

# Realized Rotated Multivariate GARCH Model<sup>1</sup>

Denis Pelletier

North Carolina State University<sup>2</sup>

and

Ji Shen

North Carolina State University<sup>3</sup>

June 15, 2015

<sup>1</sup>xxxxxxx.

<sup>2</sup>Department of Economics, Box 8110, North Carolina State University, Raleigh, NC 27695-8110, USA. Email:  
denis.pelletier@ncsu.edu

<sup>3</sup>xxxxxxx

**Abstract**

xxxxxx ?xxxxxxxxxx.

*JEL classification:* C10, C32

*Keywords:* xxxxxx, xxxxxx, xxxxxx.

# 1 Introduction

## 2 Realized Multivariate GARCH model

### 2.1 Definitions and Notations

In this section we introduce the Realized multivariate GARCH model. Assuming we have  $k$  assets,  $r_t$  is the  $k \times 1$  vector of returns for the  $k$  assets in day  $t$ . We will assume  $E(r_t|\mathcal{F}_{t-1}) = 0$ , which is a legitimate assumption from the empirical work. Otherwise we could always reinterpret  $r_t$  as the return minus its constant mean to meet this assumption. The key interest of financial application resides on the conditional variance,  $H_t = \text{var}(r_t|\mathcal{F}_{t-1})$ , where  $r_t$  is a vector of time series of returns,  $\mathcal{F}_{t-1}$  is the filtration till time  $t-1$ . In the GARCH (1, 1) model the conditional variance is a function of lagged itself and squared daily return. In the present framework,  $H_t$  will depend on  $X_{t-1}$ , which represents a realized measure of volatility, such as matrix of realized covariance, bipower variation or realized kernel. The measurement equation link the realized measure to the latent make the model complete. Thus the Realized GARCH model fully specifies the dynamic system of daily returns and realized measure.

The main challenges in multivariate volatility modelling is to ensure that the conditional covariance matrix is positive semidefinite. The BEKK model proposed by Engle and Kroner(1995) could approach this challenge . We could specify a particular BEKK-type system as follows:

$$\begin{aligned} R_t &= H_t^{1/2} Z_t \\ Z_t &\overset{i.i.d}{\sim} \mathcal{N}(0, I_k) \\ E[P_t|\mathcal{F}_{t-1}] &= E[R_t R_t'|\mathcal{F}_{t-1}] = H_t \\ P_t &= H_t^{1/2} \xi_t H_t^{1/2} \end{aligned} \tag{1}$$

where conditional covariance follows a GARCH(1,1) process

$$H_t = CC' + B H_{t-1} B' + A X_{t-1} A' \tag{2}$$

and  $X$  can be updated by the contemporaneous  $H$  in a logarithm way with a linear function  $f(\cdot)$  as follow

$$\text{logm } X_t = f(\text{logm } H_t) \tag{3}$$

We refer to first two equation as the return equation and the Realized GARCH equation. Since the realized measure  $X_t$  could be interpreted as a measurement of conditional covariance  $H_t$ , the last equation is referred as the measurement equation. In (1),  $\xi_t = Z_t Z_t'$  is  $k \times k$  symmetric innovation matrix which is outer product of innovation vector  $Z_t$ , satisfying  $E_{t-1}[\xi_t] = I_k$ , where  $I_k$  is an identity matrix. In (2), we define the symmetric square root of a positive semidefinite matrix  $H$ , denoted by  $H^{1/2}$ , using the spectral decomposition such that  $H^{1/2} = U \Lambda^{1/2} U'$  where  $U$  is a matrix containing the eigenvectors of  $H$ , and  $\Lambda^{1/2}$  is a diagonal matrix containing the square root of the eigenvalues of  $H$ .

### 2.1.1 Principal Matrix Logarithm

In (3), We use  $\logm$  before matrix to represent the principal matrix logarithm. Let  $M$  is a  $k \times k$  positive definite matrix.  $N = \logm(M)$  denotes unique  $k \times k$  matrix such that  $\expm(N) = M$  and every eigenvalue of  $N$  has imaginary part lying strictly between  $-\pi$  and  $\pi$ . So logarithm of a matrix is a generalization of the scalar logarithm and in some sense an inverse function of the matrix exponential. The exponential of  $N$  is the given by the power series

$$\expm(N) = \sum_{i=0}^{\infty} \frac{N^i}{i!}$$

The above series always converges, so the exponential of matrix is well-defined. We calculate logarithm of matrix with a routine matrix function in matlab  $\logm$ .

## 2.2 Parameterization

### 2.2.1 GARCH Equation

The  $k \times k$  matrices  $B$ ,  $A$  each have  $k^2$  free parameters, while  $C$  is  $k \times k$  a lower triangular matrix with  $k^*$  free parameters. The parameterization in (2) guarantee that  $H_t$  is positive semidefinite for all  $t$  assuming the starting value  $H_0$  is positive semidefinite. Moreover, if  $C$  is full rank matrix, then  $H_t$  is positive definite for all  $t$ . The coefficient matrix  $B$  determines the persistence of the Realized MGARCH model.

In the BEKK-type parameterization in (2) has  $O(k^2)$  parameters. To avoid the curse-of-dimensionality problem here we could impose that  $A$  and  $B$  are diagonal matrices, which yields the diagonal Realized MGARCH model. In this case, the equations for each element in the matrix  $H_t$  would characterize a bunch of univariate Realized GARCH models in which the conditional variance or covariance is driven by their own lags and the corresponding realized measure. If the elements of  $A$ ,  $B$  are unrestricted which

is a full parameterization, the evolution of every element in  $H_t$  will be influenced by their own lags as well as every cross-asset effects Realized MGARCH model.

For demonstration, we write out the 2 by 2 case of (2), the diagonality of coefficient matrices are assumed.

$$\begin{pmatrix} h_{11,t} & h_{12,t} \\ h_{21,t} & h_{22,t} \end{pmatrix} = \begin{pmatrix} c_{11} & 0 \\ c_{21} & c_{22} \end{pmatrix} \begin{pmatrix} c_{11} & 0 \\ c_{21} & c_{22} \end{pmatrix}' \\ + \begin{pmatrix} b_{11} & 0 \\ 0 & b_{22} \end{pmatrix} \begin{pmatrix} h_{11,t-1} & h_{12,t-1} \\ h_{21,t-1} & h_{22,t-1} \end{pmatrix} \begin{pmatrix} b_{11} & 0 \\ 0 & b_{22} \end{pmatrix} \\ + \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix} \begin{pmatrix} x_{11,t-1} & x_{12,t-1} \\ x_{21,t-1} & x_{22,t-1} \end{pmatrix} \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix}$$

### 2.2.2 Measurement equation

Note that the BEKK model could be easily written in vectorized model, we write the measurement equation into:

$$x_t = d + \Gamma h_t + u_t \quad (4)$$

We define  $x_t = \text{vech}(\log X_t)$  and similarly  $h_t = \text{vech}(\log H_t)$  where the vech operator stacks the lower triangular part including the main diagonal of a  $(k \otimes k)$  symmetric matrix into a  $(k^* \times 1)$  vector, where  $k^* = k(k+1)/2$ . Moreover,  $\bar{\Gamma} = L_k(\Gamma \otimes \Gamma)D_k$ , where the elimination ( $L_k$ ) and duplication ( $D_k$ ) matrices are linear operational matrices of zeros and ones which could be found in some matrix algebra textbook, like Magnus and Neudecker (1999). Hence,  $x_t$ ,  $d$ ,  $h_t$  and  $u_t$  are all  $k^* \times 1$  vectors, which  $d$  is the intercept vector and  $u_t$  is the innovation vector with mean zero normal distribution. The coefficient matrix  $\Gamma$  is a  $k^* \times k^*$  matrix. This matrix characterize the correlation between realized measure and all the elements of conditional covariance matrix. This vectorized form is easy to estimate since it's similar structure as the seemingly unrelated regression. Although the equation (4) has  $O(k^4)$  parameters, it is an easy task by using computer. The problem of high dimensionality is not about burden of computation but the prediction variation produced by too many parameters. We could use shrinkage and variable selection method in the estimation to reduce elements in coefficient matrix  $\bar{\Gamma}$ .

For deomonstration, we extend the previous example by writting out (4) of 2 by 2 case in vech form

$$\begin{pmatrix} x_{11,t} \\ x_{21,t} \\ x_{22,t} \end{pmatrix} = \begin{pmatrix} d_{11,t} \\ d_{21,t} \\ d_{31,t} \end{pmatrix} + \begin{pmatrix} \gamma_{11,t} & \gamma_{12,t} & \gamma_{13,t} \\ \gamma_{21,t} & \gamma_{22,t} & \gamma_{23,t} \\ \gamma_{31,t} & \gamma_{32,t} & \gamma_{33,t} \end{pmatrix} \begin{pmatrix} h_{11,t} \\ h_{21,t} \\ h_{22,t} \end{pmatrix} + \begin{pmatrix} u_{11,t} \\ u_{21,t} \\ u_{22,t} \end{pmatrix}$$

We could tell from this example that in the measurement equation that each element in the conditional covariance matrix is function of every component in the realized covariance.

### 2.3 Covariance targeting

The covariance targeting parameterization was introduced in Engle and Mezrich (1996) for the univariate GARCH model. The usefulness of covariance targeting in large models is that it allows for 2-step estimation which estimate the unconditional moments with empirical moments. It will substantially reduce the number of parameters to be estimated through the non-linear optimization. Since unconditional covariance of daily return and realized return are  $Var[r_t] = E[H_t] = P$  and  $E[X_t] = M$ . In 2, the covariance-targeting for  $H_t$ :

$$H_t = (P - BPB' - AMA') + B H_{t-1} B + A X_{t-1} A \quad (5)$$

assuming  $(P - BPB' - AMA')$  is positive semidefinite. From 5, it is difficult to specify sufficient parameter restrictions to ensure that  $(P - BPB' - AMA')$  is positive semidefinite. This feature has restrained us from estimate more flexible dynamics in this model other than just assume A and B are scalars. We could fit the model with rotated variables to extend the idea of covariance targeting in multivariate models of any dimension. Specifically, since  $r_t = P^{1/2}z_t$ , we define the rotated return as

$$\begin{aligned} z_t &= P^{-1/2}r_t \\ Var(z_t) &= I_k \end{aligned} \quad (6)$$

The conditional covariance of the rotated returns is  $Var[z_t|\mathcal{F}_{t-1}] \equiv \tilde{H}_t$ , noting that  $H_t = P^{1/2}\tilde{H}_tP^{1/2}$ . Similarly, for realized measure of covariance, we have  $X_t = M^{1/2}\tilde{X}_tM^{1/2}$ . So, we got  $\tilde{H}_t = P^{-1/2}H_tP^{-1/2}$ ,  $\tilde{X}_t = M^{-1/2}X_tM^{-1/2}$ . Basically, we could scale the Realized MGARCH model with the unconditional covariance matrices. A covariance-targeting realized MGARCH model is:

$$\begin{aligned} \tilde{H}_t &= (I_k - \tilde{B}\tilde{B}' - \tilde{A}\tilde{A}') + \tilde{B} \tilde{H}_{t-1} \tilde{B} + \tilde{A} \tilde{X}_{t-1} \tilde{A} \\ E[\tilde{H}_t] &= E[\tilde{X}_t] = I_k \end{aligned} \quad (7)$$

This model has  $2k$  dynamic parameters when  $B$  and  $A$  are diagonal. It is easy to impose the positive semidefinite of  $(I_k - BB' - AA')$  for this specification. For the conditional covariance matrix of unrotated returns  $H_t$ , the BEKK model in 6 and 7 implies the original model in (2) and the relation between coefficients in (2) and (7)

$$B = P^{1/2} \tilde{B} P^{-1/2}, \quad A = P^{1/2} \tilde{A} M^{-1/2}, \quad CC' = P^{1/2} (I_k - BB' - AA') P^{1/2} \quad (8)$$

It is clear the rotated BEKK implies a BEKK specification for the original variables in a constrained version since  $\tilde{A}$  and  $\tilde{B}$  depend on the spectral decomposition of  $P$  and  $M$ . Bauwens et.al(2006) discuss the invariance of multivariate models to linear transformation of return. In their definition, invariance means the transformed model is in the same series and keep the same dynamic specification(e.g scalar,diagonal or full parameterization). In the case of diagonal specification, our model is clearly not invariant to linear transformation. Suppose  $A$  is diagonal, then from 8  $\tilde{A}$  is a full asymmetric matrix, which means we get a full parameterized GARCH model for the unrotated variables. For the rotated model given in 6 and 7, we could easily impose the condition to ensure the long-run covariance stationary and also the positive definite of  $(I_k - BB' - AA')$ .

**Assumption 1** *In the Realized rotated MGARCH model given by (6) and (7),  $\rho(\tilde{A} \otimes \tilde{A} + \tilde{B} \otimes \tilde{B}) \leq 1$ .*

This estimation is easily to impose in the estimation.

## 2.4 The realized kernel

In this section, we outline what is realized kernel and how to calculate it. We assume that the  $k \times 1$  log-price vector,  $P(t)$ , is governed by the following multivariate diffusion process  $dP_t = M(t)dt + H(t)^{1/2}dW(t)$ , where  $M(t)$  and  $H(t)$  denote the  $k \times 1$  instantaneous drift vector and the  $k \times k$  positive definite square-root of the covariance matrix, respectively, while  $W(t)$  denotes a  $N$ -dimensional vector of independent Brownian motions. Traditionally the realized covariance,  $RC(\Delta) \equiv \sum_{j=1}^{N(\Delta)} R_{t-1+j\Delta} R'_{t-1+j\Delta}$  is used as a consistent estimator of integrated covariance of the continuous-time stochastic volatility process on day  $t$ ,  $IC_t = \int_{t-1}^t H(\tau)d\tau$ . However, several problems make the Realized Covariance being a noisy proxy of the latent variance. Specifically, transaction prices are contaminated by the microstructure noise such as bid-ask spread. And according to Epps (1979), the RC would have a downward bias due to the presence of zero returns by the absence of trades of some assets in sampling interval. Barndorff-Nielsen et al. (2011) proposed the realized kernel which applied an idea of refresh time" sampling

procedure to adapt the sampling scheme to the trading intensity of the least active asset at any given time. The kernel also includes well-chosen weight function for the lead and lag returns in the computation of the covariance matrix. This ensures consistency of the estimator, while also guaranteeing positive semi-definiteness that the estimate of the covariance matrix. Formally, the realized kernel is defined as

$$RK_t = \Gamma_0 + \sum_{i=1}^H K\left(\frac{i}{H+1}\right) (\Gamma_i + \Gamma'_i) \quad (9)$$

$$\Gamma_j = \sum_{i=j+1}^m \tilde{r}_{i,t} \tilde{r}'_{i-j,t} \quad (10)$$

where  $\tilde{r}$  are refresh time returns, m is the number of refresh time returns,  $K(\cdot)$  is a kernel weighting function and H is a parameter which controls the bandwidth. Refresh time return are computed by sampling all prices using last-price interpolation only when all assets have traded. The recommended kernel is non-flat Parzen's kernel

$$K(x) = \begin{cases} 1 - 6x^2 + 6x^3 & 0 < x \leq 1/2 \\ 2(1-x)^3 & 1/2 < x \leq 1 \\ 0 & x > 1 \end{cases}$$

The optimal bandwidth is chosen by minimise the mean square error, the object is achieved by setting  $H = c * \xi^{4/5} n^{3/5}$ , where  $c^*=3.5134$  is a number related to weighting function,  $\xi^2 = \omega^2 / \sqrt{IQ}$  denotes the noise-to-signal ratio, where  $\omega^2$  is a measure of variance of micro-structure noise and IQ is the integrated quarticity, n is the number of refreshed prices. In practice, the sampling interval for estimator of  $\omega$  and  $IQ$  are respectively 3 and 10 minutes.

Jittering is another technique need to mention in computation of realized kernel. It basically means to average n prices at the very beginning and end of the day to eliminate end-effects to get the consistent realized kernel. The larger is m the smaller is the end-effects, however increasing m has the drawback that it reduces the sample size. So optimal amount of jittering is to minimize the MSE caused by end-effect plus the asymptotic variance increased by reduce n observations. Empirically, the optimal value of n ranges from 1 to 2, we set it as 2 according to our data.



### 3 Adaptive LASSO

In this section, we would show that the adaptive lasso can be used to estimate the measurement equation in the Realized MGARCH model.

#### 3.1 Definition and Notation

Since the measurement equation (3) is a fully parameterized linear model, the coefficient matrix has  $k^{*2}$  parameters to estimate, where  $k^{*} = k(k+1)/2$ . In fact, the equation (3) could be decomposed to a group of equations seemingly unrelated, each with one dependent variable corresponding to elements of realized matrix  $X_t$  which is a symmetric matrix. So we got  $k^{*}$  univariate linear equations. Each equation has  $k^{*}$  parameters to links the realized covariance with all the conditional covariance cross different assets. In practice, the coefficient  $\Gamma$  is usually a sparse matrix around the identity matrix. We could use adaptive Least Absolute Selection and Shrinkage Operator (lasso) to select most significant variable and reduce others to zero to achieve the parsimony of the model.

The lasso is a regularization technique for simultaneous estimation and variable selection in linear regression (Tibshirani 1996). It just means estimating the Least Squares parameters subject to an L1 norm penalty to the slackness parameter. Since Meinshausen and Bhlmann (2004) also showed the conflict of optimal prediction and consistent variable selection in the lasso. Zou(2006) also has shown that lasso estimator is possibly inconsistent, we will adopt the adaptive lasso which enjoys the oracle properties by utilizing the adaptively weighted L1 penalty proposed in the same paper.

We define adaptive lasso(alasso) estimates  $\beta^{alasso}$  are given by

$$\beta^{alasso} = \arg \min_{\beta} \left\| y - \sum_{j=1}^p x_j \beta_j \right\|^2 + \lambda \sum_{j=1}^p \hat{w}_j |\beta_j| \quad (11)$$

$\hat{w}_j$  is a vector of weights, with  $\hat{w}_j = 1/\|\hat{\beta}\|$ , where  $\hat{\beta}$  is the unpenalized OLS estimator. Note equation (11) is a convex optimization problem, and thus it does not suffer from the multiple local minimal issue, and its global minimizer can be easily found after the alasso solution path being efficiently computed by the LARS algorithm (Efron et al., 2004), which only require computational effort of one single OLS fit.

### 3.2 Least angle regression

There are several algorithm to compute the solution of shrinkage linear regression problem, such as quadratic programming approach in the original lasso paper by Tibshirani (1996), least angle regression(LARS) by Efron(2004) and so on. They provide pretty much similar results and by minor modification could generate identical solutions. We follow the routine proposed by Zou(2006) to use LARS algorithm to do the computation. In LARS, we chose the step length in each step such that any inactive variable becomes equally important as the active variable in terms of correlation with the residual vector. That variable is then assigned to the active set and a new direction is calculated. Specifically, the algorithm starts with the empty set of active variables which is just a trivial constant model and move towards the full OLS regression. The correlation between each variable and the response is measured, and the variable with the highest correlation becomes the first to include into the model. The first direction is then towards the least squares solution using this single active variable. The angles between the variables and the residual vector shrink along this direction until another variable obtains the same correlation with the residual vector as the active variable. Then the new variable moves to the active set and the new direction is towards the least squares solution of the two active variables, and so on. After  $p$  steps, the full OLS solution will be reached. The LARS algorithm is efficient since there is a closed form solution for the step length at each stage. Denoting the model estimate of  $y$  at iteration  $k$  by  $\hat{y}^{(k)}$  and the least squares solution including the newly added active variable  $\hat{y}_{OLS}^{(k+1)}$ , the walk from  $\hat{y}^{(k)}$  towards  $\hat{y}_{OLS}^{(k+1)}$  can be formulated  $(1 - \gamma)\hat{y}^{(k)} + \gamma\hat{y}_{OLS}^{(k+1)}$  where  $0 \leq \gamma \leq 1$ . Estimating  $\hat{y}_{OLS}^{(k+1)}$ , the position where the next active variable is to be added, then amounts to estimating  $\gamma$ . We seek the smallest positive  $\gamma$  where correlations become equal, that is

$$x'_{i \in \mathcal{I}}(y - (1 - \gamma)\hat{y}^{(k)} - \gamma\hat{y}_{OLS}^{(k+1)}) = x'_{j \in \mathcal{A}}(y - (1 - \gamma)\hat{y}^{(k)} - \gamma\hat{y}_{OLS}^{(k+1)})$$

where  $\mathcal{I}$  and  $\mathcal{A}$  represent inactive set and active set of variable respectively. Solving this expression for  $\gamma$ , we get

$$\gamma_{i \in \mathcal{I}} = \frac{(x_i - x_j)'(y - \hat{y}_{(k)})}{(x_i - x_j)'(\hat{y}_{OLS}^{(k+1)} - \hat{y}_{(k)})} = \frac{(x_i - x_j)'\epsilon}{(x_i - x_j)'\mathbf{d}}$$

where  $\epsilon$  is residual and  $\mathbf{d} = \hat{y}_{OLS}^{(k+1)} - \hat{y}_{(k)}$  is the direction of the step, and  $j \in \mathcal{A}$ . Now,  $\mathbf{d}$  is the orthogonal projection of  $\epsilon$  onto the plane spanned by the variables in  $\mathcal{A}$ , therefore we have  $x'_j\epsilon = x'_j\mathbf{d} \equiv c$ , representing the angle at the current breakpoint  $\hat{y}_{(k)}$ . Furthermore, the sign of the correlation between

variables is irrelevant. Therefore, we have

$$\gamma = \min_{i \in \mathcal{I}} \left( \frac{x'_i \epsilon - c}{x'_i \mathbf{d} - c}, \frac{x'_i \epsilon + c}{x'_i \mathbf{d} + c} \right), \quad 1 \leq \gamma \leq 1$$

where the two terms are for correlations/angles of equal and opposite sign respectively. The coefficients at this next step are given by

$$\beta^{(k+1)} = (1 - \gamma)\beta^{(k)} + \gamma\beta_{OLS}^{(k+1)}$$

Given these key pieces of the LAR algorithm, we state the entire procedure in Algorithm 1.

---

**Algorithm 1:** LARS algorithm

---

- 1 Initialize the coefficient vector  $\beta^{(0)} = 0$  and the fitted vector  $\hat{y}^{(0)} = 0$ ,
  - 2 Initialize the active set  $\mathcal{A} = \emptyset$  and the inactive set  $\mathcal{I} = 1 \dots p$
  - 3 for  $k = 0$  to  $p - 2$  do
  - 4 Update the residual  $\epsilon = y - \hat{y}^{(k)}$
  - 5 Find the maximal correlation  $c = \max_{i \in \mathcal{I}} |x'_i \epsilon|$
  - 6 Move variable corresponding to  $c$  from  $\mathcal{I}$  to  $\mathcal{A}$
  - 7 Calculate the least squares solution  $\beta_{OLS}^{(k+1)} = (X'_{\mathcal{A}} X_{\mathcal{A}})^{(-1)} X'_{\mathcal{A}} y$
  - 8 Calculate the current direction  $\bar{\gamma} = X_{\mathcal{A}} \beta_{OLS}^{(k+1)} - \hat{y}^{(k)}$  Calculate the step length  
 $\gamma = \min_{i \in \mathcal{I}}^+ \left( \frac{x'_i \epsilon - c}{x'_i \mathbf{d} - c}, \frac{x'_i \epsilon + c}{x'_i \mathbf{d} + c} \right), \quad 1 \leq \gamma \leq 1$
  - 9 Update regression coefficients  $\beta^{(k+1)} = (1 - \gamma)\beta^{(k)} + \gamma\beta_{OLS}^{(k+1)}$
  - 10 Update the fitted vector  $\hat{y}_{(k+1)} = \hat{y}^{(k)} + \gamma \mathbf{d}$
  - 11 end
  - 12 Let  $\beta^{(p)}$  be the full least squares solution  $\beta^{(p)} = (X' X)^{(-1)} X' y$
  - 13 Output the series of along the path  $B = [\beta^{(0)} \dots \beta^{(p)}]$
- 

### 3.3 Information Criterion

Each step of Algorithm 1 adds a covariate to the model until the full least squares solution is reached. It is natural to parameterize this process by the size of the coefficients at each step of the algorithm. Here we define  $s$  as a standardized version of parameter size.

$$s(\beta) = \sum_{i=1}^p |\hat{\beta}_i| / \sum_{i=1}^p |\hat{\beta}_i^{ols}|, \quad s \in [0, 1] \quad (12)$$

Note that  $s$  has a one-to-one relationship with the regulation parameter  $\lambda$ . Picking a suitable model for a particular analysis thus means selecting a suitable value of  $s$ . The question of how to choose a suitable  $s$  involves the trade-off between bias and variance, along with the subjective desire for parsimony.

Specifically, we pick up the optimal model to minimize the information criteria. The 2 popular criterion are Akaike information criterion (AIC) and Bayesian information criterion (BIC) for LASSO proposed by Zou(2006).

$$\begin{aligned} AIC^{(p)} &= \left\| y - X\beta^{(p)} \right\|^2 + 2\sigma_\epsilon^2 df^{(p)} \\ BIC^{(p)} &= \left\| y - X\beta^{(p)} \right\|^2 + \log(n)\sigma_\epsilon^2 df^{(p)} \end{aligned} \quad (13)$$

where  $df^{(k)}$  represents the number of degrees of freedom at each step of the LARS algorithm and approximately equals to the number of non-zero elements of parameters vector  $\beta$ , the LARS algorithm returns the sequence  $df_{LAR}^k = |\mathcal{A}| = k, k = 1, \dots, p$ . The measure  $\sigma_\epsilon^2$  represents the residual variance of OLS model,  $\sigma_\epsilon^2 = \frac{1}{n-k} \|Y - X'\beta^{ols}\|^2$ .

These two criterion has different asymptotic property. It is well known that AIC tends to select the model with the optimal prediction performance, while BIC tends to identify the true sparse model if the true model is in the candidate list (Hastie, and Tibshirani 2007). The intention of using measurement equation is to update realized volatility measure beyond one step to make multiple-period prediction. The accuracy of prediction with measurement equation is our main concern. Thus, AIC would be the better choice for forecast. With the entire LASSO solution path computed by LARS algorithm, we can immediately select the optimal model by identify the smallest value of AIC and the corresponding  $s(\beta)$ . From another perspective, We can not use RSS on the training data to determine these parameters, since we would always pick those that fit most parameters and hence gave smallest residuals. Such models are unlikely to predict future data well due to larger variance. The fully parameterized OLS estimate is derived from taking least squares, hence the RSS of linear regression with OLS fit is smallest, we demonstrate such point with the different fit of measurement equation (see table 1).

OLS	Shrinkage Based AIC	Shrinkage Based BIC
348.53	357.39	365.62

Table 1: Residual of Sum-of-Squares(RSS) of measurement equation (4)

With all the pieces together, we could solve the adaptive lasso problem defined in (11) for  $\hat{\beta}_j$  in , the computation details are given following in Algorithm 2:

---

**Algorithm 2:** Using LARS algorithm for adaptive lasso

---

- 1 Define new covariates  $x_j^* = x_j/\hat{w}_j, j = 1, 2, \dots, p$ . where  $\hat{w}_j$  is the adaptive weight as defined above.
- 2 Solve lasso problem with LARS algorithm ,

$$\beta^{alasso} = \arg \min_{\beta} \left\| y - \sum_{j=1}^p x_j^* \beta_j \right\|^2 + \lambda \sum_{j=1}^p |\beta_j|$$

which generate a sequence of solutions  $B = [\beta^{(0)} \dots \beta^{(p)}]$

- 3 Then we substitute lasso estimators from step 2 into the information criterion equation (13). This provides us a selection criteria value for each step.
  - 4 Pick up the  $\tilde{\beta}$  which minimizes AIC(BIC)
  - 5 Output  $\hat{\beta}_j = \tilde{\beta}_j/\hat{w}_j, j = 1, 2, \dots, p$ . This is the adaptive lasso estimate.
- 

## 4 Estimation and Inference

### 4.1 Estimation for MGARCH Equation

In this section we discuss the asymptotic properties of the quasi-maximum likelihood estimator within the realized MGARCH model given by (1) and (2). We could estimate the parameters in rotated model, then reverse the linear transformation to retract the unrotated measure. In traditional GARCH MODELS, the vector of daily returns is usually modeled as  $R_t = H_t^{1/2} z_t$ , with  $z_t \stackrel{i.i.d}{\sim} \mathcal{N}(0, I_k)$ , which motivates quasi-maximum likelihood estimation (QMLE). In our model, since  $\xi_t = z_t z_t'$ , the assumption that  $z_t \stackrel{i.i.d}{\sim} \mathcal{N}(0, I_k)$  implies that  $\xi$  follows a Wishart distribution. The choice of Wishart distribution is suitable for the model where the support of the random variable of interest is confined to the space of positive semidefinite matrices.

The Realized MGARCH equation is parameterized with a finite-dimensional  $(\delta \times 1)$  parameter vector  $\theta \in \Theta \subset \mathbb{R}^\delta$ . Decompose  $\theta = (\theta'_s, \theta'_d)'$ , where  $\theta_s = (p', m')'$  denote the model's unconditional moment, and let  $\theta_d$  denote the vector of dynamic parameters indexing  $\tilde{A}$  and  $\tilde{B}$  in (8). We denote the true parameter vector with  $\theta_0$ . The parameters are estimated in 2 steps. The static parameters are to be

estimated by a moment estimator

$$p' = \text{vech}(P), \quad m' = \text{vech}(M)$$

where

$$\hat{P} = T^{-1} \sum_{t=1}^T r_t r_t', \quad \hat{M} = T^{-1} \sum_{t=1}^T X_t$$

This estimate is then decomposed into  $P^{1/2}$  and  $M^{1/2}$ . Then we construct the time series of rotated returns and realized kernel  $\hat{z}_t = \hat{P}^{-1/2} r_t, \hat{X}_t = \hat{M}^{-1/2} X_t, t = 1, 2, \dots, T$ . Then  $\theta_d$  will be estimated by QMLE in the second step. The asymptotics of the QML estimator in this 2-step procedure is a direct application of two-step GMM estimation discussed in Newey and McFadden (1994).

The log-likelihood for the  $t^{th}$  observation will be denoted by  $l_t(\theta)$ . Inference for the Realized MGARCH equation will be based on QMLE of the following log-likelihood function of Wishart density which is similar with HEAVY model in Noureldin, etc (2011).

$$l_t(\theta_d; \theta_s) = C - \frac{1}{2} (\log |\tilde{H}_t| + \text{tr}(\tilde{H}_t^{-1} z_t z_t')) \quad (14)$$

where  $C$  is a constant with respect to  $\theta$ . We assume the initial values  $H_0$  is known and is positive semidefinite. The QML estimator is  $\hat{\theta}_d$ , where

$$\hat{\theta}_d = \arg \max_{\theta_d \in \Theta} L_t(\theta_d; \theta_s) = \arg \max_{\theta_d \in \Theta} \sum_{t=1}^T l_t(\theta_d; \theta_s)$$

To compute the asymptotic standard errors, we construct a group of conditions of moment for each  $t^{th}$  observation,  $g_t = (g'_{p,t}, g'_{m,t}, g'_{\theta_d,t})$ , where  $g_{p,t} = p - \text{vech}(r_t r_t')$ ,  $g_{m,t} = m - \text{vech}(X_t)$  and  $g_{\theta_d,t} = \frac{\partial \log L_t(\theta_s, \theta_d)}{\partial \theta'_d}$ . In the true parameter values,  $E[g_t | \mathcal{F}_{t-1}] = 0$ . Under standard regularity conditions, as  $T \rightarrow \infty$  we have

$$\sqrt{T}(\hat{\theta} - \theta_0) \xrightarrow{d} \mathbb{N}(0, \mathcal{I}^{-1} \mathcal{J} (\mathcal{I}^{-1})') \quad (15)$$

where  $\hat{\theta} = (\hat{\theta}'_s, \hat{\theta}'_d)$ , and

$$\mathcal{J} = \text{Var} \left[ \frac{1}{\sqrt{T}} \sum_{t=1}^T g_t \right], \quad (16)$$

$$\mathcal{I} = E \left[ \frac{\partial g_t}{\partial \theta} \right] = \begin{bmatrix} \frac{\partial g_{p,t}}{\partial p} & 0 & \frac{\partial g_{\theta_d,t}}{\partial p} \\ 0 & \frac{\partial g_{m,t}}{\partial m} & \frac{\partial g_{\theta_d,t}}{\partial m} \\ 0 & 0 & \frac{\partial g_{\theta_d,t}}{\partial \theta} \end{bmatrix} = \begin{bmatrix} I_{k^*} & 0 & \frac{\partial^2 \log L_t(\theta_s, \theta'_d)}{\partial p \partial \theta'_d} \\ 0 & I_{k^*} & \frac{\partial^2 \log L_t(\theta_s, \theta'_d)}{\partial m \partial \theta'_d} \\ 0 & 0 & \frac{\partial^2 \log L_t(\theta_s, \theta'_d)}{\partial \theta_d \partial \theta'_d} \end{bmatrix} \quad (17)$$

## 4.2 Estimation of the Multivariate Measurement Equation

Since the measurement equation is over-parameterized. We will use alasso method to estimate and shrink the parameters in measurement equation. As we mentioned before, the measurement equation is actually a system of equations estimated similarly as using SUR method. To be concrete, we write (4) as the system of equations.

$$\begin{bmatrix} X_1 \\ \vdots \\ X_{k^*} \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_{k^*} \end{bmatrix} + H \begin{bmatrix} \tilde{\Gamma}_1 \\ \vdots \\ \tilde{\Gamma}_{k^*} \end{bmatrix} + \begin{bmatrix} u_1 \\ \vdots \\ u_{k^*} \end{bmatrix} \quad (18)$$

which  $X_1$  is a  $T \times 1$  vector,  $d_1$  is a scalar,  $H$  is a  $T \times k^*$  matrix, the  $k^* \times 1$  coefficient vector  $\tilde{\Gamma}_1$  is the transpose of the first row of coefficient matrix  $\tilde{\Gamma}$  in (19). So we have  $k^*$  equations which share the same predictor matrix  $H$ . Then we apply lasso method in each equation to get corresponding rows in coefficient matrix  $\tilde{\Gamma}$ .

Note that the estimation method used in the lasso requires to first normalize each predictor  $\tilde{\Gamma}$  by subtracting the mean and scaling so that  $\sum_{i=1}^k \|h_{i,j}\| = 1$ . So the (4) could be estimated in the normalized form:

$$\tilde{x}_t = \tilde{\Gamma} \tilde{h}_t + u_t \quad (19)$$

where  $x_t$  is centered and  $\tilde{h}$  is centered and normalized such that each variable has zero mean and unit Euclidean length.  $\tilde{\Gamma} = \Gamma e$ , which  $e$  is a  $k^* \times k^*$  matrix with same rows and each element in row represent the Euclidean length of each column in  $H$ . The intercept in (4) could be calculated as  $d = \bar{x}_t - \Gamma \bar{h}_t$ .

The formula for variance of ALASSO is very complicated as showed in Zou(2006).

$$\text{cov}(\hat{\beta}_{\mathcal{A}}) = \sigma^2 (X'_{\mathcal{A}} X_{\mathcal{A}} + \lambda_n \sum (\hat{\beta}_{\mathcal{A}}))^{-1} X'_{\mathcal{A}} X_{\mathcal{A}} (X'_{\mathcal{A}} X_{\mathcal{A}} + \lambda_n \sum (\hat{\beta}_{\mathcal{A}}))^{-1}$$

where  $\mathcal{A}$  is the active set including all non-zero variable selected by alasso, We use it to denote sub-matrices such as the  $(n \times |\mathcal{A}|)$  matrix  $X_{\mathcal{A}}$ , consisting of the columns (variables) of  $X$  corresponding to the indices in  $\mathcal{A}$  and  $\sum(\hat{\beta}) = \text{diag}(\frac{\hat{w}_1}{|\beta_1|}, \dots, \frac{\hat{w}_k}{|\beta_k|})$ .  $\sigma^2$  is the variance of residual of OLS fit which we could easily estimate with sample variance. The difficult part is that  $\lambda$  is hard to extract from LARS algorithm. According to the fundamental theorem of statistics, The empirical distribution (ED) of a set of independent drawings of a random variables generated by some data generating process (DGP) converges to the true CDF of the under the DGP. Using this theorem, we could calculate the sample

covariance with bootstrapping method proposed by Efron(1979). Bootstrapping is the practice of estimating the properties of an estimator -say, its variance- by measuring those properties when sampling from an approximating distribution (the bootstrap DGP). Since we dont know the true DGP generated this parameters, we could approximate it with ED. The algorithm is shown below

---

**Algorithm 3:** The Bootstrap algorithm

---

- 1 Select N pairs of independent bootstrap samples  $(y_1^*, x_1^*), (y_2^*, x_2^*), \dots, (y_n^*, x_n^*)$  with replacement from original data
  - 2 Estimate the alasso parameters for each bootstrap sample  $\hat{\tau}(n), n = 1, 2, \dots, N$ .
  - 3 Estimate the standard error  $se(\hat{\tau})$  by the empirical moment of the N group of parameters
  - 4  $s\hat{e}_N = \left[ \frac{1}{N-1} \sum_{n=1}^N \{\hat{\tau}(n) - \bar{\hat{\tau}}\} \right]$  with  $\bar{\hat{\tau}} = \frac{1}{N} \sum_{n=1}^N \hat{\tau}(n)$
- 

## 5 Empirical illustration

The model is applied to high-frequency prices for ten stocks from the financial sector of the S&P 100<sup>1</sup>. These are: xxxxx. The sample period is mm/dd/yyyy to mm/dd/yyyy and the source of data is the TAQ database. We focus on the noise-robust realized kernel of Barndorff-Nielsen et al.(2011) as our choice for  $X_t$ . For the daily return and realized kernel, we focus on open-to-close returns which ignore the overnight effect.

### 5.1 Realized Volatility Description

### 5.2 Daily Log-Return Description

### 5.3 Estimation Result for realized GARCH model

Application of rotated GARCH method to our model yields estimate of the diagonal coefficient matrix  $\tilde{A}$  and  $\tilde{B}$  in equation (7). We report the estimates along with their standard errors in table(3) and (4). We also could retract the true parameters A and B according to (8). Note that, the GARCH equation could also written into vech form

$$h_t = \bar{C} + \bar{B} h_{t-1} + \bar{A} x_{t-1} \quad (20)$$

---

<sup>1</sup>The ticker symbols are xxxxx



$\bar{B}$  usually referred as smoothing parameters in literature. The coefficients are easier for interpretation when expressed in terms of parameters of the vech representations in (20). Notice that the diagonal elements of  $\bar{A}$  and  $\bar{B}$  are squares and cross products of diagonal elements in  $A$  and  $B$ . We get the average value of diagonal elements in  $\bar{B}$  and  $\bar{A}$  are respectively, 0.5803 and 0.2731. The estimate of  $\bar{B}$  implies that elements of  $H_t$  will be smooth, although less smooth than the traditional GARCH model which has a smoothing parameter around 0.9 according to literature. The estimate of  $\bar{A}$  indicate the realized measure of variance play a important role in updating  $H_t$ .

#### 5.4 Adaptive lasso under different selective criterion

We compare the two different criterion, AIC and BIC we introduced in section 3.2 and their relationship of relative bound  $s$  with df and RSS of model below. Figure (1) shows the LASSO estimates as a function of standardized bounds. LASSO continuously shrinks the coefficients toward zero as  $s$  decreases; and some coefficients shrink to exactly zero if the constraints is sufficiently large. In figure (2), the number of variables decreases as relative bound decreases (penalty constraints increases). In both figures, We show how different the model will be by selecting variables according to different criterion. The broken line represents AIC criterion, we select model at AIC=362,  $s=0.32$ ; The solid line represents BIC criterion, we select model at BIC=371,  $s=0.05$ . The value of  $s$  validates the sparsity of true model. Roughly, this corresponds to keeping just 30% of the predictors. In the figure (3), the mean square of error of alasso estimate decreases as relative bound decreases (penalty constraints increases). We could find an elbow shape in the curve, which means part of parameters don't contribute a lot for reducing the MSE. As a result, the optimal selection value of  $s$  always happens near the elbow area.

#### 5.5 Estimation Result of Measurement equation

With the structure of measurement equations, each realized variance (covariance) is the function of corresponding conditional variance (covariance) and the conditional covariance of pairs of all other assets. Assume there are three asset  $A$ ,  $B$  and  $C$ , the realized variance of asset  $A$  is updated by the conditional variance of  $A$  and conditional covariance of  $A$  and  $B$ , the one of  $A$  and  $C$ , and also the one of  $B$  and  $C$ . Apparently the conditional covariance of one asset should be the main driving power in updating its counterpart in realized measure. As a result, the coefficient matrix should have larger elements in the diagonal position. Most elements that alasso shrinks are in the off-diagonal position.

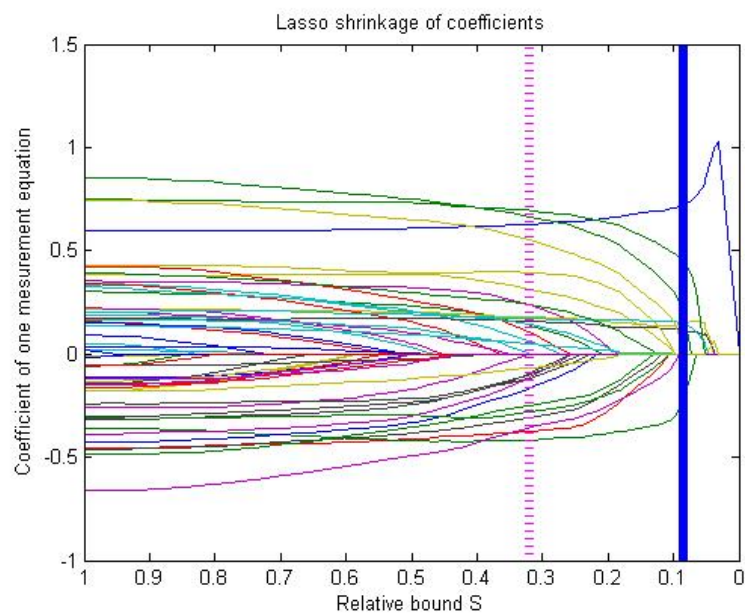


Figure 1: Lasso shrinkage of coefficient

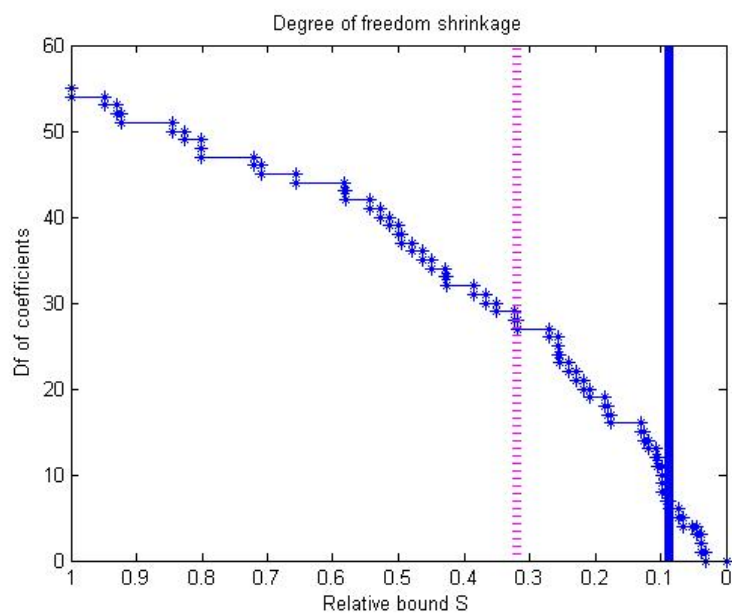


Figure 2: Degree of freedom shrinkage

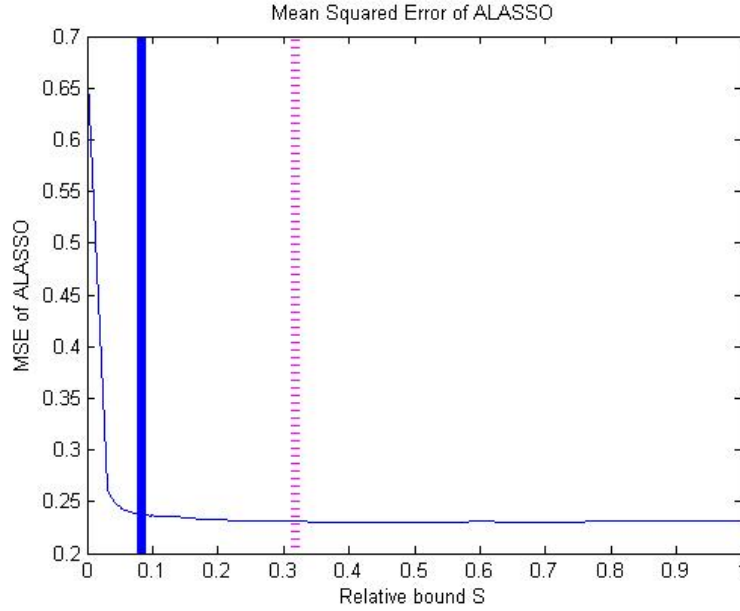


Figure 3: Mean Squared Error of ALASSO

Even some diagonal elements will also shrink since the lagged driving force in updating covariance is smaller than the one of variance. In the case of 10 assets, we got coefficient matrix in size 55\*55, in the diagonal position the elements in 1, 11, 20, 28, 35, 41, 46, 50, 53, 55 rows are index into the coefficient of variance of each asset which much larger than others. In the table(2), we compared elements of coefficient matrix corresponding to variance of each asset by using OLS and alasso using AIC and BIC criterion, the number in parenthesis are standard errors. In the last two row, we give the non-zero percentage of the coefficient matrix and the MSE of fit of OLS and alasso under different select criterion. The OLS has 100% which means it didn't shrink any variables, AIC has 38.68% larger than BIC's 13.69%, which means BIC tends to emphasis on parsimony of model. However, AIC tends to get the more accurate fit in terms of MSE, according to our result, AIC has same MSE 0.231 with OLS and less than BIC's 0.237. This result is consistent with our statement in section 3.2. In the last third row, we showed the non-zero percentage in diagonal position of coefficient matrix. AIC has 83.64%, BIC has 72.73%, which indicate the coefficient matrix estimated by alasso is basically a diagonal matrix compared with the high shrinkage percentage in off-diagonal positions.

The last fourth row report the mean of coefficient for each variance, the value for AIC is 0.889, the biggest, since it shrink most. This estimate indicate  $H_t$  is a good predictor to update  $X_t$ .

## **6 Conclusion**

## **Appendix**

Table 2: Estimation of coefficient  $\Gamma$ 

Index in diagonal position	LS	AIC	BIC
1	0.65(0.05)	0.72(0.03)	0.72(0.03)
11	0.76(0.05)	0.79(0.03)	0.80(0.02)
20	0.88(0.07)	0.93(0.05)	1.24(0.03)
28	0.73(0.09)	0.76(0.05)	0.91(0.06)
35	0.75(0.07)	0.83(0.03)	0.75(0.03)
41	0.96(0.04)	0.97(0.02)	0.91(0.03)
46	0.82(0.07)	0.87(0.05)	1.02(0.06)
50	0.62(0.08)	0.65(0.04)	1.00(0.03)
53	0.69(0.06)	0.71(0.04)	0.83(0.02)
55	0.81(0.09)	0.87(0.03)	0.71(0.03)
mean	0.766	0.810	0.889
Non-zero pct	100%	83.64%	72.73%
Overall Non-zero pct	100%	38.68%	13.69%
MSE	0.231	0.231	0.237

Table 3: Parameter Estimates of the "A" matrix in rotated realized MGARCH model

	ABT	DOW	IBM	AXP	BK	C	CAT	COF	USB	WFC
ABT	0.7213 (0.0627)									
DOW		0.4775 (0.0150)								
IBM			0.5150 (0.0503)							
AXP				0.6125 (0.0308)						
BK					0.5311 (0.0143)					
C						0.6585 (0.0011)				
CAT							0.4642 (0.0361)			
COF								0.6701 (0.0154)		
USB									0.7723 (0.0292)	
WFC										0.5135 (0.0089)

Table 4: Parameter Estimates of the "B" matrix in rotated realized MGARCH model

	ABT	DOW	IBM	AXP	BK	C	CAT	COF	USB	WFC
ABT	0.5835 (0.1073)									
DOW		0.8747 (0.0049)								
IBM			0.8012 (0.0402)							
AXP				0.7454 (0.0271)						
BK					0.8321 (0.0068)					
C						0.7512 (0.0003)				
CAT							0.8547 (0.0199)			
COF								0.7172 (0.0141)		
USB									0.6006 (0.0453)	
WFC										0.8514 (0.0032)