

Contents

- runVARHFI.m
- VAR order.
- Loading data from data folder.
- Creating database containing monthly data for VAR.
- Prepare VAR.
- Construct dummy observations for BVAR.
- Estimate a p-th order BVAR.
- Conditional forecasts (unbalanced panel).
- Converting data to quarterly
- Adding 1-step-ahead BVAR predictions to historical data
- and calibrating std deviations of prediction errors
- based on estimated uncertainty
- Making 2 parameterizations
- Removing the influence of HFIs in first parameterization
- Running Kalman filter
- Plotting the historical data and models with
- and without HFIs

runVARHFI.m

```
3: close all;  
4: clear all;  
5:  
6: readDefinitions;
```

```
9: m = loadstruct('GPM_HFI.mat');  
10: h = loadstruct('data.mat');
```

VAR order.

```
13: p = 4;
```

Loading data from data folder.

```
16: D = load('data\Vintage2008M01_1.mat');
```

Creating database containing monthly data for VAR.

```

19: d = struct();
20: d.GROWTH_ = D.GDPH*400;
21: d.PIE_ = D.PCU*400;
22: d.RS_ = D.FFED;
23: d.UNR_ = D.LR;
24: d.BLT_ = 1/4*(D.FTCRE + D.FTCIL + D.FTCIS + D.FTCNMH);
25: d.IP = D.IP;

```

Prepare VAR.

```

29: % List of variables.
30: list = fieldnames(d);
31: ny = length(list);

```

Construct dummy observations for BVAR.

```

36: % Estimate std devs of VAR variables.
37: x = db2tseries(d,list);
38: range = get(x,'nanstart') : get(x,'nanend');
39: nper = length(range);
40: stdevs = std(x(range));
41: % Create Litterman's priors.
42: % The BVAR.litterman(rho,mu*std,lambda,ny,p) has the following input arguments:
43: % * rho = 0 is a white noise prior, rho = 1 is a random walk prior,
44: % * mu is the weight on the prior (can be a vector of numbers in which case
45: % each variable gets different weight).
46: % * lambda between 0 and Inf: priors on the k-th lag coefficients get
47: % k^lambda times bigger weights. If lambda = 0, all lags are treated
48: % equally. The higher the lambda, the more the coefficients are pulled
49: % towards the prior (zero in this case).
50: dummyobs = BVAR.litterman(1,1*stdevs,1,ny,p);

```

Estimate a p-th order BVAR.

```

54: w1 = VAR();
55: [w1,data2] = estimate(w1,d,list,range,'order',p,'bvar',dummyobs);
56:
57: fcastrange = range(end) + (1:36);

```

Conditional forecasts (unbalanced panel).

```
60: cf1 = forecast(w1,d,fcastrange,d);
```

Converting data to quarterly

```
63: dq = dbfun(@(x) convert(x,'q'),d);
64: dp = dbfun(@(x) convert(x,'q'),cf1.mean);
65: ds = dbfun(@(x) convert(x,'q'),cf1.std);
```

Adding 1-step-ahead BVAR predictions to historical data and calibrating std deviations of prediction errors based on estimated uncertainty

```
70: for i = 1 : length(list)-1;
71:     nn = strcat(list{i},'1_');
72:     nn1 = strcat('std_RES_',nn);
73:     last = get(dq.(list{i}),'end');
74:     dq.(nn) = tseries();
75:     dq.(nn)(last+1) = dp.(list{i})(last+1);
76:     m.(nn1) = ds.(list{i})(last+1);
77: end;
```

Making 2 parameterizations

```
80: m = set(m,'nalt',2);
```

Removing the influence of HFIs in first parameterization

```
83: m(1).std_RES_GROWTH_1_ = Inf;
84: m(1).std_RES_PIE_1_ = Inf;
85: m(1).std_RES_RS_1_ = Inf;
86: m(1).std_RES_UNR_1_ = Inf;
87: m(1).std_RES_BLT_1_ = Inf;
```

Running Kalman filter

```
90: [m,smooth] = filter(m,dq,starthist:endhist+horizon);
```

Plotting the historical data and models with and without HFIs

```

95: list = {'GROWTH','PIE','RS','UNR'};
96: for i = 1 : length(list);
97:     subplot(2,2,i);
98:     plot(endhist-12:endhist,[h.(strcat(list{i},'_'')) smooth.mean.(list{i})]);
99:     title(list{i});
100:    if i == 1;
101:        legend({'Data', 'No HFI', 'HFI'},'Location','SouthWest');
102:    end;
103:    highlight(qq(2008,1));
104: end;

```

