Lecture 2
Value Function Iteration with Discretization

Peifan Wu

UBC

Stochastic Growth Model

Consider the following problem:

$$V(z,k) = \max_{c,k,l} u(c) + v(1-l) + \beta \mathbb{E} V(z',k')$$

s.t.

$$c + k' = \exp(z) f(k.l) + (1-\delta) k$$

$$z' = \rho z + \epsilon'$$

$$c, k' \geqslant 0$$

$$l \in (0,1)$$

$$\epsilon \sim \mathcal{N}(0,\sigma^2)$$

We want to solve policy functions $c(z,k), k(z,k), l(z,k)$

We solve them by

- Discretization of the state space (Z,K)
- Value Function Iteration

Discretize AR(1) Process

Two dominant methodologies:

1. Tauchen method (Tauchen, EL 1986)
   - Not restricted to normal innovations
2. Rouwenhorst method (Kopecky-Suen, RED 2010)
   - Works better as $\rho \to 1$
   - Replicate exactly unconditional mean, variance, and first-order autocorrelation
   - Multivariate version: Terry-Knotek (EL, 2011), Galindev-Lkhagvasuren (JEDC, 2010)

Tauchen Method

AR(1) process $y' = \rho y + \epsilon$, $\epsilon \sim \mathcal{N}\left(0, \sigma_\epsilon^2\right)$

Choice of Points: The maximum value y_N should be multiple m (e.g. $m = 3$) of the unconditional standard deviation,

$$y_N = m\sqrt{\frac{\sigma_\epsilon^2}{1-\rho^2}}$$

Let $y_1 = -y_N$ and $\{y_2, y_3, \ldots, y_{N-1}\}$ equally spaced.

Transition Probabilities: $d = y_i - y_{i-1}$, π_{ij} is the transition probability from i to j,

$$\pi_{i1} = F\left(\frac{y_1 + d/2 - \rho y_i}{\sigma_\epsilon}\right)$$

$$\pi_{ij} = F\left(\frac{y_j + d/2 - \rho y_i}{\sigma_\epsilon}\right) - F\left(\frac{y_j - d/2 - \rho y_i}{\sigma_\epsilon}\right)$$

$$\pi_{iN} = 1 - F\left(\frac{y_N - d/2 - \rho y_i}{\sigma_\epsilon}\right)$$

Rouwenhorst Method

AR(1) process $y' = \rho y + \epsilon$, $\epsilon \sim \mathcal{N}\left(0, \sigma_\epsilon^2\right)$

Choice of Points: Let $y_1 = -y_N$ and $\{y_2, y_3, \ldots, y_{N-1}\}$ equally spaced,

$$y_N = \sqrt{\frac{\sigma_\epsilon^2}{1-\rho^2}}\sqrt{N-1}$$

Transition Matrix: Let $p = \frac{1+\rho}{2}$

- For $N = 2$,

$$\Pi_2 = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}$$

- For $N \geqslant 3$,

$$\Pi_N = p\begin{bmatrix} \Pi_{N-1} & \mathbf{0} \\ \mathbf{0}' & 0 \end{bmatrix} + p\begin{bmatrix} 0 & \mathbf{0}' \\ \mathbf{0} & \Pi_{N-1} \end{bmatrix} + (1-p)\begin{bmatrix} \mathbf{0}' & 0 \\ \Pi_{N-1} & \mathbf{0} \end{bmatrix} + (1-p)\begin{bmatrix} \mathbf{0} & \Pi_{N-1} \\ 0 & \mathbf{0}' \end{bmatrix}$$

Discretize Capital Grid

- Evenly spaced grid of N points:

$$k_i = k_1 + (i-1)\,\eta \text{ where } \eta = \frac{k_N - k_1}{N-1}$$

- In general, more points where policy functions have curvature.
- A grid denser close to k_1:

$$k_i = k_1 + \exp\left[(i-1)\,\eta\right] \text{ where } \eta = \frac{\log\left(k_N - k_1\right)}{N-1}$$

- How do we choose k_1, k_N? Solve the steady state k^*, choose $k_1 < k^* < k_N$
- Grid size depends on the size and persistence of z. Keep k_1 away from zero.

Choice of Leisure

- At every (k_i, k_j, z_s) corresponding to a grid point for the variables (k, k', z) we have the intratemporal FOC:

$$u_c\left(\exp(z) f(k,l) + (1-\delta) k_i - k_j\right) \cdot \exp(z) f_l(k,l) = -v_l(1-l)$$

- RHS is increasing in l and the LFS decreasing in l
- $l = 0$ ruled out – Inada condition on f; $l = 1$ ruled out – Inada condition on v;
- Use bisection method to solve for l
- Call the solution $l(k_i, k_j, z_s)$

Algorithm

1. Define a 3-dimensional array R of size N, N, S with typical element (k_i, k_j, z_s) containing the return function
$$R\left(k_i, k_j, z_s\right) = u\left(\exp\left(z_s\right) f\left(k_i\right) + \left(1-\delta\right) k_i - k_j\right) + v\left(1 - l\left(k_i, k_j, z_s\right)\right)$$
Check whether the argument of u at point (i, j, s) is negative: if so, set $R\left(k_i, k_j, z_s\right)$ to a very large negative number
2. Start with an initial guess of the value function matrix V^0. Either the null array, or
$$V^0\left(k_i, z_s\right) = \frac{u\left(\exp\left(z_s\right) f\left(k_i, l^*\right) - \delta k_i\right) + v\left(1 - l^*\right)}{1-\beta}$$
3. Denote the number of iterations by t. Update value function by selecting
$$V^{t+1}\left(k_i, z_s\right) = \max_j \left\{ R\left(k_i, k_j, z_s\right) + \beta \sum_{s'=1}^{S} \Pi\left(z_s, z_s'\right) V^t\left(k_j, z_s'\right) \right\}$$
4. Store the $\arg\max$ (Optional: Howard improvement step)
5. Check convergence: report success if $||V^{t+1} - V^t|| < \epsilon$ otherwise go back to 3.

Additional Checks

1. Check that the policy function isn't constrained by the discrete state space.
   - Relax the bounds of the grid for k, redo the value function iteration if constrained.

2. Check that the error tolerance is small enough.
   - If a small reduction in ϵ results in large changes in the value of policy functions, the tolerance is too high.
   - Reduce ϵ until the solutions are insensitive to further reductions.

3. Check that grid size is dense enough.
   - The grid is too sparse if an increase in grid density results in a different solution.
   - Keep increasing grid size until the solutions are insensitive to further increases.

Discretization is considerably low, but it's the most robust method.

Howard Improvement Step

- Updating policy function is the most time-consuming.

- Idea: Update the value function without updating the policy function occasionally.
- The policy function tends to converge faster than the value function.

- Implementation: Choose H, iterate value function H times using the existing policy function without updating.
- Too high H may result in a value function moving further from the true one. A good idea is to increase H after each iteration, or use the Howard algorithm only after a few steps of the value function iteration.