Compare Second Moment Properties in Model and Data

compare_model_and_data.m

by Jaromir Benes

May 27, 2015

Summary

Compute and compare several second-moment properties of the estimated model and the data. Describe the data using an estimated VAR; this also allows to evaluate sampling uncertainty of the empirical estimates using bootstrap methods.

Contents

1 Clear Workspace . . . 2
2 Load Estimated Model Object, and Historical Database . . . 2
3 Estimate VAR and BVAR . . . 2
4 Compare Transition Matrices . . . 4
5 Compare Residuals . . . 5
6 Resample From Estimated VAR . . . 6
7 Compare ACF From Model and Data . . . 7
8 Compare Frequency Selective ACF . . . 8
9 Compare VAR and Model Spectra . . . 10
10 Help on IRIS Functions Used in This Files . . . 11

1 Clear Workspace

Clear workspace, close all graphics figures, clear command window, and check the IRIS version.

```
clear;
close all;
clc;
irisrequired 20140315;
```

2 Load Estimated Model Object, and Historical Database

Load the model object estimated in `estimate_params`, and the historical database created in `read_data`. Run `estimate_params` and `filter_hist_data` at least once before running this m-file.

```
load estimate_params.mat mest;
load read_data.mat d startHist endHist;
```

3 Estimate VAR and BVAR

Estimate an unrestricted 2nd-order VAR, and a 2nd-order Bayesian VAR with Litterman-type priors. First, create an empty VAR object [2] specifying the names of the endogenous variables. The names are identical to the names of measurement variables in the DSGE model [1]. Second, call the function `estimate` with an input database. For bayesian VARs, create prior dummy observations before running `estimate` [3].

```
ylist = get(mest,'yList'); [1]

p = 2;

v = VAR(ylist) %#ok<NOPTS> [2]
[v,vdata] = estimate(v,d,startHist:endHist,'order=',p);
v %#ok<NOPTS>

X = BVAR.litterman(0,sqrt(30),0) %#ok<NOPTS> [3]

bv = VAR(ylist) %#ok<NOPTS>
[bv,bvdata] = estimate(bv,d,startHist:endHist,'order=',p, ...
    'BVAR=',X,'stdize=',true);
bv %#ok<NOPTS>
```

```
v =
        empty VAR object
        variables: [4] 'Short' 'Infl' 'Growth' 'Wage'
        exogenous: [0]
        instruments: [0]
        groups: implicit
        comment: ''
        user data: empty
        export files: [0]

v =
        VAR(2) object: [1] parameterisation(s)
        variables: [4] 'Short' 'Infl' 'Growth' 'Wage'
        exogenous: [0]
        instruments: [0]
        groups: implicit
        comment: ''
        user data: empty
        export files: [0]

X =
  bvarobj with properties:

    name: 'litterman'
      y0: @litterman/y0
      y1: @litterman/y1
      k0: @litterman/k0
      g1: @litterman/g1
bv =
        empty VAR object
        variables: [4] 'Short' 'Infl' 'Growth' 'Wage'
        exogenous: [0]
        instruments: [0]
        groups: implicit
        comment: ''
        user data: empty
        export files: [0]

bv =
        VAR(2) object: [1] parameterisation(s)
        variables: [4] 'Short' 'Infl' 'Growth' 'Wage'
        exogenous: [0]
        instruments: [0]
        groups: implicit
        comment: ''
```

```
        user data: empty
        export files: [0]
```

4 Compare Transition Matrices

Get and print the transition matrices from the plain VAR and the BVAR objects. The transition matrices are Ny-by-Ny-by-P matrices, where Ny is the number of variables, and P is the order of the VAR.

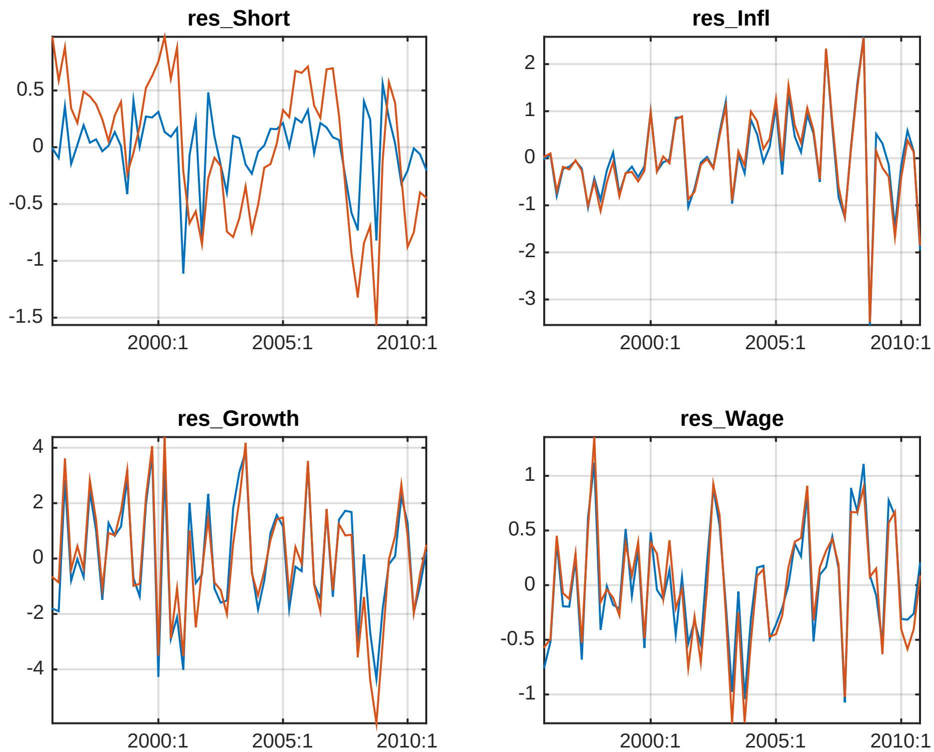
```
A = get(v,'A*');
BA = get(bv,'A*');

disp('Unrestricted VAR transition matrix');
A(:,:,1)
A(:,:,2)
disp('BVAR transition matrix');
BA(:,:,1)
BA(:,:,2)
```

```
Unrestricted VAR transition matrix
ans =
    1.5407   -0.0653    0.0334    0.0759
   -0.0501    0.3143    0.0641   -0.1158
    1.6851   -0.3895    0.0969   -1.0528
    0.2922   -0.0748   -0.0505    0.3510
ans =
   -0.5799    0.0513   -0.0048   -0.1275
    0.0950    0.2476    0.0344   -0.1642
   -0.8984   -0.3549    0.1728   -0.8460
   -0.0739    0.1175    0.0473   -0.0704
BVAR transition matrix
ans =
    0.3343    0.0721    0.1029    0.2844
    0.0191    0.2178    0.0421   -0.0576
    0.2147   -0.1388    0.1927   -0.4495
    0.0882    0.0086   -0.0114    0.2253
ans =
    0.2589    0.0683    0.0710    0.1545
    0.0089    0.1812    0.0341   -0.1151
    0.1136   -0.2504    0.1935   -0.4159
    0.0778    0.0724    0.0336    0.0733
```

5 Compare Residuals

Plot and compare the estimated residuals from the plain VAR and the BVAR. Use the output data, vdata and bvdata returned from estimate. These databases containing both the endogenous variables and estimated residuals. By default, the residuals are named res_XXX where XXX is the name of the respective variable,

```
elist = get(v,'eList');

figure();
for i = 1 : 4
    name = elist{i};
    subplot(2,2,i);
    plot(vdata.(name));
    hold all;
    plot(bvdata.(name));
    title(name,'interpreter','none');
    grid on;
    axis tight;
end
grfun.bottomlegend('Unrestricted VAR(2)','BVAR(2)');
```



6 Resample From Estimated VAR

Use a wild bootstrap to generate `N=500` of VARs; a wild bootstrap is robust to potential heteroscedasticity of residuals. Note that some of the resampled VAR parameterisations may be explosive, and remove them from the VAR object.

```
N = 1000;
Y = resample(v,vdata,Inf,N,'wild=',true,'progress=',true);
size(Y)

Nv = VAR(ylist);
Nv = estimate(Nv,Y,Inf,'order=',p);

inx = isstationary(Nv);
sum(inx)
Nv = Nv(inx);
```

```
[--IRIS VAR.resample progress------------]
[****************************************]
ans =
     1     1
ans =
   999
```

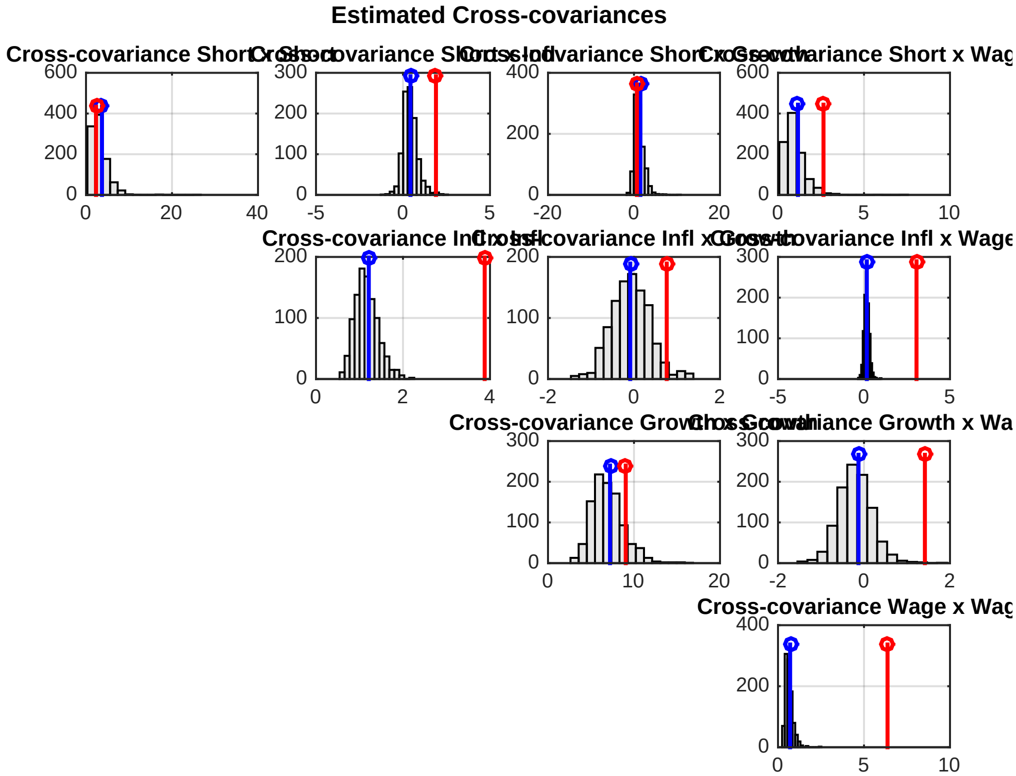
7 Compare ACF From Model and Data

Compute and plot the autocovariance/autocorrelation functions (ACF) for the estimated VAR, the resampled VARs, and the model.

Comments on the code below:

4 `helper_plot_acf` is a helper function (for plotting ACFs) created just for this exercise (it is not part of the IRIS toolbox).

```
[Cv,Rv] = acf(v,'order=',1);
[CNv,RNv] = acf(Nv,'order=',1);
[Cm,Rm] = acf(mest,'order=',1,'select=',ylist);

figure();
for i = 1 : length(ylist)
    for j = i : length(ylist)
        subplot(4,4,(i-1)*4+j);
        helper_plot_acf(CNv(i,j,1,:),Cv(i,j,1),Cm(i,j,1)); 4
        title(sprintf('Cross-covariance %s x %s',ylist{i},ylist{j}));
    end
end

grfun.bottomlegend( ...
    'VAR: Bootstrap','VAR: Point Estimate','Model: Asymptotic');
grfun.ftitle('Estimated Cross-covariances');
```



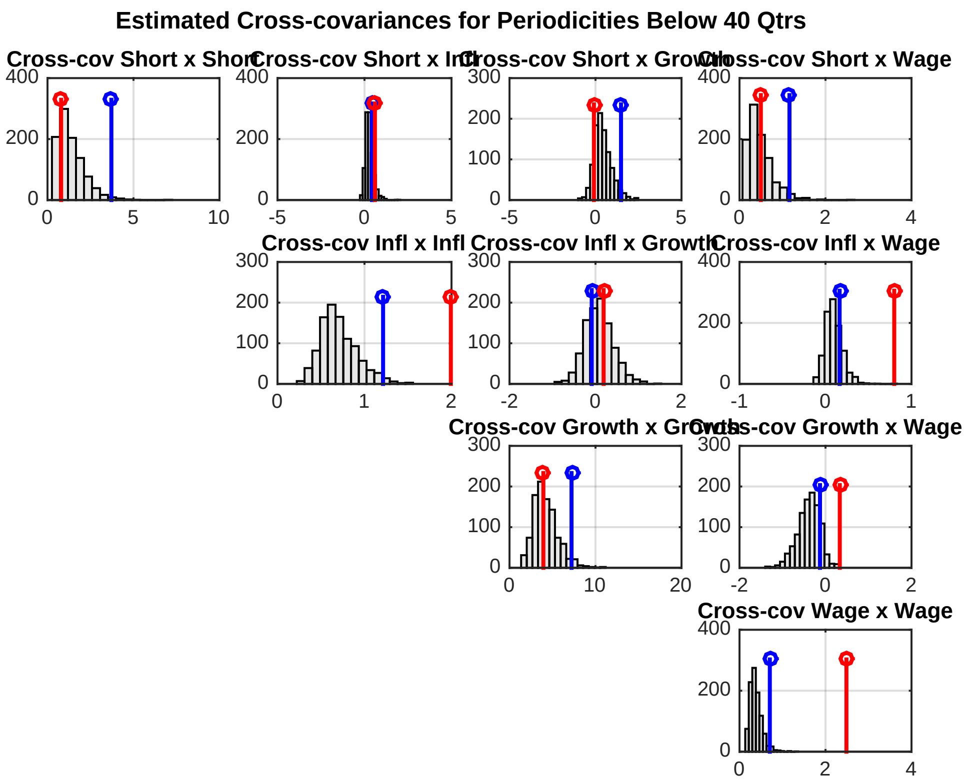
8 Compare Frequency Selective ACF

Use the 'filter=' option to compute the ACF (both from the strucural model and the VAR) that corresponds to cyclical fluctuations with periodicity between 4 and 40 quarters (1 to 10 years).

```
[Cv1,Rv1] = acf(v,'order=',1,'filter=','per <= 40 & per > 4');
[Cv2,Rv2] = acf(v,'order=',1,'filter=','per > 40');
[Cv3,Rv3] = acf(v,'order=',1,'filter=','per <= 4');

maxabs(Cv1+Cv2+Cv3 - Cv)

[CNv1,RNv1] = acf(Nv,'filter=','per <= 40 & per > 4','progress=',true);

[Cm1,Rm1] = acf(mest,'filter=','per <= 40 & per > 4','select=',ylist);

figure();
for i = 1 : length(ylist)
    for j = i : length(ylist)
```

```
        subplot(4,4,(i-1)*4+j);
        helper_plot_acf(CNv1(i,j,1,:),Cv(i,j,1),Cm1(i,j,1));
        title(sprintf('Cross-cov %s x %s',ylist{i},ylist{j}));
    end
end

grfun.bottomlegend( ...
    'VAR: Bootstrap','VAR: Point Estimate','Model: Asymptotic');
grfun.ftitle( ...
    'Estimated Cross-covariances for Periodicities Below 40 Qtrs');
```

```
ans =
   9.1038e-14
[--IRIS VAR.acf progress-----------------]
[****************************************]
```



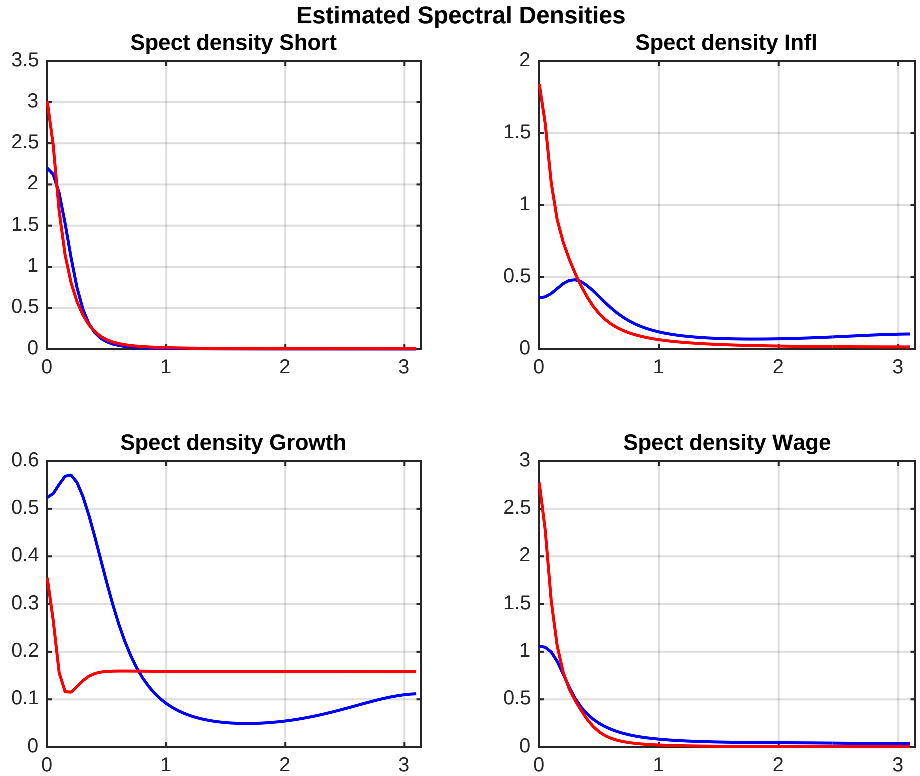
9 Compare VAR and Model Spectra

Compute and plot the power spectra and spectral densities for the estimated VAR and for the model.

Comments on the code below:

5 `helper_plot_xsf` is a helper function (for plotting the spectral densities) created just for this exercise in this directory (it is not part of the IRIS toolbox); it can be opened and viewed in the Matlab editor.

```
freq = 0 : 0.05 : pi;
[Pv,Sv] = xsf(v,freq);
[Pm,Sm] = xsf(mest,freq,'select=',ylist);

figure();

for i = 1 : length(ylist)
    subplot(2,2,i);
    helper_plot_xsf(freq,Sv(i,i,:),Sm(i,i,:)); 5
    title(sprintf('Spect density %s',ylist{i}));
end

grfun.bottomlegend('VAR: Point Estimate','Model: Asymptotic');
grfun.ftitle('Estimated Spectral Densities');
```



10 Help on IRIS Functions Used in This Files

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

```
help VAR
help VAR/estimate
help VAR/get
help VAR/isstationary
help VAR/resample
help VAR/subsasgn
help VAR/acf
help VAR/xsf
help model/acf
help model/xsf
help grfun/bottomlegend
help grfun/ftitle
```