More on Kalman Filter

more_on_kalman_filter.m

by Jaromir Benes

May 27, 2015

Summary

Run more advanced Kalman filter exercises. Split the data sample into two sub-samples, and pass information from one to the other. Run the filter with time-varying std deviations of some shocks. Evaluate the likelihood function and the contributions of individual time periods to the overall likelihood.

Contents

1 Clear Workspace

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

```
clear;
close all;
clc;
irisrequired 20140401;
%#ok<*NOPTS>
```

2 Load the estimated model object and the historical database

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

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

3 Split the Kalman filter into sub-samples

With the range split into two or more sub-samples, and the Kalman filter-smoother executed successively on each of them (using the most recent result as the initial condition for the next run), the smoothed data estimates will differ from those obtained previously (running the filter once on the whole range). This is because the individual runs of the filter of data will be based on different information sets.

The only exception is, obviously, the last sub-sample, which is by construction based on information from the entire range 1..T, and hence identical to the information set when the filter is run just once on the entire range.

When running the Kalman filter on the last sub-sambple, [2], the output database from the previous run, `f1`, is used to set up initical condition for the filter (instead of the default asymptotic distribution). This is allowed by the fact that the database `f1` contains both the point estimates and the MSE matrices [1].

```
[~,f0,v0,~,pe0] = filter(mest,d,startHist:endHist+10, ...
    'relative=',false);

N = 15;

```

```
[~,f1,v1,~,pe1] = filter(mest,d,startHist:endHist-N, ...
    'relative=',false);

f1 [1]

[~,f2,v2,~,pe2] = filter(mest,d,endHist-N+1:endHist, ...
    'relative=',false,'initcond=',f1); [2]
```

```
f1 =
    mean: [1x1 struct]
     std: [1x1 struct]
     mse: [49x13x13 tseries]
```

Print differences between the smoothed data.

```
disp('Smoothed estimates differ for the first sub-sample');
dbfun(@maxabs,f0.mean,f1.mean)
dbfun(@maxabs,pe0,pe1)

disp('Smoothed estimates are identical for the last sub-sample');
dbfun(@maxabs,f0.mean,f2.mean)
dbfun(@maxabs,pe0,pe1)
```

```
Smoothed estimates differ for the first sub-sample
ans =
      Short: 0
       Infl: 0
     Growth: 0
       Wage: 0
          Y: 0.0026
          N: 0.0018
          W: 0.0040
          Q: 0.0037
          H: 0.0030
          A: 0.0020
          P: 0.0016
          R: 2.8923e-06
         Pk: 0.0074
         Rk: 1.6092e-04
     Lambda: 0.0025
         dP: 1.2129e-04
        d4P: 5.0502e-04
         dW: 1.0965e-04
        RMC: 0.0042
         Mp: 0
```

```
         Mw: 0
         Ey: 4.0660e-04
         Ep: 2.7381e-04
         Ea: 7.2526e-05
         Er: 1.1497e-04
         Ew: 8.1722e-04
      alpha: 0
       beta: 0
      gamma: 0
      delta: 0
          k: 0
         pi: 0
        eta: 0
        psi: 0
        chi: 0
        xiw: 0
        xip: 0
       rhoa: 0
       rhor: 0
     kappap: 0
     kappan: 0
     Short_: 0
      Infl_: 0
    Growth_: 0
      Wage_: 0
ans =
     Short: 0
      Infl: 0
    Growth: 0
      Wage: 0
Smoothed estimates are identical for the last sub-sample
ans =
      Short: 0
       Infl: 0
     Growth: 0
       Wage: 0
          Y: 1.1102e-15
          N: 5.5511e-16
          W: 1.3323e-15
          Q: 6.6613e-16
          H: 1.5543e-15
          A: 6.6613e-16
          P: 8.8818e-16
          R: 2.2204e-16
         Pk: 1.7764e-15
         Rk: 9.7145e-17
```

```
      Lambda: 6.1062e-16
          dP: 0
         d4P: 6.6613e-16
          dW: 2.2204e-16
         RMC: 4.4409e-16
          Mp: 0
          Mw: 0
          Ey: 1.5092e-16
          Ep: 1.4919e-16
          Ea: 7.6111e-17
          Er: 5.0307e-17
          Ew: 1.7326e-16
       alpha: 0
        beta: 0
       gamma: 0
       delta: 0
           k: 0
          pi: 0
         eta: 0
         psi: 0
         chi: 0
         xiw: 0
         xip: 0
        rhoa: 0
        rhor: 0
      kappap: 0
      kappan: 0
      Short_: 0
       Infl_: 0
     Growth_: 0
       Wage_: 0
ans =
     Short: 0
      Infl: 0
    Growth: 0
      Wage: 0
```

4 Run Kalman Filter with Time Varying Std Devs of Some Shocks

Use the option `'vary='` to temporarily change some of the std deviations (or also cross-correlations) within the filtered sample. The estimates of unobservables and shocks will obviously change: Compare the estimated `Ep` shocks from the previous Kalman filter (with fixed std deviations) and the Kalman filter with time-varying `std_Ep` 3.

```
j = struct();
j.std_Ep = tseries();
j.std_Ep(endHist-9:endHist-5) = linspace(0.00,0.4,5);

[~,f1] = filter(mest,d,startHist:endHist,'vary=',j);

[j.std_Ep,f1.mean.Ep,f0.mean.Ep]
```

```
ans =
        tseries object: 73-by-3
        1995Q2:  NaN  -0.001157487  -0.001196067
        1995Q3:  NaN  -0.002590165 -0.0009767161
        1995Q4:  NaN -0.0009209099  0.0005911832
        1996Q1:  NaN  -0.001203951  0.0002191625
        1996Q2:  NaN  -0.003722911   -0.00237482
        1996Q3:  NaN -0.0002861726   0.001000267
        1996Q4:  NaN  -0.001900961   -0.00066432
        1997Q1:  NaN -0.0008219273  0.0003749682
        1997Q2:  NaN  -0.001888046 -0.0007225996
        1997Q3:  NaN  -0.003426181  -0.002285464
        1997Q4:  NaN -0.0007973953  0.0003239697
        1998Q1:  NaN   -0.00431276  -0.003206483
        1998Q2:  NaN -0.0009048461  0.0001897054
        1998Q3:  NaN -0.0003599312  0.0007255343
        1998Q4:  NaN  -0.003075113  -0.001996665
        1999Q1:  NaN -8.654346e-05  0.0009865065
        1999Q2:  NaN  -0.001562549 -0.0004936259
        1999Q3:  NaN  -0.002426179  -0.001360379
        1999Q4:  NaN -0.0008936189  0.0001698536
        2000Q1:  NaN   0.003011416   0.004073206
        2000Q2:  NaN  -0.003600386  -0.002539746
        2000Q3:  NaN  0.0003680103   0.001427957
        2000Q4:  NaN  -0.001476624   -0.00041696
        2001Q1:  NaN   0.001857329   0.002917102
        2001Q2:  NaN -4.249273e-05   0.001017789
        2001Q3:  NaN  -0.004300725  -0.003239501
        2001Q4:  NaN  -0.001348049 -0.0002853849
        2002Q1:  NaN   -4.10024e-05   0.001023694
        2002Q2:  NaN  6.980887e-05   0.001137266
        2002Q3:  NaN  -0.001063959  7.173713e-06
        2002Q4:  NaN  0.0005949262   0.001670892
        2003Q1:  NaN  0.0008644262   0.001946706
        2003Q2:  NaN  -0.005435136  -0.004344641
        2003Q3:  NaN   0.001448608   0.002549764
        2003Q4:  NaN  -0.001156774 -4.180462e-05
```

```
      2004Q1:  NaN   0.002780643   0.003913488
      2004Q2:  NaN -0.0003734721  0.0007824826
      2004Q3:  NaN  -0.001849481 -0.0006636789
      2004Q4:  NaN -0.0003791976  0.0008451136
      2005Q1:  NaN   0.001949039   0.003222973
      2005Q2:  NaN  -0.002871675  -0.001533901
      2005Q3:  NaN   0.003881292   0.005301027
      2005Q4:  NaN  -0.001889049 -0.0003643783
      2006Q1:  NaN  -0.001222535  0.0004359943
      2006Q2:  NaN   0.001262913   0.003091348
      2006Q3:  NaN  -0.001509774  0.0005328667
      2006Q4:  NaN  -0.004214108  -0.001903936
      2007Q1:  NaN   0.006224942    0.00886485
      2007Q2:  NaN  -0.004118933  -0.001080369
      2007Q3:  NaN  -0.005108801  -0.001602072
      2007Q4:  NaN   -0.00606329  -0.002031908
      2008Q1:  NaN  -0.001233677   0.003338597
      2008Q2:  NaN -0.0008340396   0.004203497
      2008Q3:    0             0    0.00524072
      2008Q4:  0.1   -0.01983928   -0.01478495
      2009Q1:  0.2   0.001312445   0.005910009
      2009Q2:  0.3   -0.00626424  -0.002208381
      2009Q3:  0.4  -0.003039358  0.0004822602
      2009Q4:  NaN  -0.007102349  -0.004065646
      2010Q1:  NaN -0.0008566004   0.001759127
      2010Q2:  NaN -0.0006241695    0.00163414
      2010Q3:  NaN   -0.00214986 -0.0001933592
      2010Q4:  NaN  -0.007734862  -0.006036008
      2011Q1:  NaN           NaN             0
      2011Q2:  NaN           NaN             0
      2011Q3:  NaN           NaN             0
      2011Q4:  NaN           NaN             0
      2012Q1:  NaN           NaN             0
      2012Q2:  NaN           NaN             0
      2012Q3:  NaN           NaN             0
      2012Q4:  NaN           NaN             0
      2013Q1:  NaN           NaN             0
      2013Q2:  NaN           NaN             0
  ''    'Cost Push Shock'    'Cost Push Shock'
      user data: empty
      export files: [0]
```

5 Evaluate Likelihood Function and Contributions of Individual Time Periods

Run the function `loglik` to evaluate the likelihood function. This function calls the very same Kalman filter as the function `filter`. The first output argument returned by `loglik` is minus the logarithm of the likelihood function; this value is also used as a criterion to be minimized (which means maximizing likelihood) within the function `estimate`.

Set the option `'objDecomp='` to `true` $\boxed{4}$ to obtain not only the overall likelihood, but also the contributions of individual time periods. They are stowed in a column vector with the overall likelihood at the top; the length of the vector is therefore nPer+1 $\boxed{5}$ where nPer is the number of periods over which the filter is run.

```
range = startHist:endHist+10;
nPer = length(range)

mll = loglik(mest,d,range,'relative=',false, ...
    'objDecomp=',true); 4

size(mll) 5
```

```
nPer =
    73
ans =
    74     1
```

Because there were no observations available in the input database `d` in the last 10 periods of the filter range, i.e. `endHist+1:endHist+10`, the contributions of these last 10 periods are zero.

```
mll
```

```
mll =
  352.9616
    9.9502
    5.0392
    4.4839
    3.6867
    6.1651
    4.3730
    3.9290
    3.9484
    4.9862
    5.5249
    4.5445
    5.3864
```

```
    3.8457
    4.2385
    6.1810
    4.0254
    3.6727
    4.3557
    6.0551
    5.3304
    6.4054
    4.7708
    4.3596
    8.0966
    6.5429
    6.4398
    6.2083
    4.5690
    3.7561
    4.4403
    5.2425
    5.6449
    5.4568
    6.0201
    5.2686
    5.5156
    4.3413
    3.6576
    4.0793
    4.4000
    3.7106
    5.1213
    3.7563
    5.0647
    4.1894
    4.9355
    4.4621
    7.4583
    3.8507
    3.8373
    6.0690
    9.7745
    6.4634
    8.8160
   21.1469
   11.6320
    7.8564
    6.4835
```

```
    4.9257
    4.1563
    4.2287
    3.9868
    6.0988
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
```

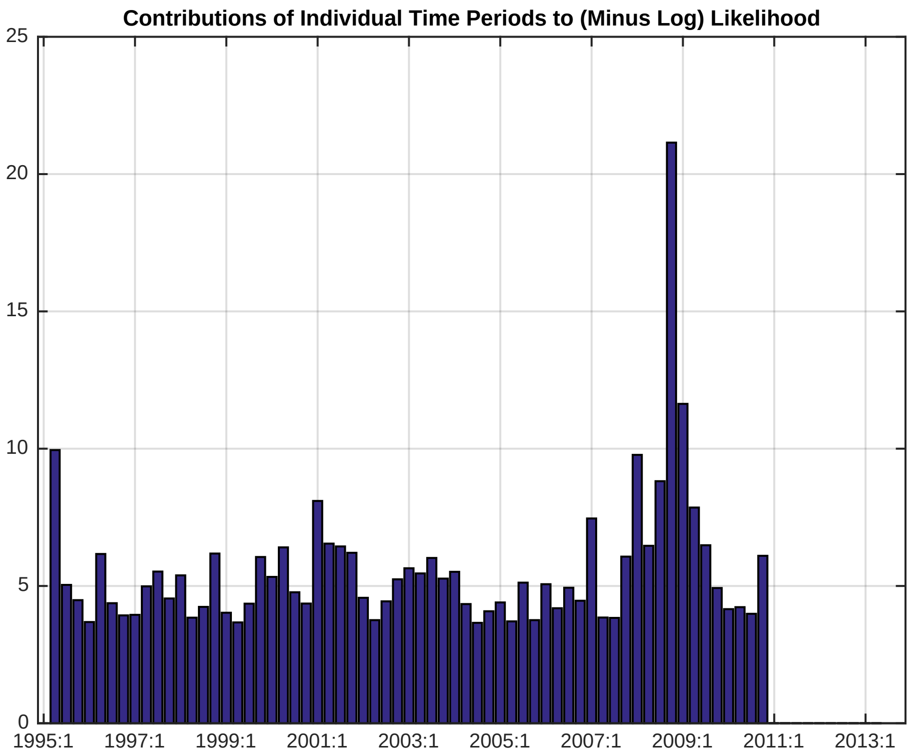
Adding up the individual contributions reproduces, of course, the overall likelihood. The following two numbers are the same (up to rounding errors):

```
mll(1)
sum(mll(2:end))
```

```
ans =
  352.9616
ans =
  352.9616
```

Visualize the contributions by converting them to a tseries object, and plotting as a bar graph. Large bars denote periods where the model performed poorly (rememeber, this is MINUS the log likelihood, ie. the larger the number the smaller the actual likelihood). Again, the last 10 periods are zeros because no observations were available in the input database in those.

```
x = tseries(range,mll(2:end));
bar(x);
grid on;
title('Contributions of Individual Time Periods to (Minus Log) Likelihood');
```



6 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 model/filter
help model/loglik
help dbase/dbfun
help tseries/tseries
help tseries/bar
help maxabs
```