This version: 9/2021. Users of this Matlab code should cite Qu, Z and F. Zhuo (2021), "Likelihood Ratio Based Tests for Regime Switching," *Review of Economic Studies*, 88, 937–968. The first part of this note is a description of the proposed methods to help understand the code.

1 The proposed methods

The model under the null hypothesis is

$$y_t = x_t'\gamma + u_t, \tag{1}$$

where x_t is a k-by-1 vector of regressors, with the first element equal to 1, and $u_t \sim i.i.d.N(0, \sigma^2)$. An important example is the autoregressive distributed lags (ADL) model, in which case

$$x_t = \begin{bmatrix} 1 \\ y_{t-1} \\ ... \\ y_{t-p} \\ z_{t-1} \\ ... \\ z_{t-q} \end{bmatrix}.$$

The model under the alternative hypothesis is

$$y_t = x_{1,t}'\delta_1 \mathbf{1}_{\{S_t=1\}} + x_{1,t}'\delta_2 \mathbf{1}_{\{S_t=2\}} + x_{2,t}'\beta + u_t, \tag{2}$$

where $x_{1,t}$ is a subvector of x_t whose coefficients are allowed to switch between regimes 1 and 2; $x_{2,t}$ contains the rest of x_t, with coefficients staying constant across regimes. The intercept is always allowed to switch. The variance is not allowed to switch in this version of the code.

For example, suppose the null hypothesis specifies an ADL(1,1) model, i.e.,

$$y_t = [1, y_{t-1}, z_{t-1}]\gamma + u_t,$$

where

$$[1, y_{t-1}, z_{t-1}]\gamma = [1, y_{t-1}, z_{t-1}] \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} = \gamma_1 + \gamma_2 y_{t-1} + \gamma_3 z_{t-1}.$$

Then, we have the following cases regarding the alternative hypothesis:

1. If only the intercept is allowed to switch, then the model under the alternative hypothesis is

$$y_t = \delta_1 \mathbf{1}_{\{S_t=1\}} + \delta_2 \mathbf{1}_{\{S_t=2\}} + [y_{t-1}, z_{t-1}]\beta + u_t,$$

with δ_1 and δ_2 being two scalar parameters.

2. If the intercept and the coefficient of y_{t-1} are allowed to switch, then the model under the alternative hypothesis is

$$y_t = [1, y_{t-1}]\delta_1 \mathbf{1}_{\{S_t=1\}} + [1, y_{t-1}]\delta_2 \mathbf{1}_{\{S_t=2\}} + z_{t-1}\beta + u_t,$$

with δ_1 and δ_2 being 2-by-1 vectors.

3. If the intercept and the coefficient of z_{t-1} are allowed to switch, the model under the alternative hypothesis is

$$y_t = [1, z_{t-1}]\delta_1 \mathbf{1}_{\{S_t=1\}} + [1, z_{t-1}]\delta_2 \mathbf{1}_{\{S_t=2\}} + y_{t-1}\beta + u_t,$$

with δ_1 and δ_2 being 2-by-1 vectors.

4. If all the regression coefficients are allowed to switch, then the model under the alternative hypothesis is

$$y_t = [1, y_{t-1}, z_{t-1}]\delta_1 \mathbf{1}_{\{S_t=1\}} + [1, y_{t-1}, z_{t-1}]\delta_2 \mathbf{1}_{\{S_t=2\}} + u_t.$$

In this case, the β term is absent.

The Matlab code covers all four cases outlined above.

The null and alternative hypotheses can be summarized as

$$\begin{aligned} H_0 &: \delta_1 = \delta_2 = \delta_* \text{ for some unknown } \delta_*; \\ H_1 &: (\delta_1, \delta_2) = (\delta_1^*, \delta_2^*) \text{ for some unknown } \delta_1^* \neq \delta_2^* \text{ and } (p, q) \in (0,1) \times (0,1). \end{aligned}$$

To compute the test statistic, let $\widetilde{\gamma}$ and $\widetilde{\sigma}^2$ denote the MLE of the null model (1), i.e.,

$$(\widetilde{\gamma}, \widetilde{\sigma}^2) = \arg\max_{\gamma, \sigma^2} \mathcal{L}^N(\widetilde{\gamma}, \widetilde{\sigma}^2).$$

Let $\mathcal{L}^A(p, q, \delta_1, \delta_2, \beta, \sigma^2)$ be the log likelihood function under the alternative hypothesis for (2) evaluated at some $0 < p, q < 1$. The log likelihood ratio at this (p, q) is then equal to

$$LR(p, q) = 2\left[\max_{\delta_1, \delta_2, \beta, \sigma^2} \mathcal{L}^A(p, q, \delta_1, \delta_2, \beta, \sigma^2) - \mathcal{L}^N(\widetilde{\gamma}, \widetilde{\sigma}^2)\right]. \tag{3}$$

The test statistic is computed as

$$\text{SupLR}(\Lambda_\epsilon) = \sup_{(p,q)\in\Lambda_\epsilon} LR\,(p, q)\,,$$

where Λ_ϵ is a compact set, given by

$$\Lambda_\epsilon = \{(p, q) : p + q \geq 1 + \epsilon \text{ and } \epsilon \leq p, q \leq 1 - \epsilon \text{ with } \epsilon > 0\}. \tag{4}$$

2 The Matlab code

The main program is `user.m`. Running this program will produce the following results: (1) the test statistic, (2) the critical values and the p-value, (3) the parameter estimates under the null and alternative hypotheses, and (4) the smoothed regime probability implied by these estimates. The program `user.m` is divided into several small sections for ease of understanding and implementation. The first two sections are for loading the data and specifying the test. They are case-specific, and thus need to be modified according to the application. The remaining sections are generic, and the user should not need to modify them.

Under the default specification (using the US GDP series), the first section of the code is

```
load GDP;
D=GDP;
y = D(:,1);
T = size(y,1);
x = ones(T,2);
x(:,2) = D(:,2);
```

In this case, `y` is the GDP series, a T-by-1 vector; `x` is a T-by-2 matrix, starting with a column vector of ones, followed by the lagged values of `y`. In a general application, `x` is a T-by-k vector, starting with a column of ones, followed by the variables whose coefficients are allowed to switch, and then followed by those whose coefficients do not switch.

The second section of the code is as follows:

```
trm=0.02; %the trimming parameter
nrep=5000; %the number of simulation replications for critical values
nd=1; %the number of regression coefficients allowed to switch
rng(12345); %random number generator seed, for replication purposes.
```

Here, the important parameters are `trm` and `nd`,where `trm` is the ϵ parameter in (4) and `nd` is the number of regression coefficients allowed to switch, a value less than or equal to k.

The remaining Matlab programs are functions to support `user.m`. They need to be in the same directory as `user.m` for the program to run.

Computational time. The test is fast to compute, requiring no more than a few minutes on a typical desktop computer. However, simulating the critical values is time consuming. It can take between half an hour and a few hours, or even a day, depending on the computer. Since the code allows for parallel computation, using multiple cores can substantially reduce the computational time.

Comments are welcome. Please contact to qu@bu.edu.