Communications for Statistical Applications and Methods
2022, Vol. 29, No. 1, 53–64

https://doi.org/10.29220/CSAM.2022.29.1.053
Print ISSN 2287-7843 / Online ISSN 2383-4757

Sparse vector heterogeneous autoregressive model with nonconvex penalties

Andrew Jaeho Shin[a], Minsu Park[a], Changryong Baek[1,a]

[a]Department of Statistics, Sungkyunkwan University, Korea

Abstract

High dimensional time series is gaining considerable attention in recent years. The sparse vector heterogeneous autoregressive (VHAR) model proposed by Baek and Park (2020) uses adaptive lasso and debiasing procedure in estimation, and showed superb forecasting performance in realized volatilities. This paper extends the sparse VHAR model by considering non-convex penalties such as SCAD and MCP for possible bias reduction from their penalty design. Finite sample performances of three estimation methods are compared through Monte Carlo simulation. Our study shows first that taking into cross-sectional correlations reduces bias. Second, nonconvex penalties performs better when the sample size is small. On the other hand, the adaptive lasso with debiasing performs well as sample size increases. Also, empirical analysis based on 20 multinational realized volatilities is provided.

Keywords: sparse vector heterogeneous autoregressive (VHAR) model, nonconvex penalty, adaptive lasso, smoothly clipped absolute deviations (SCAD), minimax concave penalty (MCP), realized volatility

1. Introduction

The importance of modeling and forecasting multivariate realized volatility is increasing rapidly. It is due to its deep relationship with financial domain. For example, forecasting co-movements of multinational stock markets, investment management, and risk management are deeply related with multivariate volatility. A groundbreaking study by Andersen *et al.* (2003) showed that the realized volatility (RV), defined as 5-minutes-low-frequency sum of intraday squared returns, provides the best approximation of volatility. In this study, we are interested in the high-dimensional modeling of volatility dynamics in RVs such as slow decaying autocorrelations known as long memory.

The heterogeneous autoregressive model (HAR) model first proposed by Corsi (2009) became very popular due to its simple structure but outstanding forecasting performance. The HAR model has been extended by incorporating jumps, leverage effects, GARCH-type errors etc. Kim and Baek (2020) proposed factor augmented HAR model by incorporating multinational RVs by factors. Later, Baek and Park (2020) considered direct modeling of high-dimensional RVs by considering lasso type of penalization based on the vector heterogeneous autoregressive model (VHAR) model given by

$$Y_t^{(d)} = \mathbf{\Phi}^{(d)} Y_{t-1}{}^{(d)} + \mathbf{\Phi}^{(w)} Y_{t-1}{}^{(w)} + \mathbf{\Phi}^{(m)} Y_{t-1}{}^{(m)} + \epsilon_t, \quad \epsilon_t \sim WN(0, \mathbf{\Sigma}), \quad t = 23, \ldots, T, \tag{1.1}$$

This work was supported by the Basic Science Research Program from the National Research Foundation of Korea (NRF-2019R1F1A1057104).

[1] Corresponding author: Department of Statistics, Sungkyunkwan University, 25-2, Sungkyunkwan-ro, Jongno-gu, Seoul 03063, Korea. E-mail: crbaek@skku.edu

Published 31 January 2022 / journal homepage: http://csam.or.kr

where ${Y_t}^{(d)}$ is k-dimensional RV time series at time t, $\mathbf{\Phi}^{(d)}$, $\mathbf{\Phi}^{(w)}$, and $\mathbf{\Phi}^{(m)}$ are $k\times k$ coefficient matrices, $\mathbf{\Sigma}$ is $k \times k$ covariance matrix and

$${Y_{t-1}}^{(w)} = \frac{1}{5}\sum_{j=1}^{5} {Y_{t-j}}^{(d)}, \quad {Y_{t-1}}^{(m)} = \frac{1}{22}\sum_{j=1}^{22} {Y_{t-j}}^{(d)}$$

are respectively the component-wise weekly and monthly RV averages. They showed that the sparse VHAR model improves forecasting and provides insight on the connectedness of multinational RVs.

This paper studies further on the sparse modeling in VHAR. Baek and Park (2020) uses adaptive lasso penalty of Zou (2006) due to its convexity, fastness, and continuity. Despite its advantages, however, lasso is not free from bias, so they proposed debiasing procedure after sparse estimation using adaptive lasso. On the other hand, Fan *et al.* (2001) raised the concern of the effect of the bias, which increases the possibility of incorrect selection of non-zero coefficients, in more general penalized estimators. They proposed the so-called smoothly clipped absolute deviations (SCAD) method by carefully designing the non-convex penalty function and showed that it achieves oracle property under certain conditions. However, due to non-convexity, SCAD has computational and analytical difficulties. Later, Zhang (2010) proposed and studied nearly unbiased penalization method called the minimax concave penalty (MCP), which has fast algorithm in finding solution.

In high-dimensional time series modeling perspective, recently, Zhu (2020) carried out a research concerning sparse vector autoregressive model estimation with nonconvex penalties. Zhu (2020) established some oracle properties of the penalized VAR model estimation with temporal and cross-sectional dependence. However, Zhu (2020) cannot accommodate long memory feature in the model and ignores cross-sectional dependence structure in the estimation, so may not suitable for high dimensional financial time series. This study fills above mentioned gap by considering sparse VHAR modeling with non-convex penalties such as SCAD and MCP for possible bias reduction and consistency. In particular, we consider loss function based on likelihood following the idea of Baek and Park (2020) also inspired by Davis *et al.* (2016).

This paper is organized as follows. Section 2 introduces sparse estimation of VHAR model by applying adaptive lasso, SCAD and MCP penalties. We also provide data adaptive penalty parameter selection method tailored to VHAR. Section 3 is devoted to Monte Carlo simulations to evaluate finite sample performances of the proposed methods. In Section 4, our methods are applied to 20 multinational RVs. We conclude in Section 5 with discussions.

2. Model and methods

2.1. VHAR model

The VHAR model (1.1) can be compactly written as

$$\mathbb{Y} = \mathbb{A}\mathbb{X} + \mathbb{Z}, \tag{2.1}$$

where $\mathbb{Y} = (Y_{23}^{(d)}, \ldots, Y_T^{(d)})$, $\quad \mathbb{A} = (\mathbf{\Phi}^{(d)}, \mathbf{\Phi}^{(w)}, \mathbf{\Phi}^{(m)})$, and $\mathbb{X} = (X_{22}, \ldots, X_{T-1})$, with

$$X_t = \begin{pmatrix} Y_t^{(d)} \\ Y_t^{(w)} \\ Y_t^{(m)} \end{pmatrix}$$

and $\mathbb{Z} = (\epsilon_{23}, \ldots, \epsilon_T)$. Then, vectorization further reduce it as

$$\mathrm{vec}(\mathbb{Y}) = (\mathbb{X}' \otimes I_k)\mathrm{vec}(\mathbb{A}) + \mathrm{vec}(\mathbb{Z}).$$

Define the design matrix $\mathbf{X} = (\mathbb{X}' \otimes I_k)$ which is the kronecker product of transpose of matrix $\mathbb{X}$ and the $k \times k$ identity matrix. Then, the VHAR model (1.1) is represented as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\alpha} + \mathbf{z}, \tag{2.2}$$

where the response vector is $\mathbf{Y} = \mathrm{vec}(\mathbb{Y})$, the parameter vector $\boldsymbol{\alpha} = \mathrm{vec}(\mathbb{A})$, and the error vector $\mathbf{z} = \mathrm{vec}(\mathbb{Z})$.

Now, the penalization method can be represented as

$$\hat{\alpha} = \operatorname*{arg\,min}_{\alpha \in \mathbb{R}^q} \frac{1}{N} \left\| \mathbf{Y} - \mathbf{X}\boldsymbol{\alpha} \right\|_2^2 + \sum_{j=1}^{q} p_\lambda(|\alpha_j|), \tag{2.3}$$

where $q = 3k^2$ is the number of parameters, $p_\lambda(\cdot)$ is the penalty function and $\lambda \geq 0$ is the tuning parameter. We consider three penalization methods, adaptive lasso proposed by Zou (2006), SCAD studied by Fan *et al.* (2001) and MCP of Zhu (2020) as detailed in the below.

2.2. Adaptive Lasso penalty

First, note that the ℓ_1 penalty, $p_\lambda(|\alpha_j|) = \lambda|\alpha_j|$, is the original lasso estimator. The adaptive lasso proposed by Zou (2006) improves original lasso estimation by considering weight in the penalty given as

$$p_\lambda(|\alpha_j|) = \lambda w_j |\alpha_j|.$$

Zou (2006) showed that the weight vector is a key factor for its oracle properties and consistency. The weight w_j is proposed as

$$w_j = \frac{1}{\left|\hat{\alpha}_j\right|^\gamma}, \tag{2.4}$$

where $\hat{\alpha_j}$ is j^{th}component parameter estimates for some $\gamma > 0$. It is important to note that the adaptive lasso estimation function (2.4) is a convex optimization problem. Thus, it does not have multiple local minimum issue and shows advantage in computation.

In practice, to derive the weights (2.4), we can use the standard lasso estimator or least squares estimator (OLS),

$$\hat{\boldsymbol{\alpha}}^{\mathrm{OLS}} = \operatorname*{arg\,min}_{\alpha} \left\| \mathbf{Y} - \mathbf{X}\boldsymbol{\alpha} \right\|_2^2 = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \tag{2.5}$$

as an initial estimator to calculate the weights. However, this procedure assumes the innovation covariance matrix as an identity matrix rather than $\boldsymbol{\Sigma}$, thus neglects the cross-sectional dependence. To incorporate the cross-sectional dependence, Davis *et al.* (2016) modified the adaptive lasso estimation function as

$$\hat{\alpha}^{\mathrm{AL}} = \operatorname*{arg\,min}_{\alpha} \left\| \left(\boldsymbol{I_T} \otimes \boldsymbol{\Sigma}^{-\frac{1}{2}}\right)\mathbf{Y} - \left(\mathbf{X}' \otimes \boldsymbol{\Sigma}^{-\frac{1}{2}}\right)\alpha \right\|_2^2 + \lambda \sum_{j=1}^{q} \mathbf{w}_j |\alpha_j|. \tag{2.6}$$

In fact, the loss function of (2.6) is derived from the negative Gaussian likelihood, so it extends usual adaptive lasso to likelihood-based time series setting. The solution is found by applying iterative procedure between $\boldsymbol{\Sigma}$ and $\boldsymbol{\alpha}$. For instance, we can use a function of OLS estimator,

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{T-22}\left(\mathbb{Y} - \hat{\mathbb{A}}\mathbb{X}\right)\left(\mathbb{Y} - \hat{\mathbb{A}}\mathbb{X}\right)' \tag{2.7}$$

for innovation covariance matrix $\boldsymbol{\Sigma}$, and find $\boldsymbol{\alpha}$ by applying standard adaptive lasso algorithm. Then, the covariance matrix (2.7) is updated with new estimator and so on. The detailed derivation of (2.6) and algorithm can be found in the Appendix of Davis *et al.* (2016).

However, Fan *et al.* (2001) asserted that l_1 penalty gives larger bias for larger non-zero coefficients. Thus, in this paper, in order to reduce the bias which comes from the penalization, we adopt the two-stage estimation procedure of Baek and Park (2020) , originally proposed in Baek *et al.* (2018). First, sparse locations are estimated via adaptive lasso estimator $\hat{\boldsymbol{\alpha}}^{AL}$. Then we derive the final estimates of parameters by constrained generalized least squares (GLS) estimation. By following the derivation in Lütkepohl (2005) , final generalized least squares (GLS) estimator is given as

$$\hat{\boldsymbol{\alpha}}^{\mathrm{GLS}} = R\left((\mathbf{X}R)'\left(I_{T-22}\otimes\hat{\Sigma}^{-1}\right)(\mathbf{X}R)\right)^{-1}(\mathbf{X}R)'\left(I_{T-22}\otimes\hat{\Sigma}^{-1}\right)\mathbf{Y}, \tag{2.8}$$

where the sparsity constraint is $\boldsymbol{\alpha} = R\boldsymbol{\gamma}$. For example,

$$\alpha := \begin{pmatrix}\alpha_1\\ \alpha_2\\ 0\\ \alpha_3\end{pmatrix} = \begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{pmatrix} =: R\boldsymbol{\gamma}.$$

This shows that $\boldsymbol{\gamma}$ carries only the non-zero coefficients of the VHAR model.

2.3. Smoothly clipped absolute deviations (SCAD) estimation

For the SCAD estimation for VHAR, the squared error loss is given by

$$\left\|\left(\boldsymbol{I_T}\otimes\boldsymbol{\Sigma}^{-\frac{1}{2}}\right)\mathbf{Y} - \left(\mathbf{X}'\otimes\boldsymbol{\Sigma}^{-\frac{1}{2}}\right)\alpha\right\|_2^2, \tag{2.9}$$

to incorporate spatial dependence. The penalty function $p_\lambda(\cdot)$ proposed by Fan *et al.* (2001), can be written as

$$\frac{d}{d\theta}\mathbf{p}_\lambda^{\mathrm{SCAD}}(\theta) = \lambda\left\{I(\theta\le\lambda) + \frac{(\gamma\lambda-\theta)_+}{(\gamma-1)\lambda}I(\theta>\lambda)\right\} = \begin{cases}\lambda, & |\theta|\le\lambda,\\ \frac{\gamma\lambda-|\theta|}{\gamma-1}, & \lambda<|\theta|<\gamma\lambda,\\ 0, & \gamma\lambda\le|\theta|,\end{cases} \tag{2.10}$$

for some $\gamma > 2$ and $\theta > 0$. This term can be rewritten as quadratic spline function with knots at λ and $\gamma\lambda$

$$\mathbf{p}_\lambda^{\mathbf{SCAD}}(\theta) = \begin{cases}\lambda|\theta|, & |\theta|\le\lambda,\\ \frac{2\gamma\lambda|\theta|-\theta^2-\lambda^2}{2(\gamma-1)}, & \lambda<|\theta|<\gamma\lambda,\\ \frac{\lambda^2(\gamma+1)}{2}, & \gamma\lambda\le|\theta|,\end{cases} \tag{2.11}$$

where λ is the tuning parameter and γ is convexity parameter which controls the convexity of the penalty. In this study, we set $\gamma = 3.7$ following the nonconvex estimation on sparse VAR model by Zhu (2020).

It is notable that SCAD penalty is identical with the lasso until $|\theta| = \lambda$, then it transforms smoothly to a quadratic function until $|\theta| = \gamma\lambda$, after which it gives a constant value for all $|\theta| > \gamma\lambda$. Also, the derivative of the SCAD penalty (2.10) gives more intuitive explanation. The SCAD penalty maintains the penalization rate and bias same as lasso for small coefficients ($\theta| < \lambda$), then steadily relaxes the penalization rate and bias as the absolute value of the coefficient increases.

2.4. Minimax concave penalty (MCP) estimation

For the MCP method for VHAR, the same squared error loss with spatial dependence

$$\left\| \left(\boldsymbol{I_T} \otimes \boldsymbol{\Sigma}^{-\frac{1}{2}}\right) \mathbf{Y} - \left(\mathbf{X}' \otimes \boldsymbol{\Sigma}^{-\frac{1}{2}}\right) \alpha \right\|_2^2 \tag{2.12}$$

is used but the penalty function $p_\lambda(\cdot)$ is given as

$$\frac{d}{d\theta}\mathbf{p}_\lambda^{\text{MCP}}(\theta) = \begin{cases} \operatorname{sign}(\theta)\left(\lambda - \frac{|\theta|}{\gamma}\right), & |\theta| \le \gamma\lambda, \\ 0, & |\theta| > \gamma\lambda, \end{cases} \tag{2.13}$$

where γ is usually set greater than 1. The penalty function (2.13) can be rewritten explicitly as

$$\mathbf{p}_\lambda^{\text{MCP}}(\theta) = \begin{cases} \lambda|\theta| - \frac{\theta^2}{2\gamma}, & |\theta| \le \gamma\lambda, \\ \frac{\gamma\lambda^2}{2}, & |\theta| > \gamma\lambda. \end{cases} \tag{2.14}$$

The MCP penalty is similar to SCAD in the sense that it begins by applying the same rate and bias of lasso penalty. And it also relaxes the rate and bias down to zero as the absolute value of coefficient increases. It is notable that while SCAD reduces the penalization rate and bias smoothly, MCP relaxes them more rapidly (Breheny and Huang, 2011).

In this study, we set $\gamma = 3$ following the default value of the R package *ncvreg* by Breheny and Huang (2011). Many values were tested as γ for both SCAD and MCP to find optimal value in our framework. Values such as $\gamma = 1.5$ (Zhu, 2020) and some other values randomly chosen, but in our setting, $\gamma = 3.7$ for SCAD and $\gamma = 3$ for MCP showed slightly better performance in terms of RMSE and ME, so these values are chosen in this paper.

2.5. Tuning parameter selection

The tuning parameter λ plays a central role in finite sample performances. Two data adaptive methods are used in the study. First, extended bayesian information criterion (eBIC) of Chen and Chen (2008) , which performs better than basic BIC in high dimensional modeling, is adopted. The weight vector for eBIC is given by $\log(\log(p))\log(n)$, where p is dimension and n is sample size. We also adopt block n-fold cross-validation (Block CV) of Baek and Park (2020) to find optimal λ for the sparse VHAR model. The block n-fold cross-validation is a variant of n-fold cross-validation in order to maintain the temporal and spatial dependence structure of time series data. Rather than splitting the sample at random, block CV splits the data into n blocks. For example, block 10-fold cross-validation with sample size $T = 100$ splits the entire data into 10 segments, $(X_1, \ldots, X_{10}), (X_{11}, \ldots, X_{20}), \ldots ,(X_{91}, \ldots, X_{100})$. Then one block of observations will be served as the test set while the rest of blocks are used as training set.

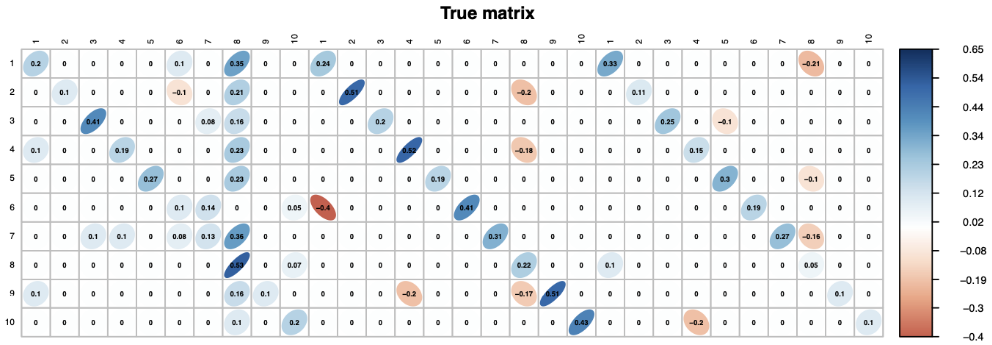

Figure 1: *True coefficient matrix.*

3. Finite sample performances

3.1. Simulation settings

In this section, we examine the finite sample performance of the sparse VHAR modeling methods. Our data generating process (DPG) is based on a 10-dimensional VHAR model with true coefficients as in Figure 1. The coefficients is generated by empirically fitting 10 dimensional RVs, so we tried to mimic long memory and sparsity level of financial market. Also, we consider Gaussian innovations

$$\epsilon_t := (\epsilon_{t,1}, \ldots, \epsilon_{t,10})' \sim N(0, \Sigma_z), \tag{3.1}$$

with two types of covariance matrix. DGP1 assumes covariance matrix as $\Sigma_z = I_{10}$, therefore, there are no spatial correlations. On the other hand, DGP2 assumes covariance matrix as

$$\mathbf{\Sigma_z} = \left(\begin{array}{cccccccccc} \delta^2 & \delta/12 & \delta/12 & \delta/16 & \delta/16 & \delta/20 & \delta/20 & \delta/24 & \delta/24 & \delta/28 \\ \delta/12 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \delta/12 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \delta/16 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \delta/16 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \delta/20 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \delta/20 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \delta/24 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \delta/24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \delta/28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

and cross-sectional correlations are considered and controlled by δ. This covariance matrix is partly referred from Davis *et al.* (2016) and it is adjusted to satisfy the consistency assumption of innovation covariance matrix $\mathbf{\Sigma} = (\sigma_{ij})$ (Baek and Park, 2020),

$$\sigma_{ii} - \sum_{j \neq i} \sigma_{ij} > 0, \quad i = 1, \ldots, k.$$

The finite sample performances are measured by (empirical) root mean squared error (RMSE) of VHAR coefficients and misspecification error (ME). RMSE is defined as

$$\text{RMSE} = \mathbb{E}\|\hat{\alpha} - \alpha\|_2, \tag{3.2}$$

where $\mathbb{E}$ represents empirical average over many replications. We also considered the misspecification error (ME)

$$\text{ME} = \mathbb{E}\|1_{\{\hat{\alpha}=0\}} - 1_{\{\alpha=0\}}\|_2^2/(3k^2), \tag{3.3}$$

Table 1: Summary of performance measures for DGP1

		DGP1 / $\Sigma = I_k$			**DGP1 / update Σ**		
		eBIC			**eBIC**		
		AdaLasso	**SCAD**	**MCP**	**AdaLasso**	**SCAD**	**MCP**
250	**RMSE**	0.1079	0.0895	0.0965	0.1090	0.0889	0.0964
	ME	0.2024	0.1593	0.1783	0.2050	0.1578	0.1766
500	**RMSE**	0.0653	0.0871	0.0946	0.0654	0.0868	0.0942
	ME	0.1336	0.1536	0.1758	0.1344	0.1531	0.1748
1000	**RMSE**	0.0519	0.0857	0.0935	0.0519	0.0857	0.0935
	ME	0.0977	0.1495	0.1743	0.0982	0.1495	0.1743
		BLOCK-CV			**BLOCK-CV**		
		AdaLasso	**SCAD**	**MCP**	**AdaLasso**	**SCAD**	**MCP**
250	**RMSE**	0.0856	0.0821	0.0822	0.0924	0.0804	0.0807
	ME	0.1877	0.1848	0.1622	0.2058	0.1894	0.1652
500	**RMSE**	0.0575	0.0764	0.0766	0.0581	0.0750	0.0752
	ME	0.1453	0.1649	0.1376	0.1504	0.1669	0.1405
1000	**RMSE**	0.0401	0.0722	0.0723	0.0401	0.0715	0.0717
	ME	0.1149	0.1537	0.1193	0.1182	0.1534	0.1191

the misspecification of non-zero coefficients. Here, $1_{\{\alpha=0\}}$ is an indicator function that returns 1 if j^{th}element is zero or returns 0 otherwise. In other words, ME denotes how accurately the sparse VHAR model derives sparseness of the true model. Thus, ME ranges between 0 and 1 and near 0 indicates sparsistency. Each of the Monte Carlo simulations is repeated 500 times respectively. Sample sizes are set as $N = 250, 500$, and 1000. We also considered the case where cross-correlations are not considered in estimation by setting $\Sigma = I_k$ in equations (2.6), (2.9) and (2.12). This is to see the improvement of methods when inter-dependence is directly considered in estimation.

Table 1 summarizes the result for DGP1 which has no spatial dependence. First of all, observe that RMSE or ME tends to decrease as sample size increases. This shows the sparsistency hold for all three proposed penalization methods considered here. Furthermore, since the simulation of DGP1 assumes $\boldsymbol{\Sigma}$ as an identity matrix (no spatial dependence), iterative process that updates the $\boldsymbol{\Sigma}$ does not make a meaningful difference in performance. Regarding tuning parameter selection method, two methods perform similarly, but Block-CV performs slightly better than eBIC overall. More interestingly, observe when the sample size is small as $N = 250$, SCAD and MCP show better performance than the adaptive lasso method. However, when the sample size gets larger and larger ($N = 500, 1000$), adaptive lasso tends to perform better than the other two. Also, it is notable that MCP shows better overall performance than SCAD.

Tables 2–4 summarize the results for DGP2 when the spatial dependence is considered. Cross-sectional correlations are controlled by δ and we set $\delta = 1, 2, 5$, and 10 with sample sizes $T = 250, 500$, and 1000. Note that both RMSE and ME tend to increase as δ increases. For example, when delta is small ($\delta = 1$ or 2), updating $\boldsymbol{\Sigma}$ does not show clear difference in performance. However, when delta is large ($\delta = 5$ or 10), updating $\boldsymbol{\Sigma}$ gives better performance overall. It means that the dependence structure cannot be ignored in estimation, and it is important to incorporate such structure in the model. However, RMSE and ME tend to decrease as sample size increases, so consistency is also achieved in DGP2.

Regarding non-convex penalty methods, we observe that SCAD and MCP performs similar but SCAD performs slightly better. When it comes to the comparison with adaptive lasso with debiasing, it depends on the sample size. That is, non-convex penalty methods outperform adaptive lasso when the sample size is small, $T = 250$ for all δ considered. However, as sample size increases, the two-stage estimation procedure with adaptive lasso shows smaller RMSE and ME. Also, Block-CV

Table 2: Summary of performance measures for DGP2 with sample size 250

		DGP2 / $\Sigma = I_k$			**DGP2 / update Σ**		
		eBIC			**eBIC**		
δ		**Adpative Lasso**	**SCAD**	**MCP**	**Adpative Lasso**	**SCAD**	**MCP**
1	**RMSE**	0.1084	0.0899	0.0969	0.1099	0.0910	0.0977
	ME	0.2044	0.1597	0.1784	0.2086	0.1630	0.1791
2	**RMSE**	0.1162	0.0872	0.0959	0.1114	0.0889	0.0974
	ME	0.2062	0.1561	0.1757	0.2085	0.1609	0.1777
5	**RMSE**	0.2674	0.0952	0.1088	0.1555	0.0925	0.1064
	ME	0.2107	0.1574	0.1729	0.2070	0.1580	0.1746
10	**RMSE**	0.5725	0.1717	0.1833	0.2836	0.1797	0.1847
	ME	0.2165	0.1683	0.1757	0.1623	0.1708	0.1767
		BLOCK-CV			**BLOCK-CV**		
δ		**Adaptive Lasso**	**SCAD**	**MCP**	**Adaptive Lasso**	**SCAD**	**MCP**
1	**RMSE**	0.0867	0.0823	0.0825	0.0934	0.0801	0.0802
	ME	0.1885	0.1854	0.1643	0.2104	0.1882	0.1661
2	**RMSE**	0.0968	0.0815	0.0821	0.0929	0.0792	0.0792
	ME	0.1802	0.1750	0.1604	0.2048	0.1848	0.1651
5	**RMSE**	0.1777	0.1095	0.1147	0.0970	0.0790	0.0799
	ME	0.1614	0.1631	0.1679	0.1935	0.1818	0.1670
10	**RMSE**	0.2889	0.1048	0.1087	0.1101	0.0995	0.1014
	ME	0.2111	0.1594	0.1667	0.1875	0.1944	0.1907

Table 3: Summary of performance measures for DGP2 with sample size 500

		DGP2 / $\Sigma = I_k$			**DGP2 / update Σ**		
		eBIC			**eBIC**		
δ		**Adpative Lasso**	**SCAD**	**MCP**	**Adpative Lasso**	**SCAD**	**MCP**
1	**RMSE**	0.0664	0.0865	0.0939	0.0659	0.0878	0.0949
	ME	0.1364	0.1520	0.1753	0.1372	0.1563	0.1775
2	**RMSE**	0.0674	0.0828	0.0924	0.0662	0.0853	0.0941
	ME	0.1366	0.1462	0.1717	0.1374	0.1541	0.1752
5	**RMSE**	0.1070	0.0880	0.1027	0.0790	0.0880	0.1034
	ME	0.1534	0.1480	0.1686	0.1481	0.1545	0.1733
10	**RMSE**	0.2042	0.1481	0.1580	0.1200	0.1030	0.1055
	ME	0.1691	0.1590	0.1683	0.1626	0.1534	0.1625
		BLOCK-CV			**BLOCK-CV**		
δ		**Adaptive Lasso**	**SCAD**	**MCP**	**Adaptive Lasso**	**SCAD**	**MCP**
1	**RMSE**	0.0576	0.076	0.0761	0.0580	0.0733	0.0735
	ME	0.1481	0.1634	0.1387	0.1550	0.1617	0.1362
2	**RMSE**	0.0622	0.0732	0.0736	0.0586	0.0721	0.0724
	ME	0.1372	0.1521	0.1353	0.1521	0.1623	0.1378
5	**RMSE**	0.1018	0.0908	0.0932	0.0655	0.0683	0.0688
	ME	0.1258	0.1451	0.1487	0.1494	0.1611	0.1449
10	**RMSE**	0.1433	0.1625	0.1661	0.0749	0.0814	0.0824
	ME	0.1350	0.1618	0.1679	0.1430	0.1709	0.1755

method generally shows smaller RMSE and ME regardless of sample sizes and δ.

To summarize, we were able to verify that all three sparse estimations of VHAR considered here showed consistency as sample size increases. However, non-convex penalty methods such as SCAD and MCP performs better in smaller sample, while adaptive lasso with debiasing outperformed other methods as sample size increases. Indeed, it is important to take the cross-correlations structure in the estimation. Also, Block-CV method that considers spatial and temporal dependence worked fine in selecting penalty parameter.

Table 4: Summary of performance measures for DGP2 with sample size 1000

		DGP2 / $\Sigma = I_k$			**DGP2 / update Σ**		
		eBIC			**eBIC**		
δ		**Adpative Lasso**	**SCAD**	**MCP**	**Adpative Lasso**	**SCAD**	**MCP**
1	**RMSE**	0.0524	0.0850	0.0931	0.0518	0.0851	0.0936
	ME	0.1023	0.1473	0.1736	0.1027	0.1478	0.1742
2	**RMSE**	0.0529	0.0801	0.0900	0.0518	0.0806	0.0905
	ME	0.1045	0.1399	0.1705	0.1066	0.1409	0.1708
5	**RMSE**	0.0712	0.0849	0.1011	0.0614	0.0845	0.1008
	ME	0.1252	0.1434	0.1669	0.1263	0.1442	0.1670
10	**RMSE**	0.1172	0.1258	0.1350	0.0842	0.1264	0.1363
	ME	0.1449	0.1466	0.1568	0.1485	0.1467	0.1581
		BLOCK-CV			**BLOCK-CV**		
δ		**Adaptive Lasso**	**SCAD**	**MCP**	**Adaptive Lasso**	**SCAD**	**MCP**
1	**RMSE**	0.0398	0.0718	0.0720	0.0392	0.0691	0.0693
	ME	0.1160	0.1470	0.1140	0.1199	0.1461	0.1150
2	**RMSE**	0.0428	0.0679	0.0682	0.0395	0.0674	0.0676
	ME	0.1070	0.1404	0.1148	0.1148	0.1427	0.1117
5	**RMSE**	0.0695	0.0735	0.0743	0.0462	0.0613	0.0619
	ME	0.0950	0.1363	0.1330	0.1063	0.1371	0.1224
10	**RMSE**	0.1004	0.1399	0.1415	0.0516	0.0716	0.0725
	ME	0.1120	0.1525	0.1600	0.0935	0.1662	0.1889

4. Real data application to RV

In this section, we apply three sparse VHAR estimation methods to 20 multinational RVs and compare their forecasting performances. The data is taken from the Oxford-Man Institute of Quantitative Finance (http://realized.oxford-man.ox.ac.kr). It is calculated based on the aggregation of 5-minute intraday returns. Since multinational stock markets have different opening days, we use linear interpolation to adjust them. The 20 multinational stock indices are S&P 500, FTSE 100, Nikkei 225, DAX, Russell 2000, All Ordinaries, DJIA, Nasdaq 100, CAC 40, Hang Seng, KOSPI, AEX Index, Swiss Market Index, IBEX 35, NIFTY 50, IPC Mexico, IBovespa, S&P/TSX Composite Index, Euro STOXX 50, and FTSE MIB. We considered the period from January 2016 to December 2019; 1087 observations in total. Figure 2 shows the RV time series plot of DJIA, SP500, KOSPI and Nikkei225 stock markets. It clearly shows long non-periodic cyclic behavior which means strong positive long-term correlations.

To evaluate forecasting performance, we split the entire data into two sets. In-sample set is the train set which is the first 1037 periods and out-of-sample set is the test set from the last 50 periods. Then, we use 1-step-ahead out-of-sample forecasting method to obtain forecasts. The overall performance is measured by mean squared prediction error (MSPE), and mean absolute percentage error (MAPE) given by

$$
\begin{aligned}
\text{MSPE} &= \frac{1}{T_0} \sum_{t=T-T_0+1}^{T} \left(Y_t^{(d)} - \hat{Y}_t^{(d)} \right)^2, \\
\text{MAPE} &= \frac{1}{T_0} \sum_{t=T-T_0+1}^{T} \left| \frac{Y_t^{(d)} - \hat{Y}_t^{(d)}}{Y_t^{(d)}} \right|,
\end{aligned}
$$

where T_0 is the out-of-sample(test) set size and $\hat{Y}_t^{(d)}$ is the 1-step-ahead estimated value based on the data from 1 to $t - 1$.

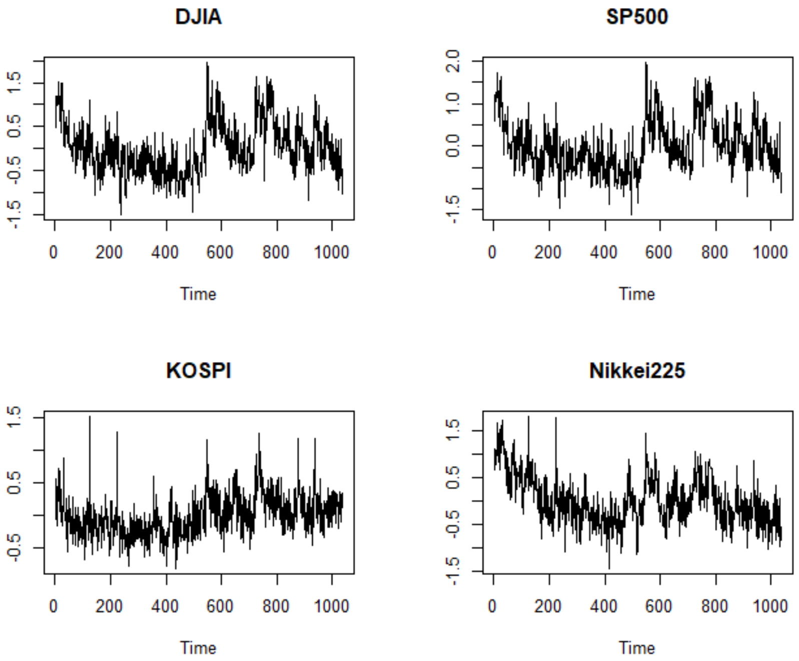

Figure 2: *RV time series plot of DJIA, SP500, KOSPI and Nikkei225.*

Table 5: Forecasting performance of Adaptive Lasso, MCP, and SCAD

	Adaptive Lasso		MCP		SCAD	
	MSPE	MAPE	MSPE	MAPE	MSPE	MAPE
AEX.Index	0.0663	2.6531	0.0705	1.7001	0.0679	1.6877
ALL.Ordinaries	0.0560	2.1621	0.0538	2.4850	0.0575	2.6645
Bovespa.Index	0.0477	1.2657	0.0519	1.1414	0.0518	1.1443
DJIA	0.0827	1.3385	0.0836	1.4511	0.0846	1.4141
CAC.40	0.0604	1.2546	0.0722	1.1056	0.0661	1.2040
FTSE.MIB	0.0659	1.6785	0.0785	1.5099	0.0744	1.5501
FTSE.100	0.0535	1.5471	0.0487	1.5234	0.0485	1.5213
DAX	0.0837	1.3915	0.0893	1.1656	0.0897	1.1640
S.P.TSX.Composite.Index	0.0538	1.4974	0.0569	1.5285	0.0568	1.5256
Hang.Seng	0.0419	2.3259	0.0397	3.3698	0.0414	3.0527
IBEX.35	0.0400	1.6957	0.0478	1.2418	0.0477	1.2430
Nasdaq.100	0.1136	1.1525	0.1199	1.2329	0.1195	1.2239
KOSPI.Composite.Index	0.0356	2.7994	0.0393	3.0088	0.0502	1.5590
IPC.Mexico	0.0441	0.9211	0.0317	1.5395	0.0327	1.4836
Nikkei.225	0.0932	1.3550	0.1018	1.1960	0.0977	1.2371
S.P.CNX	0.0363	2.9844	0.0338	2.0442	0.0320	1.8573
Russel.2000	0.0990	3.5046	0.1055	3.4162	0.1056	3.4271
S.P.500	0.1023	0.8713	0.1013	0.8732	0.1025	0.8813
Swiss.Market.Index	0.0492	5.0024	0.0510	3.8228	0.0507	3.8682
Euro.STOXX.50	0.0802	1.3878	0.0937	1.1979	0.0892	1.2206
Average	0.0653	1.9394	0.0685	1.8277	0.0683	1.7465

Table 5 shows performance measures according to each estimation method. The penalty parameter is chosen by 10-fold block-CV following the result of simulation study. The convexity parameters γ were set as 3.7 and 3 for SCAD and MCP, respectively. It shows some interesting results. First, observe that the MSPE of adaptive lasso is the smallest, but MCP, and SCAD give very similar MSPEs as well. However, nonconvex penalization methods give slightly smaller error in light of MAPE. This may show that it is hard to say one method is superior to others since it depends on the performance measure. That is, the adaptive lasso method may still have some bias even with debiasing procedure. On the other hand, non-convex penalty methods can reduce bias from penalty design. Therefore, ensembling forecasts from all three methods seems to be an interesting open question.

5. Concluding remarks

In this study, we considered sparse VHAR model tailored to high-dimensional long memory time series. In particular, we focused on the SCAD and MCP where the estimation bias is reduced by considering well-designed nonconvex penalty. On the other hand, we also considered two-stage estimation procedure, debiasing with adaptive lasso. We compared them through extensive Monte Carlo simulation and applied to 20 multinational RVs to compare forecasting. Several findings are in order. First, taking into cross-sectional correlations reduced bias in all cases considered. In terms of bias reduction, however, there are some pros and cons depending on methods. Our simulation study shows that nonconvex penalties perform better when the sample size is small while adaptive lasso procedure with debiasing outperforms as sample size increases. Hence, in practice, it is expected that the adaptive lasso with debiasing would be most reliable when the sample size is sufficient. Put it on the other way around, noncovex penalty methods is preferred for a relatively small sample size compared to dimension. From a forecasting perspective, it seems that the risk measure is more important than methods since nonconvex methods give smaller MAPE while adaptive lasso gives the smallest MSPE. Therefore, one can select method depending on their risk measure or ensembling forecast from all three methods can be used as well. Lastly, it remains an interesting future work on how to incorporate other high-dimensional approaches such as applying the dynamic factor approach (Baek *et al.*, 2018), sure independence screening (SIS), elastic net, and many other methods to this domain.

6. Acknowledgement

We would like to thank an Associate Editor and anonymous referees for many useful comments and suggestions.

References

Andersen TG, Bollerslev T, Diebold FX, and Labys P (2003). Modeling and forecasting realized volatility, *Econometrica*, **71**, 579–625.

Baek CR, Davis RA, and Pipiras V (2018). Periodic dynamic factor models: estimation approaches and applications, *Electronic Journal of Statistics*, **12**, 4377–4411

Baek CR and Park MS (2020). Sparse vector heterogeneous autoregressive modeling for realized volatility, *Journal of the Korean Statistical Society*, **50**, 1–16.

Breheny P and Huang J (2011). Coordinate descent algorithms for nonconvex penalized regression with applications to biological feature selection, *The Annals of Applied Statistics*, **5**, 232.

Chen J and Chen Z (2008). Extended Bayesian information criteria for model selection with large model spaces, *Biometrika*, **95**, 759–771.

Corsi F (2009). A simple approximate long-memory model of realized volatility, *Journal of Financial Econometrics*, **7**, 174–196.

Davis RA, Zang PF, and Zheng T (2016). Sparse vector autoregressive modeling, *Journal of Computational and Graphical Statistics*, **25**, 1077–1096.

Fan, Jianqing and Li, Runze (2001). Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American statistical Association*, **96**, 1348–1360.

Kim DW and Baek CR (2020). Factor-augmented HAR model improves realized volatility forecasting, *Applied Economics Letters*, **27**, 1002–1009.

Lütkepohl H (2005). *New Introduction to Multiple Time Series Analysis*, Springer Science & Business Media, Springer, New York.

Zhang CH (2010). Nearly unbiased variable selection under minimax concave penalty, *The Annals of Statistics*, **38**, 894–942.

Zhu X (2020). Nonconcave penalized estimation in sparse vector autoregression model, *Electronic Journal of Statistics*, **14**, 1413–1448.

Zou H (2006). The adaptive lasso and its oracle properties, *Journal of the American Statistical Association*, **101**, 1418–1429.

Received July 02, 2021; Revised August 18, 2021; Accepted September 23, 2021