.98	0.98	0.99	0.96	-	-	0.98	1
					ar(0.9)	0.99	1	0.98	0.99	-	-	1	0.99
50	10	5	5	1	ar(0)	1	0.98	1	0.84	-	-	0.98	0.96
					ar(0.5)	0.98	0.98	0.99	0.86	-	-	0.98	1
					ar(0.9)	0.98	0.99	0.94	0.96	-	-	0.93	0.99
100	20	10	5	0	ar(0)	0.99	0.99	1	0.99	-	-	0.99	1
					ar(0.5)	1	0.98	0.99	0.99	-	-	0.99	0.98
					ar(0.9)	1	0.99	1	1	-	-	0.99	1
100	20	10	5	1	ar(0)	1	0.98	0.96	0.99	-	-	0.96	0.99
					ar(0.5)	0.98	0.98	0.99	0.98	-	-	0.97	0.98
					ar(0.9)	1	0.99	1	1	-	-	0.99	1
	parameters				Ω								
50	10	5	5	0	ar(0)	-	0.92	-	-	-	-	-	0.92
					ar(0.5)	0.30	0.90	-	-	-	-	0.32	0.88
					ar(0.9)	0.50	0.68	-	-	-	-	0.52	0.63
50	10	5	5	1	ar(0)	-	0.95	-	-	-	-	-	0.67
					ar(0.5)	0.58	0.80	-	-	-	-	0.52	0.68
					ar(0.9)	0.73	0.57	-	-	-	-	0.65	0.47
100	20	10	5	0	ar(0)	-	0.98	-	-	-	-	-	0.98
					ar(0.5)	0.40	0.98	-	-	-	-	0.42	0.98
					ar(0.9)	0.86	0.58	-	-	-	-	0.86	0.67
100	20	10	5	1	ar(0)	-	0.97	-	-	-	-	-	0.50
					ar(0.5)	0.57	0.87	-	-	-	-	0.55	0.53
					ar(0.9)	0.69	0.71	-	-	-	-	0.56	0.48

The results of the average running time are given in Table 3. Algorithm runtimes are measured in CPU time (seconds). Each algorithm was run on a Windows 10 laptop with an Intel(R) Core(TM) i7-7500U CPU 2.70GHz and 8GB of RAM, and all simulations were computed using R-4.1.0. There is no significant difference in the running time of these models for different values of ρ_ε. Therefore only the results for $\rho_\varepsilon = 0$ are shown.

Because the algorithm of Cov-SR4 is the most complex, its running time is the longest, while the algorithm of SRRR is the simplest, so its running time is the shortest.

Table 3. Comparison of four methods in terms of running time with rank $r = 3$. Average running time(s) comparison based on 50 simulation runs.

n	p	p_0	m	q	Cov-SR4	SRRR	R4	Cov-SRRR2
50	10	5	5	0	64.58	4.44	51.12	47.39
50	10	5	5	1	103.19	4.65	81.51	77.03
100	20	10	5	0	176.41	4.51	116.28	68.22
100	20	10	5	1	241.49	4.47	146.68	123.85

5. Real Data Applications

5.1. Stock Data

The weekly log returns of nine stocks from 2004 available from the R package MRCE in the literature [27] are considered. This data are also analyzed by reference [14]. The data is modeled with a first-order vector autoregressive model both in references [14,27],

$$\boldsymbol{Y} = \tilde{\boldsymbol{Y}}\boldsymbol{B} + \boldsymbol{C} + \boldsymbol{\mathcal{E}}, \tag{27}$$

where $\boldsymbol{Y}$ and $\tilde{\boldsymbol{Y}}$ are $(n-1) \times q$ matrix. $\boldsymbol{Y}$ has rows $\boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$ and the predictor $\tilde{\boldsymbol{Y}}$ has rows $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n-1}$. $\boldsymbol{y}_i$ means the log returns for the nine companies at week i. For the convenience of calculation, we expanded the data by 100 times.

Following the approach of reference [14], using the weekly log-returns in the first 26 weeks ($n = 26$) for training and those in the last 26 weeks for forecast. All of the four methods introduced in our paper are used to analyze this data. All of the methods resulted in unit-rank models. The row-wise outliers detect method is used. The Cov-SR4 method and R4 method automatically detected week sixteen as outliers. This outlier corresponds to a real major market disturbance in the automotive industry. Our robust approach automatically takes into account outlier samples, leading to a more reliable model. The mean squared prediction errors for Cov-SR4, SRRR, R4, and Cov-SRRR2 are 7.008, 7.526, 8.543, and 7.022, respectively. The row sparsity is used to estimate coefficient matrix $\boldsymbol{B}$ in our method. The estimation of the unit lag coefficient matrix $\boldsymbol{B}$ for the approximate Cov-SR4 method is reported in Table 4. The cells on the rows of Exxon, GM, and IBM are zeros in our results which are the same as the coefficient matrix estimated by MRCE (Rothman et al., 2010). The only difference is that the coefficient of Walmart is also zeros in our work. From the coefficient matrix $\boldsymbol{B}$, it can be found that GE, CPhillips, Citigroup, and AIG have a significant effect on the stock prices of all companies.

Table 4. Estimated coefficient matrix $\boldsymbol{B}$ of Cov-SR4 for stock data..

	Wal	Exx	GM	Ford	GE	CPhil	Citi	IBM	AIG
Walmart	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Exxon	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
GM	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Ford	−0.008	−0.004	−0.016	−0.002	−0.006	−0.006	−0.002	−0.005	−0.006
GE	0.047	0.028	0.100	0.012	0.035	0.040	0.009	0.030	0.037
CPhillips	−0.138	−0.082	−0.292	−0.034	−0.103	−0.116	−0.028	−0.089	−0.107
Citigroup	0.133	0.079	0.281	0.033	0.099	0.112	0.027	0.085	0.103
IBM	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
AIG	0.130	0.077	0.276	0.032	0.097	0.110	0.026	0.084	0.101

The estimation for the inverse error covariance matrix for the Cov-SR4 method is reported in Table 5 and this result is also similar to the inverse error covariance matrix

estimated by MRCE in reference [27]. The non-zero term in Table 5 implies that the errors of the two firms are correlated to a given set of errors of the other firms. We see that AIG (an insurance company) is estimated to be partially correlated with most other companies. And companies with similar businesses are also more likely to be partially correlated, such as Ford and GM, both in the automotive industry, GE and IBM, both in the technology industry, and CPhillips and Exxon, both related to the oil industry. These results are meaningful in financial market practical applications.

Table 5. Inverse error covariance estimate of Cov-SR4 for stock data.

	Wal	**Exx**	**GM**	**Ford**	**GE**	**CPhil**	**Citi**	**IBM**	**AIG**
Walmart	1547.025	0.000	−214.459	0.000	0.000	91.522	0.000	0.000	0.000
Exxon	0.000	3037.352	0.000	0.000	0.000	−936.069	0.000	0.000	−45.210
GM	−214.459	0.000	2116.922	−1103.131	−167.819	0.000	−255.571	0.000	−20.347
Ford	0.000	0.000	−1103.131	1007.451	0.000	0.000	0.000	0.000	0.000
GE	0.000	0.000	−167.819	0.000	2031.920	0.000	−118.701	−1013.879	−118.590
CPhillips	91.522	−936.069	0.000	0.000	0.000	2233.820	0.000	0.000	−96.543
Citigroup	0.000	0.000	−255.571	0.000	−118.701	0.000	2918.238	0.000	−257.842
IBM	0.000	0.000	0.000	0.000	−1013.879	0.000	0.000	3203.400	0.000
AIG	0.000	−45.210	−20.347	0.000	−118.590	−96.543	−257.842	0.000	1668.546

5.2. US COVID-19 State-Level Mortality Rate Analysis

As the coronavirus is spreading, causing millions of deaths and hundreds of millions of people getting sick, it is extremely important to know the case fatality rate (CRF). Data comes from the r package covid19nytimes. This package contains the daily number of cases of infections and deaths in various states in the United States. Naive estimates of CFR from the reported numbers of total confirmed cases and total deaths are difficult to interpret due to the advances in treatment technology and differences in medical resources. The daily data is so irregular the features once we are settled on the appropriate lag time, we can look at the fatality rate per identified case. This is one possible measure of fatality rate over time. The 7-day moving averages were used to smooth the series. Divide the number of deaths in a moving average of seven days by the number of infections in a moving average of seven days as the daily mortality rate.

Data from eight states in the southeastern United States are used, such as Alabama, Florida, Georgia, Louisiana, Mississippi, North Carolina, South Carolina, and Tennessee. Data for 110 days from 8 April 2020 to 29 July 2020 are used. This data is also modeled with a first-order vector autoregressive model,

$$\boldsymbol{Y} = \tilde{\boldsymbol{Y}}\boldsymbol{B} + \boldsymbol{C} + \boldsymbol{\mathcal{E}}, \tag{28}$$

where $\boldsymbol{Y}$ and $\tilde{\boldsymbol{Y}}$ are $(n-1) \times q$ matrix. $\boldsymbol{Y}$ has rows $\boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$ and the predictor $\tilde{\boldsymbol{Y}}$ has rows $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n-1}$. $\boldsymbol{y}_i$ means the daily mortality rate for the eight states at day i. For the convenience of calculation, we also expanded the data by 100 times.

Similar to the previous example, the daily mortality rate for the first 60 days ($n = 60$) is used for training, and the daily mortality rate for the 50 days after is used for prediction. All four methods described in our paper are also used to analyze this data. The model is best when the rank of all methods is 6. The row-wise outliers detect method is used. Both Cov-SR4 and R4 methods automatically detect that 25 April 2020 is an outlier. According to the CDC, the outlier mortality rate for that day's data may be due to a decrease in COVID-19 mortality compared to last week but may rise again as more death certificates are counted. The outlier has a surprising effect on both coefficient estimation and model prediction. This can be seen from $||\hat{\boldsymbol{B}} - \tilde{\boldsymbol{B}}||_F / ||\tilde{\boldsymbol{B}}||_F \approx 61.6\%$ and $||\boldsymbol{X}\hat{\boldsymbol{B}} - \boldsymbol{X}\tilde{\boldsymbol{B}}||_F / ||\boldsymbol{X}\tilde{\boldsymbol{B}}||_F \approx 14.5\%$, where $\hat{\boldsymbol{B}}$ and $\tilde{\boldsymbol{B}}$ denote the Cov-SR4 and the Cov-SRRR2 estimates, respectively. The mean squared prediction errors for Cov-SR4, SRRR, R4, and Cov-SRRR2 are 0.130, 0.167, 0.133,

and 0.156, respectively. The robust method of Cov-SR4 and R4 shows better performance. The estimation of the coefficient matrix $\boldsymbol{B}$ for the Cov-SR4 method is reported in Table 6. From the coefficient matrix $\boldsymbol{B}$, a significant positive correlation can be found for CRF in the southeastern states of the United States. The estimation for the inverse error covariance matrix for the Cov-SR4 method is reported in Table 7, which cannot be obtained from the R4 method. The non-zero term in Table 7 implies that Louisiana is partially correlated with most other states. South Carolina is also partially correlated with Alabama and Georgia. These results have practical application to the study of CRF due to COVID-19 in the southeastern United States.

Table 6. Estimated coefficient matrix $\boldsymbol{B}$ of Cov-SR4 for US COVID-19 data.

	Ala	FL	Ge	Lou	Mis	NC	SC	Ten
Alabama	0.550	0.055	0.027	−0.201	−0.076	−0.127	0.015	0.024
Florida	0.207	0.222	0.009	0.016	0.127	−0.021	0.333	−0.074
Georgia	0.037	0.116	0.677	0.032	0.110	0.151	0.031	−0.027
Louisiana	0.088	0.119	−0.044	0.738	−0.001	0.210	0.017	−0.011
Mississippi	−0.125	0.316	0.141	−0.134	0.778	−0.106	−0.225	0.094
North Carolina	−0.090	0.012	0.220	0.932	−0.092	0.380	−0.075	0.051
South Carolina	0.050	0.207	0.008	−0.010	0.088	0.072	0.789	0.025
Tennessee	0.234	−0.206	0.116	0.222	0.078	0.232	0.113	0.826

Table 7. Inverse error covariance estimate of Cov-SR4 for US COVID-19 data.

	Ala	FL	Ge	Lou	Mis	NC	SC	Ten
Alabama	5.119	0.000	0.000	0.000	0.000	0.000	0.188	0.000
Florida	0.000	5.355	0.000	−0.059	0.000	0.000	0.000	0.000
Georgia	0.000	0.000	5.590	0.722	0.000	0.000	−0.097	0.000
Louisiana	0.000	−0.059	0.722	1.482	−0.171	0.000	0.000	0.000
Mississippi	0.000	0.000	0.000	−0.171	7.379	0.000	0.000	0.000
North Carolina	0.000	0.000	0.000	0.000	0.000	7.503	0.000	0.000
South Carolina	0.188	0.000	−0.097	0.000	0.000	0.000	4.171	0.000
Tennessee	0.000	0.000	0.000	0.000	0.000	0.000	0.000	19.458

6. Discussion

In this paper, we propose a block coordinate descent algorithm to solve the sparse robust reduced-rank regression with covariance estimation and prove the monotonicity of the new algorithm. This method can explicitly detect outliers, account for general correlations among error terms, and incorporate variable selection. In the multiple response variable regression model, the new method outperformed other methods in terms of prediction error and variable selection. Both simulation studies and real data analysis demonstrate the effectiveness of the new models and algorithms.

Although the mean-shift method is used to detect the outliers in the new model, in the future, we will consider the general robust loss to conquer leverage outliers. The method of order statistics used in reference [16] is a new approach to outlier detection. In addition, for non-Gaussian distributed multivariate response variables, how to capture the dependence among response variables is a very challenging but widely used scenario. And the application of this dependence to regression models of multivariate non-Gaussian response variables is a topic worthy of further study. As far as I know, there are few studies on statistical tests in this direction.

Author Contributions: Data curation, W.L.; Formal analysis, W.L.; Writing—original draft, W.L. and G.L.; Writing—review and editing, W.L., G.L. and Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: The research is supported by the Natural Science Foundation of China (Nos. 11803159, 11271136, 81530086, 11671303, 11201345, 71771089), the 111 Project of China (No. B14019), the Shanghai Philosophy and Social Science Foundation (No. 2015BGL001) and the National Social Science Foundation Key Program of China (No. 17ZDA091).

Data Availability Statement: Stock data and COVID-19 data are available from the R packages MRCE and covid19nytimes, respectively.

Acknowledgments: The authors thank the editor, the associate editor, and the reviewers for their helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ancha, X.; Shirong, Z.; Yincai, T. A unified model for system reliability evaluation under dynamic operating conditions. *IEEE Trans Reliab.* **2021**, *70*, 65–72.
2. Chunling, L.; Lijuan, S.; Ancha, X. Modelling and estimation of system reliability under dynamic operating environments and lifetime ordering constraints. *Reliab. Eng. Syst. Saf.* **2022**, *218*, 108136.
3. Hu, J.; Chen, P. Predictive Maintenance of Systems Subject to Hard Failure Based on Proportional Hazards Model. *Reliab. Eng. Syst. Saf.* **2020**, *196*, 106707. [CrossRef]
4. Chen, P.; Ye, Z.S. Approximate Statistical Limits for a Gamma Distribution. *Int. J. Qual. Eng. Technol.* **2017**, *49*, 64–77. [CrossRef]
5. Anserson, T.W. Estimating linearrestrictions onregression coefficientsformultivariate normal distributions. *Ann. Math. Stat.* **1951**, *22*, 327–351. [CrossRef]
6. Izenman, A.J. Reduced-rank regression for the multivariate linear model. *J. Multivar. Anal.* **1975**, *5*, 248–264. [CrossRef]
7. Obozinski, G.; Wainwright, M.J.; Jordan, M.I. Support union recovery in high-dimensional multivariate regression. *Ann. Stat.* **2011**, *39*, 1–47. [CrossRef]
8. Negahban, S.; Wainwright, M.J. Estimation of (near) lowrank matrices with noise and high-dimensional scaling. *Ann. Stat.* **2011**, *39*, 1069–1097. [CrossRef]
9. Chen, K., Chan, K.S.; Stenseth, N.C. Reduced rank stochastic regression with a sparse singular value decomposition. *J. R. Stat. Soc. B.* **2012**, *74*, 203–221. [CrossRef]
10. Chen, L.; Huang, J.Z. Sparse Reduced-Rank Regression for Simultaneous Dimension Reduction and Variable Selection in Multivariate Regression. *J. Am. Stat. Assoc.* **2012**, *107*, 1533–1545. [CrossRef]
11. Bunea, F.; She, Y.; Wegkamp, M.H. Optimal selection of reduced rank estimators of high-dimensional matrices. *Ann. Stat.* **2011**, *39*, 1282–1309. [CrossRef]
12. Bunea, F.; She, Y.; Wegkamp, M.H. Joint variable and rank selection for parsimonious estimation of high-dimensional matrices. *Ann. Stat.* **2012**, *40*, 2359–2388. [CrossRef]
13. Chen, L.; Huang, J.Z. Sparse reduced-rank regression with covariance estimation. *Comput. Stat.* **2016**, *26*, 461–470. [CrossRef]
14. She, Y.; Chen, K. Robust reduced-rank regression. *Biometrika* **2012**, *104*, 633–647. [CrossRef]
15. Jäntschi, L. A test detecting the outliers for continuous distributions based on the cumulative distribution function of the data being tested. *Symmetry* **2019**, *11*, 835. [CrossRef]
16. Jäntschi, L. Detecting extreme values with order statistics in samples from continuous distributions. *Mathematics* **2020**, *8*, 216. [CrossRef]
17. She, Y.; Owen, A.B. Outlier detection using nonconvex penalized regression. *J. Am. Stat. Assoc.* **2011**, *106*, 626–639. [CrossRef]
18. Yuan, M.; Lin, Y. Model selection and estimation in the gaussian graphical model. *Biometrika* **2007**, *94*, 19–35. [CrossRef]
19. Reinsel, G.C.; Velu, R.P. Reduced-Rank Regression Model. In *Multivariate Reduced-Rank Regression, Theory and Applications*, 1st ed.; Springer: New York, NY, USA, 1998; pp. 15–55.
20. Yuan, M.; Lin, Y. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. B.* **2006**, *68*, 49–67. [CrossRef]
21. Mazumder, R.; Hastie, T. The graphical lasso: New insights and alternatives. *Electron. J. Stat.* **2012**, *6*, 2125–2149. [CrossRef]
22. Friedman, J.; Hastie, T.; Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **2007**, *9*, 432–441. [CrossRef]
23. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B.* **1996**, 267–288. [CrossRef]
24. She, Y. Thresholding-based iterative selection procedures for model selection and shrinkage. *Electron. J. Stat.* **2009**, *3*, 384–415. [CrossRef]
25. She, Y. An iterative algorithm for fitting nonconvex penalized generalized linear models with grouped predictors. *Comput. Stat. Data An.* **2012**, *56*, 2976–2990. [CrossRef]
26. Trevethan, R. Sensitivity, specificity, and predictive values: Foundations, pliabilities, and pitfalls in research and practice. *Front. Public Health* **2017**, *5*, 307. [CrossRef]
27. Rothman, A.; Levina, E.; Zhu, J. Sparse multivariate regression with covariance estimation. *J. Comput. Graph. Stat.* **2010**, *19*, 947–962. [CrossRef]