

Program ssvs.m

Step 1. Generate VAR data

Simulate VAR DGP from model:

$$Y_t = Y_{t-1}\Phi + \varepsilon_t$$

where Y_t is of dimensions $[T \times p]$ with $T=600$ and $p=6$.

with coefficients:

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Note that no CONSTANT is included})$$

$$\Psi = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

, where $\Sigma = \Psi'\Psi$ so that Ψ is the Cholesky decomposition (upper triangular matrix) of Σ .

These are the defaults for the function *simvardgp.m*; that is if you write:

```
y = simvardgp();
```

You get a matrix Y generated from the specification above.

Another option is to provide yourself the number of lags, observations, series, phi and psi parameters in the following way:

```
y = simvardgp(T,N,L,PHI,PSI);
```

For more information about the inputs in *simvardgp.m* see the documentation inside the file *or* type:

```
help simvardgp
```

in the MATLAB command window.

Step 2. Data transformations

The model we are using in the SSVS algorithm is of the form:

$$y_t' = z_t' C + \sum_{j=1}^L y_{t-j}' A_j + \varepsilon_t',$$

where for $t=1,2,\dots,T$ and z_t is an h -dimensional vector of exogenous variables, the lag L is a known positive integer, the regression coefficients C and A_j are $h \times p$ and $p \times p$ unknown matrices and the error terms are iid Normal with covariance matrix Σ .

Define $x_t' = (z_t', y_{t-1}', \dots, y_{t-L}')'$. The VAR model above can be written in matrix form:

$$Y = X\Phi + \varepsilon,$$

where $Y = \begin{pmatrix} y_1' \\ \vdots \\ y_T' \end{pmatrix}$, $X = \begin{pmatrix} x_1' \\ \vdots \\ x_T' \end{pmatrix}$, $\Phi = \begin{pmatrix} C \\ A_1 \\ \vdots \\ A_L \end{pmatrix}$, $\varepsilon = \begin{pmatrix} \varepsilon_1' \\ \vdots \\ \varepsilon_T' \end{pmatrix}$. Here Y and ε are $T \times p$ matrices, Φ is a

$(h+Lp) \times p$ matrix, x_t' is a $1 \times (h+Lp)$ row vector, and X is a $T \times (h+Lp)$ matrix of observations.

In the program (and the VAR DGP) we assume that there is no constant term and exogenous (z_t) variables, so that the model is of the form (in vector notation):

$$Y_t = Y_{t-L}\Phi + \varepsilon_t$$

for appropriately defined vectors.

The only thing you have to choose in the program is the number of lags L . All other information like the dimensions of matrix Y_t (the dimensions are T and p) are extracted automatically. Matrix of lagged Y 's is generated automatically using function *mlag.m*

Step 3. Names and storage spaces

Before we proceed with the definition of priors, we set some Gibbs related preliminaries (number of iterations, burn-in draws, thin value) and then the storage matrices for the draws and some "intermediate" variables used in estimation of the posteriors.

Step 4. Priors

Let $n = (h+Lp)p$, the total number of regression coefficients. Denote $\phi = \text{vec}(\Phi) = (\phi_1, \phi_2, \dots, \phi_n)'$. For $j = 2, \dots, p$, let $\eta_j = (\psi_{1j}, \dots, \psi_{j-1,j})'$. Write $\eta = (\eta_2', \dots, \eta_p')'$ and $\psi = (\psi_{11}, \dots, \psi_{pp})'$.

That is, $\psi = (\psi_{11}, \dots, \psi_{pp})'$ contains all the diagonal elements of Ψ , while the vectors $\eta_j = (\psi_{1j}, \dots, \psi_{j-1,j})'$ are obtained as (for a 6×6 matrix Ψ):

$$\Psi = \begin{bmatrix} \Psi_{11} & \Psi_{12} & \Psi_{13} & \Psi_{14} & \Psi_{15} & \Psi_{16} \\ 0 & \Psi_{22} & \Psi_{23} & \Psi_{24} & \Psi_{25} & \Psi_{26} \\ 0 & 0 & \Psi_{33} & \Psi_{34} & \Psi_{35} & \Psi_{36} \\ 0 & 0 & 0 & \Psi_{44} & \Psi_{45} & \Psi_{46} \\ 0 & 0 & 0 & 0 & \Psi_{55} & \Psi_{56} \\ 0 & 0 & 0 & 0 & 0 & \Psi_{66} \end{bmatrix}$$

$\begin{matrix} \eta_2 & \eta_3 & \eta_4 & \eta_5 & \eta_6 \end{matrix}$

We propose hierarchical priors on (ϕ, η, ψ) . Here we assume that all the ϕ parameters are restricted, so that if m is the number of restrictions, in that case $m = n = (h + Lp)p$.

Inside the program:

- All the matrices that are used to store the draws are of the form “*name*” plus the suffix “*_draws*”

- **PHI_M** is the MLE of Φ . This is estimated as:

$$\hat{\Phi}_M = (X'X)^{-1}X'Y$$

- **SSE** is the MLE sum of squared errors (of residuals) such that:

$$\hat{\Sigma}_M = \frac{1}{T} SSE$$

- **phi_m_vec** is the vector created from stacking the columns of PHI_M

Note: the vector $\psi = (\psi_{11}, \dots, \psi_{pp})'$ is not defined in the program. We will see later that we will sample from its squared elements!

(i) Priors on $\phi = \text{vec}(\Phi) = (\phi_1, \phi_2, \dots, \phi_n)'$:

Let $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_n)$ be a vector of 0–1 variables and let $D = \text{diag}(h_1, h_2, \dots, h_n)$ where

$$h_i = \begin{cases} \tau_{0i}, & \text{if } \gamma_i = 0, \\ \tau_{1i}, & \text{if } \gamma_i = 1, \end{cases}$$

with preselected constants $\tau_{0i} < \tau_{1i}$. The prior for ϕ_m (remember, $m = n = (h + Lp)p$) conditional on γ is

$$(\phi_m | \gamma) \sim N_m(\mathbf{0}, DRD),$$

where R is a preselected correlation matrix. Under this prior, each element ϕ_m has the distribution

$$(\phi_i | \gamma_i) \sim (1 - \gamma_i)N(0, \tau_{0i}^2) + \gamma_i N(0, \tau_{1i}^2)$$

Inside the program:

- **tau_0** and **tau_1** are τ_{0i} and τ_{1i} respectively. These are (0.1 , 5) as suggested by the authors.
- **f_m** is the mean of ϕ_m . This is always zero, but we define a variable for it, anyway.
- **h_i** is $h_i = \begin{cases} \tau_{0i}, & \text{if } \gamma_i = 0, \\ \tau_{1i}, & \text{if } \gamma_i = 1, \end{cases}$
- **D** is $\mathbf{D} = \text{diag}(h_1, h_2, \dots, h_n)$
- **R** is the preselected correlation matrix \mathbf{R} . This is the identity matrix as suggested by the authors.
- **DRD_j** is the prior covariance matrix \mathbf{DRD} .

(ii) Priors on $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_n)$:

We assume that the elements of $\boldsymbol{\gamma}$ are independent Bernoulli $p_i \in (0,1)$ random variables so that

$$P(\gamma_i = 1) = p_i, \quad P(\gamma_i = 0) = 1 - p_i, \quad i = 1, 2, \dots, m(=n)$$

Inside the program:

- **gamma** is the vector $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_n)$. Starting values = all elements equal to 1.
- **p_i** is $p_i \in (0,1)$ which is set to 0.5, like in the paper.

(iii) Priors on $\boldsymbol{\eta} = (\psi_{12}, \psi_{13}, \psi_{23}, \dots, \psi_{p-1,p})'$:

For $j = 2, \dots, p$, let $\boldsymbol{\omega}_j = (\omega_{1j}, \dots, \omega_{j-1,j})'$ be a vector of 0–1 variables, and let $\mathbf{D}_j = \text{diag}(h_{1j}, \dots, h_{j-1,j})$ where

$$h_{ij} = \begin{cases} \kappa_{0ij}, & \text{if } \omega_{ij} = 0, \\ \kappa_{1ij}, & \text{if } \omega_{ij} = 1, \end{cases}$$

with preselected constants $\kappa_{0ij} < \kappa_{1ij}$. Letting \mathbf{R}_j be a preselected $(j-1) \times (j-1)$ correlation matrix, the prior we consider for $\boldsymbol{\eta}_j$, conditional on $\boldsymbol{\omega}_j$, is

$$(\boldsymbol{\eta}_j \mid \boldsymbol{\omega}_j) \sim N_{j-1}^{iid}(\mathbf{0}, \mathbf{D}_j \mathbf{R}_j \mathbf{D}_j), \quad \text{for } j = 2, \dots, p.$$

Under this prior, each element of $\boldsymbol{\eta}_j$ has distribution

$$(\eta_{ij} \mid \omega_{ij}) \sim (1 - \omega_{ij})N(0, \kappa_{0ij}^2) + \omega_{ij}N(0, \kappa_{1ij}^2), \quad \text{for } i = 1, \dots, j-1.$$

Inside the program:

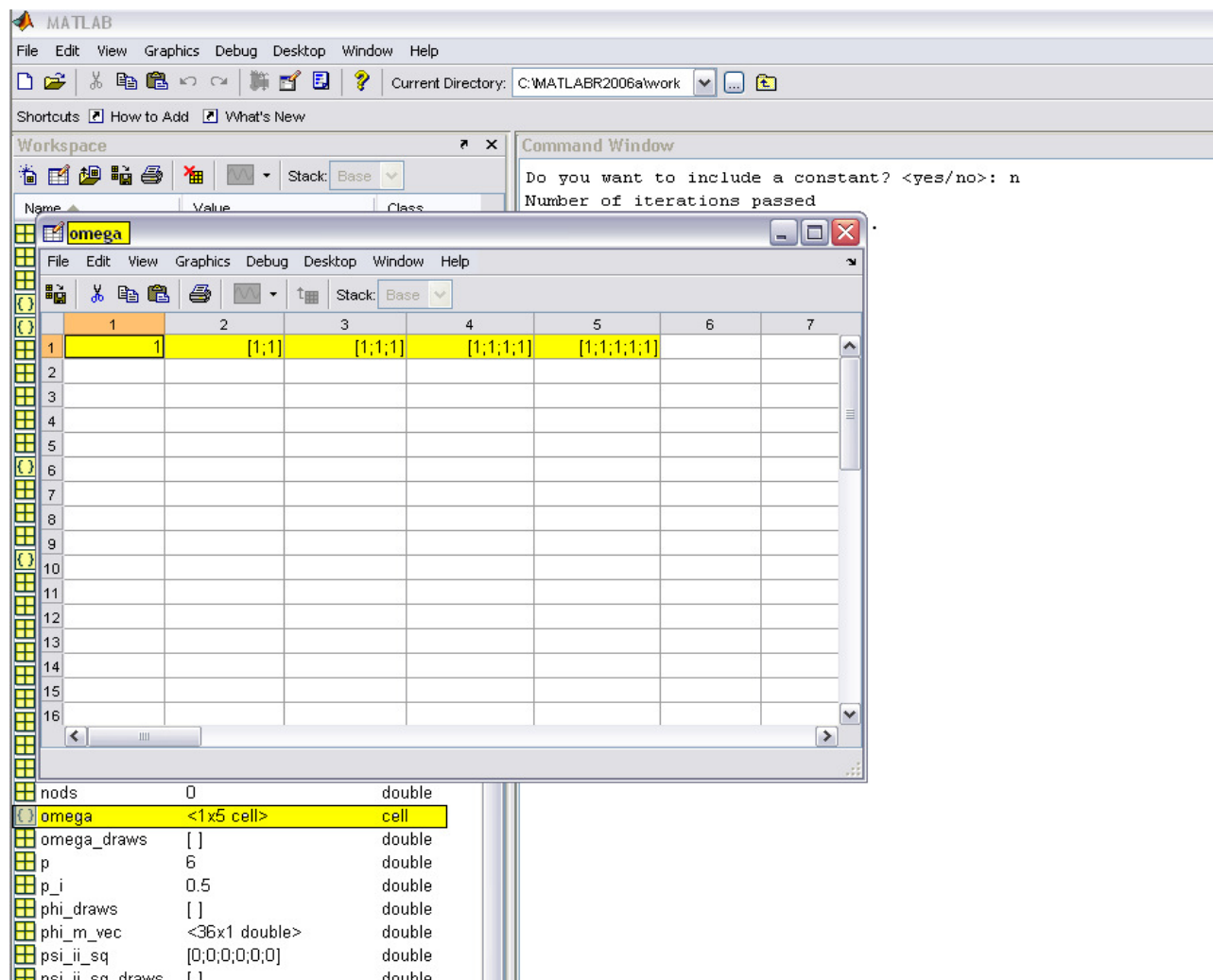
- **kappa_0** and **kappa_1** are κ_{0ij} and κ_{1ij} respectively. These are (0.1 , 5) as suggested by the authors.
- **D_j** is $\mathbf{D}_j = \text{diag}(h_{1j}, \dots, h_{j-1,j})$
- **DiRiDi** is $\mathbf{D}_j \mathbf{R}_j \mathbf{D}_j$, that is the prior variance-covariance matrix
- **R_j** is a cell array containing the $(j-1) \times (j-1)$ preselected correlation matrices \mathbf{R}_j . This is the identity matrix as suggested by the authors.
- **h** is a cell array containing $h_{ij} = \begin{cases} \kappa_{0ij}, & \text{if } \omega_{ij} = 0, \\ \kappa_{1ij}, & \text{if } \omega_{ij} = 1, \end{cases}$ elements

A short note on cell arrays:

While a conventional cell can accept one element (number – scalar), like it is in an Excel spreadsheet, MATLAB allows you to define *cell arrays*. In a cell array each cell can contain from a simple scalar to a vector or even a matrix. Thus when we define in the program:

```
for kk_1 = 1:(p-1)
    omega{kk_1} = ones(kk_1,1); % Omega_j
end
```

this will create a cell array where the first cell (for $kk_1 = 1$) is 1×1 vector of ones (and thus the number/scalar 1), the second cell (for $kk_1 = 2$) is 2×1 vector of ones, the third cell (for $kk_1 = 3$) is 3×1 vector of ones, and so on. This will look like in MATLAB as below:



Standard operators are INVALID with cell arrays. For example:

`5*omega` or `5*omega(2,1)`
will not work. What is valid is

`5*omega{2}` or `5*cell2mat(omega(2))`

The function *cell2mat* converts a cell to a matrix. I used this method instead of using *name{cell_number}* (like above in `omega{2}`)

Note:

The same “trick” is used for all the other matrices that their dimension is growing with there index *j*.

(iv) Priors on $\omega = (\omega_2, \dots, \omega_p)'$. We assume that the elements of ω are independent Bernoulli $q_{ij} \in (0,1)$ random variables so that

$$P(\omega_{ij} = 1) = q_{ij}, \quad P(\omega_{ij} = 0) = 1 - q_{ij}, \quad i = 1, \dots, p, \quad j = 1, \dots, p-1$$

Inside the program:

- **omega** is a cell array containing $\omega_j = (\omega_{1,j}, \dots, \omega_{j-1,j})'$
- **q_ij** is $q_{ij} \in (0,1)$

(v) Priors on $\psi = (\psi_{11}, \dots, \psi_{pp})'$:

Assume that $\psi_{ii}^2 \stackrel{ind}{\sim} \text{gamma}(a_i, b_i)$ distributions. Here (a_i, b_i) are positive constants. So for $i = 1, \dots, p$, ψ_{ii} has the prior density

$$[\psi_{ii}] = \frac{2b_i^{a_i}}{\Gamma(a_i)} \psi_{ii}^{2(a_i-1)} \exp(-b_i \psi_{ii}^2), \quad \text{for } \psi_{ii} > 0.$$

Inside the program:

- **psi_ii_sq** is a $p \times 1$ vector with elements ψ_{ii}^2
- **a_i** and **b_i** are (a_i, b_i)

We have finished with the prior specification. Now we define some matrices that we will use in order to update the priors and thus get the posteriors. These matrices as logical all come from the (conditional) likelihood function(s).

Remember we have defined $\hat{\Sigma}_M = \frac{1}{T} SSE$ (ML estimator of error covariance)? Write $SSE = S(\Phi) = (s_{ij})$. For $j = 2, \dots, p$, define $s_j = (s_{1j}, \dots, s_{j-1,j})'$. Let S_j be the upper-left $j \times j$ submatrix of $SSE = S(\Phi)$.

Inside the program:

- **S** is a cell array containing S_j for $j = 2, \dots, p$
 - **s** is a cell array containing $s_j = (s_{1j}, \dots, s_{j-1,j})'$ for $j = 2, \dots, p$
-

Step 5. Sampling from the posterior

1. Draw $(\psi_{(k)} | \phi_{(k-1)}, \gamma_{(k-1)}, \omega_{(k-1)}; \mathbf{Y})$ from the gamma distribution

$$(\psi_{ii}^2 | \boldsymbol{\phi}, \boldsymbol{\gamma}, \boldsymbol{\omega}; \mathbf{Y}) \sim \text{gamma}\left(a_i + \frac{1}{2}T, B_i\right),$$

where

$$B_i = \begin{cases} b_1 + \frac{1}{2}s_{11}, & \text{if } i = 1, \\ b_i + \frac{1}{2}\{s_{ii} - s_i'[S_{i-1} + (\mathbf{D}_i \mathbf{R}_i \mathbf{D}_i)^{-1}]^{-1} s_i\}, & \text{if } i = 2, \dots, p. \end{cases}$$

2. Draw $(\eta_{(k)} | \psi_{(k)}, \phi_{(k-1)}, \gamma_{(k-1)}, \omega_{(k-1)}; \mathbf{Y})$ from normal distribution

$$(\eta_j | \boldsymbol{\psi}, \boldsymbol{\phi}, \boldsymbol{\gamma}, \boldsymbol{\omega}; \mathbf{Y}) \sim N_{j-1}(\boldsymbol{\mu}_j, \mathbf{A}_j),$$

where

$$\begin{aligned} \boldsymbol{\mu}_j &= -\psi_{jj} \left\{ \mathbf{S}_{j-1} + (\mathbf{D}_j \mathbf{R}_j \mathbf{D}_j)^{-1} \right\}^{-1} \mathbf{s}_j \\ \mathbf{A}_j &= \left\{ \mathbf{S}_{j-1} + (\mathbf{D}_j \mathbf{R}_j \mathbf{D}_j)^{-1} \right\}^{-1} \end{aligned}$$

3. Draw $(\omega_{(k)} | \eta_{(k)}, \psi_{(k)}, \phi_{(k-1)}, \gamma_{(k-1)}, \omega_{(k-1)}; \mathbf{Y})$ from Bernoulli distribution

$$\omega_{ij} \sim \text{Bernoulli}(u_{ij1}/u_{ij1} + u_{ij2}),$$

where

$$\begin{aligned} u_{ij1} &= \frac{1}{\kappa_{1ij}} \exp\left(-\frac{\psi_{ij}^2}{2\kappa_{1ij}^2}\right) q_{ij} \\ u_{ij2} &= \frac{1}{\kappa_{0ij}} \exp\left(-\frac{\psi_{ij}^2}{2\kappa_{0ij}^2}\right) (1 - q_{ij}). \end{aligned}$$

4. Draw $(\phi_{(k-1)} | \gamma_{(k-1)}, \boldsymbol{\Sigma}_{(k)}, \omega_{(k-1)}; \mathbf{Y})$ from normal distribution, where $\boldsymbol{\Sigma}_{(k)}$ is computed from $\psi_{(k)}$ and $\eta_{(k)}$

$$(\boldsymbol{\phi} | \boldsymbol{\gamma}, \boldsymbol{\eta}, \boldsymbol{\omega}, \boldsymbol{\psi}; \mathbf{Y}) \sim N_m(\boldsymbol{\mu}, \mathbf{A}),$$

where

$$\begin{aligned} \boldsymbol{\mu} &= \left\{ (\boldsymbol{\Psi} \boldsymbol{\Psi}') \otimes (\mathbf{X} \mathbf{X}') + (\mathbf{D} \mathbf{R} \mathbf{D})^{-1} \right\}^{-1} \left\{ \left\{ (\boldsymbol{\Psi} \boldsymbol{\Psi}') \otimes (\mathbf{X} \mathbf{X}') \right\} \hat{\boldsymbol{\phi}}_M + (\mathbf{D} \mathbf{R} \mathbf{D})^{-1} \boldsymbol{\phi}_0^{(\gamma)} \right\} \\ \mathbf{A} &= \left\{ (\boldsymbol{\Psi} \boldsymbol{\Psi}') \otimes (\mathbf{X} \mathbf{X}') + (\mathbf{D} \mathbf{R} \mathbf{D})^{-1} \right\}^{-1} \end{aligned}$$

where $\boldsymbol{\phi}_0^{(\gamma)}$ is the prior mean of $\boldsymbol{\phi}$ and $\hat{\boldsymbol{\phi}}_M$ are the elements of the “stacked” matrix of MLE coefficients, that is $\text{vec}(\hat{\boldsymbol{\phi}}_M) = \text{vec}((\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{Y})$ or `phi_m_vec` in the program.

5. Draw $(\gamma_{(k)} | \psi_{(k)}, \phi_{(k)}, \omega_{(k)}; \mathbf{Y})$ from Bernoulli distribution

$$\gamma_i \sim \text{Bernoulli}(u_{i1}/u_{i1} + u_{i2})$$

where

$$u_{i1} = \frac{1}{\tau_{0i}} \exp\left(-\frac{\phi_i^2}{2\tau_{0i}^2}\right) p_i$$

$$u_{i2} = \frac{1}{\tau_{1i}} \exp\left(-\frac{\phi_i^2}{2\tau_{1i}^2}\right) (1 - p_i)$$

And that's it! Program should converge to these conditionals quickly (20,000 iterations) as the authors claim.