Estimating all possible SUR models with permuted exogenous data matrices derived from a VAR process☆

Cristian Gatu a, ∗, Erricos J. Kontogiorghes b, c

a Institut d’informatique, Université de Neuchâtel, Émile-Argand 11, CH-2007 Neuchâtel, Switzerland
b Department of Public and Business Administration, University of Cyprus, P.O. Box 20537, CY-1678 Nicosia, Cyprus
c School of Computer Science and Information Systems, Birkbeck College, University of London, Malet Street, London WC1E 7HX, UK

Received 12 October 2004; accepted 14 March 2005
Available online 4 June 2005

Abstract

The Vector Autoregressive (VAR) process with zero coefficient constraints can be formulated as a Seemingly Unrelated Regressions (SUR) model. Within the context of subset VAR model selection a computationally efficient strategy to generate and estimate all G! SUR models when permuting the exogenous data matrices is proposed, where G is the number of the regression equations. The combinatorial algorithm is based on orthogonal transformations, exploits the particular structure of the modified models and avoids the estimation of these models afresh by utilizing previous computation. Theoretical measurements of complexity are derived to prove the efficiency of the proposed algorithm.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Subset Vector Autoregressive model selection; Seemingly Unrelated Regressions models; Least squares; QR decomposition; Combinatorial algorithms

☆ This work is in part supported by the Swiss National Foundation for Research Grants 101312-100757/1, 200020-100116/1 and 101412-105978.

Corresponding author.
E-mail addresses: cristian.gatu@unine.ch (C. Gatu), erricos@ucy.ac.cy (E.J. Kontogiorghes).

0165-1889/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.jedc.2005.03.006
1. Introduction

Consider the Vector Autoregressive (VAR) model in compact form

\[ Y = XB + U, \quad (1) \]

which can be equivalently written as

\[ \text{vec}(Y) = (I_G \otimes X) \text{vec}(B) + \text{vec}(U), \quad \text{vec}(U) \sim (0, \Sigma \otimes I_M). \quad (2) \]

Here vec is the vector operator, \( Y = (y_1 \ldots y_G) \in \mathbb{R}^{M \times G} \) are the response vectors, \( X \in \mathbb{R}^{M \times K} \) is the lagged exogenous data matrix having full-column rank and block-Toeplitz structure, \( B = (b_1 \ldots b_G) \in \mathbb{R}^{K \times G} \) is the coefficient matrix and \( U = (u_1 \ldots u_G) \in \mathbb{R}^{M \times G} \) are the disturbances. The expectation of \( U \) is zero, i.e. \( \mathbb{E}(u_i) = 0, \quad \mathbb{E}(u_i u_j^T) = \sigma_{ij} I_M \quad (i,j = 1, \ldots, G) \) and \( \Sigma = [\sigma_{ij}] \in \mathbb{R}^{G \times G} \) has full rank (Kontoghiorghes and Clarke, 1995; Kourouklis and Paige, 1981; Lütkepohl, 1993; Paige, 1978, 1979; Srivastava and Giles, 1987). The Ordinary and Generalized Least Squares estimators of (2) are the same and given by

\[ \hat{B} = (X^T X)^{-1} X^T Y. \]

An important problem in the VAR modeling is the identification of the lag structure or, equivalently, the specification of the subset VAR models (Lütkepohl, 1993; Winker, 2000). A subset VAR model can be seen as a VAR model with zero-restrictions on some of the coefficients (ZR-VAR). In order to identify a “good” sub-model, several subset VAR models are enumerated by enforcing some coefficients to be zero, and compared with respect to some criterion (Akaike, 1969, 1974; Bozdogan and Bearse, 2003; Hannan and Quinn, 1979; Schwarz, 1978). Imposing zero-restrictions on some coefficients is equivalent to selecting the non-zero ones. Let \( S_i \in \mathbb{R}^{K \times k_i} \quad (i = 1, \ldots, G) \) denote a selection matrix such that \( b_i = S_i^T b_i \) corresponds to the selected coefficients of \( b_i \)—the \( i \)-th column of \( B \). Furthermore, let \( X_i = X S_i \) which are the columns of \( X \) that correspond to the selected coefficients of \( b_i \). Thus, the ZR-VAR model is equivalent to the Seemingly Unrelated Regressions (SUR) model

\[ \text{vec}(Y) = (\bigoplus_{i=1}^G X_i) \text{vec}(\{\beta_i\}_G) + \text{vec}(U), \quad \text{vec}(U) \sim (0, \Sigma \otimes I_M), \quad (3) \]

where \( \bigoplus_{i=1}^G X_i = \text{diag}(X_1, \ldots, X_G) \), \( \{\beta_i\}_G \) denotes the set \( \{\beta_1, \ldots, \beta_G\} \) and \( \text{vec}(\{\beta_i\}_G) = (\beta_1^T \ldots \beta_G^T)^T \). For notational convenience the direct sum \( \bigoplus_{i=1}^G \) and the set \( \{\cdot\}_G \) are abbreviated to \( \bigoplus_i \) and \( \{\cdot\} \), respectively (Foschi and Kontoghiorghes, 2003; Kontoghiorghes, 2000; Kontoghiorghes and Clarke, 1995; Srivastava and Giles, 1987).

Ideally, the enumeration and estimation of all possible \( 2^G - 1 \) ZR-VAR models is required. In this context, an extension of the Dropping Columns Algorithm which derives all subset VAR models by efficiently moving from one sub-model to another has been previously proposed (Gatu and Kontoghiorghes, 2003, 2005). The algorithm generates a regression tree and avoids estimating each sub-model afresh.
However, this strategy is computationally infeasible even for modest values of $K$ and $G$. An alternative is to consider sub-classes of these models. Within this context, two special cases of subset VAR models have been proposed (Gatu and Kontoghiorghes, 2005). The first special case derives $(K + 2)^{K-1} - 1$ sub-models by deleting the same coefficient from each block of the VAR model (2). Notice that selecting some coefficients is equivalent to deleting the remaining ones. The second case is based on deleting proper subsets of coefficients from each block of the regressors and generates $P_{\min}(K, G)$ models, when $G$ is greater than one. Both cases require $O(G)$ less computational complexity than generating the models afresh. Here, a third special case of subset VAR models which derives efficiently $G!$ new subset models by permuting the exogenous matrices $X_1, \ldots, X_G$ in the SUR model (3) is considered.

In the next section, the numerical solution of the permuted SUR model is presented. Particular cases of permuted SUR models together with their numerical solutions which take advantage of the structure of the models are also described. An efficient algorithm for estimating all $G!$ permuted SUR models is given in Section 3. Theoretical measures of complexity are derived and employed to illustrate the efficiency of the proposed combinatorial algorithm. Finally, Section 4 concludes.

2. Numerical estimation of the permuted SUR model

The best linear unbiased estimator (BLUE) of the SUR model (3) comes from the solution of the Generalized Linear Least Squares Problem (GLLSP)

$$\arg\min_{V, \beta} \| V \|_F^2 \quad \text{subject to } \vec(Y) = (\oplus_i X_i) \vec(\beta_i) + \vec(V C^T).$$

(4)

Here $\Sigma = C C^T$, the random $V \in \mathbb{R}^{M \times G}$ is defined as $V C^T = U$ which implies $\vec(V) \sim (0, I_{GM})$, and $\| \cdot \|_F$ denotes Frobenius norm i.e., $\| V \|_F^2 = \sum_{i=1}^{M} \sum_{j=1}^{G} V_{ij}^2$ (Kontoghiorghes and Clarke, 1995; Kourouklis and Paige, 1981; Paige, 1978, 1979). The upper-triangular $C \in \mathbb{R}^{G \times G}$ is the Cholesky factor of $\Sigma$. For the solution of (4) consider the Generalized QR Decomposition (GQRD) of the matrices $\oplus_i X_i$ and $C \otimes I_M$:

$$Q^T (\oplus_i X_i) = \left( \begin{array}{c} \oplus_i R_i \\ 0 \end{array} \right)^{K^*}_{GM-K^*},$$

(5a)

and

$$Q^T (C \otimes I_M) P = W = \begin{pmatrix} W_{11} & W_{12}^{GM-K^*} \\ W_{12} & W_{22} \end{pmatrix}^{K^*}_{GM-K^*},$$

(5b)
where $K^* = \sum_{i=1}^G k_i$, $Q$ and $P$ are $GM \times GM$ orthogonal matrices and $\oplus R_i$ and $W$ are upper triangular of order $K^*$ and $GM$, respectively. Now, since $\|V\|_F^2 = \|P^T \text{vec}(V)\|^2$ and writing

$$Q^T \text{vec}(Y) = \left( \begin{array}{c} \text{vec}(\{y_{Ai}\}) \\ \text{vec}(\{y_{Bi}\}) \end{array} \right)_{GM-K^*} \quad \text{and}$$

$$P^T \text{vec}(V) = \left( \begin{array}{c} \text{vec}(\{v_{Ai}\}) \\ \text{vec}(\{v_{Bi}\}) \end{array} \right)_{GM-K^*},$$

the GLLSP (4) is equivalent to

$$\arg\min_{\{v_{Ai}\},\{v_{Bi}\},\{\beta_i\}} \sum_{i=1}^G (\|v_{Ai}\|^2 + \|v_{Bi}\|^2) \quad \text{subject to}$$

$$\begin{pmatrix} \text{vec}(\{y_{Ai}\}) \\ \text{vec}(\{y_{Bi}\}) \end{pmatrix} = \begin{pmatrix} \oplus_i R_i \\ 0 \end{pmatrix} \text{vec}(\{\beta_i\}) + \begin{pmatrix} W_{11} & W_{12} \\ 0 & W_{22} \end{pmatrix} \begin{pmatrix} \text{vec}(\{v_{Ai}\}) \\ \text{vec}(\{v_{Bi}\}) \end{pmatrix}, \quad (6)$$

where $\| \cdot \|$ denotes the Euclidian norm (Kontoghiorghes, 2004). From the constraint in (6) it follows that $\text{vec}(\{v_{Bi}\})$ and BLUE of $\text{vec}(\{\beta_i\})$, say $\hat{\beta}^{(i)} \in \mathbb{R}^{k_i}$, are computed from the solution of the triangular system

$$\begin{pmatrix} \oplus_i R_i & W_{12} \\ 0 & W_{22} \end{pmatrix} \begin{pmatrix} \hat{\beta}^{(i)} \\ \text{vec}(\{v_{Bi}\}) \end{pmatrix} = \begin{pmatrix} \text{vec}(\{y_{Ai}\}) \\ \text{vec}(\{y_{Bi}\}) \end{pmatrix}. \quad (7)$$

The GQRD (5) is the main computational component for obtaining the BLUE of (3). In (5a), $R_i \in \mathbb{R}^{k_i \times k_i}$ is the upper-triangular factor in the QR Decomposition (QRD) of $X_i$. That is,

$$Q_i^T X_i = \left( \begin{array}{c} R_i \\ 0 \end{array} \right)_{M-k_i}^{k_i} \quad \text{with} \quad Q_i^T = \left( \begin{array}{c} Q_{Ai}^T \\ Q_{Bi}^T \end{array} \right)_{M-k_i}^{k_i} \quad (i = 1, \ldots, G), \quad (8)$$

where $Q_i \in \mathbb{R}^{M \times M}$ is orthogonal and $Q$ in (5a) is defined by

$$Q = (\oplus_i Q_{Ai} \oplus_i Q_{Bi}) = \begin{pmatrix} Q_{A1} & \cdots & Q_{B1} \\ \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots \\ Q_{AG} & \cdots & Q_{BG} \end{pmatrix}. \quad (9)$$
For the RQ Decomposition (RQD) (5b), initially the matrix $Q$ is premultiplied by $Q^T(C \otimes I_M)$ to give

$$Q^T(C \otimes I_M)Q = \begin{pmatrix} W^{(1,0)} & W^{(2,0)} \\ W^{(3,0)} & W^{(4,0)} \end{pmatrix} \equiv W_0$$

Here

$$\begin{pmatrix} 0 \ W^{(2,0)} & \cdots & 0 \\ 0 \ 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 \ 0 & \cdots & 0 \end{pmatrix} = \begin{pmatrix} c_{ii}IM \\ c_{ij}Q^T_iQ_j \end{pmatrix}$$

for $i,j = 1, \ldots, G$, $i < j$. Finally, the RQD of (10) is computed in $G - 1$ steps. The $i$th ($i = 1, \ldots, G - 1$) step computes the factorization
The matrix $P_i$ is the product of $i$ orthogonal matrices. That is, $P_i = P_{i,1}^* \cdots P_{i,j}^*$. Now,

$$
P_{i,j}^* = \begin{pmatrix}
  k_{i+1} & \lambda_{ij} & M-k_{i-j+1} & \mu_{ij} \\
  P_{ij}^{(A)} & 0 & 0 & 0 \\
  0 & I_{\lambda_{ij}} & 0 & 0 \\
  P_{ij}^{(C)} & 0 & 0 & 0 \\
  0 & 0 & 0 & I_{\mu_{ij}}
\end{pmatrix}
$$

where $\lambda_{ij} = \sum_{l=1}^{i-j} (M - k_l)$, $\mu_{ij} = \lambda_{i,0} - \lambda_{i,j-1}$ and $j = 1, \ldots, i$. Furthermore,

$$
W_{i-j+1, i-j+1}^{(i,j)} P_{ij} = \begin{pmatrix}
  k_{i+1} & M-k_{i-j+1} \\
  0 & 0 \\
  W_{i-j+1, i-j+1}^{(i,j)} & I_{\mu_{ij}}
\end{pmatrix},
$$

$W_{i-j+1, i-j+1}^{(i,j)}$ is upper-triangular and

$$
P_{ij} = \begin{pmatrix}
  P_{ij}^{(A)} & P_{ij}^{(B)} \\
  P_{ij}^{(C)} & P_{ij}^{(D)}
\end{pmatrix}.
$$

Fig. 1 illustrates this strategy to compute (11), where $G = 5$. An arc indicates the columns affected from the application of $P_{ij}^*$ ($i = 1, \ldots, G-1$ and $j = 1, \ldots, i$). Notice that $P_{ij}^*$ can be considered as block-version of a Givens rotation (Foschi et al., 2003; Foschi and Kontoghiorghes, 2003; Kontoghiorghes, 2000; Yanev and Kontoghiorghes, 2004, 2005a,b).
Thus, $W$ in (5b) is given by

$$W = \begin{pmatrix}
W_{1,1}^{(1,G-1)} & W_{1,2}^{(1,G-1)} & \cdots & W_{1,G}^{(1,G-1)} \\
0 & W_{2,2}^{(1,G-1)} & \cdots & W_{2,G}^{(1,G-1)} \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & W_{G,G}^{(1,G-1)} \\
0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 \\
\end{pmatrix}. $$

The matrices $Q, P_1, \ldots, P_{G-1}$ and thus $P$ are not constructed explicitly.
An equivalent strategy for computing the RQD of (10) annihilates the blocks of form \( W^{(3,j)} \) row-by-row and from bottom-to-top. This is performed in \( G - 1 \) steps. The \( i \)th step computes the factorization

\[
\begin{pmatrix}
W^{(1,i-1)}_{G-i,G-i+1} & \cdots & W^{(1,i-1)}_{G-i,G} \\
W^{(1,i-1)}_{G-i+1,G-i+1} & \cdots & W^{(1,i-1)}_{G-i+1,G} \\
\vdots & \ddots & \vdots \\
0 & \cdots & W^{(1,i-1)}_{G,G} \\
W^{(3,i-1)}_{G-i,G-i+1} & \cdots & W^{(4,i-1)}_{G-i,G} 
\end{pmatrix}
\]

and the product

\[
\begin{pmatrix}
W^{(r,i-1)}_{1,G-i+1} & \cdots & W^{(r,i-1)}_{1,G} \\
\vdots & \ddots & \vdots \\
W^{(r,i-1)}_{G-i,1,G-i+1} & \cdots & W^{(r,i-1)}_{G-i,1,G} \\
\end{pmatrix}
\begin{pmatrix}
W^{(r+1,i-1)}_{1,G-i+1} & \cdots & W^{(r+1,i-1)}_{1,G} \\
\vdots & \ddots & \vdots \\
W^{(r+1,i-1)}_{G-i,1,G-i+1} & \cdots & W^{(r+1,i-1)}_{G-i,1,G} \\
\end{pmatrix}
\]

where \( r = 1, 3 \). The matrix \( \tilde{P}_i \) is the product of \( i \) orthogonal matrices. That is, \( \tilde{P}_i = \tilde{P}^*_{i,1} \cdots \tilde{P}^*_{i,i} \). Here,

\[
\tilde{P}^*_{i,j} = \begin{pmatrix}
\hat{\lambda}_{ij} & k_{G-i+j} & \bar{\mu}_{ij} & M-k_{G-i} \\
I_{\hat{\lambda}_{ij-1}} & 0 & 0 & 0 \\
0 & \tilde{P}^{(A)}_{i,j} & \tilde{P}^{(B)}_{i,j} & k_{G-i+j} \\
0 & 0 & I_{\bar{\mu}_{ij}} & 0 \\
0 & \tilde{P}^{(C)}_{i,j} & 0 & \tilde{P}^{(D)}_{i,j} & M-k_{G-i}
\end{pmatrix},
\]

where now, \( \hat{\lambda}_{ij} = \sum_{l=1}^{j} k_{G-i+l}, \bar{\mu}_{ij} = \hat{\lambda}_{ij} - \tilde{\lambda}_{ij}, \) and \( j = 1, \ldots, i \). Furthermore,

\[
(\begin{pmatrix}
W^{(3,i-1)}_{G-i,G-i+j} & \cdots & W^{(4,i-1)}_{G-i,G} \\
\end{pmatrix}) \tilde{P}_{i,j} = \begin{pmatrix}
k_{G-i+j} & M-k_{G-i} \\
W^{(i,j)}_{G-i,G-i} & \cdots & W^{(i,j)}_{G-i,G-i} \\
\end{pmatrix},
\]
\[ W_{G-i,G-i}^{(i,j)} \] is upper-triangular and

\[
\tilde{P}_{ij} = \begin{pmatrix}
\tilde{P}_{ij}^{(A)} & \tilde{P}_{ij}^{(B)} \\
\tilde{P}_{ij}^{(C)} & \tilde{P}_{ij}^{(D)}
\end{pmatrix}.
\]

Fig. 2 shows the stages of this annihilation strategy, where \( G = 5 \) and the \( i \)th stage \((i = 1, \ldots, G - 1)\) performs \( i \) annihilations. An arc is drawn between the columns of the blocks to be annihilated and the pivot column.

### 2.1. Permutated SUR models

Let \( \zeta = (\xi_1, \ldots, \xi_G) \) denote a permutation of the indices \((1, \ldots, G)\). Consider the estimation of the permutated ZR-VAR model

\[
\text{vec}(Y) = (\oplus_i X_{\xi_i})\text{vec}(\beta_{\xi_i}) + \text{vec}(U), \quad \text{vec}(U) \sim (0, \Sigma \otimes I_M),
\]

after (3) has been estimated. As in (4) the BLUE of (13) comes from the solution of the GLLSP

\[
\arg\min_{V, \beta_{\xi_i}} \| V \|_F^2 \quad \text{subject to} \quad \text{vec}(Y) = (\oplus_i X_{\xi_i})\text{vec}(\beta_{\xi_i}) + \text{vec}(VC^T),
\]

which requires the computation of the GQRD of the matrices \( \oplus_i X_{\xi_i} \) and \((C \otimes I_M)\). From (5a), (8) and (9) it follows that the QRD of \( \oplus_i X_{\xi_i} \) is given by

\[
Q_{\xi_i}^T(\oplus_i X_{\xi_i}) = \begin{pmatrix}
R_{\xi_i} & \\
0 & \end{pmatrix},
\]

where \( Q_{\xi_i} = (\oplus_i Q_{A_{\xi_i}} \oplus_i Q_{B_{\xi_i}}) \). The RQD of \( Q_{\xi_i}^T(C \otimes I_M) \) needs to be computed afresh. This is equivalent in constructing and re-triangularizing the matrix \( Q_{\xi_i}^T(C \otimes I_M)Q_{\xi_i} \) which has the same structure as \( W_0 \) in (10).

The computational burden of computing the RQD of \( Q_{\xi_i}^T(C \otimes I_M)Q_{\xi_i} \) can be reduced significantly in some cases. Consider the permutation \( \xi \) of the
indices 1, . . . , G such as \( \xi_i = i \) for \( i = 1, \ldots, t \) and \( 1 \leq t \leq G - 2 \). In this case, the matrix \( Q_t^i (C \otimes I_M)Q_{\xi} \) has the structure

\[
\hat{W}_0 = \begin{pmatrix}
\hat{W}^{(1,0)}_i \\
\hat{W}^{(2,0)}_i \\
\hat{W}^{(3,0)}_i \\
\hat{W}^{(4,0)}_i
\end{pmatrix},
\]

where

\[
\begin{pmatrix}
\hat{W}^{(1,0)}_{ij} \\
\hat{W}^{(2,0)}_{ij} \\
\hat{W}^{(3,0)}_{ij} \\
\hat{W}^{(4,0)}_{ij}
\end{pmatrix} = c_{ij} Q^T_{\xi_i} Q_{\xi_j}
\equiv
\begin{pmatrix}
\hat{W}^{(1,0)}_{ij} \\
\hat{W}^{(2,0)}_{ij} \\
\hat{W}^{(3,0)}_{ij} \\
\hat{W}^{(4,0)}_{ij}
\end{pmatrix}
\text{ for } i, j = 1, \ldots, t \text{ and } i \leq j.
\]

(16)

Given the RQDs (11) for \( i = 1, \ldots, t - 1 \), it follows that

\[
\hat{W}_{t-1} = \begin{pmatrix}
\hat{W}^{(1,t-1)}_{ij} \\
\hat{W}^{(2,t-1)}_{ij} \\
\hat{W}^{(3,t-1)}_{ij} \\
\hat{W}^{(4,t-1)}_{ij}
\end{pmatrix} = \begin{pmatrix}
\hat{W}^{(1,t-1)}_{ij} & \hat{W}^{(2,t-1)}_{ij} \\
0 & \hat{W}^{(4,t-1)}_{ij}
\end{pmatrix}
\]

if \( 1 \leq i \leq j \leq t \),

\[
\text{otherwise.}
\]

(17)

Thus, for obtaining the RQD of \( Q_t^i (C \otimes I_M) \) only the RQDs (11) for \( i = t, \ldots, G - 1 \) need to be computed.

Consider now the case where \( \xi_i = i \) for \( i = G - t + 1, \ldots, G \) and \( 1 \leq t \leq G - 2 \). The matrix \( Q_t^i (C \otimes I_M)Q_{\xi} \) has the structure

\[
\hat{W}_0 = \begin{pmatrix}
\hat{W}^{(1,0)}_i \\
\hat{W}^{(2,0)}_i \\
\hat{W}^{(3,0)}_i \\
\hat{W}^{(4,0)}_i
\end{pmatrix},
\]

where

\[
\begin{pmatrix}
\hat{W}^{(1,0)}_{ij} \\
\hat{W}^{(2,0)}_{ij} \\
\hat{W}^{(3,0)}_{ij} \\
\hat{W}^{(4,0)}_{ij}
\end{pmatrix} = c_{ij} Q^T_{\xi_i} Q_{\xi_j}
\equiv
\begin{pmatrix}
\hat{W}^{(1,0)}_{ij} \\
\hat{W}^{(2,0)}_{ij} \\
\hat{W}^{(3,0)}_{ij} \\
\hat{W}^{(4,0)}_{ij}
\end{pmatrix}
\text{ for } i, j = G - t + 1, \ldots, G
\]

and \( i \leq j \).

(18)
The computation (12) can be utilized to derive

\[
\tilde{W}_{ij}^{(t-1)} = \begin{cases} 
\left( \begin{array}{c}
\tilde{W}^{(1,0)}_{ij} \\
\tilde{W}^{(2,0)}_{ij} \\
\tilde{W}^{(3,0)}_{ij} \\
\tilde{W}^{(4,0)}_{ij}
\end{array} \right) & \text{if } 1 \leq i \leq G \text{ and } 1 \leq j \leq G - t, \\
\begin{array}{c}
\tilde{W}^{(1,t-1)}_{ij} \\
\tilde{W}^{(2,t-1)}_{ij} \\
0 \\
\tilde{W}^{(3,t-1)}_{ij} \\
\tilde{W}^{(4,t-1)}_{ij}
\end{array} & \text{if } G - t + 1 \leq i \leq j \leq G,
\end{cases}
\]

(19)

Here, the blocks \(\tilde{W}_{ij}^{(l-1)}\) \((l = 1, 2, 3, 4)\) are obtained in \(t - 1\) steps by setting \(\tilde{W}_{ij}^{(l,0)} = \tilde{W}_{ij}^{(l,0)}\) and computing the modifications

\[
\begin{pmatrix}
\tilde{W}_{1,G-i+1}^{(r,i-1)} & \cdots & \tilde{W}_{1,G}^{(r,i-1)} \\
\vdots & \ddots & \vdots \\
\tilde{W}_{G-i-1,1,G-1}^{(r,j-1)} & \cdots & \tilde{W}_{G-i-1,G-1}^{(r,j-1)} \\
\tilde{W}_{G-i-1,1,G-1}^{(r+1,j-1)} & \cdots & \tilde{W}_{G-i-1,G-1}^{(r+1,j-1)}
\end{pmatrix} \hat{P}_i
\]

(20)

where \(r = 1, 3\) and \(i = 1, \ldots, t - 1\). After \(\tilde{W}_{t-1}\) in (19) has been constructed the RQD of \(Q_{\xi}^1(C \otimes I_M)\) is obtained by computing (12) for \(i = t, \ldots, G - 1\).

3. Generating all the permuted ZR-VAR models

There are \(G!\) possible ways of permuting the matrices \(X_1, \ldots, X_G\) in the ZR-VAR model (3). The estimation of the resulting models is derived by enumerating the \(G!\) permutations \(\xi = (\xi_1, \ldots, \xi_G)\) and solving the corresponding GLLSP (14). For simplicity let “model \(\xi\)” denote the model in (13) corresponding to a permutation \(\xi\). Recall that for the solution of (14) only the RQD of \(Q_{\xi}^1(C \otimes I_M)\) is required since the QRD of \(\oplus \xi X_{\xi_i}\) is available.

Algorithms which generate the sequence of all permutations have been extensively discussed (Knuth, 1973; Reingold et al., 1977). A convenient method to generate all permutations is to derive one permutation from another by applying an adjacent transposition (Reingold et al., 1977, pp. 168–171). This approach yields an efficient
The algorithm for generating all the permuted ZR-VAR models. The order of permutations is illustrated in Fig. 3 for the case $G = 4$. The adjacent transposed indices are shadowed.

Let $\tau^{(j)}$ ($j = 1, \ldots, (G - 1)!$) denote the permutations of the indices $\{1, 2, \ldots, G - 1\}$ such that they derive from one to another by applying an adjacent transposition. The superscript indicates the generation order. The permutations $\xi$ of order $G$ are obtained by inserting the new element $G$ in all possible positions of each $\tau$. That is, $\xi^{(j)} = (\tau^{(j)}, G)$, $\xi^{(j+1)} = (\tau^{(j+1)}, G)$, $\xi^{(j+2)} = (\tau^{(j)}, G, \tau^{(j+1)})$, $\xi^{(j+3)} = (\tau^{(j+1)}, G, \tau^{(j+2)})$, $\xi^{(j+4)} = (\tau^{(j)}, G, \tau^{(j+1)}, G)$, $\xi^{(j+5)} = (\tau^{(j+1)}, G, \tau^{(j+2)}, G)$, $\xi^{(j+6)} = (\tau^{(j)}, G, \tau^{(j+1)}, \tau^{(j+2)}, G)$, $\xi^{(j+7)} = (\tau^{(j+1)}, G, \tau^{(j+2)}, \tau^{(j+3)}, G)$, $\xi^{(j+8)} = (\tau^{(j)}, G, \tau^{(j+1)}, \tau^{(j+2)}, \tau^{(j+3)}, G)$, $\xi^{(j+9)} = (\tau^{(j+1)}, G, \tau^{(j+2)}, \tau^{(j+3)}, \tau^{(j+4)}, G)$, $\xi^{(j+10)} = (\tau^{(j)}, G, \tau^{(j+1)}, \tau^{(j+2)}, \tau^{(j+3)}, \tau^{(j+4)}, G)$, $\xi^{(j+11)} = (\tau^{(j+1)}, G, \tau^{(j+2)}, \tau^{(j+3)}, \tau^{(j+4)}, \tau^{(j+5)}, G)$, $\xi^{(j+12)} = (\tau^{(j)}, G, \tau^{(j+1)}, \tau^{(j+2)}, \tau^{(j+3)}, \tau^{(j+4)}, \tau^{(j+5)}, G)$, $\xi^{(j+13)} = (\tau^{(j+1)}, G, \tau^{(j+2)}, \tau^{(j+3)}, \tau^{(j+4)}, \tau^{(j+5)}, \tau^{(j+6)}, G)$, and so on...

The sequence of $G!$ permutations is divided in $(G - 1)!/2$ sub-sequences of length $2G$. They have the form $\xi^{(j+1)}, \xi^{(j+2)}, \ldots, \xi^{(G+1)}$, $\xi^{(G+2)}, \ldots, \xi^{(j+G)}$ where $j = 1, \ldots, (G - 1)!$ and $(j \mod 2) = 1$. For the estimation of the model $\xi^{(j+G+1)}$ the RQD of $W_0$ in (10) is computed by using (11) for $i = 1, \ldots, G$. The next $G - 2$ models, $\xi^{(j+G+2)}, \ldots, \xi^{(j+G-1)}$ are derived by applying the transpositions $(t + 1, t + 2)$, where $t = G - 2, \ldots, 1$. It follows that $\xi^{(j-1)} = \xi^{(j+G+1)}$ for $i = 1, \ldots, t$. Thus, for the estimation of the model $\xi^{(j+G)}$ the construction of $\hat{W}_{i-1}$ in (17) is followed by computation (11) for $i = t, \ldots, G - 1$ and $t = G - 2, \ldots, 1$. Finally, for the estimation of the model $\xi^{(G)}$ there is no computation which can be re-utilized and, thus, the RQD of $W_0$ is computed afresh.

**Algorithm 1** Generate the sequence of $G!$ permutations of $\{1, 2, \ldots, G\}$ by adjacent transpositions.

1: **procedure** Permutations($G$, $\Pi$)  
2: \hspace{1cm} Let $\zeta_0 \leftarrow i$; \hspace{0.5cm} $\theta_i \leftarrow i$; \hspace{0.5cm} $\delta_i \leftarrow -1$; \hspace{0.5cm} where $i = 1, \ldots, G$  
3: \hspace{1cm} Let $\delta_1 \leftarrow 0$; \hspace{0.5cm} $\zeta_0 \leftarrow G + 1$; \hspace{0.5cm} $\zeta_{G+1} \leftarrow G + 1$; \hspace{0.5cm} $j \leftarrow 0$; \hspace{0.5cm} $i \leftarrow G + 1$  
4: \hspace{1cm} while ($i \neq 1$) do  
5: \hspace{1.5cm} $j \leftarrow j + 1$; \hspace{0.5cm} $\Pi_{ij} \leftarrow \zeta^T$ \hspace{1cm} [a new permutation $\zeta$ has been generated]
6: \( i \leftarrow G \)
7: while \((\xi_{\theta,i} + \delta, i > i)\) do
8: \( \delta_i \leftarrow -\delta_i; \quad i \leftarrow i - 1 \)
9: end while
10: \( \xi_{\theta_i} \leftarrow \xi_{\theta,i} + \delta_i; \quad \theta_{\xi_i} \leftarrow \theta_i \)
11: end while
12: end procedure

Similarly, for the estimation of the model \( \tilde{\xi}^{(G+1)} \) the RQD of \( W_0 \) in (10) is computed using (12a) and (12b) for \( i = 1, \ldots, G - 1 \). The models \( \tilde{\xi}^{(G+2)}, \ldots, \tilde{\xi}^{(G+G-1)} \) result by applying the transpositions \( (G - t - 1, G - t) \), where \( t = G - 2, \ldots, 1 \). Thus, \( \tilde{\xi}^{(G+G-t)} = \tilde{\xi}^{(G+1)} \) for \( i = G - t + 1, \ldots, G \). Furthermore, for the estimation of the model \( \tilde{\xi}^{(G+G-t)} \) the computation of \( \tilde{W}_{t-1} \) in (19) and that of (12) are required for \( i = t, \ldots, G - 1 \) and \( t = G - 2, \ldots, 1 \).

This procedure is summarized by Algorithm 2 which first calls Algorithm 1 to construct the matrix \( \Pi \) containing the sequence of \( G! \) permutations. This may result into storage problems, but Algorithms 1 and 2 can be combined to offset this problem. I.e., once a new permutation \( \xi \) is generated (line 5 of Algorithm 1) the corresponding permuted model is estimated (lines 5–28 of Algorithm 2). Here, for the clarity and emphasis of the main computational steps of Algorithm 2 the index of the permutations are assumed to be generated by the Algorithm 1. Initially, the QRD (5a) is computed (line 3). Then, each iteration of the repetitive structure (lines 5–28) estimates a sequence of \( 2 \times G \) permuted models. Specifically, it efficiently forms the RQD (5b) by utilizing previous computation and obtains the BLUE of the corresponding permuted models (13).

Algorithm 2 Estimating all \( G! \) permuted ZR-VAR models, where \( G \) is the order of the VAR process.

1: procedure EstimatePerm( \( G, \Theta_i X_i, \text{vec}([y_i]), C, \{\hat{\beta}^{(i)}\}_{G!} \) )
2: \( \text{call Permutations}(G, \Pi) \)
3: \( \text{Compute the QRD (5a) using (8)} \)
4: \( \text{Let } j \leftarrow 0; \quad d \leftarrow -1 \)
5: while \( (j < G!) \) do \hspace{1cm} \( \text{[each iteration estimates } 2G \text{ permuted models]} \)
6: \( j \leftarrow j + 1; \quad \tilde{\xi} \leftarrow \Pi_j; \quad \text{[a new model } \xi \text{ is considered]} \)
7: \( \text{Construct } Q_{\tilde{\xi}} \equiv (\oplus_i Q_{A_{\tilde{\xi}}^i} \quad \oplus_i Q_{B_{\tilde{\xi}}^i}); \quad \text{Compute } \)
8: \( \tilde{W}_0 \equiv Q_{\tilde{\xi}}^T(C \otimes I_M)Q_{\tilde{\xi}} \)
9: if \( (d < 0) \) then
10: \( \text{Compute the RQDs (11), } i = 1, \ldots, G - 1 \text{ and thus obtain the RQD (5b)} \)
11: Compute the BLUE \( \hat{\beta}^{(\tilde{\xi})} \) of the ZR-VAR (13) using (7)
12: for \( t = G - 2, \ldots, 0 \) do
\[ j \leftarrow j + 1; \quad \xi \leftarrow \Pi J_\xi; \quad \text{[a new model } \xi \text{ is considered]} \]

Construct \( \hat{W}_{\xi}^{(t-1)} \) in (17)

Compute the RQDs (11), \( i = t, \ldots, G - 1 \) and thus obtain the RQD (5b)

Compute the BLUE \( \hat{\beta}_\xi \) of the ZR-VAR (13) using (7)

\[ j \leftarrow j + 1; \quad \xi \leftarrow \Pi J_\xi; \quad \text{[a new model } \xi \text{ is considered]} \]

Construct \( \hat{W}_{\xi}^{(t-1)} \) in (19)

Compute the RQDs (12a) and perform (12b), \( i = t, \ldots, G - 1 \) and thus obtain the RQD (5b)

Compute the BLUE \( \hat{\beta}_\xi \) of the ZR-VAR (13) using (7)

\[ \text{end for} \]

else

Compute the RQDs (12a) and perform (12b), \( i = 1, \ldots, G - 1 \) and thus obtain the RQD (5b)

Compute the BLUE \( \hat{\beta}_\xi \) of the ZR-VAR (13) using (7)

for \( t = 0, \ldots, G - 2 \) do

\[ j \leftarrow j + 1; \quad \xi \leftarrow \Pi J_\xi; \quad \text{[a new model } \xi \text{ is considered]} \]

Construct \( \hat{W}_{\xi}^{(t-1)} \) in (19)

Compute the RQDs (12a) and perform (12b), \( i = t, \ldots, G - 1 \) and thus obtain the RQD (5b)

Compute the BLUE \( \hat{\beta}_\xi \) of the ZR-VAR (13) using (7)

\[ \text{end for} \]

\[ \text{end if} \]

\[ d \leftarrow -d \]

\[ \text{end while} \]

\[ \text{end procedure} \]

3.1. Complexity considerations

For the estimation of the \( G! \) permuted ZR-VAR models the most demanding computation is the GQRD (5). The QRD (5a) is available throughout the generating process. Thus, for the complexity analysis only the RQD (5b) will be taken into account. Furthermore, for the derivation of the theoretical measures of complexity it will be assumed that \( k_1 = k_2 = \cdots = k_G = pG \), where \( p \geq 1 \) and \( pG \leq M \). All quantities denote floating point operations (flops) (Golub and Van Loan, 1996).

The complexities of computing afresh the RQD of \( W_0 \) and the GQRD (5) are given, respectively, by

\[ C_W = 4G^3M^3/3 \]

and

\[ C_{\text{GQRD}} = 2G^3(2M^3 + 6pGM^2 + 3pM - p^3G^3)/3. \]

In Section 2 two strategies for re-triangularizing \( W_0 \) in (10) have been presented. Each of these strategies consists of \( G - 1 \) steps that correspond to computations (11) and (12). The number of flops required by the \( i \)th \( (i = 1, \ldots, G - 1) \) step of these
computations is given, respectively, by
\[
St_1(i) = i(M - pG)(M^2(3i + 1) + M(2pG(3i + 2) + 3i - 3) + pG(pG(3i + 7) + 3i + 9))/3
\]
and
\[
St_2(i) = 2i(M - pG)(M^2(3G - 3i - 1) + M(pG(3G - 1) + 3G - 3i - 3)) + pG(pG(3i + 2) + 3i + 3)/3.
\]
The complexity of constructing \( W_0 \) in (10) is
\[
C_{W_0} = GM^2(G - 1)(2M - 1)/2.
\]
Thus, the overall complexities of computing the RQD of \( W_0 \) using the column-wise and row-wise block-annihilation strategies are given, respectively, by
\[
C_1 = C_{W_0} + \sum_{i=1}^{\frac{G-1}{2}} St_1(i)
\]
\[
= G(G - 1)(2M^3(G + 3) + M^2(2G(p + 1) + 2pG^2 - 7) - 2pGM(pG^2 - 2pG - 6) - 2p^2G^2(pG^2 + 3pG + G + 4))/6
\]
\[
\approx G^2((G + 3)M^3 + pG^2M^2 - p^3G^4)/3
\]
and
\[
C_2 = C_{W_0} + \sum_{i=1}^{\frac{G-1}{2}} St_2(i)
\]
\[
\]
\[
\approx G^2((G + 3)M^3 + 2pG^2M^2 - p^2G^3M - 2p^3G^4)/3.
\]
Recall (see Section 2) that the estimation of the sub-sequences of \( G \) models \( \zeta^{(G-1)}, \ldots, \zeta^{(G)} \) \( (j = 1, \ldots, (G - 1)! \) and \( (j \mod 2) = 1 \) \) can be obtained by employing (11) and utilizing previous computations. This requires
\[
C_3(G) = C_{W_0} + 2C_1 + \sum_{i=1}^{\frac{G-2}{2}} \sum_{t=i}^{\frac{G-1}{2}} St_1(i)
\]
\[
\approx G^3(M - pG)(M + pG)^2/4.
\]
For the estimation of the remaining \( G \) models i.e., \( \zeta^{(G+1)}, \ldots, \zeta^{(2G)} \), the computation (12) is required. In addition, (20) is computed which has complexity
\[
Ext(i) = 2iM(G - i - 1)(M - pG)(M + pG + 1).
\]
Thus, the complexity of the estimation procedure of the $G$ models in this case is given by

$$C_4(G) = C_{W_0} + 2C_2 + \sum_{t=1}^{G-2} \left( \sum_{i=1}^{t-1} \text{Ext}(i) + \sum_{i=t}^{G-1} \text{St}_2(i) \right)$$


The orders of complexities of the various methods for computing the RQD of $W_0$ in (10) are summarized in Table 1. Fig. 4 plots the ratios between the complexities of computing the RQD of $W_0$ using the column- and row-wise strategies and the one of computing the RQD of $W_0$ afresh for $G = 5, \ldots, 100$ and $M = 500, \ldots, 10000$. Fig. 5 shows the complexity ratios of generating the two sub-sequences of $G$ models. Here, the ratios are between the strategies which utilize previous computation and computing the RQDs afresh. Notice that these plots confirm the orders of complexity shown in Table 1.
Summarizing, the complexity of generating the $G!$ ZR-VAR permuted models i.e., computing the RQD (5b) by employing Algorithm 2 is given by

$$C_{Tot} = \frac{(G-1)!((C_3(G) + C_4(G))/2}{2}.$$ 

Thus, the efficiency obtained by employing the Algorithm 2, rather than estimating the models afresh is given by the ratio

$$C_{Tot}/(G!C_W) \approx 0.25.$$ 

Now, taking into account the computational complexity of the QRD (5a) the latter becomes

$$C_{Tot}/(G!C_{GQRD}) \approx 0.19.$$ 

4. Conclusions

Within the context of subset VAR model selection a computationally efficient algorithm for estimating SUR models with permuted exogenous data matrices has been proposed. The main computation required for the estimation of the models is the generalized QR decomposition (5). The particular properties of the models are investigated and used to derive computationally efficient means for estimating the models. Specifically, the QR factorization (5a) of the exogenous data matrices is computed only once for the original model, and it is available for the estimation of the remaining $G! - 1$ models. Furthermore, two efficient strategies for computing the RQD (5b) have been presented. These strategies are adapted to the special cases of the permuted models so that the previous computation is re-utilized. These special cases arise when the $G!$ models are obtained such that the permutations of the block matrices derive one from another by applying an adjacent transposition. The ratio between the complexity—in term of flops—of the proposed strategy and that of deriving the estimators afresh is approximatively $1/4$. This ratio is obtained without
taking into consideration that the QRD (5a) is available throughout the generating process. Otherwise, the efficiency of the proposed algorithm would have been even greater (approximatively 1/5). The implementation and the application of the proposed algorithm need to be pursued.

The problem of estimating the SUR models with permuted exogenous data matrices arises in subset VAR modeling (Bozdogan and Bearse, 2003; Gatu and Kontoghiorghes, 2005; Lütkepohl, 1993; Maringer, 2004; Winker, 2001). The estimation of all subset VAR models is infeasible even for modest dimensions of the original model. However, an algorithm that derives all possible subset VAR models by deleting a single coefficient at one time has been previously introduced (Gatu and Kontoghiorghes, 2005). The algorithm generates a regression tree (Gatu and Kontoghiorghes, 2003). The tree structure suggests that a branch-and-bound approach should be designed. Thus, the best-subset VAR models could be derived without computing the whole regression tree which generates all sub-models (Gatu and Kontoghiorghes, 2005; Hand, 1981). The branch-and-bound method which employs a cutting test based on statistical criteria such as BIC, AIC or ICOMP is currently investigated (Akaike, 1969, 1974; Bozdogan and Bearse, 2003; Schwarz, 1978).

Alternative approaches to the exhaustive methods merit also to be investigated. The first is the use of heuristics and genetic algorithms (Bozdogan and Bearse, 2003; Winker, 2000). A second alternative is to search through restricted subsets of models. The proposed strategy for estimating the permuted ZR-VAR models complements the recently reported methods in Gatu and Kontoghiorghes (2005) that generate subset VAR models by deleting the same variables, or proper subsets of variables, from each block in (3). In all of these cases, the computational burden of deriving the generalized QR decomposition (5) and, thus, estimating the models, is significantly reduced. In order to make the applicability of these methods feasible, a common framework (i.e., tree-structure) should be designed. This will allow a full investigation of the subset VAR models. The development of such framework is currently under investigation.

References


