Final Assignment

Kenji Sato

27 May, 2017

1 Deadline and submission

June 9, 2017, 18:00
Late submission not accepted.

2 Assignment

Complete your **lrem** project that you launched for `hw08` and do a simulation in vignettes.

In addition, do the following.

- Write `README.md`. The content must include
  - installation manual (`devtools::use_readme_rmd()` will help); and
  - a short description about what the package does.
- Write a short paper accompanying the package (as a *vignette*).

2.1 Specifications

Your package should be able to solve a linear rational expectations (LRE) models.

A. Autonomous LRE System

Your package solves LRE model without inputs.

$$E\mathbb{E}_t \begin{bmatrix} x^1_{t+1} \\ x^2_{t+1} \end{bmatrix} = A \begin{bmatrix} x^1_t \\ x^2_t \end{bmatrix} \tag{1}$$

where $x^1 \in \mathbb{R}^{n_1}$ is the vector of predetermined variables and $x^2 \in \mathbb{R}^{n_2}$ the vector of non-predetermined variables.

The solution to Eq. (1)

$$
\begin{aligned}
x^1_{t+1} &= h(x^1_t) + \xi_{t+1} \\
x^2_t &= g(x^1_t)
\end{aligned}
\tag{2}
$$

is characterized by two functions (g, h). See the lecture slides for the formula.

Suggested API

We would suggest that you implement this algorithm with the following API. Whether or not you use this API, document usage properly.

`lre_auto`	
Purpose	Compute (g, h) from (A, E, n_1)
Arguments	`A`: matrix A
	`E`: matrix E
	`nx`: number of predetermined variables, n_1, OR
	`x0`: initial vector, x^1_0
Return Value	`list(g = g, h = h)`: list of two functions (g, h) in (2)

Users should be able to use this function with the following code combination after properly defined `A` and `E`.

```r
sol <- lre_auto(A, E, nx = 3)
g <- sol$g
h <- sol$h
```

B. LRE System with AR Inputs

Let Φ be a (strictly) stable matrix and ϵ be i.i.d. shocks.

$$
\begin{aligned}
E\mathbb{E}_t \begin{bmatrix} x^1_{t+1} \\ x^2_{t+1} \end{bmatrix} &= A \begin{bmatrix} x^1_t \\ x^2_t \end{bmatrix} + Bu_t \\
u_{t+1} &= \Phi u_t + \epsilon_{t+1}
\end{aligned}
\tag{3}
$$

This system satisfies

$$
\begin{bmatrix} I & O \\ O & E \end{bmatrix} \mathbb{E}_t \left[\begin{array}{c} u_{t+1} \\ \hline x^1_{t+1} \\ x^2_{t+1} \end{array}\right] = \begin{bmatrix} \Phi & O \\ B & A \end{bmatrix} \left[\begin{array}{c} u_t \\ \hline x^1_t \\ x^2_t \end{array}\right]
\tag{4}
$$

and under the assumptions discussed in the class (4) is solved by

$$\begin{aligned} \begin{bmatrix} u_{t+1} \\ x^1_{t+1} \end{bmatrix} &= h(u_t, x^1_t) + \begin{bmatrix} I \\ 0 \end{bmatrix} \epsilon_{t+1} \\ x^2_t &= g(u_t, x^1_t), \end{aligned} \tag{5}$$

where g and h here are computed by the same algorith used for autonomous system but with the extended coefficient matrices.

(If you feel unhappy with the order of (u_t, x^1_t) and want it to be (x^1_t, u_t), you may arrange it freely. You will need to define the coefficient matrices more carefully.)

Provide an R function that computes (g, h) in (5).

Suggested API

We would suggest that you implement this algorithm with the following API. Whether or not you use this API, document usage properly.

`lre_ar`	
Purpose	Compute (g, h) from (A, E, B, Φ, n_1)
Arguments	`A`: matrix A
	`E`: matrix E
	`B`: matrix B
	`Phi`: matrix Φ
	`nx`: number of predetermined variables, n_1, OR
	`x0`: initial vector, x^1_0
Return Value	`list(g = g, h = h)`: list of two functions (g, h) in (5)

Hint: Let n_ϵ be the size of exogenous shocks. You can construct extended coefficient matrices in (4) from `A`, `E`, `B` and `Phi`. Pass the obtained matrices to `lre_auto`.

C. Simulate

You need to extend that old `simulate` function to accept shock process ξ_t or ϵ_t.

Suggested API

We would suggest that you implement simulation function with the following API. Whether or not you use this API, document usage properly.

`simulate`	
Purpose	Simulation given (g, h, x_0^1, ξ)
Arguments	`g`: function g in (2) or (5)
	`h`: function h in (2) or (5)
	`x0`: initial x_0^1; understood as (u_0, x_0^1) in case of (5)
	`t`: Integer, simulation length
	`e`: vector or matrix, each row `e[k, ]` corresponds to ξ_{k+1}.
Return Value	`out`: matrix of simulation output

Either one of `e` or `t` can accept `NULL`.

- If `e` is `NULL`, it is understood as zero shocks.
- If `t` is `NULL`, `nrow(e) + 1` might be used as `t`.

If `nrow(e)` is smaller than `t - 1`, you might need to extend matrix `e` to have `t - 1` rows and fill the empty elements with zeros.

D. Impulse Response

Let x_0^1 and x_0^2 be on the steady state, i.e., they are zero.

To make code as simple as possible, let's define an impulse response as a response to

$$\xi_1 \neq 0, \xi_2 = 0, \dots \tag{6}$$

Users are responsible to set, for example,

$$\xi_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{7}$$

Suggested API

We would suggest that you implement this algorithm with the following API. Whether or not you use this API, document usage properly.

`impulse`	
Purpose	Impulse response given (g, h, ξ_1)

`impulse`	
Arguments	`g`: function g in (2) or (5)
	`h`: function h in (2) or (5)
	`x0`: steady state. For linear models, it's the origin.
	`t`: Integer, simulation length
	`e1`: i-th element `e[i]` corresponds to a shock to i-th endogenous variable
Return Value	`out`: matrix of simulation output

It seems redundant to pass `x0` because it should be zero. This is due to the unfortunate fact that our `g` and `h` don't have information on the system size.

2.2 Exmaple 1

See lrem-hansen-rbc.html

Try to replicate this example.

2.3 Example 2

Simulate the canonical New-Keynesian model that consists of the following three equations.

Phillips curve (AS curve)

$$\pi_t = \beta \mathbb{E}_t \pi_{t+1} + \kappa x_t + u_t^S$$

IS curve

$$x_t = \mathbb{E}_t x_{t+1} - \frac{1}{\sigma} \left(i_t - \mathbb{E}_t \pi_{t+1} \right) + u_t^D$$

Monetary policy rule

$$i_t = \alpha \pi_t + \iota x_t$$

Assume AR shocks with $|\rho_S|, |\rho_D| < 1$, $\mathbb{E}_t \epsilon_{t+1}^S = 0$, and $\mathbb{E}_t \epsilon_{t+1}^D = 0$:

$$\begin{aligned} u_{t+1}^S &= \rho_S u_t^S + \epsilon_{t+1}^S \\ u_{t+1}^D &= \rho_D u_t^D + \epsilon_{t+1}^D \end{aligned}$$

- π_t: log inflation rate
- x_t: output gap
- i_t: log gross nominal interest rate
- σ: intertemporal elasticity of substitution

- β: discount rate
- κ: slope of the Phillips curve
- α and ι: monetary policy parameters

Simulate the impulse response with the following parameter values.

```
alpha <- 2
iota <- 0
beta <- 0.99
sigma <- 1
kappa <- 0.132
rhoS <- 0.9
rhoD <- 0.9
```

All three endogenous variables, π, x and i, are non-predetermined.

3 Assessment

The TA and lecturer (henceforth, the users) will ask themselves the following questions.

- **Installation**: Does installation go smoothly?
- **Vignettes**: Is the vignette(s) written reasonably well?
- **Functionality**: Does the package work as expected?
- **Documentation**: Is the package well documented?
- **Format**: Does the code written (aesthetically) well?

3.1 Installation

The users will install your package on their computers following the instruction provided in `README.md`.

`README.md` should provide a sunccinct description about who wants to use the package and what it provides to solve a problem. Desirably, it provides a short example. Long example should be written in a vignette.

3.2 Vignettes

In the vignette(s), you will discuss how the package is supposed to be used to solve a real problem.

The users check what vignettes the package has by

```
vignette(package = "lrem")
```

Then they read them with

```
vignette("lrem-vignette-topic", package = "lrem")
```

Since your package contains several functions, the users want to know how those functions can be combined to solve their tasks.

See http://r-pkgs.had.co.nz/vignettes.html for more details.

3.3 Functionality

In general, the package should be able to solve the problems `README.md` and vignettes declare that it does.

The users will

- redo the computation described in the vignettes; and
- try to solve problems they have, which are unknown to the package creator.

Since this project is for a course, your package is supposed to have the functionalities discussed in Section 2.

3.4 Documentation

The users read the help page (for example, by `?lre_auto`) very very often. Documentation is important. See http://r-pkgs.had.co.nz/man.html to learn how to write documentation.

3.5 Format

Sensible users read the code (at least partially) to become confident that the package provides what they really want. If your code doesn't stick to a consistent and reasonable style, the users might stop using it because they think the package might be unreliable.

Format is important also because ill-formatted code is difficult to maintain.

Follow this style http://adv-r.had.co.nz/Style.html if you have no particular style yet.

4 Remarks

4.1 Collaboration

Because the repositories you make are public, everybody else can see your work. I don't deny you to learn from other students' works but keep in mind that their works are copyright protected unless otherwise explicitly stated.

Be a sensible user not a pirate.

4.2 Need help?

If you have questions about this assignment, ask on Slack.