Machine Learning application in Multivariate GARCH model [1]

City University of Hong Kong

Jianxiang CHEN

Abstract

The volatility clustering is one of the core observation in financial econometrics and Multivariate GARCH models are well documented in the financial volatility literature. A new estimation methods under the family of VEC model is proposed in this paper. We introduced the penalization into the maximal likelihood function of multivariate model. The innovation provides the model ability to perform well in high dimensional matrix. The model implementation on simulation and empirical financial data are studied to address the model accuracy and robustness. We also embedded the estimation results into GMV portfolio to evaluate the model performance in real market. The sharpe ratio and annualized volatility is shown to verify the model is consistent and well-conditioned numerically.

1 Introduction

The estimation and prediction of time-varying financial volatility is one of the most important issues in financial econometrics. Since the work of ARCH (Engle, 1982)[1], the second-order moment time series models have been widely applied in various fields of finance, such as hedging ratio, portfolio theory, asset pricing and risk measurement. Eventually, financial institutions also developed many portfolio selection strategies based on GARCH-type models. Indeed, the univariate GARCH models need to iteratively estimate volatility many times,

[1]I am willing to express my wondrously thanks to my thesis supervisor Dr. Cui Liyuan for her wise guidance and continuous encouragement. Then I would like to thank the City University of Hong Kong to provides me such an opportunities to apply what I learned from school to experience the financial research.

and the study of MGARCH(multivariate GARCH) models has been widely focused by the academia since late 1980s.

In 1988, Bollerslev et al[2] proposed a general formulation of covariance matrix time series. However, this model suffered from the curse of parameter number, $N(N+1)(N(N+1)+1)/2$, which restricts its application in bivariate case. And it is difficult to permit the positivity of estimated covariance matrices. To overcome this defect, Engle and Kroner(1995)[3] introduced BEKK. It guarantees the positivity from strong assumption on model structure. Later, Alexander et al(1997)[4] proposed the orthogonal GARCH model, van der Weide(2002)[5] proposed the general orthogonal GARCH model, Vrontos et al(2003) proposed the full-factor MGARCH model, Bollerslev(1990)[6] proposed CCC(Constant Conditional Correlation) model and Engle(2002)[7] proposed DCC(Dynamic Conditional Correlation) model. These models permit the positivity and can be applied in larger dimension cases through imposed different constraints. However, since the complexity of financial market, their assumptions limit their feasibility and can only fit the data follows some specific probability distributions. To overcome this issue and extend the model to high dimensional study, we provide a alternative time series model based on the VEC(1988)[2] model.

The paper is organized in the following way. In section 2, we provide an overview of covariance matrix estimation and review some popular covariance matrix models. In section 3, we derived our new model and its estimation method. In section 4 and 5, we evaluate the performance of our model in simulation and empirical studies. In section 6, the conclusions are given.

2 Overview of MGARCH models

Since the article focuses on the estimation methods of covariance of financial random variables, this section will illustrate some of the most popular and fundamental multivariate GARCH models and discuss their assumptions, advantages and disadvantages.

2.1 Model setup

Consider a financial time series $\{y_t\}_{t=1}^{T}$,

$$\begin{aligned} y_t &= \mu_t(\theta) + \varepsilon_t, \quad t = 1, ..., T \\ \varepsilon_t &= H_t^{1/2}(\theta) z_t, \quad t = 1, ..., T, \end{aligned} \tag{1}$$

where θ is an information variable, y_t is an $p \times 1$ vector, $E_{t-1}(y_t) = \mu_t(\theta)$ and we assuming the mean and variance of the normal random variable z_t as following.

$$\begin{aligned} \mathrm{E}\left(z_t\right) &= 0 \\ \mathrm{Var}\left(z_t\right) &= I_N \end{aligned} \tag{2}$$

Therefore, the covariance matrix of y_t can be expressed in the following way,

$$\begin{aligned} \mathrm{Var}\left(y_t | I_{t-1}\right) = \mathrm{Var}_{t-1}\left(y_t\right) &= \mathrm{Var}_{t-1}\left(\varepsilon_t\right) \\ &= H_t^{1/2} \mathrm{Var}_{t-1}\left(z_t\right) \left(H_t^{1/2}\right)' \end{aligned} \tag{3}$$

,the matrix H_t is positive definite.

2.2 Models

(i) VEC Model

The VEC(1,1) model is defined as,

$$h_t = c + A\eta_{t-1} + Gh_{t-1} \tag{4}$$

where,

$$\begin{aligned} h_t &= \text{vech}\left(H_t\right) \\ \eta_t &= \text{vech}\left(\varepsilon_t \varepsilon_t'\right) \\ \text{eigen}(A+G) &< 1 \end{aligned} \tag{5}$$

and vech(.) denotes the operator that stacks the lower triangular portion of a $N \times N$ matrix as a $N(N+1)/2 \times 1$ vector. A and G are square parameter matrices of order $(N+1)N/2$ and c is a $(N+1)N/2 \times 1$ parameter vector. However, the number of parameters needed to estimated is relatively large and can be expressed as follows.

$$\begin{aligned} \text{num}(VEC) &= \text{num}(c) + \text{num}(A) + \text{num}(G) \\ &= \tilde{N}/2 + \tilde{N}^2/4 + \tilde{N}^2/4 \\ &= N(N+1)(N(N+1)+1)/2 \end{aligned} \tag{6}$$

which causes VEC model usually applied in bivariate case. And we assume $\tilde{N} = N(N+1)$.

To reduce the estimation complexity, Bollerslev et al.(1988)[2] proposed an extended form of VEC called DVEC model. The DVEC(1,1) is defined as follows,

$$H_t = C^\circ + A^\circ \odot \left(\varepsilon_{t-1}\varepsilon_{t-1}'\right) + G^\circ \odot H_{t-1} \tag{7}$$

and the symmetric $N \times N$ matrices A°, g° and C° as the matrices implied by the relations $A = \text{diag}\left[\text{vech}\left(A^\circ\right)\right], G = \text{diag}\left[\text{vech}\left(G^\circ\right)\right]$ and $c = \text{vech}\left(C^\circ\right)$. The positive definiteness of H_t can be ensured by the positivity of $A^\circ, G^\circ, C^\circ$

and initial H_0.

(ii) BEKK Model

The BEKK(1,1,K) model[3] is defined as,

$$H_t = C^*C^* + \sum_{k=1}^{K} A_k^* \varepsilon_{t-1}\varepsilon_{t-1}' A_k^* + \sum_{k=1}^{K} G_k^* H_{t-1} G_k^* \tag{8}$$

where C^*, A_k^* and G_k^* are $N \times N$ matrices and the elements of A_k^* and G_k^* and diagonal elements of upper triangular matrix, C^*, are restricted to be positive.

Similar to VEC model, the number of parameters in BEKK model is also very large which equals to N(5N + 1)/2 in with $K = 1$. Analogically, it has diagonal and scalar version.

(iii) CCC Model

The CCC model is defined as,

$$H_t = D_t R D_t = \left(\rho_{ij}\sqrt{h_{iit}h_{jt}}\right) \tag{9}$$

where

$$D_t = \text{diag}\left(h_{11t}^{1/2} \dots h_{NNt}^{1/2}\right) \tag{10}$$

h_{iit} can be defined as any univariate GARCH model, and

$$R = (\rho_{ij})$$

is a symmetric positive definite matrix with $\rho_{ii} = 1, \forall i$.

There are $N(N+5)/2$ parameters in CCC and H_t is positive definite if all N conditional variances are positive and R is positive definite.

(iv) DCC Model

The DCC(Engle,2002)[7] model is defined as,

$$H_t = D_t R_t D_t \tag{11}$$

$$R_t = \text{diag}\left(q_{11,t}^{-1/2} \dots q_{NN,t}^{-1/2}\right) Q_t \, \text{diag}\left(q_{11,t}^{-1/2} \dots q_{NN,t}^{-1/2}\right) \tag{12}$$

where Q_t is a $N \times N$ symmetric positive definite matrix Q_t is expressed as,

$$Q_t = (1 - \alpha - \beta)\bar{Q} + \alpha u_{t-1} u'_{t-1} + \beta Q_{t-1} \tag{13}$$

where $\bar{Q}$ is the $N \times N$ unconditional variance matrix of u_t, and α and β are non-negative scalar satisfying $\alpha + \beta < 1$.

(iv) ISEE Model

The innovated scalable efficient estimation is an important large precision matrix estimation proposed by Fan and Lv(2016)[15]. It assumes that the return of a portfolio follows multivariate normal distribution and each row of its precision matrix has finite nonzero off-diagonal entries which they proofed that the number of nonzero entries should be smaller than a certain value K. There are more detailed assumptions and proofs in their paper. And based on the assumption, Fan derived an efficient penalization method which take the balance of accuracy and computational cost.

Instead of computing the inverse of estimated covariance, they start from calculating the innovated data matrix and try to directly get the precision matrix by conducting the lasso regression on each block of the innovated data. Let's denote the precision matrix as Ω, the innovated data matrix can be written as follows.

$$\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_n)^T = \boldsymbol{\varepsilon}\Omega$$

Then the portfolio is defined as the union of finite subsets, $\bigcup_{l=1}^{L} A_l = \{1, \cdots, p\}$ and $A_l \cap A_m = \emptyset$ for any $1 \leq l \neq m \leq L$. The sub-matrix of innovated data can be estimated by solving the following regression problem.

$$\mathbf{X}_A = \mathbf{X}_{A^c}\mathbf{C}_A + \mathbf{E}_A \tag{14}$$

For $j \in A_l$,

$$\left(\widehat{\boldsymbol{\beta}}_j, \widehat{\theta}_j^{1/2}\right) = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^{p-1}|A|, \sigma \geq 0} \left\{ \frac{\|\mathbf{X}_j - \mathbf{X}_{A^c}\boldsymbol{\beta}\|_2^2}{2n\sigma} + \frac{\sigma}{2} + \lambda \|\boldsymbol{\beta}_*\|_1 \right\} \tag{15}$$

Since $\widehat{\boldsymbol{\Omega}}_{A,A} = \left(n^{-1}\widehat{\mathbf{E}}_A^T\widehat{\mathbf{E}}_A\right)^{-1}$ and $\widehat{\mathbf{E}}_j = \mathbf{X}_j - \mathbf{X}_{A^c}\widehat{\boldsymbol{\beta}}_j$, the innovated matrix equals to $\mathbf{E}_A\boldsymbol{\Omega}_{A,A}$. In the end, we can recover the precision matrix by equation, $\widehat{\Omega}_{\text{ISEE,ini}} = n^{-1}\widehat{\mathbf{X}}^T\widehat{\mathbf{X}}$.

3 Main Result

3.1 General form

$$\text{vech}\left(H_t\right) = \text{vech}\left(C^*\right) + \sum_{i=1}^{q} A_i \,\text{vech}\left(\varepsilon_{t-i}\varepsilon_{t-i}'\right) + \sum_{j=1}^{p} B_j \,\text{vech}\left(H_{t-j}\right) \tag{16}$$

where $C^\star, A$ and B are positive definite matrices combined low rank matrices and diagonal matrices. As for estimation, we applied maximum likelihood estimation and assume that $\{\varepsilon_t\}_{t=0}^T$ follows multivariate normal distribution.

$$l_T(\theta) = -\frac{1}{T}\sum_{t=1}^{T} \log|H_t(\theta)| - \frac{1}{T}\sum_{t=1}^{T} \varepsilon_t' H_t^{-1}(\theta)\varepsilon_t \tag{17}$$ loglik

3.2 Estimation Method

According to the last section, the optimal solution is searched in a general structure on θ,

$$A = A_L + A_D, B = B_L + B_D, \text{ and } C = C_L + C_D, \tag{18}$$

where A_L, B_L and C_L are low-rank positive definite matrices. A_D, B_D and C_D are positive definite diagonal matrices.

3.2.1 ADMM

[16] The estimation procedure can be transferred as following constraint optimization problem.

$$\left(\hat{\zeta}, \hat{\theta}^{+}\right) = \underset{\zeta,\theta}{\operatorname{argmin}} \{ \} \frac{1}{T} \sum_{t=1}^{T} \log |H_t(\theta)| + \frac{1}{T} \sum_{t=1}^{T} \operatorname{tr}\left(H_t^{-1}(\theta) R_t\right) + \lambda_1 \sum_{Z=A,B,C} \|Z_L\| \\ +\lambda_2 \sum_{Z=A,B,C} |Z_D|_1 : \zeta_{Z_L} = Z_L, \zeta_{Z_S} = Z_D, \zeta_{Z_L} \succeq 0, \zeta_{Z_D} \succeq \epsilon I_{d,d}, Z = A, B, C \Bigg\} \tag{19}$$

loss

The augmented Lagrangian function of eq.19 can be written as following and $\mu = (\mu_{A_L}, \mu_{A_D}, \mu_{B_L}, \mu_{B_D}, \mu_{C_L}, \mu_{C_D})$ is its penalty parameter. λ_1 and λ_2 are tuning parameters which should be selected by Cross Validation described

in Section 3.2.4.

$$\begin{aligned}\mathrm{L}(\zeta,\theta;\Lambda) = \frac{1}{T}\sum_{t=1}^{T}\log|H_t(\theta)| + \frac{1}{T}\sum_{t=1}^{T}\mathrm{tr}\left(H_t^{-1}(\theta)R_t\right)\lambda_1 + \sum_{Z=A,B,C}\|Z_L\|_* \\ + \lambda_2\sum_{Z=A,B,C}|Z_D|_1 - \sum_{Z=A,B,C}\langle\Lambda_{Z_L},\zeta_{Z_L}-Z_L\rangle - \sum_{Z=A,B,C}\langle\Lambda_{Z_D},\zeta_{Z_D}-Z_S\rangle \\ + \sum_{Z=A,B,C}\frac{1}{2\mu_{Z_L}}\|\zeta_{Z_L}-Z_L\|_F^2 + \sum_{Z=A,B,C}\frac{1}{2\mu_{Z_D}}\|\zeta_{Z_D}-Z_S\|_F^2\end{aligned} \tag{20}$$

However, the term $\frac{1}{T}\sum_{t=1}^{T}\log|H_t(\theta)|$ is a concave function which is difficult to find its global optimal solution. So, we use the majorization-minimization approach applied by Bien and Tibshirani(2011)[9] in covariance matrix estimation. The method transforms the log-likelihood function with lasso penalty into a convex function through iteratively solving convex approximations to the original nonconvex problem. Specifically, we replaced $\log|H_t(\theta)|$ by the sum of $\log|H_t(\theta_o)|$ and $\mathrm{tr}\left\{H_t^{-1}(\theta_o)\left[H_t(\theta)-H_t(\theta_o)\right]\right\}$ in the last step since $\log|H_t(\theta)| \le \log|H_t(\theta_o)| + \mathrm{tr}\left\{H_t^{-1}(\theta_o)\left[H_t(\theta)-H_t(\theta_o)\right]\right\}$.

$$\begin{aligned}\tilde{\mathbf{L}}(\zeta,\theta;\Lambda) = \frac{1}{T}\sum_{t=1}^{T}\mathrm{tr}\left[H_t^{-1}\left(\hat{\theta}^{(j-1)}\right)H_t(\theta)\right] + \frac{1}{T}\sum_{t=1}^{T}\mathrm{tr}\left(H_t^{-1}(\theta)R_t\right) + \lambda_1\sum_{Z=A,B,C}\|Z_L\|_* \\ + \lambda_2\sum_{Z=A,B,C}|Z_D|_1 - \sum_{Z=A,B,C}\langle\Lambda_{Z_L},\zeta_{Z_L}-Z_L\rangle - \sum_{Z=A,B,C}\langle\Lambda_{Z_D},\zeta_{Z_S}-Z_D\rangle \\ + \sum_{Z=A,B,C}\frac{1}{2\mu_{Z_L}}\|\zeta_{Z_L}-Z_L\|_F^2 + \sum_{Z=A,B,C}\frac{1}{2\mu_{Z_D}}\|\zeta_{Z_D}-Z_D\|_F^2\end{aligned} \tag{21}$$

Therefore, the combination of ADMM and majorization-minimization can be numerically solved by gradient descent method which showed in the next section.

3.2.2 General Gradient Descent

ζ Step:

$$\begin{aligned}
\zeta_{Z_L}^{i+1} &= \arg\min_{\zeta Z_L \succeq 0} L\left(\zeta, \theta^i; \Lambda^i\right) \\
&= \underset{\zeta Z_L \succeq 0}{\operatorname{argmin}} - \left\langle \Lambda_{Z_L}^{(i)}, \zeta_{Z_L} - Z_L^{(i)} \right\rangle + \frac{1}{2\mu_{Z_L}} \left\| \zeta_{Z_L} - Z_L^{(i)} \right\|_F^2 \\
&= \underset{\zeta_{Z_L} \succeq 0}{\operatorname{argmin}} \left\| \zeta_{Z_L} - \left(Z_L^{(i)} + \mu_{Z_L} \Lambda_{Z_L}^{(i)} \right) \right\|_F^2 \\
&= \sum_{k=1}^{d} \max\left(\eta_{k,Z_L}, 0\right) \nu_{k,Z_L} \nu'_{k,Z_L}, \text{ for } Z = A, B, C
\end{aligned} \tag{22}$$

step1

In the last line of eq. 22, we conduct the eigenvalue decomposition. And the $\max\left(\eta_{k,Z_L}, 0\right)$ makes sure that the eigenvalues are positive and the matrix $\zeta_{Z_L}^{i+1}$ is a positive definite matrix.

θ Step:

$$\begin{aligned}
Z_L^{(i+1)} &= \arg\min_{Z_L} \tilde{\mathrm{L}}\left(\zeta^{i+1}, Z_S^i, Z_L; \Lambda^i\right) \\
&= \arg\min_{ZL} \left\{ \tilde{l}_T(\theta) + \lambda_1 \left\| Z_L \right\|_* - \left\langle \Lambda_L^i, \Theta_L^{i+1} \right\rangle + \frac{1}{2\mu_L} \left\| \Theta_L^{i+1} - L \right\|_F^2 \right\} \\
&= \underset{\Omega_{Z_L}}{\operatorname{argmin}} \left\{ \frac{1}{2\Delta_k} \left\| \Omega_{Z_L} - \left(Z_L - \Delta_k \nabla_{Z_L} F(\theta) \right) \right\|_F^2 + \lambda_1 \left\| Z_S \right\|_* \right\}
\end{aligned} \tag{23}$$

Consider the singular value decomposition on $(Z_L - \Delta_k \nabla_{Z_L} F(\theta))$,

$$Z_L^{i+1} := U \operatorname{diag}\left\{ \left[\max\left(\sigma_i - \lambda_1 \Delta_k, 0\right) \right]_{1 \le i \le p} \right\} U' \tag{24}$$

$$Z_S^{(i+1)} := \mathbf{S}\left(Z_S^{(i+1)} - \Delta_k \nabla_{Z_S} F(\theta) \right), \lambda_2 \Delta_k \Big) \tag{25}$$

where $\nabla_{Z_S} F_T(\theta) = \nabla_{Z_S} \tilde{l}_T(\theta) + \Lambda_{Z_S}^{(i)} + \frac{1}{\mu_{Z_S}} \left(Z_S - \zeta_{Z_S}^{(i+1)} \right)$ and $\nabla_Z \bar{l}_T(\theta)$ is computed by majorization-minization(See Appendix 1).

Λ Step:

$$\Lambda_{Z_L}^{i+1} = \Lambda_{Z_L} - \frac{1}{\mu_{Z_L}} \left(\zeta_{Z_L}^{i+1} - Z_L^{i+1} \right) \tag{26}$$

$$\Lambda_{Z_S}^{i+1} = \Lambda_{Z_S} - \frac{1}{\mu_{Z_S}} \left(\zeta_{Z_S}^{i+1} - Z_S^{i+1} \right) \tag{27}$$

3.2.3 Termination condition

Define

$$\begin{aligned} r_i &= \sum_{Z=A_L,B_L,C_L,A_S,B_S,C_S} \zeta_Z^i - Z^i \\ s_i &= \sum_{Z=A_L,B_L,C_L,A_S,B_S,C_S} Z^i - Z^{i-1} \end{aligned} \tag{28}$$

Then stop condition is $\|r_i\| < \varepsilon_r$ and $\|s_i\| < \varepsilon_s$, where both ε_r and ε_s are very small tuning parameters.

3.2.4 Tunning selection

Cross Validation

We set $k \in [-3, 3]$ and $\lambda_1, \lambda_2 \in \{10^k\}_{k=-3,-2,-1,0,1,2,3}$, and CV function given in eq. 29. Then we substitute each pair of λ_1 and λ_2 into the estimation algorithm and select the optimal λ_1 and λ_2 with largest log-likelihood value.

$$CV = \mathrm{L}(\zeta, \theta; \Lambda) \tag{29}$$ gcv

4 Simulation Studies

We evaluated the estimation performance in this section under different dimensions and time-intervals through simulation studies. Since the number of parameters will eventually rise with the increase of dimension which means more

sample points needed in the estimation process, we organized the simulation examples as follows: (i) $P = 30$, $T = 100$ and 500; (ii) $P = 50$, $T = 500$ and 1000; (iii) $P = 100$, $T = 500$ and 1000.

These simulation combinations are intently designed to test the vertical model performance from low-dimension to high-dimension and able to explore the model application in the different portfolios and stock indexes. For example, the first simulation combination can be regarded as the half-year and two-year log-return of DJIA, the second combination can be deemed as a mid-sized investment portfolio, and the last one expressed the potential application on S&P 100 approximately in two years and five years. In the next section, we will introduce the generating methods of simulated log-return and 4 types of covariance structures.

4.1 Data Generating Process

In order to simulate log-return sequences of multiple assets, we follow the setup in model setup section, $\varepsilon_t = H_t^{1/2}(\theta) z_t$. And recall the covariance structure in VEC model, $\text{vech}(H_t) = \text{vech}(C^*) + \sum_{i=1}^{q} A_i \,\text{vech}\left(\varepsilon_{t-i}\varepsilon_{t-i}'\right) + \sum_{j=1}^{p} B_j \,\text{vech}(H_{t-j})$. Then we designed 4 types of $Z = (A, B, C)$ with respect to different market conditions. Let's denote d as the dimension of matrix Z.

DGP 1

In DGP 1, we assume a 3×3 matrix, Ω can be written as,

$$\Omega_{3\times 3} = \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{31} & \omega_{32} & \omega_{33} \end{pmatrix}$$

and $\omega_{ii} \sim U(0,1), \omega_{ij,i\neq j} = 0.2^{|\omega_{ii}-\omega_{jj}|}$. And Σ is a $d \times d$ diagonal matrix which

defined as follows.

$$diag(\Sigma) = (|\sigma_1|, |\sigma_2|, ..., |\sigma_d|), |\sigma_i| \sim N(0, 0.5^2)$$

So, the matrix Z can be expressed in the following equation.

$$Z_d = \boldsymbol{\beta}\Omega\boldsymbol{\beta}' + \Sigma, \text{ for } Z = A, B, C$$

,where $\boldsymbol{\beta} = (\beta_{\mathbf{1}}, \beta_{\mathbf{2}}, \beta_{\mathbf{3}})$ and $\beta_{\mathbf{i}} = (b_1^{(i)}, b_2^{(i)}, ...b_k^{(i)}, ..., b_d^{(i)})$. In addition, $b_k^{(1,2)} \sim U(\beta_{max}, \beta_{min})$ and $b_k^{(3)} \sim N(0, 0.5^2)$ for $i = 1, 2, ..., \alpha$.

DGP 2

In DGP 2, we assume the dimension of Ω becomes an integer, α, randomly selected from 1 to 4. The elements of Ω is still defined as $\omega_{ii} \sim U(0, 1), \omega_{ij,i\neq j} = 0.2^{|\omega_{ii} - \omega_{jj}|}$. And Σ is a $d \times d$ diagonal matrix which defined as follows.

$$diag(\Sigma) = (|\sigma_1|, |\sigma_2|, ..., |\sigma_d|), |\sigma_i| \sim N(0, 0.5^2)$$

So, the matrix Z can be expressed in the following equation.

$$Z_d = \boldsymbol{\beta}\Omega\boldsymbol{\beta}' + \Sigma, \text{ for } Z = A, B, C$$

where $\boldsymbol{\beta} = (\beta_{\mathbf{1}}, ..., \beta_{\alpha})$, $\beta_{\mathbf{i}} = (b_1^{(i)}, b_2^{(i)}, ..., b_d^{(i)})$ and $b_k^{(i)} \sim U(\beta_{max}, \beta_{min})$ for $i = 1, 2, ..., \alpha$.

DGP 3

In DGP 3, we introduce a nonlinear structure about factor $\boldsymbol{\beta}$ and keep the Ω setup unchanged as it in DGP 2. So, the matrix Z can be expressed in the following equation.

$$Z_d = \sum_{k=1}^{2} \boldsymbol{\beta}_k \Omega \boldsymbol{\beta}_k' + \Sigma, \text{ for } Z = A, B, C$$

where $\boldsymbol{\beta_p} = (\beta_{\mathbf{p},\mathbf{1}}, ..., \beta_{\mathbf{p},\alpha})$, $\beta_{\mathbf{p},\mathbf{i}} = (b_{p,1}^{(i)}, b_{p,2}^{(i)}, ..., b_{p,k}^{(i)}, ..., b_{p,d}^{(i)})$ with $b_{1,k}^{(i)} \sim U(\beta_{max}, \beta_{min})$ and $b_{2,k}^{(i)} \sim N(0, 0.5^2)$ for $i = 1, 2, ..., \alpha$.

DGP 4

In DGP 4, the $\boldsymbol{\beta}$ setup is used in DGP 1, which is $\boldsymbol{\beta} = (\beta_{\mathbf{1}}, \beta_{\mathbf{2}}, \beta_{\mathbf{3}})$ and $\beta_{\mathbf{i}} = (b_1^{(i)}, b_2^{(i)}, ...b_k^{(i)}, ..., b_d^{(i)})$. Similarly, $b_k^{(1,2)} \sim U(\beta_{max}, \beta_{min})$ and $b_k^{(3)} \sim N(0, 0.5^2)$ for $i = 1, 2, ..., \alpha$. Then we change the definition of Ω into following expression.

$$\Omega_{3\times 3} = \begin{pmatrix} \omega_{11} & \omega_{12} & 0 \\ 0 & 0 & \omega_{23} \\ \omega_{31} & 0 & 0 \end{pmatrix}$$

where $\omega_{ij} \sim U(0, 1)$ and Σ is a $d \times d$ diagonal matrix which defined as follows.

$$\text{diag}(\Sigma) = (|\sigma_1|, |\sigma_2|, \ldots, |\sigma_d|), |\sigma_i| \sim N\left(0, 0.5^2\right)$$

4.2 Comparison under different measurements

As seen in the tables from 1 to 4 in the Appendix 7.3, there are totally 6 types of performance measurement methods involved. Within the same data generating process, the norms under different dimensions keep the similar value and is restricted in the 10^{-2} magnitude. It implies the new estimation method is robust under various dimensions and provides relatively accurate results. Besides, all the minimal singular value of these matrices are strictly positive, which means they are invertible. And it also displayed our assumption that the estimator is searched with in a convex field of positive definite matrices since

positive definite and non-negative definite matrix has to be non singular.

Finally, the simulation results show that no matter the linearity and symmetry hold or not, the new estimation method always provides a accurate and robust estimator.

5 Empirical Studies

In empirical studies, we select the log returns of S&P 100 component stocks in 2006-2016 to forecast their covariance matrices in 2008-2018. So, there are approximately 2500 trading days and 100 individual assets considered in the sample set. Since the true value of parameter matrices and volatility about market data are unable to observed, we compare the out-of-sample performance of our method with the others through global minimum variance (GMV) portfolio allocation.[10] Then we compute the sharpe ratio and annualized volatility among different models to see advantages and disadvantages of our model. The detailed computation and allocation steps are expressed in the following subsections.

5.1 Global Minimum Variance Portfolio Allocation

Let's consider a portfolio P and denote its weight and covariance matrix by a vector $\mathbf{w} = (w_1, w_2, ...)'$ and a matrix $\mathbf{\Sigma}$. In addition, we define a constant vector $\mathbf{1} = (1, 1, ...)'$. Then we can write the portfolio variance as $\sigma_P^2 = \mathbf{w'\Sigma w}$ and combine the restriction that sum of portfolio weights equals to one. The portfolio allocation problem can be transformed as a constrained optimization problem.

$$\begin{aligned} \min \quad & \mathbf{w}'\boldsymbol{\Sigma}\mathbf{w} \\ \text{w.r.t} \quad & \mathbf{w}^{'}\mathbf{1} = 1 \end{aligned}$$

It is equivalent to minimize the equation $\mathbf{w}'\boldsymbol{\Sigma}\mathbf{w} + \lambda\left(\mathbf{w}'1 - 1\right)$, and take the partial derivatives respect to $\mathbf{w}$.

$$\frac{\partial}{\partial \mathbf{w}}\mathbf{w}'\boldsymbol{\Sigma}\mathbf{w} + \lambda\left(\mathbf{w}'\mathbf{1} - \mathbf{1}\right)$$

$$2\boldsymbol{\Sigma}\mathbf{w} + \lambda\mathbf{1} = \mathbf{0}$$

Then we get $\mathbf{w}' = -\frac{1}{2}\lambda\mathbf{1}'\boldsymbol{\Sigma}^{-1}$ and substitute the result into the constraint, $\mathbf{w}'\mathbf{1} = -\frac{1}{2}\lambda\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1} = 1$. It is obvious that the optimal weight is only related the covariance matrix and can be expressed as $\mathbf{w} = -\frac{\boldsymbol{\Sigma}^{-1}\mathbf{1}}{\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1}}$.

As we said above, the future covariance matrix of market data is not accessible. In order to find the portfolio with the lowest risk, we forecast the covariance matrix using different methods and using those predictors to decide its weight.

5.2 Evaluation of the Model Performance

We download the 10 years prices of S&P component stocks and compute their log returns. Then we rebalance the portfolio in every two years, which is regarding the returns in last two years as sample to train and forecast the covariance matrix sequence in the following two years. Among all the volatility models, we select 6 of them to conduct the comparison.

The first 5 methods are unconditional covariance matrix estimation models and the last is conditional methods. In addition to the models mentioned above, we select four regularization approaches of large covariance matrices.

The LW is a estimator about the linear transformation of sample covariance matrix proposed by Ledoit and Wolf (2004). Universal Thresholding (UT) was proposed by Bickel in 2008. Adaptive Thresholding (AT) was proposed by Cai in 2011. And Multiple Testing (MT) was developed by Bailey, Pesaran and Smith based on universal Thresholding method.

	LW	ISEE	UT	AT	MT	DCC	VEC
annual volatility	0.3280	0.3587	0.3286	0.336311	0.3588	0.3570	0.1845
sharpe ratio	0.2313	0.2018	0.2327	0.2100	0.2015	0.2065	0.4187

[1] In the above table LW is Ledoit and Wolf; DCC is Dynamic Conditional Covariance; DCP is Dynamic Conditional Precision; ISEE is Innovated Scalable Efficient Estimation; UT is Universal Thresholding; AT is Adaptive Thresholding; MT is Multiple Testing.
[2] Since the data is S&P 100 component stock price from 2008 to 2018, we rebalance the portfolio in every 2 years.

According to the above table, we can see that the decrease of volatility usually will bring increase of sharpe ratio. However, it is not always true since the global minimal variance portfolio does not necessarily keep the same expected return. And after the comparison, we find our model provide the best estimation in respect of annualized volatility and sharpe ratio.

6 Conclusion

In this paper, a penalized estimation method under multivariate GARCH family was advocated. Compared to the fundamental VEC model, this model reduced the number of estimated parameters while maintaining the accuracy and extending the dimension of covariance could be estimated. In simulation studies, there are totally 4 types of data generator and 3 time-dimension combinations tested. It proofed the ability of the new model to address different market scenarios and estimating stability.

Empirically, the model was calibrated into S%P 100 component stocks from

2008 to 2018. In order to provide a convincing comparison results, the global minimum variance portfolios were implemented through estimators acquired from those covariance methods. Compared to the other methods mentioned in empirical section, the penalized optimization based estimator is seen to give more robust portfolio weights and relative lower portfolio variance combining the higher sharpe ratios. These results showed our model hold huge potential

Our method seems quite comprehensive which inherent both the merit of conditional covariance models and thresholding strategies. Further studies can be addressed to improving the computation speed and simplifying the optimization methods.

References

[1] 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica 50: 987-1007.

[2] Bollerslev T, Engle RF, Wooldridge JM. 1988. A capital asset pricing model with time varying covariances. Journal of Political Economy 96: 116-131.

[3] Engle R, Kroner FK. 1995. Multivariate simultaneous generalized ARCH.

[4] Alexander CO, Chibumba AM. 1997. Multivariate orthogonal factor GARCH. Mimeo, University of Sussex. Andersen TG, Bollerslev T, Diebold

[5] van der Weide R. 2002 . GO-GARCH: a multivariate generalized orthogonal GARCH model. Journal of Applied Econometrics 17: 549-564

[6] Bollerslev T. 1990. Modeling the coherence in short-run nominal exchange rates: a multivariate generalized ARCH model. Review of Economics and Statistics 72: 498-505.

[7] Engle RF. 2002. Dynamic conditional correlation?a simple class of multivariate GARCH models. Journal of Business and Economic Statistics 20: 339-350.

[8] Cui, L., Hong, Y., Li, Y., & Wang, J. (2019). A Machine Learning Approach to Estimating Large Positive Definite Covariance Matrix of High Frequency Data. Available at SSRN 3414910.

[9] Bien, J., Tibshirani, R. J., 2011. Sparse estimation of a covariance matrix. Biometrika 98, 807-820

[10] Mandelbrot, B. B., The Variation of Certain Speculative Prices, The Journal of Business 36, No. 4, (1963), 394-419

[11] Zhao, Xiaolu and Nolte, Ingmar and Taylor, Stephen J., High-Frequency Covariance Matrix Estimation Using Price Durations (April 14, 2018).

[12] Bauwens, L., Laurent, S., Rombouts, J. (2006). Multivariate GARCH models: A survey. Journal of Applied Econometrics, 21(1), 79-109.

[13] Alexander CO. 2001. Market Models. John Wiley: New York.

[14] Sun, T., & Zhang, C. H. (2012). Scaled sparse linear regression. Biometrika, 99(4), 879-898.

[15] Fan, Yingying; Lv, Jinchi. Innovated scalable efficient estimation in ultra-large Gaussian graphical models. Ann. Statist. 44 (2016), no. 5, 2098–2126.

[16] Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning, 3(1), 1-122.

[17] Bailey, N., Pesaran, M.H. and Smith, L.V. (2018). A Multiple Testing Approach to the Regularization of Large Sample Correlation Matrices.USC.

[18] Cai, T. and Liu, W. (2011). Adaptive Thresholding for Sparse Covariance Matrix Estimation. Journal of the American Statistical Association 106 672 − 684

[19] Ledoit, O. andWolf, M. (2004). AWell-Conditioned Estimator for Large-Dimensional Covariance Matrices.Journal of Multivariate Analysis 88, 365− 411

A Appendix

A.1 A majorization-minimization Approach

Transform the original minimization problem

$$\min L^d = \frac{TN}{2}\log 2\pi + \frac{1}{2}\sum_{t=0}^{T}\log(|H_t(\theta)|) + \frac{1}{2}\sum_{t=0}^{T}\varepsilon_t' H_t(\theta)^{-1}\varepsilon_t \tag{30}$$

into

$$\min \tilde{L} = \frac{T}{2}\sum_{t=0}^{T}\operatorname{tr}\left(H_t^{-1}\left(\theta_0\right)H_t(\theta)\right) - \sum_{t=0}^{T}\operatorname{tr}\left(H_t^{-1}(\theta)\varepsilon_t\varepsilon_t'\right) \tag{31}$$

For each parameters in $Z = (A, B, C^*)$, partial derivatives given below:

$$d\tilde{L} = \operatorname{tr}\left(H_t^{-1}(\theta_0)dH_t(\theta)\right) - \operatorname{tr}\left(H_t^{-1}(\theta)\varepsilon_t\varepsilon_t' H_t^{-1}(\theta)dH_t(\theta)\right) \tag{32}$$

A step

$$\frac{\partial}{\partial A} = \operatorname{vec}\left(H_t^{-1}(\theta_0)\right) \oplus \operatorname{vec}(2O - I) \otimes vec\left(H_{t-1}(\theta)\right)' - \operatorname{vec}\left(H_t^{-1}(\theta)\varepsilon_t\varepsilon_t' H_t^{-1}(\theta)\right) \oplus \operatorname{vec}(2O - I) \otimes vec\left(H_{t-1}(\theta)\right)' \tag{33}$$

B step

$$\frac{\partial}{\partial B} = \operatorname{vec}\left(H_t^{-1}(\theta_0)\right) \oplus \operatorname{vec}(2O - I) \otimes vec\left(\varepsilon_{t-1}\varepsilon_{t-1}'\right)' - \operatorname{vec}\left(H_t^{-1}(\theta)\varepsilon_t\varepsilon_t' H_t^{-1}(\theta)\right) \oplus \operatorname{vec}(2O - I) \otimes vec\left(\varepsilon_{t-1}\varepsilon_{t-1}'\right)' \tag{34}$$

***C** step**

$$\frac{\partial}{\partial C^*} = \operatorname{vec}\left(H_t^{-1}(\theta_0)\right) \oplus \operatorname{vec}(2O - I) - \operatorname{vec}\left(H_t^{-1}(Z)\varepsilon_t\varepsilon_t' H_t^{-1}(\theta)\right) \oplus \operatorname{vec}(2O - I) \tag{35}$$

Note: O is defined as a square matrix with all entries equal to one and I is an identity matrix.

Algorithm 1 Majorization-minimization

```
repeat
    θ ← θ_0
    repeat
        Z = arg  min_θ − L(θ, θ_0)
    until convergence
until convergence
```

Note: The first convergence condition is norm of gradients lower than a very small value or its gradients stop to change. And the second convergence

condition is that the change of parameter matrix Z lower than a certain small value.

A.2 Algorithm

repeat

  $\zeta^{(i+1)} = (Z_L^{(i)} + \mu_{Z_L}\Lambda^{(i)})_+$

  repeat

    $H_t(\theta_0^{(j+1)}) = H_t(\theta^{(j)})$

    repeat

      $Z_L^{(k+1)} = Z_L^{(k)} - \Delta_k \nabla_{Z_L} F(\theta^{(k)})$

      $Z_D^{(k+1)} = Z_D^{(k)} - \Delta_k \nabla_{Z_D} F(\theta^{(k)})$

    until convergence

    $Z_L^{(i+1)} = SVD(Z_L^{(last \quad k)})$

    $Z_D^{(i+1)} = (Z_D^{(last \quad k)})_+$

  until convergence

  $\Lambda_{Z_L}^{(i+1)} = \Lambda_{Z_L}^{(i)} - \frac{1}{\mu_{Z_L}}(\zeta_{Z_L}^{(i+1)} - Z_L^{(i+1)})$

until convergence

where$\nabla_{Z_L} F_T(\theta) = \nabla_{Z_L} \tilde{l}_T(\theta) + \Lambda_{Z_S}^{(i)} + \frac{1}{\mu_{Z_L}} \left(Z_L - \zeta_{Z_L}^{(i+1)} \right)$ and $\nabla_{Z_D} F_T(\theta) = diag(\nabla_{Z_L} F_T(\theta))$.

A.3 Results for Simulation Studies

Table 1: Data Generating Process 1

DGP1						
P = 30	T = 100			T = 500		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.004722266	0.00473434	0.005766681	0.004747535	0.004680533	0.005799696
Sum of the singular values	3.629528109	3.64389767	0.223119758	3.629165881	3.581265546	0.224536937
$\max_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.018598518	0.009138994	0.008621478	0.064888534	0.015776259	0.00806929
$\min_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.007817147	0.007778912	0.007499908	0.007845994	0.007130042	0.007550534
Largest singular value	0.007959403	0.007959645	0.007606974	0.024659598	0.007959733	0.007584013
Smallest singular value	0.00207756	0.004882362	0.005983876	0.002142184	5.65E-05	0.006459631
P = 50	T = 500			T = 1000		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.007977024	0.007971138	0.007758027	0.002975873	0.002982025	0.002919545
Sum of the singular values	10.17240828	10.16155127	0.387881848	3.794178557	3.798373426	0.145488266
$\max_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.053422123	0.017858142	0.009033353	0.012698289	0.018303796	0.005410525
$\min_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.00797318	0.007980203	0.007824189	0.003020716	0.005754435	0.003275095
Largest singular value	0.013426951	0.007975588	0.007773943	0.002981864	0.006996329	0.00313584
Smallest singular value	0.007938974	0.00424921	0.00706683	0.00238522	0.001043712	0.001262535
P = 100	T = 500			T = 1000		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.008015496	0.007858076	0.007743163	0.094994221	0.094993749	0.094942387
Sum of the singular values	39.66195454	39.49218896	0.773430738	479.7139949	479.7100215	9.494054406
$\max_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.199197842	0.123193443	0.017857177	0.168783748	0.171728668	0.103449779
$\min_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.005348753	0.006643605	0.007750774	0.095062637	0.095234851	0.096408915
Largest singular value	0.090867439	0.043090137	0.008078052	0.095	0.095	0.095
Smallest singular value	8.65E-05	0.000483964	0.004909439	0.058994933	0.05502153	0.089054406

Note: The norm was calculated by the difference of true parameter matrices(A, B and C) and estimated parameter matrices($\hat{A}, \hat{B}$ and $\hat{C}$). In addition, all the norms in the above table have been scaled by their dimension.

Table 2: Data Generating Process 2

DGP2						
P=30	T = 100			T = 500		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.020207521	0.004730809	0.005853831	0.060267243	0.060145948	0.077409071
Sum of the singular values	4.34143169	3.639698204	0.226697412	46.40361009	46.27598979	2.99804008
$\max_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	2.376040625	0.033799567	0.007853628	0.109739401	0.114664855	0.100142562
$\min_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.008116831	0.007402967	0.007624961	0.099774976	0.099014283	0.099956912
Largest singular value	0.701433159	0.01028767	0.007639803	0.099995	0.099995	0.099995
Smallest singular value	0.00139806	0.004829876	0.007132455	0.088595788	0.023061722	0.09979284
P=50	T = 500			T = 1000		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.007801917	0.00788592	0.007611757	0.003010112	0.003056531	0.003134074
Sum of the singular values	9.916837705	9.979898545	0.380042032	3.807933384	3.816786796	0.15260954
$\max_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.018214183	0.102027483	0.008356251	0.03139546	0.045050093	0.017579229
$\min_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.006523254	0.00715301	0.007174826	0.011408969	0.01475397	0.00852686
Largest singular value	0.00797125	0.0372589	0.007749396	0.016141282	0.02497113	0.007813606
Smallest singular value	4.30E-05	0.000595461	0.004923962	0.002960858	0.002974207	0.001909422
P=100	T = 500			T = 1000		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.180657147	0.180647964	0.18058626	0.003829198	0.003444327	0.001558683
Sum of the singular values	910.7873525	911.2142087	18.04560567	15.22760152	15.17939968	0.295748217
$\max_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.708340774	0.204182447	0.182471307	0.262821635	0.190869104	0.009006906
$\min_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.164636466	0.175219138	0.180500144	0.079443511	0.05788726	0.004814394
Largest singular value	0.232879815	0.180660522	0.180620826	0.170715465	0.122507789	0.002981548
Smallest singular value	0.003838677	0.137973892	0.177044121	0.002198978	0.002981437	0.001423855

Note: The norm was calculated by the difference of true parameter matrices(A, B and C) and estimated parameter matrices($\hat{A}, \hat{B}$ and $\hat{C}$). In addition, all the norms in the above table have been scaled by their dimension.

Table 3: Data Generating Process 3

DGP3						
P=30	T = 100			T = 500		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.007950406	0.007953086	0.007571031	0.004799684	0.004800551	0.005864812
Sum of the singular values	3.693189361	3.69715133	0.227129341	3.693651898	3.692421173	0.227141431
$\max_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.028089285	0.017175178	0.007770052	0.019307789	0.024649349	0.007783771
$\min_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.007981144	0.007981297	0.007589013	0.008966327	0.008069889	0.007639403
Largest singular value	0.007959869	0.007973718	0.007577799	0.007962573	0.007975504	0.007579774
Smallest singular value	0.000232206	0.003900813	0.007418127	0.00259157	9.67E-05	0.007449341
P=50	T = 500			T = 1000		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.007969815	0.007969737	0.007745173	0.002974405	0.003017254	0.002932923
Sum of the singular values	10.15994831	10.15641026	0.387237034	3.794240447	3.809359134	0.146558249
$\max_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.035098717	0.053812577	0.00877504	0.004017121	0.032190239	0.003858301
$\min_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.014968972	0.013353165	0.007895416	0.002977467	0.012095459	0.003170396
Largest singular value	0.009411622	0.011775923	0.007762738	0.002979917	0.017543958	0.00323604
Smallest singular value	0.003186538	0.000514194	0.007189422	0.00245522	0.00217096	0.002354549
P=100	T = 500			T = 1000		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.002016055	0.002272827	0.001994775	0.003153721	0.003481496	0.001087911
Sum of the singular values	10.09333879	10.17218371	0.199477325	15.13181877	15.21505636	0.301408749
$\max_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.070694914	0.188199268	0.002025803	0.260170679	0.262836049	0.023023277
$\min_{1\le i\le n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.003368714	0.020028515	0.002003533	0.003698564	0.043786537	0.003494422
Largest singular value	0.020764376	0.070611122	0.001995	0.072851284	0.121007203	0.007099162
Smallest singular value	0.001814605	0.001162165	0.001972325	0.002236257	0.000420344	0.001837871

Note: The norm was calculated by the difference of true parameter matrices(A, B and C) and estimated parameter matrices($\hat{A}, \hat{B}$ and $\hat{C}$). In addition, all the norms in the above table have been scaled by their dimension.

Table 4: Data Generating Process 4

DGP4						
P=30	T = 100			T = 500		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.007943749	0.007954959	0.007603008	0.060376563	0.060374002	0.077436174
Sum of the singular values	3.690779199	3.698853931	0.22807709	46.48856066	46.48636566	2.999087369
$\max_{1\leq i\leq n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.026825254	0.01189435	0.008204091	0.114789699	0.12220333	0.101038988
$\min_{1\leq i\leq n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.01052483	0.008278559	0.007889037	0.100833849	0.100076053	0.100102297
Largest singular value	0.007961475	0.007971884	0.007621229	0.099995	0.099995	0.099995
Smallest singular value	0.001947912	0.006316699	0.007165089	0.090898112	0.088707948	0.099237555
P=50	T = 500			T = 1000		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.007976949	0.007969027	0.007757253	0.002975155	0.002976308	0.003113646
Sum of the singular values	10.16874578	10.15964029	0.387831267	3.792260329	3.79479258	0.152299915
$\max_{1\leq i\leq n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.053412102	0.017841656	0.009030912	0.010831043	0.003000261	0.020996829
$\min_{1\leq i\leq n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.00797291	0.007977855	0.007822936	0.00298267	0.002977218	0.005367931
Largest singular value	0.013426262	0.007974724	0.007772153	0.002980637	0.002979216	0.007564281
Smallest singular value	0.007936884	0.00424861	0.00706583	0.000463531	0.002973379	0.002941521
P=100	T = 500			T = 1000		
	A	B	C	A	B	C
$\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lvert a_{ij}\rvert^2\right)^{1/2}$	0.180657147	0.180647964	0.18058626	0.09499507	0.094992104	0.025649174
Sum of the singular values	912.3177298	912.2537817	18.05859142	479.720536	479.6930708	9.489633703
$\max_{1\leq i\leq n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.229919207	0.455256999	0.186376146	0.172073294	0.440445796	0.111295313
$\min_{1\leq i\leq n}(\sum_{j=1}^{n}\lvert a_{ij}\rvert)$	0.180664417	0.180662673	0.18064685	0.09500452	0.09500423	0.095341231
Largest singular value	0.180660548	0.180661868	0.180621647	0.095	0.095	0.095
Smallest singular value	0.162983743	0.099034756	0.177067337	0.065535992	0.038070794	0.084633703

Note: The norm was calculated by the difference of true parameter matrices(A, B and C) and estimated parameter matrices($\hat{A}, \hat{B}$ and $\hat{C}$). In addition, all the norms in the above table have been scaled by their dimension.