Description of the State Space/ Kalman Filter Toolbox

Author: Marcelo Perlin / (PhD Student) / ICMA – Reading University
Email: marceloperlin@gmail.com / m.perlin@icmacentre.ac.uk

This toolbox was designed to simulate and fit linear state space models with the kalman filter approach. The main literature I used for this particular package is Kim and Nelson (1999) and Hamilton (1994). Getting our hands dirty, a generic state space model with stochastic coefficients only is given by:

$$y_t = \beta_t x_t + \varepsilon_t \qquad \varepsilon_t \sim N(0,R)$$
$$\beta_t = \mu + F\beta_{t-1} + \nu_t \qquad \nu_t \sim N(0,Q)$$

Where *nc* is the number of independent variables and, in matrix notation, β_t is a column vector of size [*1, nc*], x_t and μ are [*nc,1*]. The *F* symbol is a matrix of size [*nc,nc*] addressing the autoregressive process at β. The main issue in this particular model is that the coefficients for x_t are stochastic, that is, are driven by probabilities. The kalman filter is the most popular approach at estimating this model. It is based on a set of iterations in order to predict and update the values at β_t equation and in the covariance matrix of β_t (*P*). The actual steps of the algorithm are not going to be given here. Please check the reference for that.

Some comments about the fitting code:

- So far it doesn't handle autoregressive processes between series of the beta matrix. This means that, at the F matrix, the estimated coefficients are in the diagonal only (all non diagonal elements are zero, no spill over effect).
- The model is estimated by Gaussian maximum likelihood with the function fminsearch. I also played around with fminunc(), but there was no improvement over fminsearch when it comes to robustness and speed.
- So far the code doesn't handle state space models with mixture of non stochastic and stochastic coefficients, that is, when you want some variables to have stochastic coefficients and others not in the same model.

For the Example_Script_1_kalmanFit.m found at the zip file, this is the model that is being estimated (in non matrix notation):

$$y_t = \beta_{1t} x_{1t} + \varepsilon_t \qquad \varepsilon_t \sim N(0, R)$$
$$\beta_{1t} = \mu_1 + F_1 \beta_{1t-1} + v_{1t} \qquad v_{1t} \sim N(0, Q_1)$$

If you have all required toolboxes (optimization, statistics) this is the output you should be getting at matlab's screen:

**** Optimization Finished ****

Final sum of log likelihood = 1690.3358 (final sum of log likelihood)
Number of Estimated parameters = 4 (number of parameter at the model)

--> Final Parameters <--

Coefficients for Column 1 at indep:

Value of u_1: 0.17544 (constant at beta calculation of column 1)
Value of F_1: 0.26981 (element (1,1) at F matrix, autoregressive coeff of b_1)
Value of Q_1: 0.11781 (variance at residue of beta_1 equation)

Value of R: 5.161e-005 (variance of residue at y equation)

Please note that different matlab versions will output different values, but the difference for the values showed before should be insignificant. If this is not even similar to the output you get, then something is wrong with your matlab. Sometimes is the .mat file that is not opening due to versions incompatibility. In that case, try running the simul and fit scripts and check if the main files are still working. Feel free to contact me for any further problems.

Comparing Against Eviews

In order to check the performance of the package I compared it to the state space modeling package of Eviews. Next I show the output from eviews when I fit the same model as in Example_Script_1_kalmanFit.m, for the same data. The specification in Eviews is given by:

```
@signal dep = sv1*indep + [var = exp(c(1))]
@state sv1 = c(3) + c(4)*sv1(-1) + [var = exp(c(2))]
```

And the output is:

Method: Maximum likelihood (Marquardt)
Sample: 1 500
Included observations: 500
Convergence achieved after 1 iteration

	Coefficient	Std. Error	z-Statistic	Prob.
C(1)	-9.875022	0.072891	-135.4760	0.0000
C(2)	-2.132854	0.250753	-8.505783	0.0000
C(3)	0.175696	0.043939	3.998607	0.0001
C(4)	0.267337	0.132094	2.023834	0.0430

	Final State	Root MSE	z-Statistic	Prob.
SV1	0.239497	0.357195	0.670493	0.5025

Log likelihood	1694.241	Akaike info criterion	-6.760965
Parameters	4	Schwarz criterion	-6.727248
Diffuse priors	0	Hannan-Quinn criter.	-6.747735

Notes that eviews takes the exponential of c(1) and c(2) in order to respect the positivity restrictions in the variances of each equation. Therefore we need to take the exponential of those coefficients in order to recover the true values. This calculation gives exp(c(1))= 5.1445e-005 and exp(c(2))= 0.1185. It’s clear that the differences in the parameter vector between Matlab and eviews are quite small, showing consistency of the package.

Building your own Model using the input optionsSpec

One of the main advantages of this submission is that you can set options for building your own econometric model. This is done by formatting the input optionsSpec according to your choices. The input optionsSpec.u_D controls for where (which regressor) to include the *u* coefficients, and which don't. The input optionsSpec.F_D controls for the process at the beta equations. That is, where (which regressor) to estimate the *F* coefficient (the autoregressive parameter at beta equation), where to place a random walk process and where exclude such autoregressive process from the estimation. An example will clear things up. For instance, let's say you have 1 column at indep (one regressor) and you want to estimate the following state space model:

$$y_t = \beta_{1t} x_{1t} + \varepsilon_t \qquad \varepsilon_t \sim N(0, R)$$
$$\beta_{1t} = F_1 \beta_{1t-1} + \nu_{1t} \qquad \nu_{1t} \sim N(0, Q_1)$$

Notes that *u_1=0* at previous model. This is the option you should be setting at optionsSpec:

```
optionsSpec.u_D={0};      % either 0 or 'e'
optionsSpec.F_D={'e'};    % either 1, 0 or 'e'
```

This particular input means that you don't want a *u* parameter at beta equation and also you want to estimate an autoregressive coefficient for the beta process. The 'e' symbol is the tag for "estimate this one".

For another example, let's say you want a pure random walk model at the beta. Here are the equations describing the system:

$$y_t = \beta_{1t} x_{1t} + \varepsilon_t \qquad \varepsilon_t \sim N(0, R)$$
$$\beta_{1t} = \beta_{1t-1} + \nu_{1t} \qquad \nu_{1t} \sim N(0, Q_1)$$

And here are the options supplied at the algorithm:

```
optionsSpec.u_D={0};      % either 0 or 'e'
optionsSpec.F_D={1};      % either 1, 0 or 'e'
```

Now, another example, lets say you have 2 regressors and you want to estimate the following model:

$$y_t = \beta_{1t} x_{1t} + \beta_{2t} x_{2t} + \varepsilon_t \qquad \varepsilon_t \sim N(0, R)$$
$$\beta_{1t} = \mu_1 + \beta_{1t-1} + v_{1t} \qquad v_{1t} \sim N(0, Q_1)$$
$$\beta_{2t} = F_2 \beta_{2t-1} + v_{2t} \qquad v_{2t} \sim N(0, Q_2)$$

This are the options at kalmanFit();

```
optionsSpec.u_D={'e' 0 };    % either 0 or 'e'
optionsSpec.F_D={ 1 'e'};    % either 1, 0 or 'e'
```

Notes that each element at `optionsSpec.u_D` controls for each regressor. This means that element 2 of `optionsSpec.u_D` is the *u* coefficient for column 2 of indep. The same structure is true of `optionsSpec.F_D`. The `optionsSpec` structure can grow as much as needed in order to accommodate different number of independent variables.

Problems and Questions

If you're having problems with the package, please send a message to my email (**marceloperlin@gmail.com**) with a nice personal introduction and an attached zip file containing:

1) the scripts you're running (the main .m file)

2) the error message (if there is one) (could be in .txt or just in the email space)

3) the data (.xls, .txt or .mat)

and I'll take a look over it.

Also, if you have any question regarding the package, feel free to contact me at my previously cited email.

Cheers.

Marcelo

References

HAMILTON, J. (1994) 'Time Series Analysis' Princeton University Press.

KIM, C., J., NELSON, C., R. (1999) 'State Space Model with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications' *The MIT press*.