Markov Processes and Recursive Representation

Diego Ascarza

RIEF

Introduction

- Can we formulate the neoclassical growth model with uncertainty using the recursive language? what are the state variables in this case?
- Individual and aggregate capital.
- History of the shocks?
- In general, the probability distribution for the shock of next period depends on all the history of past and current realizations ... unless the stochastic variable follows a Markov process.

First Order Markov Process

- A first-order Markovian stochastic process satisfies:

$$Prob(z_{t+1} \backslash z^t) = Prob(z_{t+1} \backslash z_t)$$

  An i.i.d shock can be seen as a special case of a Markovian process.

- In several applications, we will use a discrete first-order Markovian process, characterized by the state space:

$$Z = (Z_1, ..., Z_q)$$

First Order Markov Process

And the stationary transition matrix:

$$\Pi = \begin{bmatrix} \pi(Z_1, Z_1) & \pi(Z_1, Z_2) & .... & \pi(Z_1, Z_q) \\ \pi(Z_2, Z_1) & \pi(Z_2, Z_2) & .... & \pi(Z_2, Z_q) \\ .... & .... & .... & .... \\ \pi(Z_q, Z_1) & \pi(Z_q, Z_2) & .... & \pi(Z_q, Z_q) \end{bmatrix}$$

where

$$\pi(Z_i, Z_j) = Prob(z_{t+1} = Z_j \backslash z_t = Z_i)$$

and $\sum_{j=1}^{q} \pi(Z_i, Z_j) = 1$

First Order Markov Process

- Defining the probability distribution (Inconditional):

$$\pi_t = \begin{bmatrix} \pi_{1t} \\ \pi_{2t} \\ .... \\ \pi_{qt} \end{bmatrix} = \begin{bmatrix} Prob(z_t = Z_1) \\ Prob(z_t = Z_2) \\ .... \\ Prob(z_t = Z_q) \end{bmatrix}$$

The vector π_t evolves according to:

$$\pi_{t+1}^T = \pi_t^T x \Pi$$

A probability distribution π^* is invariant if ${\pi^*}^T = {\pi^*}^T x \Pi$

First Order Markov Process

- A Markov Process is asymptotically stationary if, departing from any π_0, $\text{lím}_{t\to\infty} \pi_t = \pi^*$.
- **Result:** If all the elements of Π are strictly positive, the correspondent process is asymptotically stationary and converges to a unique invariant distribution.

Continuous case

- In other cases we will work with continuous Markovian processes, like an AR(1):

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

$$\epsilon' N(\mu, \sigma^2)$$

where ϵ is a random variable, normally distributed with zero mean and variance σ^2. If $-1 < \rho < 1$ this process is stationary and its moments are constant over time.

Stochastic Recursive Competitive Equilibrium

A stochastic recursive competitive equilibrium is a set of functions $v(k, K, z)$, $c(k, K, z)$, $i(k, K, z)$ and $k'(k, K, z)$, prices $w(K, z)$ and $r(K, z)$ and law of motion $\Gamma(K, z)$ such that:

- For each threesome (k, K, z), given the functions w, r, Γ, the value function $v(k, K, z)$ solves the Bellman's equation of the consumer:

$$v(k, K, z) = max_{c,i,k'} \left\{ u(c) + \beta E_z v(k', K', z') \right\}$$

$$s.t.$$

$$c + i = w(K, z) + r(K, z)k$$

$$k' = (1 - \delta)k + i$$

$$K' = \Gamma(K, z)$$

and $c(k, K, z)$, $k'(k, K, z)$, $i(k, K, z)$ are optimal decision rules for this problem.

Stochastic Recursive Competitive Equilibrium

- For each pair (K, z), prices satisfy the marginal conditions:

$$r(K, z) = e^z f'(K)$$
$$w(K, z) = e^z f(K) - e^z f'(K)K$$

- For each pair (K, z), markets clear:

$$e^z f(K, z) = c(K, K, z) + i(K, K, z)$$

- For each (K, z), the aggregate law of motion of capital is consistent with the behavior of agents:

$$\Gamma(K, z) = k'(K, K, z)$$

Expected Value Function

The expected value function of next period is defined as:

- If z follows a discrete first-order Markovian process with state space $Z = (Z_1, ..., Z_q)$ and transition matrix Π, then:

$$\mathbb{E}_z v(k', K', z') = \sum_{j=1}^{q} \pi(z, Z_j) v(k', K', Z_j)$$

- If z follows an AR(1) with conditional density $g(z' \backslash z)$, then:

$$E_z v(k', K', z') = \int_Z v(k', K', z') g(z' \backslash z) dz'$$

Social Planner's Problem

A benevolent social planner chooses functions $v(k,z)$, $c(k,z)$, $i(k,z)$, $k'(k,z)$ that solve the Bellman's equation:

$$v(k,z) = max_{c,i,k'} \left\{ u(c) + \beta \mathbb{E}_z v(k', z') \right\}$$

$$s.t.$$

$$c + i = e^z f(k)$$
$$k' = (1-\delta)k + i$$

As before, the welfare theorems hold and the solution to this problem is equivalent to the one that is obtained in the competitive equilibrium.

Notice that the correspondent contingent plans can be found using the optimal decision rule: $k_{t+1}(z^t) = k'(k'(k'(...), z_{t-1}), z_t)$

Stochastic Dynamic Programming

Consider the following Bellman equation:

$$v(x,z) = max_y \left\{ F(x,z,y) + \beta E_z v(y,z') \right\}$$

$$s.t.$$

$$y \in \Omega(x,z)$$

where z follows a first-order Markovian process.

The results of existence, uniqueness and contraction still hold under the same conditions for X, F, Ω, β plus some extra technical assumptions about the stochastic process (Chapter 9).

These extra assumptions are satisfied automatically with discrete Markovian processes.

Value Function Iteration

- To implement numerically this method, we assume that the technological shock z follows a first-order Markovian process with state space $Z = (Z_1, ..., Z_q)$ and transition matrix Π.
- Using the contraction mapping theorem, if we depart from a function v^0, the sequence v^n defined by:

$$v^{n+1}(k,z) = max_{k'} \left\{ u[e^z f(k) + (1-\delta)k - k'] + \beta \sum_{j=1}^{q} \pi(z, Z_j) v^n(k', Z_j \right.$$

$$s.t.$$

$$k' \in [0, e^z f(k) + (1-\delta)k]$$

converges to the solution of the social planner if $n \to \infty$.

Value Function Iteration

Initial set-up:

- Define a grid of capital $K = (K_1, K_2, ...K_p)$ for the capital k.
- Define a state matrix S as follows:

$$S = \begin{bmatrix} (K_1, Z_1) & (K_2, Z_1) & .... & (K_p, Z_1) \\ (K_1, Z_2) & (K_2, Z_2) & .... & (K_p, Z_2) \\ .... & .... & .... & .... \\ K_1, Z_q) & (K_2, Z_q) & .... & (K_p, Z_q) \end{bmatrix}$$

Value Function Iteration

- Define the operator Φ:

$$\Phi(S) = \begin{bmatrix} (K_1, Z_1) \\ .... \\ (K_1, Z_q) \\ (K_2, Z_1) \\ .... \\ (K_p, Z_q) \end{bmatrix} \quad \Phi^{-1}\left(\begin{bmatrix} (K_1, Z_1) \\ .... \\ (K_1, Z_q) \\ (K_2, Z_1) \\ .... \\ (K_p, Z_q) \end{bmatrix} \right) = S$$

In Matlab, $\Phi(A)$ is written as $A(:)$; while $\Phi^{-1}(B)$ is written as $reshape(B, q, p)$.

Value Function Iteration

- Build the Matrix M of pqxp as:

$$M = \begin{bmatrix} F(S_{11}, K_1) & F(S_{11}, K_2) & .... & F(S_{11}, K_p) \\ .... & .... & .... & .... \\ F(S_{1q}, K_1) & F(S_{1q}, K_2) & .... & F(S_{1q}, K_p) \\ F(S_{21}, K_1) & F(S_{21}, K_2) & .... & F(S_{21}, K_p) \\ .... & .... & .... & .... \\ F(S_{pq}, K_1) & F(S_{pq}, K_2) & .... & F(S_{pq}, K_p) \end{bmatrix}$$

where:

$$F(S_{ij}, K_l) = F(K_i, K_j, K_l) = u[e^{Z_j} f(K_i) + (1 - \delta)K_i - K_l]$$

M stores the return function evaluated at each possible combination $[(k, z), k']$ in the grid.

Value Function Iteration

- Drop the non-feasible entries by doing:

$$F(S_{ij}, K_l) \equiv F(K_i, Z_j, K_l) = -100000$$
$$\text{if} \quad K_l > e^{Z_j} f(K_i) + (1-\delta)K_i$$

As before we are preventing the algorithm to choose not feasible choices for the planner.

- Build the auxiliar matrix E of pqxq as:

$$E = \begin{bmatrix} I_q \\ I_q \\ .... \\ I_q \end{bmatrix} \quad I_q \text{ is the identity of qxq}$$

Value Function Iteration-Algorithm

1. Propose an initial column vector $V^0 in \mathbb{R}^{pq}$ (For example, $V^0 = 0$) and initialize $s = 0$.

2. Given V^s and M, compute V^{s+1} as:

$$V^{s+1} = max \left\{ M + \beta[Ex(\Pi x \Phi^{-1}(V^s))] \right\}$$

   where the maximum is calculated by rows.

Calculate the distance between V^{s+1} and V^s. If the distance is greater than the tolerance criteria, go to step 2 with $s = s + 1$. Otherwise, the algorithm converges.

Value Function Iteration-Algorithm

- As before, we obtain an approximation to the value function of the Social Planner v at every point of the states of the grid.
- Together with the correspondent decision rule G:

$$G = argmax\{M + \beta[Ex(\Pi x \Phi^{-1}(V))]\}$$

  A column vector of pq components, where $G_i \in \{1, .., p\}$ indicates the number of the column that maximizes the row i.
- Results convenient to work with the reordered decision rule as a matrix: $\hat{G} = \Phi^{-1}(G)$.

.

Optimal Path Simulation

- The optimal trajectory of capital depends on the history of the realizations for z.
- To obtain a particular time series $k_0, k_1, ...k_T$ departing from (k_0, z_0) we need first to simulate the history $z_0, z_1, ..., z_T$ using a random numbers generator and the information of the transition matrix Π.
- Given $z_0 = Z_i$, extract z_1 from Z assigning the probability π_{ij} to the state Z_j.
- Recursively, given $z_n = Z_i$ extract z_{n+1} of Z assigning the probability π_{ij} to the state Z_j.
- Continue until you get the history $z_0, z_1, ..., z_T$.

Optimal Path Simulation

- One the history of the shocks is known, we calculate recursively the optimal path for the capital $(k_0, k_1, ..., k_T)$ using the decision rule G.

$$\begin{array}{l} k_t = K_j \\ z_t = Z_i \end{array} \rightarrow k_{t+1} = K_{\hat{G}_{ij}}$$

  And then we calculate the optimal path for the other variables. This paths represent time series and their moments can be calculated (mean, variance, correlations between them). In this way, we can compare the statistical moments of the model with the data.

- However, these time series depend on a particular realization of the shocks.

- Therefore, it is recommendable to make a large simulation ($T = 10000$) leaving the first 1000 periods so we can approximate the statistics of the invariant distribution (LLN).