316-406 ADVANCED MACROECONOMIC TECHNIQUES NOTE 5c

Chris Edmond ⟨cpedmond@unimelb.edu.au⟩

Introduction to numerical dynamic programming

We'll now turn to simple numerical methods for solving dynamic programming problems. The main method we will be interested in is based on **value function iteration**.

So, consider the stochastic growth model with Bellman equation

$$V(k,z) = \max_{k' \geq 0} \{U(c) + \beta \mathsf{E}[V(k',z')|z]\}$$

where the maximization is subject to the resource constraint

$$c + k' \leq zf(k) + (1-\delta)k$$

and the technology shock z follows an m-state Markov chain with transition matrix P and typical element

$$p_{ij} = \Pr(z' = z_j | z = z_i), \qquad i,j = 1,...,m$$

Assume that the initial distribution is degenerate so that the initial technology shock z_0 is known. Write the support of the Markov chain as

$$z \in \mathcal{Z} = (z_1 < z_2 < \cdots < z_m)$$

The vector $\mathcal{Z}$ has m elements and is ordered.

A. Discrete-state methods

Suppose that the capital stock k is constrained to belong to a discrete set of values, i.e., a **grid**

$$k \in \mathcal{K} = (k_1 < k_2 < \cdots < k_n)$$

The vector $\mathcal{K}$ has n elements and is ordered. We will discuss how the discretization is done later. For now, just note that we have a vector of possible choices for the capital stock. Again, consider the Bellman equation

$$V(k,z) = \max_{k' \geq 0} \{U[zf(k) + (1-\delta)k - k'] + \beta \mathsf{E}[V(k',z')|z]\}$$

If the current period's state is some $(k_h, z_i) \in \mathcal{K} \times \mathcal{Z}$, then this is equivalent to

$$V(k_h, z_i) = \max_{k' \in \mathcal{K}} \left\{ U[z_i f(k_h) + (1-\delta)k_h - k'] + \beta \sum_{j=1}^{m} p_{ij} V(k', z_j) \right\}$$

Notice that $V(k_h, z_i)$ is just a table (i.e., a matrix) that gives the values associated with each point in the state space $\mathcal{K} \times \mathcal{Z}$.

In what follows, I will suppose that the Markov chain has $m = 2$ elements. The generalization to bigger m is obvious, I simply want to make the exposition below as straightforward as possible. I don't want to drown in sub- and super-scripts.

Define two vectors V_1 and V_2 with typical elements

$$\begin{aligned} V_1(h) &\equiv V(k_h, z_1) \\ V_2(h) &\equiv V(k_h, z_2) \end{aligned}$$

Each of these vectors is $n \times 1$. Similarly, define two "reward" matrices, R_1 and R_2 with typical elements

$$\begin{aligned} R_1(i,h) &\equiv U[z_1 f(k_i) + (1-\delta)k_i - k_h] \\ R_2(i,h) &\equiv U[z_2 f(k_i) + (1-\delta)k_i - k_h] \end{aligned}$$

For each shock we have a square matrix $n \times n$ that computes for each capital stock k_i (row) all the possible period utilities associated with possible choices k_h (column). So we say, for example, $R_1(i,h)$ is the period reward associated with choosing capital stock k_h for tomorrow given that today's capital stock is k_i and the technology shock is z_1.

Stack the vectors V_1, V_2 into a matrix $V = [V_1, V_2]$ and define an operation T that associates vectors $[V_1, V_2]$, with new vectors $TV = [TV_1, TV_2]$. To do this, we represent the right hand side of the Bellman equation for each shock z_i by

$$\begin{aligned} TV_1 &= \max \left\{ R_1 + \beta p_{11} \mathbf{1} V_1' + \beta p_{12} \mathbf{1} V_2' \right\} \\ TV_2 &= \max \left\{ R_2 + \beta p_{21} \mathbf{1} V_1' + \beta p_{22} \mathbf{1} V_2' \right\} \end{aligned}$$

In a slight abuse of our previous notation, the prime ($'$) here indicates vector transposition. The $\mathbf{1}$ indicates an $n \times 1$ vector of ones.

Notice that the term inside the braces is an $n \times n$ matrix. When Matlab applies the max operator to an $n \times n$ matrix, it returns a $1 \times n$ row vector whose elements are the maximums of each column of the matrix. For example,

$$[u, t] = \max \begin{pmatrix} 0 & 1 & 9 \\ 1 & 3 & 7 \\ 2 & 2 & 0 \end{pmatrix}$$

in Matlab produces a 1×3 row vector u whose elements are the maximums of each column of the matrix

$$u = \begin{pmatrix} 2 & 3 & 9 \end{pmatrix}$$

and a $1 \times n$ row vector t whose elements tell you which row of the matrix attains the maximum in that column

$$t = \begin{pmatrix} 3 & 2 & 1 \end{pmatrix}$$

Begin the iteration with a guess, say $V^0 = [V_1^0, V_2^0] = 0$ and apply the matrix operation T to get $V^1 = TV^0$. (I use super-scripts here to distinguish between, say, the iterate matrix V^1 and the vector V_1 that is the first column of some arbitrary V). If

$$\max \left\{ \max\{ \left| V^0 - TV^0 \right| \right\} \} < \varepsilon$$

for some pre-specified tolerance level ε, we stop. Otherwise, we iterate for $l = 0, 1, ...$ on $V^l \mapsto TV^l = V^{l+1}$ until we have satisfied the convergence criterion,

$$\max \left\{ \max\{ \left| V^l - TV^l \right| \right\} \} < \varepsilon$$

Typically, this criterion is a number like $\varepsilon = 10^{-6}$.

B. Policy functions

Once we have solved the value function iteration problem, we can back out other objects of interest. For example, once we have the solution to the fixed point problem, call it $V = [V_1, V_2]$, we can compute policy functions $g(k_h, z_i)$ for each $(k_h, z_i) \in \mathcal{K} \times \mathcal{Z}$.

We are interested in vectors g_1 and g_2 with typical elements of the form

$$\begin{aligned} g_1(h) &\equiv g(k_h, z_1) \\ g_2(h) &\equiv g(k_h, z_2) \end{aligned}$$

These are computed from the right hand side of the Bellman equation

$$\begin{aligned} [u_1, t_1] &= \max\left(R_1 + \beta p_{11}\mathbf{1}V_1' + \beta p_{12}\mathbf{1}V_2'\right) \\ [u_2, t_2] &= \max\left(R_2 + \beta p_{21}\mathbf{1}V_1' + \beta p_{22}\mathbf{1}V_2'\right) \end{aligned}$$

Then the policy functions are

$$\begin{aligned} g_1(h) &= \mathcal{K}(t_1(h)) \\ g_2(h) &= \mathcal{K}(t_2(h)) \end{aligned}$$

for $h = 1, ..., n$. That is, the policy functions g_i are built up by evaluating the grid of capital stocks $\mathcal{K}$ at the levels of the capital stock t_i that attain the maximum.

Chris Edmond

5 September 2004