Estimating VARs with MATLAB

Barcelona GSE. Empirical Time Series Methods for Macroeconomic Analysis.
Instructor: Luca Gambetti.
TA: Nicolò Maffei Faccioli.

1 Introduction:

In this lecture notes, we will cover how to estimate Vector Auto Regressive models (henceforth, VAR) with OLS using the SUR representation, how to compute and plot the impulse response functions, how to construct bootstrap confidence bands and how to perform out-of-sample forecasts.

2 Estimation of a $VAR(p)$:

Consider the following $VAR(p)$ process:

$$Y_t = A_1 Y_{t-1} + A_2 Y_{t-2} + ... + A_p Y_{t-p} + \epsilon_t \tag{1}$$

where $\epsilon_t \sim WN(0, \Omega)$. How do we estimate (1) by OLS? First of all, notice that any $VAR(p)$ can be rewritten in terms of a $VAR(1)$, using the companion form representation. How do we do this? Define $\mathbf{Y_t} = (Y_t, Y_{t-1}, ..., Y_{t-p+1})'$, $\mathbf{\epsilon_t} = (\epsilon_t, 0, ..., 0)'$ and construct the following np x np companion matrix:

$$\mathbf{A} = \begin{bmatrix} A_1 & A_2 & ... & ... & A_p \\ I_n & 0 & ... & ... & 0 \\ 0 & I_n & ... & ... & 0 \\ 0 & 0 & I_n & ... & 0 \\ . & . & . & ... & 0 \\ . & . & . & ... & 0 \\ . & . & . & ... & 0 \\ 0 & 0 & ... & I_n & 0 \end{bmatrix} \tag{2}$$

Then the process (1) can be rewritten as follows

$$\mathbf{Y_t} = \mathbf{A Y_{t-1}} + \mathbf{\epsilon_t} \tag{3}$$

Notice that, if there was a constant in (1), then the matrix $\mathbf{A}$ would be the same as (2), the only difference is that we have to construct a vector of constants and zeros $\mathbf{C} = (c_1, ..., c_n, 0, ..., 0)'$ such that (3) looks as follows:

$$\mathbf{Y_t} = \mathbf{C} + \mathbf{A Y_{t-1}} + \mathbf{\epsilon_t} \tag{4}$$

This companion form representation will be useful to compute the coefficients (in this case matrices of coefficients: $A^0, A^1, A^2, ..., A^j$) of the Wold representation of Y_t. Notice that the process represented in (1) can be stacked in an ever more compact form, abstracting from the subscript t. This will be useful for the OLS estimation of the matrices $A_1, ..., A_p$. Notice that we can rewrite the process in (1) as (SUR representation) follows:

$$\mathbf{Y} = \mathbf{X}\Pi + \mathbf{u} \tag{5}$$

where:
1) $\mathbf{Y} = (Y_{p+1}, ..., Y_T)$ is a $(T-p)$ x n matrix, with $Y_t = (Y_{1t}, ..., Y_{nt})$.
2) $\mathbf{X} = (Y_p, ..., Y_{T-p})$ is a $(T-p)$ x np matrix, with $X_t = (Y_{t-1}, ..., Y_{t-p})$. Clearly, if we include a constant, then $\mathbf{X}$ is a $(T-p)$ x $(np+1)$ matrix, since $X_t = (1, Y_{t-1}, ..., Y_{t-p})$.
3) $\mathbf{u} = (\epsilon_{p+1}, ..., \epsilon_T)$ is a $(T-p)$ x n matrix.
4) $\Pi = (A_1, ..., A_p)$ is a np x n matrix of coefficients, $(np+1)$ x n if we include the constant $C = (c_1, ..., c_n)'$, so that $\Pi = (C, A_1, ..., A_p)$.
Therefore, after constructing this matrices, the estimation of a $VAR(p)$ model by OLS reduces to the usual formula:

$$\hat{\Pi} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{Y} \tag{6}$$

Hence, what we have to do in MATLAB, is to construct the matrices $\mathbf{Y}$ and $\mathbf{X}$ and to compute the OLS estimator, obtained in the previous formula. Suppose that you are interested in estimating a $VAR(4)$ model, including $n = 2$ variables, namely the growth rate of real gross domestic product (henceforth, $grGDPC1$) and the spread between the 10 years government rate and the effective federal funds rate (henceforth, *spread*).
To estimate the $VAR(4)$ by OLS, we proceed as follows: first we rewrite the process in a SUR representation form, as in (5). I created a function, called SUR, that does the job:

```
function [Y,X,Y_initial]=SUR(data,p,c)
% Function that computes Y and X of the SUR representation
% Author: Nicolo' Maffei Faccioli
%
% Y = X*PI + e
%
% INPUTS:
% Data: T x n dataset
% p: number of lags
%
% OUTPUTS:
% Y: T x n matrix of the SUR representation
% X: T x (n*p) matrix of the SUR representation

Y=(data(p+1:end,:));

if c==1
    X =[ones(length(lagmatrix(data,1:p)),1) lagmatrix(data,1:p)];
else X=lagmatrix(data,1:p);
end

X(1:p,:)=[];

% Discarded observations:

Y_initial=data(1:p,:);

end
```

First, we create a matrix $\mathbf{Y} = (Y_{p+1}, ..., Y_T)$, where, in our example, $Y_t = (grGDPC1_t, spread_t)$, and we create a matrix $\mathbf{X}$, using the function of MATLAB *lagmatrix*, which gives us the lags, from 1 to p (above, $1:p$), of our variables of interest. Given this function, estimating the $VAR(4)$ process is as simple as performing a basic OLS, as in (6). Including all of this in just one function:

```
function [pi_hat,Y,X,Y_initial,Yfit,err]=VAR(data,p,c)

% Function to estimate VAR by OLS, using the companion form representation
% Author: Nicolo' Maffei Faccioli
% INPUTS: data = data at hand, p = # of lags, c=1 if constant, c=0 if not.
% OUTPUTS: pi_hat = estimated coefficients organized in a n*pxn matrix
% (including the constant if c==1, not including it if c==0) ; Y = matrix
% of dependent variables ; X = matrix of regressors (lags + constant if
% c=1) ; Y_initial = initial p elements discarded from data.


[Y,X,Y_initial]=SUR(data,p,c);
pi_hat=(X'*X)\X'*Y;
Yfit=X*pi_hat; %Fitted value of Y
err=Y-Yfit; %Residuals

end
```

3 Impulse Responses of the Wold Representation:

Recall the companion form representation of a general $VAR(p)$ process in (3). As you have seen in class, substituting backwards in the companion form, we obtain:

$$\mathbf{Y_t} = \mathbf{A^j}\mathbf{Y_{t-j}} + \mathbf{A^{j-1}}\epsilon_{\mathbf{t-j+1}} + ... + \mathbf{A}\epsilon_{\mathbf{t-1}} + \epsilon_{\mathbf{t}} \tag{7}$$

If the conditions for stationarity (i.e. if all the eigenvalues of $\mathbf{A}$ are smaller than one in absolute value), then the series $\sum_{j=0}^{\infty}\mathbf{A^j}$ converges and $\mathbf{Y_t}$ has a $VMA(\infty)$ representation in terms of the Wold shock $\epsilon_{\mathbf{t}}$, given by:

$$\mathbf{Y_t} = \sum_{j=0}^{\infty}\mathbf{A^j}\epsilon_{\mathbf{t-j}} = C(L)\epsilon_{\mathbf{t}} \tag{8}$$

where $C(L) = (C_0 + C_1L + C_2L^2 + ...)$ and $C_0 = \mathbf{A^0}, C_1 = \mathbf{A^1}, ..., C_j = \mathbf{A^j}$. Notice that the matrix of coefficients of the Wold representation C_j has the following interpretation:

$$\frac{\delta Y_{t+j}}{\delta \epsilon_t'} = C_{j\,(1:n,1:n)}$$

That is, the row i, column k element of $C_{j\,(1:n,1:n)}$ identifies the consequences of a unit increase in the kth variable's innovation at date t for the value of the ith variable at time $t+j$ holding all other innovation at all dates constant. Therefore, to compute the impulse responses, in our example, we can estimate the $VAR(4)$ process by OLS as in the previous section and construct, with our estimates, the matrix $\mathbf{A}$ as in (2) and compute the matrices C_j for $j = 0, 1, 2, ..., J$, where J is simply the horizon we choose. How can we do this in MATLAB? First we can construct the matrix $\mathbf{A}$ as follows:

```
BigA=[pi_hat(2:end,:)'; eye(n*p-n) zeros(n*p-n,n)]; % BigA companion form, ...
    npxnp matrix
```

First of all, notice that we include, in the companion matrix, the estimates in $\hat{\Pi}$ from the second row onwards, for each column. This because we are excluding the constant from the matrix **A**, as pointed out in (4) . Notice that, as the matrix has to be of dimension np x np and the matrix of estimates is a np x p, then the matrices of *ones* and *zeros* have to be, respectively, $n(p-1)$ x $n(p-1)$ and $n(p-1)$ x n. Once we created the companion form matrix A, we can compute the impulse responses of the Wold representation as follows:

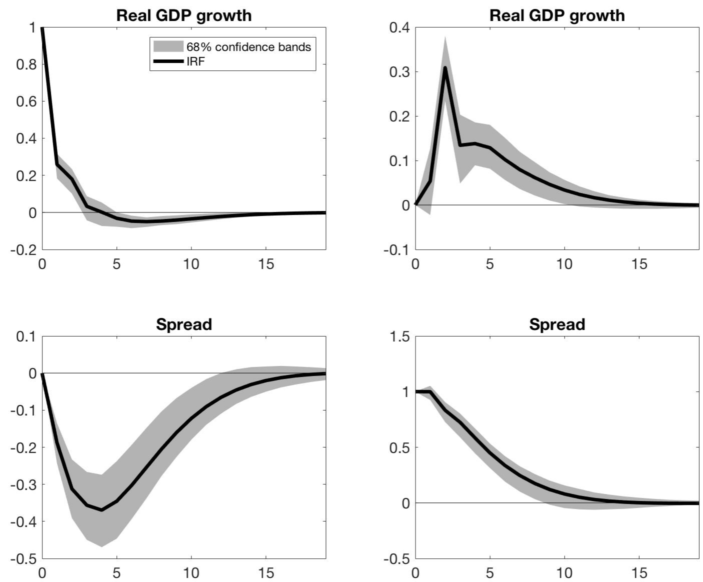
$$C_0 = A^0, C_1 = A^1, ..., C_j = A^j \rightarrow IRF = C_{j(1:n,1:n)}$$

We can perform a for loop that fills in a tensor with the impulse responses up to horizon J. In this case, I assumed $J = 19$.

```
%% Wold representation impulse responses:

C=zeros(n,n,20);

for j=1:20
    BigC=BigA^(j-1);
    C(:,:,j)=BigC(1:n,1:n); % Impulse response functions of the Wold ...
        representation
end

C_wold=reshape(permute(C,[3 2 1]),20,n*n,[]);

% Bootstrap:

hor=20;
iter=1000;
conf=68;

[HighC,LowC]=bootstrapVAR(finaldata,hor,c,iter,conf,p,n); % function that ...
    performs bootstrap

% Plot:

colorBNDS=[0.7 0.7 0.7];
VARnames={'Real GDP growth'; 'Spread'};

figure2=figure(2);

for k=1:n
    for j=1:n
    subplot(n,n,j+n*k-n)
    fill([0:hor-1 fliplr(0:hor-1)]' ,[HighC(:,j+n*k-n); ...
        flipud(LowC(:,j+n*k-n))],...
        colorBNDS,'EdgeColor','None'); hold on;
    plot(0:hor-1,C_wold(:,j+n*k-n),'LineWidth',3.5,'Color','k'); hold on;
    line(get(gca,'Xlim'),[0 0],'Color',[0 0 0],'LineStyle','-'); hold off;
    title(VARnames{k})
    legend({'68% confidence bands','IRF'},'FontSize',12)
    set(gca,'FontSize',15)
    xlim([0 hor-1]);
    end
end
```

Notice that C is a tensor of dimension $C(n, n, hor)$, i.e. it stores the n x n matrices $C_{j(1:n,1:n)}$ of coefficients for $j = 0, 1, 2, ..., J$. In the script above, we are simply telling MATLAB to transform our tensor in a hor x $n*n$ matrix, so that we can easily plot the impulse responses for each variable. For example, if we want to plot the impulse response of a shock in the spread on the growth rate of real GDP, we can simply plot $C_{wold}(:, 2)$. In Figure 1, I present the impulse response functions of $grGDPC1$ and *spread*, obtained including a constant and four lags of the dependent variables. You can play around with the code and check what happens if we don't include the constant, we include more lags or we include additional variables.

Figure 1: Impulse Response Functions



Top-left graph represents $\epsilon_1 \to grGDPC1$, top-right $\epsilon_2 \to grGDPC1$, bottom-left $\epsilon_1 \to spread$ and bottom-right $\epsilon_2 \to spread$.

How do we interpret this results? Let's consider the top-right graph, which represents the effects of a positive shock in the spread on the growth of real GDP. A 100 basis point increase in the spread is associated to almost a 30 basis points increase in real GDP growth two quarters ahead. Notice that, after 10 quarters, real GDP growth goes back to zero.

4 Bootstrap Confidence Bands:

How do we obtain the bands of the impulse responses presented in Figure 1? The bootstrap method consists on the following steps:
a) Step 1: Estimate a $VAR(p)$ with our sample and store the OLS estimates $\hat{\Pi}$ and the vector of residuals $\hat{u} = (\hat{u}_1, ..., \hat{u}_T)'$.
b) Step 2: Draw uniformly from $(\hat{u}_1, ..., \hat{u}_T)$ and set $\tilde{u}_1^{(1)}$ equal to the selected realization and construct:

$$Y_1^{(1)} = \hat{A}_1 Y_0 + \hat{A}_2 Y_{-1} + ... + \hat{A}_p Y_{-p+1} + \tilde{u}_1^{(1)}$$

c) Step 3: Taking a second draw, with replacement, from $(\hat{u}_1, ..., \hat{u}_T)$, generate $\tilde{u}_2^{(1)}$ and construct:

$$Y_2^{(1)} = \hat{A}_1 Y_1 + \hat{A}_2 Y_0 + ... + \hat{A}_p Y_{-p+2} + \tilde{u}_2^{(1)}$$

d) Step 4: Proceeding in this fashion, construct a sample of length T, $(Y_1^{(1)}, ..., Y_T^{(1)})$ and use this constructed sample to estimate a $VAR(p)$ and compute the impulse responses.
e) Step 5: Repeat steps 2 to 4 M times and store the M realizations of the impulse response functions, $C(L)^{(l)}$, $l = 1, ...M$. Then, take, for all the elements of the impulse response functions and for all the horizons, the αth and $(1-\alpha)$th percentiles to construct confidence bands.
I created a function, called *bootstrapVAR*, that does precisely these steps, with the possibility of including or excluding the constant. I didn't include the function in this PDF, as it is pretty long, but you can find it in the BOX folder. Notice that, in the previous steps, Y_1 and $\hat{u}_1$ actually refer to Y_{p+1} and $\hat{u}_{p+1}$, as $Y_0, Y_{-1}, Y_{-2}, ..., Y_{-p+1}$ are the initial discarded observations.

5 Forecasting:

How do we compute the j periods ahead forecasts of a $VAR(p)$ process? Recall the following $VAR(p)$ process:

$$Y_t = C + A_1 Y_{t-1} + A_2 Y_{t-2} + ... + A_p Y_{t-p} + \epsilon_t \tag{9}$$

Given our estimates of C and $A_1, ..., A_p$, obtained with OLS, with the methodology described above, we can estimate the j periods ahead forecasts as follows, recursively:

$$\hat{Y}_{t+1|t} = \hat{C} + \hat{A}_1 Y_t + \hat{A}_2 Y_{t-1} + ... + \hat{A}_p Y_{t-p+1}$$

$$\hat{Y}_{t+2|t} = \hat{C} + \hat{A}_1 Y_{t+1|t} + \hat{A}_2 Y_t + ... + \hat{A}_p Y_{t-p+2}$$

$$...$$

$$\hat{Y}_{t+j|t} = \hat{C} + \hat{A}_1 Y_{t+j-1|t} + \hat{A}_2 Y_{t+j-2|t} + ... + \hat{A}_p Y_{t-p+j|t}$$

Suppose that we are interested in computing an out-of-sample 1 period ahead forecast. How do we do this? First of all, we estimate the $VAR(p)$ with an initial sample T_0. Then, with the estimates obtained with the inital sample estimation, we can forecast $\hat{Y}_{T_0+1|T_0}$. Then, we update the sample with an additional observation, i.e. we define $T_0 = T_0 + 1$, estimate the $VAR(p)$ and make a new one period ahead forecast. We keep doing this until we reach the end of the sample. How is this performed in MATLAB?

```
% One period ahead:

% First of all, we have to estimate the parameters using an initial sample
% of T_0=123:

T_0=123;
ldiff=length(finaldata)-T_0;
forecasts1=zeros(ldiff,n);

% Notice that here I make a loop that repeats the steps until the end of
% the sample:

for k=1:ldiff

    [pi_hat,Y,X,Y_initial,Yfit,err]=VAR(finaldata(1:123+k-1,:),p,1);
    Ylast= reshape(flipud(Y(end-p+1:end,:))',1,[]);
    forecasts1(k,:)=pi_hat(1,:)+Ylast*pi_hat(2:end,:); % 1 period ahead

end
```

The results obtained are represented in Figure 2 and Figure 3:

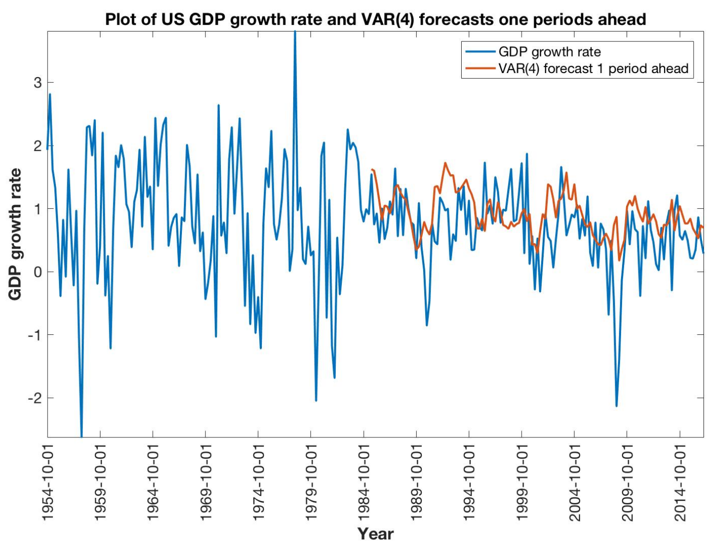


Figure 2: Actual & Forecasts - $grGDPC1$

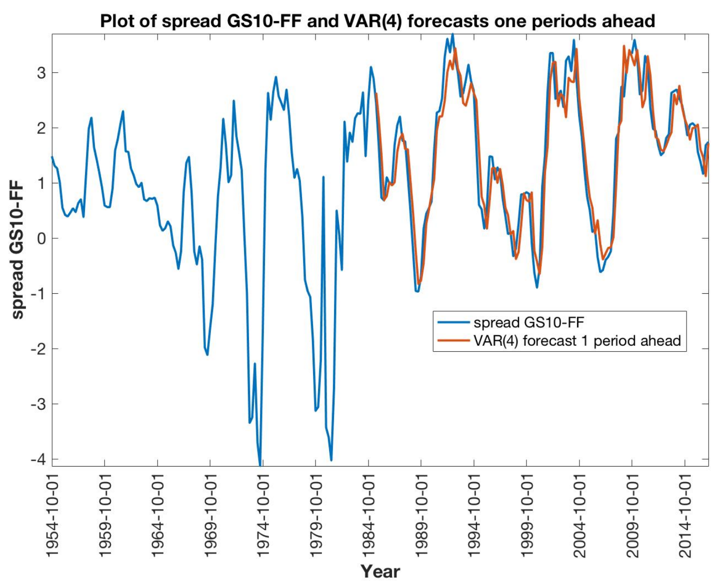


Figure 3: Actual & Forecasts - *spread*