PROJECT DYNAMIC PROGRAMMING COURSE 2024

Consider an agent who allocates her time between producing the consumption good C, and accumulating human capital H. Normalizing the labor supply of the agent to one, the accumulation law for human capital is given by:

$$\mathrm{H}_{t+1} = (1-\delta)\mathrm{H}_t + (1-L_t) \ with \ L_t \in [0,1],$$

where δ is the depreciation rate of human capital, while L_t is the share of the labor supply dedicated to the production of the consumption good. The production function reads

$$C_t = \mathrm{H}_t^{\alpha} L_t \ with \ \alpha \in (0,1).$$

The agent seeks to maximize her discounted utility, as captured by the following objective:

$$\max_{\{\mathrm{C_t}\}_{\mathrm{t}=0}^{\infty}} \sum_{\mathrm{t}=0}^{\infty} \beta^{\mathrm{t}} \mathrm{U}(\mathrm{C_t})$$

$$\begin{aligned} s.t. \ \ & (i)\ C_t = \mathrm{H}_t^{\alpha} L_t, \\ & (ii)\ \mathrm{H}_t = (1-\delta)\mathrm{H}_{t\text{-}1} + (1-L_{t\text{-}1}), \\ & (\mathrm{iii})\ \mathrm{L_t} \in [0,1]. \end{aligned}$$

1. Assuming that the agent's utility function is CRRA, so that $\mathrm{U(C)} = \mathrm{C}^{1\text{-}\sigma}/(1\text{-}\sigma)$, write a Python script that approximates her value function. You may use the code provided in the course on optimal growth as a template, or alternatively, write your own code.
2. Generate graphics illustrating the value function and policy function, as well as the values of consumption and L as a function of the stock of human capital H. Discuss your results. You can use the following parameter values for your simulations: $\sigma = .9,\ \beta = .9,\ \alpha = .4,\ \delta = .05$.
3. Write a Python script that approximates the value function associated to any *arbitrary* policy function.
4. Combine the script in 3 with a greedy procedure to update the policy function. Verify that iterating greedy updates of the policy function until convergence yields the same value function as the one derived in question 1.

Optional question:

Write a Python script using PyTorch to calibrate a neural network that approximates the path starting from an arbitrary initial stock of human capital and converging toward the steady state.

To calibrate your neural network, minimize the distance between the simulated and theoretical values of (i) the Euler equation, (ii) the accumulation law for capital, (iii) the initial stock of human capital.

When evaluating the Euler equation, ensure that the slackness condition is accounted for, as the boundary constraints ($\mathrm{L_t} \geq 0$ and $\mathrm{L_t} \leq 1$) may bind in certain states.