Materials 26 - Implementation of target criterion - Documentation of code

Laura Gáti

April 20, 2020

Overview

1 Summary of codes

I'm sending you a folder with codes. The code you want to start with is `main_file.m`. Let me describe this code briefly. This code has three sections:

1. Parameters

   Loads parameters, solves RE version of model, generates disturbances and sets up choices and figure titles and the like - here you don't need to change anything.

2. Model selection

   Here you can specify the PLM, the gain scheme, which variables to input exogenously and what those inputs should be and the assumption whether agents' beliefs incorporate the Taylor rule or not. In detail:

   - PLM: choices are

     (a) constant only: learn only intercept a for π, x, i (this is the constant-only special case of what I refer to as "vector learning")

     (b) constant only, π only: learn only intercept $a(1,1) \equiv \bar{\pi}$ for π only (this is the constant-only special case of what I call "scalar learning")

(c) slope and constant: learn entire ϕ matrix (intercept a and slope b) for π, x, i (this is the general case vector learning)

- gain: choices are
  (a) decreasing gain
  (b) constant gain
  (c) endogenous gain, CEMP's criterion
  (d) endogenous gain, CUSUM criterion
  (e) endogenous gain, smooth criterion: a note for this one: I haven't yet found the ideal functional form for this, which is why I recommend you use the CUSUM criterion.
- Choice of input variables: choices are
  (a) `s_inputs`$= [0, 0, 1]$: i only
  (b) `s_inputs`$= [0, 1, 1]$: x, i
  (c) `s_inputs`$= [1, 1, 1]$: π, x, i
- initialization of exogenous sequences: choices are:
  (a) Taylor rule: use the sequence(s) generated by the simulation with the Taylor rule as input sequence(s)
  (b) random: use random sequences as input sequences
- do agents know the Taylor rule? Need to set this in `smat.m`. Choices are:
  (a) Yes: set by setting `s1 = s1_TR` in `smat.m`.
  (b) No: set by commenting out `s1 = s1_TR` in `smat.m`.

3. An initial evaluation of loss

   Here you also don't need to do anything (but of course you can).

   - Simulates the model conditional on the Taylor rule using `sim_learnLH_clean.m` detailed below (these are the variables with the 0 subscript, that get plotted first)
   - Simulates the model given the exogenous sequence(s) you specified above in section 2 using `sim_learnLH_clean_given_seq.m` (these are the variables with the 1 subscript, that get plotted second).
   - Evaluates the objective function one time and spits out as well as plots the residuals of the NKIS, the NKPC and the TR.

The two simulation codes in detail:

1. `sim_learnLH_clean.m`: simulate model

   $\hookrightarrow$ uses `ALM.m`: if the LOM (equation A.1-A.3) of observables is written

$$\underbrace{\begin{pmatrix} 0 & 1 & \sigma \\ 1 & -\kappa & 0 \\ -\psi_\pi & -\psi_x & 1 \end{pmatrix}}_{\equiv A} \underbrace{\begin{bmatrix} \pi_t \\ x_t \\ i_t \end{bmatrix}}_{\equiv y_t} = \underbrace{\begin{pmatrix} s_1 f_b + s_2 s_t \\ s_3 f_a + s_4 s_t \\ s_5 s_t \end{pmatrix}}_{\equiv B} \quad \text{where} \quad s_t = \begin{bmatrix} r_t^n \\ \bar{i}_t \\ u_t \end{bmatrix} \tag{1}$$

   where s_i are generated by `smat.m` and are given by

$$s_1 = \begin{bmatrix} \sigma & 1-\beta & -\sigma\beta \end{bmatrix} \qquad s_2 = \sigma \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} (\mathbb{I}_{nx} - \beta h_x)^{-1} \tag{2}$$

$$s_3 = \begin{bmatrix} (1-\alpha)\beta & \kappa\alpha\beta & 0 \end{bmatrix} \quad s_4 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} (\mathbb{I}_{nx} - \beta h_x)^{-1} \tag{3}$$

$$s_5 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \text{ or if you include a mon pol shock } \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \tag{4}$$

   `ALM.m` computes $y_t = A^{-1}B$.

   Note: To impose that agents form interest-rate expectations according to the Taylor rule (which I refer to as agents "knowing the Taylor rule"), I replace s_1 in `smat.m` by

$$s_1^{old} = \begin{bmatrix} \sigma - \sigma\beta\psi_\pi & 1-\beta-\sigma\beta\psi_x & 0 \end{bmatrix} \tag{5}$$

   You need to do this within `smat.m`, but I've set it such that it displays this info assumption the first time it's called.

2. `sim_learnLH_clean_given_seq.m` simulate model given exogenous input sequence(s)

   $\hookrightarrow$ uses `A9A10.m`: This code first determines how many sequences are input, and then uses equations A.9 and A.10 to compute the rest of the observables as

$$x_t = -\sigma i_t + s_1 f_b + s_2 s_t \tag{A9}$$

$$\pi_t = \kappa x_t + s_3 f_a + s_4 s_t \tag{A10}$$

   where s_i are again computed by `smat.m`. Of course, you can again tell `smat.m` whether you want agents to know the Taylor rule or not.

A Model summary

$$x_t = -\sigma i_t + \hat{\mathbb{E}}_t \sum_{T=t}^{\infty} \beta^{T-t}\big((1-\beta)x_{T+1} - \sigma(\beta i_{T+1} - \pi_{T+1}) + \sigma r_T^n\big) \tag{A.1}$$

$$\pi_t = \kappa x_t + \hat{\mathbb{E}}_t \sum_{T=t}^{\infty} (\alpha\beta)^{T-t}\big(\kappa\alpha\beta x_{T+1} + (1-\alpha)\beta\pi_{T+1} + u_T\big) \tag{A.2}$$

$$i_t = \psi_\pi \pi_t + \psi_x x_t + \bar{i}_t \qquad \text{(if imposed)} \tag{A.3}$$

$$\text{PLM:} \qquad \hat{\mathbb{E}}_t z_{t+h} = a_{t-1} + b h_x^{h-1} s_t \quad \forall h \geq 1 \qquad b = g_x\, h_x \tag{A.4}$$

$$\text{Updating:} \qquad a_t = a_{t-1} + k_t^{-1}\big(z_t - (a_{t-1} + b s_{t-1})\big) \tag{A.5}$$

$$\text{Anchoring function:} \qquad k_t = k_{t-1} + \mathbf{g}(fe_{t-1}^2) \tag{A.6}$$

$$\text{Forecast error:} \qquad fe_{t-1} = z_t - (a_{t-1} + b s_{t-1}) \tag{A.7}$$

$$\text{LH expectations:} \qquad f_a(t) = \frac{1}{1-\alpha\beta} a_{t-1} + b(\mathbb{I}_{nx} - \alpha\beta h)^{-1} s_t \qquad f_b(t) = \frac{1}{1-\beta} a_{t-1} + b(\mathbb{I}_{nx} - \beta h)^{-1} s_t \tag{A.8}$$

This notation captures vector learning (z learned) for intercept only. For scalar learning, $a_t = \begin{pmatrix}\bar{\pi}_t & 0 & 0\end{pmatrix}'$ and b_1 designates the first row of b. The observables (π, x) are determined as:

$$x_t = -\sigma i_t + \begin{bmatrix}\sigma & 1-\beta & -\sigma\beta\end{bmatrix} f_b + \sigma \begin{bmatrix}1 & 0 & 0\end{bmatrix} (\mathbb{I}_{nx} - \beta h_x)^{-1} s_t \tag{A.9}$$

$$\pi_t = \kappa x_t + \begin{bmatrix}(1-\alpha)\beta & \kappa\alpha\beta & 0\end{bmatrix} f_a + \begin{bmatrix}0 & 0 & 1\end{bmatrix} (\mathbb{I}_{nx} - \alpha\beta h_x)^{-1} s_t \tag{A.10}$$

B Target criterion

The target criterion in the simplified model (scalar learning of inflation intercept only, $k_t^{-1} = \mathbf{g}(fe_{t-1})$):

$$\pi_t = -\frac{\lambda_x}{\kappa}\Bigg\{x_t - \frac{(1-\alpha)\beta}{1-\alpha\beta}\Big(k_t^{-1} + ((\pi_t - \bar{\pi}_{t-1} - b_1 s_{t-1}))\mathbf{g}_\pi(t)\Big)$$
$$\Big(\mathbb{E}_t \sum_{i=1}^{\infty} x_{t+i} \prod_{j=0}^{i-1}(1 - k_{t+1+j}^{-1} - (\pi_{t+1+j} - \bar{\pi}_{t+j} - b_1 s_{t+j})\mathbf{g}_{\bar{\pi}}(t+j))\Big)\Bigg\} \tag{B.1}$$

where I'm using the notation that $\prod_{j=0}^{0} \equiv 1$. For interpretation purposes, let me rewrite this as follows:

$$\pi_t = -\frac{\lambda_x}{\kappa} x_t + \frac{\lambda_x}{\kappa}\frac{(1-\alpha)\beta}{1-\alpha\beta}\Big(k_t^{-1} + fe_{t|t-1}^{eve}\mathbf{g}_\pi(t)\Big)\mathbb{E}_t \sum_{i=1}^{\infty} x_{t+i}$$
$$-\frac{\lambda_x}{\kappa}\frac{(1-\alpha)\beta}{1-\alpha\beta}\Big(k_t^{-1} + fe_{t|t-1}^{eve}\mathbf{g}_\pi(t)\Big)\Big(\mathbb{E}_t \sum_{i=1}^{\infty} x_{t+i} \prod_{j=0}^{i-1}(k_{t+1+j}^{-1} + fe_{t+1+j|t+j}^{eve})\mathbf{g}_{\bar{\pi}}(t+j)\Big) \tag{B.2}$$

Interpretation: tradeoffs from discretion in RE + effect of current level and change of the gain on future tradeoffs + effect of future expected levels and changes of the gain on future tradeoffs