PhD Macro Core Part I:
Lecture 5 – Dynamic Programming I

Min Fang
University of Florida

Fall 2024

Today

- Formal introduction to deterministic dynamic programming
  - Recursive approach to the growth model
  - Key concepts: value function, Bellman equation, Euler equation, etc.

Sequence Problem

- We are now familiar with the following sequence problem:

$$\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t), \quad 0 < \beta < 1$$

  subject to the sequence of constraints,

$$c_t \geqslant 0, \text{ and } c_t + k_{t+1} \leqslant f(k_t) = F(k_t, 1) + (1 - \delta)k_t$$

  with the initial condition $k_0 > 0$ given.

- Unless stated otherwise, assume $u(c)$ and $f(k)$ strictly increasing and strictly concave.

Value Function

- Let $v(k_0)$ denote the maximized objective function:

$$v(k_0) \equiv \max_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(f(k_t) - k_{t+1}), \quad 0 < \beta < 1$$

- This is known as a value function

- It is the value to the planner of being endowed with $k_0 > 0$ and then proceeding optimally

- The Key is that the value function has a simple recursive structure

Recursive Structure of the Value Function

- To see this recursive structure, let's break the sum up

$$
\begin{aligned}
v\left(k_0\right) &\equiv \max_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u\left(f\left(k_t\right)-k_{t+1}\right) \\
&= \max_{\{k_{t+1}\}_{t=0}^{\infty}} \left[u\left(f\left(k_0\right)-k_1\right)+\beta \sum_{t=1}^{\infty} \beta^{t-1} u\left(f\left(k_t\right)-k_{t+1}\right)\right] \\
&= \max_{k_1} \left[u\left(f\left(k_0\right)-k_1\right)+\beta \max_{\{k_{t+1}\}_{t=1}^{\infty}} \sum_{t=1}^{\infty} \beta^{t-1} u\left(f\left(k_t\right)-k_{t+1}\right)\right] \\
&= \max_{k_1} \left[u\left(f\left(k_0\right)-k_1\right)+\beta v\left(k_1\right)\right]
\end{aligned}
$$

Notation Conventions

- We do not know the value function, but we know it satisfies

$$v(k_0) = \max_{k_1} \left[u\left(f(k_0) - k_1\right) + \beta v(k_1) \right]$$

- Nothing special about periods $t = 0$ and $t = 1$, we could write

$$v(k_t) = \max_{k_{t+1}} \left[u\left(f(k_t) - k_{t+1}\right) + \beta v(k_{t+1}) \right]$$

- But for this stationary problem, there is nothing special about the time periods at all, so

$$v(k) = \max_{x} [u(f(k) - x) + \beta v(x)]$$

  where x is just a dummy variable indexing possible choices of the next period's capital stock

- Current capital stock k is the sole state variable, a sufficient statistic for past decisions

Bellman Equation

- A recursive representation like

$$v(k) = \max_x [u(f(k) - x) + \beta v(x)]$$

  is often called a Bellman equation, after Richard Bellman

- More generally, this is an example of an equation to be solved for an unknown function, i.e., given exogenous $u(c), f(k)$ and β we need to solve for an endogenous function $v(k)$

- Both macro and micro involve solving problems like this.

First Order Condition

- Only consider the RHS of the Bellman equation.
- Treating $v(x)$ as known, this is just like a two-period problem

$$\max_x [u(f(k) - x) + \beta v(x)]$$

- And implies the first order condition

$$u'(f(k) - x) = \beta v'(x)$$

  which we could imagine solving for some $x = g(k)$.

- But we don't know $v(x)$ and hence don't know $v'(x)$

Aside on Parameterized Optimization Problems

- Suppose we seek to maximize $u(x, \theta)$ by choice of x given parameter θ

$$v(\theta) \equiv \max_{x} u(x, \theta)$$

- Let $x = g(\theta)$ achieve the maximum

$$g(\theta) \equiv \operatorname*{argmax}_{x} u(x, \theta)$$

  so that

$$v(\theta) = u(g(\theta), \theta)$$

- What do we know about $v(\theta)$? What about $v'(\theta)$?

Envelope Theorem

- Suppose further that $x = g(\theta)$ is characterized by the first order condition:

$$\frac{\partial u(g(\theta),\theta)}{\partial x} = 0$$

- Then we have the envelope theorem:

$$v'(\theta) = \frac{\partial u(g(\theta),\theta)}{\partial x} g'(\theta) + \frac{\partial u(g(\theta),\theta)}{\partial \theta} = \frac{\partial u(g(\theta),\theta)}{\partial \theta}$$

- The total derivative of the value function with respect to θ is given by the partial derivative of the objective function with respect to θ evaluated at the optimum.

Envelope Condition

- Let's apply this to our dynamic programming problem

$$v(k) = \max_x [u(f(k) - x) + \beta v(x)]$$

- Therefore

$$\begin{aligned} v'(k) &= \frac{\partial}{\partial k}[u(f(k) - x) + \beta v(x)], \quad x = g(k) \\ &= u'(f(k) - x) f'(k), \quad x = g(k) \\ &= u'(f(k) - g(k)) f'(k) \end{aligned}$$

Policy Function

- In dynamic programming problems like this, the function

$$g(k) \equiv \underset{x}{\operatorname{argmax}}[u(f(k) - x) + \beta v(x)]$$

  is known as the policy function or decision rule

- Iterating on the policy function gives the sequence of capital stocks

$$\begin{aligned} k_1 &= g(k_0) \\ k_2 &= g(k_1) = g(g(k_0)) \\ &\vdots \\ k_{t+1} &= g(k_t) = g^t(k_0) \end{aligned}$$

- Properties of $g(k)$ determine the properties of the optimal sequence of k_t. Steady states satisfy $k^* = g(k^*)$. A steady state is locally stable if $|g'(k^*)| < 1$, and so on

Policy Function

- With the policy function $g(k)$, can then recover consumption

$$c(k) = f(k) - g(k)$$

- And $k' \equiv g(k)$ can be a simple notation of next period capital

- Notice that this $c(k)$ is the same as

$$c(k) \equiv \underset{c}{\operatorname{argmax}}[u(c) + \beta v(f(k) - c)]$$

Combining First Order and Envelope Conditions

- To summarize, we have the problem

$$v(k) = \max_x [u(f(k) - x) + \beta v(x)]$$

- The first order condition for this problem is

$$u'(f(k) - g(k)) = \beta v'(g(k))$$

- The envelope condition says

$$v'(k) = u'(f(k) - g(k)) f'(k)$$

- Evaluating this at $g(k)$ gives the somewhat cumbersome

$$v'(g(k)) = u'(f(g(k)) - g(g(k))) f'(g(k))$$

Euler Equation

- Hence, we can write the Euler equation

$$u'(f(k) - g(k)) = \beta u'(f(g(k)) - g(g(k)))f'(g(k))$$

- This Euler equation can also be viewed as a functional equation, to be solved for the policy function $g(k)$

- Using $k_{t+1} = g(k_t)$ and $k_{t+2} = g(g(k_t))$ etc, in sequence notation this is just the usual condition

$$u'(f(k_t) - k_{t+1}) = \beta u'(f(k_{t+1}) - k_{t+2}) f'(k_{t+1})$$

Dynamical Systems Compared

- Policy function from the Bellman equation problem:

$$k_{t+1} = g(k_t), \quad k_0 > 0 \text{ given}$$

  This is a one-dimensional dynamical system in k_t

- From the Euler equation:

$$u'(f(k_t) - k_{t+1}) = \beta u'(f(k_{t+1}) - k_{t+2}) f'(k_{t+1}), \quad k_0 > 0 \text{ given}$$

  This is a two-dimensional dynamical system in k_t

- Why the difference?

Bellman Equations vs. Euler Equations

- For a given problem, can either

  (i) attempt to solve for value function $v(k)$ from Bellman equation and then determine policy function $g(k)$, or

  (ii) attempt to solve for policy function $g(k)$ from Euler equation

- Solving Euler equations is generally faster than solving Bellman equations, so when the problem is 'well-behaved', (ii) is often preferable

- Solving Bellman equations, while slower, is generally more robust

Method of Successive Approximations

- Suppose we had some candidate value function $v_0(k)$

- Define a new value function by

$$v_1(k) = \max_x \left[u(f(k) - x) + \beta v_0(x)\right]$$

  and test whether $v_1(k)$ equals $v_0(k)$ or not

- Unless we are lucky, $v_1(k) \neq v_0(k)$. But suppose we keep iterating

$$v_{n+1}(k) = \max_x \left[u(f(k) - x) + \beta v_n(x)\right], \quad n = 0, 1, \ldots$$

- What happens to the sequence of functions v_n as $n \to \infty$?

A Suboptimal Policy

- Suppose we followed any feasible policy $g_0(k)$
- Let $v_0(k)$ be the value of this generally suboptimal policy

$$v_0(k_0) = \sum_{t=0}^{\infty} \beta^t u\left(f(k_t) - g_0(k_t)\right)$$

such that, for arbitrary k,

$$v_0(k) = u\left(f(k) - g_0(k)\right) + \beta v_0\left(g_0(k)\right)$$

A Suboptimal Policy

- Then we have

$$\begin{aligned} v_1(k) &= \max_x \left[u(f(k) - x) + \beta v_0(x)\right] \\ &\geqslant \left[u\left(f(k) - g_0(k)\right) + \beta v_0\left(g_0(k)\right)\right] \\ &= v_0(k) \end{aligned}$$

  (can do no worse by choosing optimally today)

- Likewise

$$\begin{aligned} v_2(k) &= \max_x \left[u(f(k) - x) + \beta v_1(x)\right] \\ &\geqslant \max_x \left[u(f(k) - x) + \beta v_0(x)\right] \\ &= v_1(k) \end{aligned}$$

- Continuing in this way, $v_{n+1}(k) \geqslant v_n(k)$ for $n = 0, 1, \ldots$

Bellman operator

- Let Tv denote the function created by the RHS of the Bellman equation

$$Tv(k) \equiv \max_x [u(f(k) - x) + \beta v(x)]$$

- T is an operator that takes as an input a function v and returns a new function Tv

- In this notation

$$Tv_n(k) \equiv \max_x \, [u(f(k) - x) + \beta v_n(x)]$$

- Can iterate on Bellman operator to get $v_{n+1} = Tv_n$ for $n = 0, 1, \ldots$

Iterating on Bellman Operator

- Notice that solving the Bellman equation is equivalent to solving the fixed point problem

$$v = Tv$$

- We have seen that

$$v_{n+1} = Tv_n \geqslant v_n$$

- Does this increasing sequence v_n converge to a limit v as $n \to \infty$?

Next

- Sketch of mathematical background:
  - Contraction mapping theorem
  - Blackwell's sufficient conditions (for a contraction)

References

- [SLR] Sargent and Ljungqvist, Recursive Macroeconomic Theory (4th)
  - Chapter 3: Dynamic Programming
- [DK] Dirk Krueger, Macroeconomic Theory (2015)
  - Chapter 5: Dynamic Programming
- [AKMM] Azzimonti, Krusell, McKay, and Mukoyama, Macroeconomics
  - Chapter 4: Dynamic Optimization