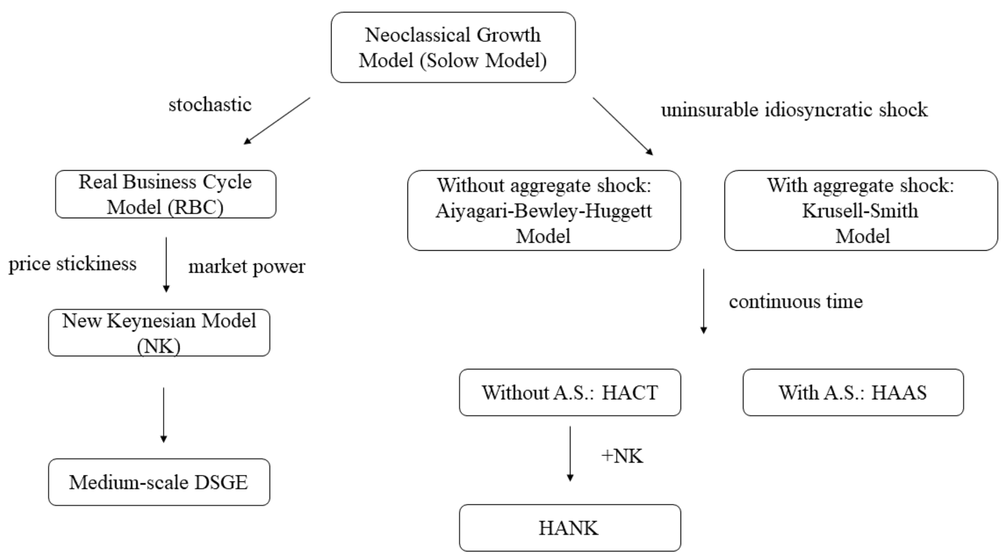
Macroeconomic Analysis with Machine Learning and Big Data

Lecture 9: HACT with Aggregate Shocks

Instructor: Yucheng Yang *Scribe: Jieyu Guo, Zehou Zhang*

Disclaimer: *These notes have not been subjected to careful scrutiny and may contain errors. Please use it at your own risk.*

1 Recap: Structural Models in Macroeconomics



2 Model Setup

I follow Ahn *et al.* (2018, NBER Macro Annual) to introduce the basic model setup of HACT with aggregate shocks. I will introduce their solution method, as well as the solution method proposed by Fernandez-Villaverde *et al.* (2019) in this note.

- Household (HH): heterogenous over asset a and effective productivity z

$$E_0 \int_0^\infty e^{-\rho t} \frac{C_{jt}^{r\theta}}{1-\theta} dt$$

$$a_{jt} = w_t z_{jt} + r_t a_{jt} - C_{jt}$$

$$z_{jt} \in \{z_L, z_H\}, z_L < z_H$$

$$\text{Assume } \overline{N} \equiv \int_0^1 z_{jt} d_j \text{ is constant } \forall t$$

- Firm: representative in competitive market.

$$Y_t = e^{H_t} K_t^{\alpha} N_t^{1-\alpha}$$

Log productivity H_t is the only source of **aggregate shocks** in this model and follow the Ornstein-Uhlenbeek process:

$$dH_t = -\eta H_t dt + \sigma dW_t$$

As firms are in the competitive market, the factor prices are:

$$r_t = \alpha e^{H_t} K_t^{\alpha-1} \overline{N}_t^{1-\alpha}$$

$$w_t = (1-\alpha) e^{H_t} K_t^{\alpha} \overline{N}^{-\alpha}$$

- Hamilton-Jacobi-Bellman (HJB) equation:
  - individual state variables: a, z_j
  - aggregate state variables: $H, g(a, z)$
  - Remark: in HACT, $g_t(a, z)$ is deterministic given initial conditions, so $g_t(a, z)$ was not in the state variables. With aggregate shocks, $g_t(a, z)$ is stochastic and have many possible realizations, so we have it in the state variable.
  - The fully recursive formulation of the HH problem (**the HJB equation**) is:

$$\begin{aligned} \rho v(a, z, g_t, H_t) = \max_c u(c) &+ \partial_a v(a, z, g_t, H_t)[\omega(g, H)Z + r(g_t, H_t)a - c] \\ &+ \lambda_z[v(a, z', g_t, H_t) - v(a, z, g_t, H_t)] \\ &+ \partial_H v(a, z, g_t, H_t)(-\eta H_t) + \frac{1}{2}\partial_{HH}^2 v(a, z, g_t, H_t)\sigma^2 \\ &+ \int \frac{\delta v(a, z, g_t, H_t)}{\delta g(a, z)} (\kappa_H g)(a, z) da dz \end{aligned} \tag{1}$$

  where $\frac{\delta v(a,z,g,H)}{\delta g(a,z)}$ is the functional derivative, and κ_H is the Kolmogorov Forward operator to be defined later.

- **Kolmogorov Forward (KF) Equation**:

$$\kappa_H g_t(a, z) = \frac{dg_t}{dt}$$

where the Kolmogorov Forward operator κ_H is defined as

$$(\kappa_H g)(a, z) = -\partial_a[s(a, z, g_t, H_t)g(a, z)] - \lambda_z g(a, z') + \lambda_{z'} g(a, z')$$

Comparing to HACT, the HJB equation here with the functional derivative component is too complicated to solve. Fortunately, we can simply this formulation with expectation terms getting in. Using Ito's formula, we have

$$dv(a,z,H_t,g_t) = \partial_H v(a,z,g_t,H_t)(-\eta H_t)dt + \frac{1}{2}\partial_{HH} v(a,z,g_t,H_t)\sigma^2 dt + \sigma\partial_H v(a,z,g_t,H_t)dW_t + \int \frac{\delta v}{\delta g}\kappa_H g_t(a,z)dadzdt \tag{2}$$

Taking expectation to get rid of the Brownian motion, we have

$$\mathbb{E}_t dv(a,z,g_t,H_t) = \partial_H v(a,z,g_t,H_t)(-\eta H_t)dt + \frac{1}{2}\partial_{HH} v(a,z,g_t,H_t)\sigma^2 dt + \int \frac{\delta v}{\delta g}\kappa_H g_t(a,z)dadzdt \tag{3}$$

Plug into Equation 1, **the HJB equation** can be finally written as:

$$\rho v_t(a,z) = \max_c u(c) + \partial_a v_t(a,z)(\omega_t Z + r_t a - c) + \lambda_z(v_t(a,z') - v_t(a,z)) + \frac{1}{dt}\mathbb{E}_t dv_t(a,z) \tag{4}$$

This formulation with expectation term reminds us of the RBC models. It turns out that one of the solution method approximate the problem with a linear rational expectation system problem and solve it with the techniques we have used before.

3 Solution Method I: Linearization around the Steady State Equilibrium

The solution method proposed by Ahn *et al.* (2018) consists of three steps.

- Step 1: Solve the Steady State Equilibrium without Aggregate Shock.

  This step solve the steady state equilibrium without aggregate shocks (i.e. $H_t \equiv 0$) but with idiosyncratic shocks. The HJB and KF equations, and the market clear condition are

  $$\rho v(a,z) = \max_c u(c) + \partial_a v(a,z)(\omega Z + ra - c) + \lambda_z(v(a,z') - v(a,z))$$

  $$0 = -\partial_a[s(a,z)g(a,z)] - \lambda_z g(a,z') + \lambda_{z'} g(a,z')$$

  $$K = \int a g(a,z)dadz$$

  It can be solved with the same method as HACT.

- Step 2: Linearize Equilibrium Conditions:
  This step is to compute a first-order Taylor expansion of the model's discretized equilibrium conditions around steady state.

– discretized equilibrium for finite difference:

$$\begin{aligned}
\rho\overrightarrow{v_t} &= \overrightarrow{u}(\overrightarrow{v_t}) + A(\overrightarrow{v_t}, \overrightarrow{p_t})\overrightarrow{v_t} + \frac{1}{d_t}E_t dv_t \\
\frac{d\overrightarrow{g_t}}{d_t} &= \overrightarrow{A}(\overrightarrow{v_t}; \overrightarrow{p_t})^T\overrightarrow{g_t} \\
dH_t &= -\eta H_t dt + \sigma dW_t \\
\overrightarrow{p_t} &= \overrightarrow{F}(\overrightarrow{g_t}, \overrightarrow{H_t})
\end{aligned} \tag{5}$$

– first-order Taylor expansion

$$E_t\begin{bmatrix} d\hat{v}_t \\ d\hat{g}_t \\ dZ_t \\ 0 \end{bmatrix} = \begin{bmatrix} B_{vv} & 0 & 0 & B_{vp} \\ B_{gv} & B_{gg} & 0 & B_{gp} \\ 0 & 0 & -\eta & 0 \\ 0 & B_{pg} & B_{pH} & -I \end{bmatrix} \times \begin{bmatrix} \hat{v}_t \\ \hat{g}_t \\ H_t \\ \hat{p}_t \end{bmatrix} d_t \tag{6}$$

– plug the pricing equation $\hat{p}_t = B_{pg}\hat{g}_t + B_{pZ}Z_t$ into the remaining equations of the system:

$$E_t\begin{bmatrix} d\hat{v}_t \\ d\hat{g}_t \\ dZ_t \end{bmatrix} = \underbrace{\begin{bmatrix} B_{vv} & B_{vp}B_{pg} & B_{vp}B_{pH} \\ B_{gv} & B_{gg} + B_{gp}B_{pg} & B_{gp}B_{pH} \\ 0 & 0 & -\eta \end{bmatrix}}_{\mathbf{B}} \times \begin{bmatrix} \hat{v}_t \\ \hat{g}_t \\ H_t \end{bmatrix} d_t \tag{7}$$

- Step 3: Solve Linear System
  - Schur decomposition of the matrix B to identify the stable and unstable roots of the system.
  - If the Blanchard and Kahn (1980) condition holds, i.e. the number of stable roots equals the number of state variables $\hat{g}_t$ and Z_t, then we can compute the solution:

$$\begin{aligned}
\hat{v}_t &= D_{vg}\hat{g}_t + D_{vH}H_t, \\
\frac{d\hat{g}_t}{dt} &= (B_{gg} + B_{gp}B_{pg} + B_{gv}D_{vg})\hat{g}_t + (B_{gp}B_{pH} + B_{gv}B_{vH})H_t, \\
dH_t &= -\eta H_t dt + \sigma dW_t, \\
\hat{p}_t &= B_{pg}\hat{g}_t + B_{pH}H_t.
\end{aligned} \tag{8}$$

- Approximated Solution

$$v_t(a_j, z_j) = v(a_j, z_j) + \sum_{k=1}^{I}\sum_{l=1}^{2} D_{vg}[i, j; k, l](g_t(a_k, z_l) - g(a_k, z_l)) + D_{vz}[i, j]H_t \tag{9}$$

Optimal consumption is then given by:

$$c_t(a_i, z_j) = (\partial_a v_t(a_i, z_j))^{-1/\theta}$$

4 Solution Method II: Deep Neural Network Approximation

Fernandez-Villaverde *et al.* (2019) proposes a different method that can better address the nonlinearities in the model. To deal with the functional derivative component in the original HJB equation as below,

$$\begin{aligned}\rho v(a,z,g_t,H_t) = &\max_c u(c) + \partial_a v(a,z,g_t,H_t)[\omega(g,H)Z + r(g_t,H_t)a - c] \\ &+ \lambda_z[v(a,z',g_t,H_t) - v(a,z,g_t,H_t)] \\ &+ \partial_H v(a,z,g_t,H_t)(-\eta H_t) + \frac{1}{2}\partial^2_{HH} v(a,z,g_t,H_t)\sigma^2 \\ &+ \int \frac{\delta v(a,z,g_t,H_t)}{\delta g(a,z)}(\kappa_H g)(a,z)dadz\end{aligned}$$

they use finite moments $\{m_{1,H},...,m_{N,H};m_{1,L},...,m_{N,L}\}$ to approximate the whole distribution of $g(\cdot,\cdot)$, and use a deep neural network to approximate the nonlinear law of motion $h(\cdot)$ of those moments over time.

In the algorithm, every time we have a guess for the law of motion of those moments $h_n(\cdot)$. Then we solve the HJB given $h_n(\cdot)$ and simulate time series data for the moments $\{m_{1,H},...,m_{N,H};m_{1,L},...,m_{N,L}\}$. With the simulated time series, we learn a new law of motion $h_{n+1}(\cdot)$ until it converges in some sense.

5 Course Summary