Description of the MS-Regress Matlab package

Author: Marcelo Perlin / (PhD Student) / ICMA – Reading University

Email: marceloperlin@gmail.com / m.perlin@icmacentre.ac.uk

The purpose of this document is to report the models that this markov switching regression package can handle. The mathematical notation behind the markov switching/Hamilton filter is not going to be exposed here since it is not the point. This definitely isn't an introductory document on the markov switching filter. My advice to those who are just starting out is to first read through the papers and books at the bibliography and then come back here to check the application of the algorithm and the general computational structure (input/output) of the model.

The Mean Equation

This particular matlab package is quite flexible when it comes to the specification of markov switching models. The engine of the estimation function is similar to the one of the autoregressive package since I just modified the last one in order to get the new one. The central point of this flexibility resides in the input argument S, which controls for where to include markov switching effect.

For instance, if you have 2 explanatory variables (x_{1t} , x_{2t}) and if S=[1 1], then the model for the mean equation is:

$$y_t = \beta_1^{S_t} x_{1t} + \beta_2^{S_t} x_{2t} + \sigma^{S_t} \varepsilon_t$$

S_t represents the state at time *t*, that is, $S_t = 1....K$, where K is the number of states

σ^{S_t} is the model's standard deviation at state S_t

$\beta_i^{S_t}$ is the beta coefficient for explanatory variable *i* at state S_t where *i* goes from *1* to *n*

ε_t is the residue which follows a particular distribution (in this case Normal or Student), with zero mean and std=var=1

Now, if S=[0 1], then the model is:

$$y_t = \beta_1 x_{1t} + \beta_2^{S_t} x_{2t} + \sigma^{S_t} \varepsilon_t$$

Notes that, with this change in input argument S, only the second coefficient and the model's standard deviation are switching according to the markov chain. Also, you should notes that the order in S is the same order as in the mean equation. For instance, if S=[1 0], then the mean equation becomes:

$$y_t = \beta_1^{S_t} x_{1t} + \beta_2 x_{2t} + \sigma^{S_t} \varepsilon_t$$

As an example for the markov switching fitting function, the following options at MS_Regress_Fit():

```
dep=some_variables(:,1);                     % Defining dependent variable
Constant=ones(length(dep),1);                % Defining a constant vector
indep=[Constant some_variables(:,3:4)];      % Defining some explanatory
k=3;                                         % Number of States
S=[1 0 0];                                   % Defining which ones do switch
advOpt.distrib='Normal';                     % Defining the distribution
advOpt.std_method='white';                   % Defining method for std errors
```

reefers to the estimation of the equations:

$$y_t = \beta_1^{S_t=1} x_{1t} + \beta_2 x_{2t} + \beta_3 x_{3t} + \sigma^{S_t=1} \varepsilon_t \quad \text{if } S_t = 1$$

$$y_t = \beta_1^{S_t=2} x_{1t} + \beta_2 x_{2t} + \beta_3 x_{3t} + \sigma^{S_t=2} \varepsilon_t \quad \text{if } S_t = 2$$

$$y_t = \beta_1^{S_t=3} x_{1t} + \beta_2 x_{2t} + \beta_3 x_{3t} + \sigma^{S_t=3} \varepsilon_t \quad \text{if } S_t = 3$$

with:

$$P = \begin{bmatrix} p_{11} & p_{21} & p_{31} \\ p_{12} & p_{22} & p_{32} \\ p_{13} & p_{33} & p_{33} \end{bmatrix}$$

as the transition matrix, which controls the probability of a switch from state *j* (column *j*) to state *i* (row *i*). The sum of each column in *P* is equal to one, since they represent full probabilities of the process for each state. Notes that those options are the same provided at the example script Example_MS_Regress_Fit.m.

Another flexibility of the package is that, since I wrote it for dealing with a generic type of regression, you can set any kind of explanatory variable in the model as long as they are observed (you have it available for the whole time period studied). This includes autoregressive components, constants or just plain regression on other variables. This means that you can set up your own econometric model without changing any part of the original code.

If you run the script Example_MS_Regress_Fit.m with matlab version[1] Matlab_7.2.0.232 (2006a), this is the output you should be getting if you have all the proper packages installed (optimization, statistics). Notes that I've also added some explanations for the results (in red).

[1] Different versions of matlab will output different numbers. The difference is quite small (insignificant) for the coefficients values, but increases significantly for the standard errors. For more information, please check the excel file within the zip file

***** MS Optimizations terminated. *****

Final log Likelihood: 3687.5802 (maximum log likelihood)
Number of parameters: 17 (number of parameters in the model)
Distribution Assumption -> Normal (the distribution assumption. You can change it at advOpt.distrib)
Method for standard error calculation -> white (the method for calculation of standard errors ('white' for white(1984) or 'newey_west' for Newey and West (1987). Change it at advOpt.std_method

-----> Final Parameters <-----

---> Non Switching Parameters <--- (the parameter that don't switch (the 0 options at input S))

Non Switching Parameter at Indep collumn 2
Value: 0.078412 (value of coefficient)
Std error: 0.26981 (standard error)
Non Switching Parameter at Indep collumn 3
Value: 0.63409 (value of coefficient)
Std error: 0.98937 (standard error)

---> Switching Parameters <--- (the parameters that do switch (the 1 options at input S))

State 1
Standard Deviation: 0.0069734 (the model's standard deviation (the sigma) at state 1)
Std Error: 0.0045348 (corresponding standard error)
State 2
Standard Deviation: 0.0071614 (the model's standard deviation (the sigma) at state 2)
Std Error: 0.0024444 (corresponding standard error)
State 3
Standard Deviation: 0.0040062 (the model's standard deviation (the sigma) at state 3)
Std Error: 0.003222 (corresponding standard error)

Switching Parameters for Indep collumn 1

State 1
Value: 0.0059847 (value of the switching variable Indep column 1 at state 1)
Std error: 0.0021562 (associated standard error)
State 2
Value: -0.010234 (value of the switching variable Indep column 1 at state 2)
Std error: 0.0029159 (associated standard error)
State 3
Value: -0.00021858 (value of the switching variable Indep column 1 at state 3)
Std error: 0.00077067 (associated standard error)

---> Transition Probabilities Matrix <--- (the matrix that controls the probability of each regime switch)

0.639	0.199	0.102
0.066	0.585	0.027
0.294	0.214	0.869

If this isn't the output you're getting then something is seriously wrong with your Matlab. From the emails I get, usually it is a version problem, where the .mat doesn't work. My suggestion is then to check if the main routines that don't use .mat files are working (any of the simul and fit scripts). Or just go out there and upgrade your software at www.mathworks.com.

The Distribution Equation (likelihood shape)

Notes that in the latest update I've added the choice for distribution assumption for the model (this is the dist parameter at the function). You can choose either the Normal distribution or the Student distribution (dist='Normal', dist='t').

The likelihood of the Normal distribution, for the case of generic state *j*, is given by:

$$L\left(y_t \mid \theta, S_t = j\right) = \frac{1}{\sigma^j \sqrt{2\pi}} \cdot \exp\left(-\frac{\left(y_t - \overline{y_t^j}\right)^2}{2\left(\sigma^j\right)^2} \right)$$

Where $\overline{y_t^j}$ is the fitted mean equation at time *t* for state *j*.

And the likelihood for the student distribution is:

$$L\left(y_t \mid \theta, S_t = j\right) = \frac{\Gamma\left(\frac{v^j + 1}{2}\right)}{\Gamma\left(\frac{v^j}{2}\right) \sigma^j \sqrt{\pi v^j} \cdot \left(1 + \frac{\left(y_t - \overline{y_t^j}\right)^2}{v^j \left(\sigma^j\right)^2}\right)^{\left(\frac{v^j+1}{2}\right)}}$$

You can see that for the *t* distribution there is an extra parameter (v^j) (degree of freedom) which is also switching states and is estimated from data. Those last equations are then used for maximum log likelihood estimation of the model as a constrained optimization problem (function fmincon.m).

Using it for your own Data

You probably want to apply the package to you own series. This topic will set some advices of how you can do this if the model is not converging to a solution.

1. If you're dealing with returns (log or arithmetic), don't multiply them by 100 (just use them in the original form). I'm not exactly sure why, but this is something that people do when estimating in GAUSS and Ox and it is supposed to help the estimation. If you do this at Matlab, the fmincon function (the one which maximizes log likelihood) gets really messed up because the bounds of the parameters are badly shaped for this kind of issue, making the fmincon get stuck at –inf values. If you don't believe me, just try to estimate the Example_Script_Fit with x=ret*100.

2. Try to make your explained and explanatory series to behave around zero. For instance, if the model is not converging, then try diminishing the mean from the original time series. This will help the optimizing function (fmincon()) to find the solution. As you probably know, this is a gentle transformation.

3. Always try to estimate simple models. For instance, don't try to estimate any model with $k>3$ and number of explanatory variables higher than 4. The model's size (number of parameters) grows exponentially as n and k grows. For instance, if $k=5$ and $n=4$ (4 explanatory variables), then the model has 55 parameters to be estimated from data, which is definitely too much for fmincon(). Don't get me wrong, the package will try to estimate it, but the solution is probably a local maximum and you can't really trust the output you get. If you're still not convinced, then Google "parsimonious models" (go ahead, no one is watching ☺).

If, after those steps, you're still having problems converging to a solution, 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 full data (explained and explanatory) (.xls, .txt or .mat)

and I'll take a look over it. I'll try to reply in less than 48 hours if the problem is simple (usually is). Also, if you have any question regarding the package, feel free to contact me at my previously cited email.

Cheers.

Marcelo

References

DUEKER, M., NEELY, (2001) ‘C., J., Can Markov Switching Models Predict Excess Foreign Exchange Returns?’ *FRB of St. Louis Working Paper No. 2001-021F*. Available at SSRN: http://ssrn.com/abstract=648227.

HAMILTON, J., D. (2005) ‘Regime Switching Models’ *Palgrave Dictionary of Economic.*
KIM, C., J., NELSON, C., R. (1999) ‘State Space Model with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications’ *The MIT press*.

NEWEY, B., WEST, D (1987) ‘A Simple, Positive semidefinite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix’ *Econometrica*, ed. 55, p. 347-370.

WHITE, H. (1984) ‘Asymptotic Theory for Econometricians’ New York: Academic Press.