Identify structural VAR

identify_structural_VAR.m

15 March 2014

Summary

Use a simple identification scheme based on Choleski decomposition to calculate a structural VAR from the estimated reduced-form VAR. Check the properites of the structural shocks, and run shock (impulse) response simulation.

Contents

1 Clear Workspace

```
clear;
close all;
clc;
%#ok<*NOPTS>
```

2 Load Estimated Reduced-Form VAR and Data

Load the estimated reduced-form VAR object and its data.

```
load estimate_simple_VAR.mat v vd;
```

3 Identify Structural VAR

Use simple Cholesky decomposition to identify a structural VAR; this is the default identification scheme in the functio `SVAR` [1] . This gives a SVAR with a lower triangular matrix, `B`, of instantaneous shock multipliers. Use the option `'ordering='` [2] to change the order of shocks; in that case, the matrix `B` is a permuted lower triagular matrix [3].

The function `SVAR` also returns a new database, `sd1` or `sd2`, where the shocks are recomputed according to the identification scheme:

$$
\begin{gathered}
y_t = A_1 y_{t-1} + A_2 y_{t-2} + B u_t \\
\mathrm{E}[u_t u_t'] = \mathrm{I} \\
B B' = \Omega
\end{gathered}
$$

Compare this form with the reduced-form VAR equation in `estimate_simple_VAR`.

```
[s1,sd1] = SVAR(v,vd); [1]

[s2,sd2] = SVAR(v,vd, ...
    'ordering=',{'yy','pp','r','mm'}); [2]

get(s1,'B')
get(s2,'B') [3]
```

```
ans =
    0.3066         0         0         0
   -0.0531    0.4491         0         0
    0.6116   -0.2517    1.8987         0
   -1.2372    0.2909   -0.5229    4.5498
```

```
ans =
    0.2913   -0.0213    0.0932         0
         0    0.4464   -0.0724         0
         0         0    2.0106         0
   -0.9905    0.2928   -0.9065    4.5498
```

4 Covariance of Structural Shocks

The identifying restrictions used to set up a structural VAR above included the assumption of uncorrelated structural shocks. Compute the sample covariance and correlation matrix of the reduced-form residuals (i.e. forecast errors from the reduced-form VAR, contained in the database `vd`), and those of the structural shocks: first manually [5] and then using the function `acf` [6].

```
xv = [vd.res_r, ... [4]
    vd.res_pp, ...
    vd.res_yy, ...
    vd.res_mm];
xv(:,:).' * xv(:,:) / length(xv) [5]
acf(xv,Inf,'demean=',false,'smallSample=',false) [6]

xs = [sd2.res_r, ... [7]
    sd2.res_pp, ...
    sd2.res_yy, ...
    sd2.res_mm];
xs(:,:).' * xs(:,:) / length(xs)
acf(xs,Inf,'demean=',false,'smallSample=',false)
```

```
ans =
    0.0940   -0.0163    0.1875   -0.3793
   -0.0163    0.2045   -0.1455    0.1963
    0.1875   -0.1455    4.0424   -1.8227
   -0.3793    0.1963   -1.8227   22.5898
ans =
    0.0940   -0.0163    0.1875   -0.3793
   -0.0163    0.2045   -0.1455    0.1963
    0.1875   -0.1455    4.0424   -1.8227
   -0.3793    0.1963   -1.8227   22.5898
ans =
    1.0000    0.0000   -0.0000   -0.0000
    0.0000    1.0000   -0.0000    0.0000
   -0.0000    0.0000    1.0000   -0.0000
   -0.0000         0   -0.0000    1.0000
ans =
    1.0000    0.0000    0.0000   -0.0000
```

```
    0.0000    1.0000   -0.0000   -0.0000
    0.0000   -0.0000    1.0000   -0.0000
   -0.0000   -0.0000   -0.0000    1.0000
```

5 Asymptotic ACF for endogenous variables

The asymptotic properties of the endogenous variables remain exactly the same in both the reduced-form VAR, `v`, and the structural VAR, `s`. Calculate and compare the asymptotical autocovariance (`CV`, `CS`) and autocorrelation (`RV`, `RS`) matrices up to second order for the VAR [8] and the SVAR [9].

The matrices are all Ny-by-Ny-K, where Ny is the number of variables, and K is the maximum order requested (i.e. 2) plus 1 (for the contemporaneous matrices).

Show the contemporaneous coveriances and correlations (i.e. the first pages in the `CV`, `RV`, `CS`, and `RS`). Verify that the matrices are identical for the VAR and the SVAR [10].

```
[CV,RV] = acf(v,'order=',2); [8]
[CS,RS] = acf(s2,'order=',2); [9]

size(CV)

CV(:,:,1)
CS(:,:,1)
RV(:,:,1)
RS(:,:,1)

maxabs(CV-CS) [10]
maxabs(RV-RS) [10]
```

```
ans =
     4     4     3
ans =
    1.6975    0.3064    0.0353   -4.4722
    0.3064    0.3395   -0.3028   -0.3304
    0.0353   -0.3028    5.6821   -3.0184
   -4.4722   -0.3304   -3.0184   36.6055
   Rows: r pp yy mm
Columns: r pp yy mm
ans =
    1.6975    0.3064    0.0353   -4.4722
    0.3064    0.3395   -0.3028   -0.3304
    0.0353   -0.3028    5.6821   -3.0184
   -4.4722   -0.3304   -3.0184   36.6055
   Rows: r pp yy mm
Columns: r pp yy mm
```

```
ans =
    1.0000    0.4036    0.0114   -0.5673
    0.4036    1.0000   -0.2180   -0.0937
    0.0114   -0.2180    1.0000   -0.2093
   -0.5673   -0.0937   -0.2093    1.0000
   Rows: r pp yy mm
Columns: r pp yy mm
ans =
    1.0000    0.4036    0.0114   -0.5673
    0.4036    1.0000   -0.2180   -0.0937
    0.0114   -0.2180    1.0000   -0.2093
   -0.5673   -0.0937   -0.2093    1.0000
   Rows: r pp yy mm
Columns: r pp yy mm
ans =
   2.4869e-14
ans =
   5.9397e-15
```

6 Simulate Shock Response Function

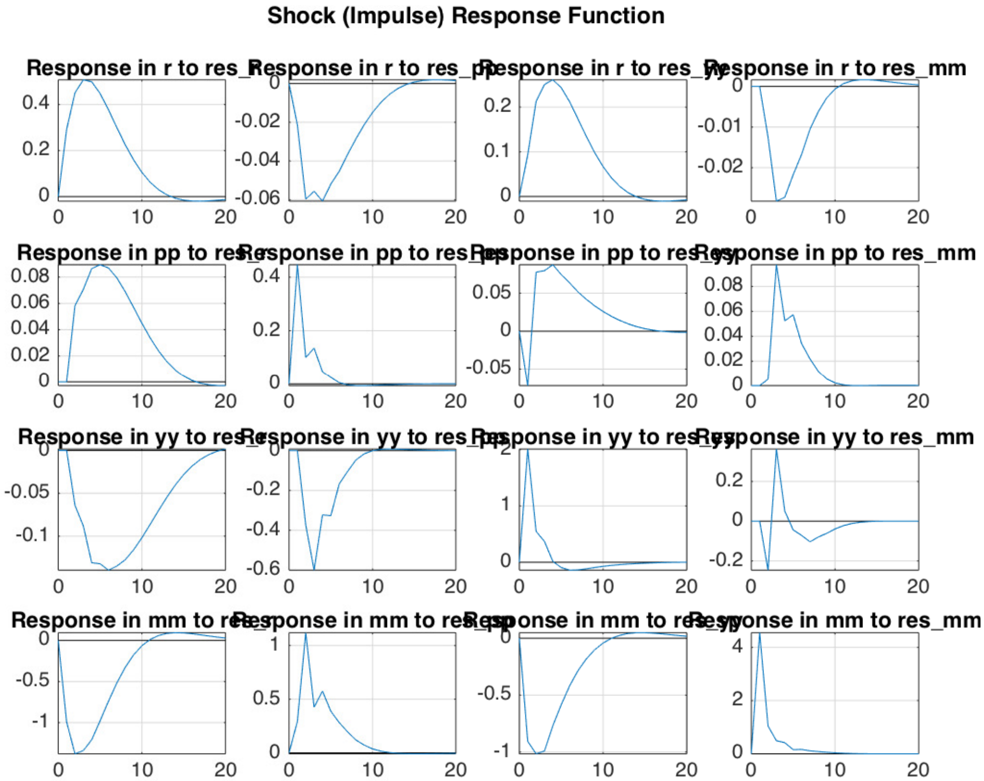
Run the function `srf` [11] to calculate shock (impulse) responses. The function returns two databases : `s` is a database with plain shock responses [12], `sc` is a database with cumulative responses [13]. The option `'presample='` is used to fill the output time series with zeros before the shock period; this is for reporting purposes only.

Each variables in the output databases has 4 columns, i.e. the responses to the 4 shock [14]. Plot the plain shock responses in a 4-by-4 figure.SSSSSS

```
[s,sc] = srf(s2,1:30,'presample=',true); [11]
s [12]
sc [13]

yNames = get(s2,'yNames')
eNames = get(s2,'eNames') [14]

figure();
count = 0;
for i = 1 : 4
    for j = 1 : 4
        % Response of the i-th variable to the j-th shock.
        count = count + 1;
        subplot(4,4,count);

        plot(0:20,s.(yNames{i}){:,j});
```

```
        axis tight;
        grid on;
        grfun.zeroline();
        title(['Response in ',yNames{i},' to ',eNames{j}], ...
            'interpreter','none');
    end
end

grfun.ftitle('Shock (Impulse) Response Function');
```

```
s =
         r: [32x4 tseries]
        pp: [32x4 tseries]
        yy: [32x4 tseries]
        mm: [32x4 tseries]
     res_r: [32x4 tseries]
    res_pp: [32x4 tseries]
    res_yy: [32x4 tseries]
    res_mm: [32x4 tseries]
sc =
         r: [32x4 tseries]
        pp: [32x4 tseries]
        yy: [32x4 tseries]
        mm: [32x4 tseries]
     res_r: [32x4 tseries]
    res_pp: [32x4 tseries]
    res_yy: [32x4 tseries]
    res_mm: [32x4 tseries]
yNames =
    'r'    'pp'    'yy'    'mm'
eNames =
    'res_r'    'res_pp'    'res_yy'    'res_mm'
```



7 Help on IRIS Functions Used in This File

Use either `help` to display help in the command window, or `idoc` to display help in an HTML browser window.

```
help SVAR
help SVAR/SVAR
help VAR/acf
help SVAR/srf
help grfun/ftitle
help grfun/zeroline
help utils/maxabs
```