“Macroeconometrics - PS 3”

Valerio Pieroni

Vector Autoregression (VAR)

Consider the VAR(4)

$$Y_t = c + A_1 Y_{t-1} + A_2 Y_{t-2} + A_3 Y_{t-3} + A_4 Y_{t-4} + \epsilon_t,$$

where Y_t is a $n \times 1$ vector and A_p are $n \times n$ matrices.

- GDP: GDP growth rate.
- UNR: Unemployment rate.
- INF: Infaltion rate using GDP deflator.
- FFR: Federal funds rate.
- SPR: Spread $i_{10,t} - i_t$.

VAR Estimation

- We have same regressor in each of the $n = 5$ equations.
- Use OLS equation by equation.
- Use OLS on the stacked representation.

Consider the VAR(4)

$$Y_t = c + A_1 Y_{t-1} + A_2 Y_{t-2} + A_3 Y_{t-3} + A_4 Y_{t-4} + \epsilon_t,$$

where Y_t is a $n \times 1$ vector and A_p are $n \times n$ matrices.

$$Y_{T \times n} = X_{T \times np} A_{np \times n} + \varepsilon_{T \times n}$$

VAR Estimation

```
function [beta, residuals] = VAR(y,p,c)

% Function to estimate a VAR(p) with or without constant using OLS
% Inputs:    y = T x N matrix of endogeneous variables
%            p = VAR lag order
%            c = 1 if constant required
% Outputs:   beta = (Np+1 x N) matrix of estimated coefficients (Np x N) if
%                   no constant is included
%            residuals = (T-p x N) matrix of OLS residuals

[T, ~] = size(y);
yfinal = y(p+1:T,:);
if c == 1
    X = [ones(T-p,1), lagmakerMatrix(y,p)];
else
    X = lagmakerMatrix(y,p);
end

beta = (X'*X)\X'*yfinal;
residuals = yfinal - X*beta;

end
```

VAR Estimation

```matlab
function x = lagmakerMatrix(y,p)
% Function to create the matrix of regressors according to the
% SUR representaiton for a VAR
% Inputs:    y = T x N matrix of endogeneous variables
%            p = VAR lag order
% Outputs:   x = T-p x Np matrix of lagged dependent variables as regressors

[T, N] = size(y);
x = zeros(T-p, N*p);
counter = 0;
for i=1:p
    for j=1:N
        counter = counter + 1;
        x(:,counter) = y(p+1-i:T-i,j);
    end
end
end
```

Compute IRFs

- Check stationarity: eigenvalues of the companion matrix $|\lambda_i| < 1$.
- Then the process admits the multivariate Wold representation

$$Y_t = \nu + \Theta(L)\epsilon_t.$$

- Compute IRFs as $\Theta_0, \Theta_1, \Theta_2$, ... using I_n, F_n, F_n^2, ... (why?)
- F_n are the first $n \times n$ entries in the companion form $F_{np \times np}$.

Compute IRFs

```
function irfwold = woldirf(beta, c, p, horizon)
% Function to compute the matrices of the Wold representation of a
% stationary VAR(p)
% Inputs:   beta = (Np+1 x N) matrix of estimated coefficients (Np x N) if
%                       no constant is included
%           c = 1 if constant required
%           p = VAR lag order
%           horizon = how many of the Wold matrices (+1) will be computed
% Outputs:  irfwold = N x N x horizon + 1 array of Wold coefficient
%                         matrices

[BigA, N] = companionMatrix(beta,c,p);
irfwold = zeros(N,N,horizon + 1);
for h=1:horizon+1

    temp = BigA^(h-1);
    irfwold(:,:,h) = temp(1:N, 1:N);

end
end
```

Compute IRFs

Finally we compute bootstrapped 68% confidence intervals.

```


nboot1 = 1000;
nboot2 = 2000;
prc = 68;
cumulate = [];

% Function to compute bootstrapped Wold IRFs using the resampling method
% with Kilian correction
function [bootwold, upper, lower, boot_beta] = bootstrapWold_corrected()
function [bootwold, upper, lower, boot_beta] = bootstrapWold()
function [ynext] = bootstrapVAR()
```

Identification of MP Shocks

- Structural VAR: VAR + parameters restrictions.
- MP shock affects contemporaneously only the spread (partial identification).
- Order the variables as before.

$$Y = \begin{bmatrix} GDP \\ UNR \\ INF \\ FFR \\ SPR \end{bmatrix}, \qquad \Theta_0 = \begin{bmatrix} \theta_{11,0} & \theta_{12,0} & \theta_{13,0} & 0 & \theta_{15,0} \\ \theta_{21,0} & \theta_{22,0} & \theta_{23,0} & 0 & \theta_{25,0} \\ \theta_{31,0} & \theta_{32,0} & \theta_{33,0} & 0 & \theta_{35,0} \\ \theta_{41,0} & \theta_{42,0} & \theta_{43,0} & \theta_{44,0} & \theta_{45,0} \\ \theta_{51,0} & \theta_{52,0} & \theta_{53,0} & \theta_{54,0} & \theta_{55,0} \end{bmatrix}$$

Identification of MP Shocks

- Compute the variance covariance matrix $\Omega = (T - np - p)^{-1}\epsilon_t\epsilon_t'$.
- Choleski factorization $\Omega = PP'$.
- Identify the structural shocks w_t and compute IRFs

$$Y_t = \nu + \Theta(L)PP^{-1}\epsilon_t = \nu + \Theta(L)Pw_t$$

- By construction $E(w_t w_t') = I_n \Rightarrow$ structural shocks are orthogonal.

Identification of MP Shocks

```
% compute omega from OLS residuals
S=chol(omega,'lower'); % Cholesky factorization, lower triangular matrix

D=zeros(N,N,hor+1);
for i=1:hor+1
    D(:,:,i)=C(:,:,i)*S; % Cholesky wold respesentation
end

% Alternatively use the function choleskyIRF
DD = choleskyIRF(C, S);
```

Identification of MP Shocks

```
function [cholirf] = choleskyIRF(wold, S, scaling)
% Function to compute the point estimate of the IRF of a VAR identified
% Inputs:   wold    = (N x N x horizon +1) array of Wold IRFs
%           S       = N x N lower triangular matrix Cholesky factor
%           scaling = 2 x 1 vector where the first argument is the variable
%                     to be used for scaleing and the second is the shock size
% Outputs:  chol = N x N x horizon + 1 array of Cholesky identified IRFs

[N,~, horizon] = size(wold);
cholirf = zeros(N,N,horizon);
for h=1:horizon
    cholirf(:,:,h) = wold(:,:,h) * S;
end

if nargin > 2
    cholirf = cholirf ./ (cholirf(scaling(1),scaling(1),1) * 1/scaling(2));
end
end
```

Identification with Long Run Restrictions

- Identify MP shocks using Cholesky and long run restrictions.
- Compute IRFs to the structural MP shocks.

- The long-run level effects are given by $D := I_n + F_n + F_n^2 + ...$
- If the eigenvalues of F_n are such that $|\lambda_i| < 1$

$$D = (I_n - F_n)^{-1}.$$

- Using the Cholesky lower triangular factorization $D\Omega D' = SS'$.
- Recover the selection matrix $P = (I_n - F_n)S'$
- Structural shocks $w_t := P^{-1}\epsilon_t$.