Chapter 11. Dynamic Programming

Xiaoxiao Hu

May 17, 2022

11.A. Life-cycle saving problem revisited

- Consider life-cycle saving problem
- Assume

1. wage w_t is 0, i.e., $w_t = 0$ for all t;
2. interest rate is 0, i.e., $r_t = 0$ for all t;
3. utility function takes the form $u(c) = \ln(c)$;
4. no discouting, i.e., $\beta = 1$;
5. terminal stock $k_{T+1} = 0$.

Life-cycle saving problem (finite-horizon)

- Time is discrete: $t = 0, 1, 2, ..., T$
- Decision is on how much of income to spend on consumption in each period.
- Unspent income is saved and overspent income is on debt.
- $c_t \geq 0$: consumption in period t and
- k_{t+1}: accumulated savings or debts at the beginning of period $t + 1$.

Life-cycle saving problem (finite-horizon)

- Budget constraint in period t is

$$c_t + k_{t+1} = k_t$$

- $k_0 > 0$ given.
- $k_{T+1} = 0$ imposed.

Life-cycle saving problem (finite-horizon)

- Individual only derives utility from consumption and
- chooses consumption path to maximize total value of utilities in period $t = 0$:

$$U(c_0, c_1, ..., c_T) = \sum_{t=0}^{T} \ln(c_t).$$

The maximization principle

- Maximization problem is:

$$\max_{\substack{c_0, c_1, \dots, c_T \\ k_1, k_2, \dots, k_T}} \sum_{t=0}^{T} \ln(c_t)$$

$$\text{s.t. } c_t + k_{t+1} = k_t \text{ for all } t = 0, ..., T$$

- We could solve it using method in Chapter 10.

The maximization principle

- Define Hamiltonian:

$$H(c_t, k_t, \pi_{t+1}, t) = \ln(c_t) + \pi_{t+1}(-c_t)$$

- FOCs:

$$\frac{\partial H}{\partial c_t} = \frac{1}{c_t} - \pi_{t+1} = 0 \text{ for all } t = 0, ..., T$$

$$\pi_{t+1} - \pi_t = -\frac{\partial H^*}{\partial k_t} = 0 \text{ for all } t = 1, ..., T$$

$$k_{t+1} - k_t = \frac{\partial H^*}{\partial \pi_{t+1}} = -c_t \text{ for all } t = 0, ..., T$$

The maximization principle

1. Euler Equation:

$$c_{t+1} = c_t \text{ for all } t = 0, ..., T \implies c_t = c_0 \text{ for all } t = 0, ..., T$$

2. From constraints, we have $\sum_{t=0}^{T} c_t = k_0$.
3. From 1 and 2, solution is $c_t^* = \frac{k_0}{T+1}$ for all $t = 0, ..., T$ and $k_{t+1}^* = \frac{T-t}{T+1} k_0$ for all $t = 0, ..., T-1$.

Define the problem recursively

- Now let us look at the problem from a different angle.
- Define maximum value at $t = 0$ as a function of initial stocks:

$$V_0(k_0) = \max_{\substack{c_0, c_1, ..., c_T \\ k_1, k_2, ..., k_T}} \{u(c_0) + u(c_1) + ... + u(c_T)\}$$

subject to budget constraints for all $t = 0, ..., T$ and terminal condition $k_{T+1} = 0$.

Define the problem recursively

- By previously calculated optimal consumption path

$$c_t^* = \frac{k_0}{T+1}$$

for all $t = 0, ..., T$, we have

$$V_0(k_0) = (T+1)\ln\Big(\frac{k_0}{T+1}\Big).$$

Define the problem recursively

- Given k_1, we could similarly define maximum value at $t = 1$ as a function of k_1:

$$V_1(k_1) = \max_{\substack{c_1,...,c_T \\ k_2,...,k_T}} \{u(c_1) + u(c_2) + ... + u(c_T)\}$$

subject to budget constraints for all $t = 1, ..., T$ and terminal condition $k_{T+1} = 0$.

- We could use maximum principle to solve this problem.
- Maximum value is $V_1(k_1) = T \ln\left(\frac{k_1}{T}\right)$.

Define the problem recursively

- Next, consider a two-period problem:

$$W(k_0) = \max_{c_0, k_1} \{\ln(c_0) + V_1(k_1)\} \text{ s.t. } c_0 + k_1 = k_0$$

- Solving the problem, we have

$$W(k_0) = (T+1) \ln\Big(\frac{k_0}{T+1}\Big) = V_0(k_0)$$

- It suggests:

$$V_0(k_0) = \max_{c_0, k_1} \{\ln(c_0) + V_1(k_1)\} \text{ s.t. } c_0 + k_1 = k_0$$

Define the problem recursively

- Similarly, we could define

$$V_2(k_2) = \max_{\substack{c_2,...,c_T \\ k_3,...,k_T}} \{u(c_2) + u(c_3) + ... + u(c_T)\}$$

subject to budget constraints for all $t = 2, ..., T$ and terminal condition $k_{T+1} = 0$,

- and verify

$$V_1(k_1) = \max_{c_1,k_2} \{\ln(c_1) + V_2(k_2)\} \text{ s.t. } c_1 + k_2 = k_1$$

Define the problem recursively

- This argument works for all $t = 0, ..., T - 1$:

$$V_t(k_t) = \max_{c_t, k_{t+1}} \{\ln(c_t) + V_{t+1}(k_{t+1})\}$$

$$\text{s.t. } c_t + k_{t+1} = k_t$$

- This equation, called Bellman Equation, expresses the value function as a combination of a flow payoff and a (discounted) continuation payoff.

- Such a method of optimization over time as a succession of static programming problems is called Dynamic Programming.

Life-cycle saving problem (infinite-horizon)

- Bellman Equation holds for infinite-horizon problems as well.
- As an example, we consider an infinite-horizon version of this simplified life-cycle saving problem.

Life-cycle saving problem (infinite-horizon)

- For the problem to be well-defined, we need discounting.
- Let discount factor be $\beta \in (0, 1)$.
- So objective function becomes

$$U(c) = \sum_{t=0}^{\infty} \beta^t \ln(c_t).$$

- Budget constraint in period t is still

$$c_t + k_{t+1} = k_t.$$

- $k_0 > 0$ given.

The maximization principle

- Define Hamiltonian:

$$H(c_t, k_t, \pi_t, t) = \beta^t \ln(c_t) + \pi_{t+1}(-c_t)$$

- FOCs:

$$\frac{\partial H}{\partial c_t} = \beta^t \frac{1}{c_t} - \pi_{t+1} = 0 \text{ for all } t = 0, ..., T$$

$$\pi_{t+1} - \pi_t = -\frac{\partial H^*}{\partial k_t} = 0 \text{ for all } t = 1, ..., T$$

$$k_{t+1} - k_t = \frac{\partial H^*}{\partial \pi_{t+1}} = -c_t \text{ for all } t = 0, ..., T$$

- Transversality condition: $\lim_{T \to \infty} \pi_{T+1} k_{t+1} = 0$

The maximization principle

1. Euler Equation:

$$c_{t+1} = \beta c_t \text{ for all } t = 0, ..., T$$

$$\implies c_t = \beta^t c_0 \text{ for all } t = 0, ..., T$$

2. From constraints: $\sum_{t=0}^{\infty} c_t + \lim_{T\to\infty} k_{T+1} = k_0$.

3. $\pi_{t+1} = \beta^t / c_t$ and transversality condition

   - $\lim_{T\to\infty} \frac{\beta^T k_{T+1}}{c_T} = 0$
   - By 1, $c_T = \beta^T c_0$.
   - So, we have $\lim_{T\to\infty} \frac{k_{T+1}}{c_0} = 0 \implies \lim_{T\to\infty} k_{T+1} = 0$.

The maximization principle

4. From 1, 2 and 3,

$$\sum_{t=0}^{\infty} \beta^t c_0 = k_0 \implies c_0 = (1-\beta)k_0$$

- $c_t = \beta^t(1-\beta)k_0$ and $k_{t+1} = \beta^{t+1}k_0$.
- In each period t, $c_t = (1-\beta)k_t$ and $k_{t+1} = \beta k_t$.
- Above equations that express c_t and k_{t+1} as functions of k_t are called policy functions.

Define the problem recursively

Similar to finite-horizon case, we show that Bellman Equation holds:

$$V_t(k_t) = \max_{c_t, k_{t+1}} \{\ln(c_t) + \beta V_{t+1}(k_{t+1})\}$$

$$\text{s.t. } c_t + k_{t+1} = k_t$$

Define the problem recursively

In period t, value function is

$$V_t(k_t) = \max_{\substack{\{c_{t+j}\}_{j=0}^{\infty} \\ \{k_{t+j+1}\}_{j=0}^{\infty}}} \sum_{j=0}^{\infty} \beta^j \ln(c_{t+j})$$

subject to budget constraints

$$c_{t+j} + k_{t+j+1} = k_{t+j} \text{ for all } j \geq 0.$$

- We could drop time subscript t in V_t since functional forms of value functions are same in each period.

Define the problem recursively

Solving it using maximum principle, we have

$$c_{t+j} = \beta^j (1 - \beta) k_t,$$

which gives value function

$$V(k_t) = \frac{\ln(1 - \beta) + \ln(k_t)}{1 - \beta} + \frac{\beta \ln(\beta)}{(1 - \beta)^2}.$$

Define the problem recursively

- Now define

$$W(k_t) = \max_{c_t, k_{t+1}} \{\ln(c_t) + \beta V(k_{t+1})\}$$

$$\text{s.t. } c_t + k_{t+1} = k_t$$

- Solving the problem, we have

$$W(k_t) = \frac{\ln(1-\beta) + \ln(k_t)}{1-\beta} + \frac{\beta \ln(\beta)}{(1-\beta)^2} = V(k_t).$$

- Thus, Bellman Equation holds.

11.B. Dynamic Programming

- We will briefly show that Bellman Equation holds in a general setting.
  - solution to initial problem solves Bellman equation.
  - solution to Bellman Equation is also a solution to initial problem.
- Our discussions will be focused on infinite-horizon discrete-time models.
- In fact, dynamic programming is especially useful for when time is discrete (and there is uncertainty).

Dynamic Programming

We reformulate initial problem into a Sequence Problem.

Definition (Sequence Problem)**.** Sequence problem is of the form:

$$V(x_0) = \sup_{\{x_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t F(x_t, x_{t+1}) \tag{SP}$$

$$\text{s.t. } x_{t+1} \in \Gamma(x_t) \text{ for all } t = 0, 1, 2, ...$$

$$x_0 \in X \text{ given.}$$

Dynamic Programming

Formulating Life-cycle saving problem (infinite-horizon) into a sequence problem, we have:

$$V(k_0) = \sup_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \ln(k_t - k_{t+1})$$

$$\text{s.t. } k_{t+1} \in [0, k_t] \equiv \Gamma(k_t) \text{ for all } t = 0, 1, 2, ...$$

$$k_0 > 0 \text{ given.}$$

Definition 11.B.1 (Bellman Equation).

$$V(x_t) = \sup_{x_{t+1} \in \Gamma(x_t)} \{F(x_t, x_{t+1}) + \beta V(x_{t+1})\} \text{ for all } x_t \in X \tag{BE}$$

- Bellman equation expresses value function as a combination of a flow payoff $F(x_t, x_{t+1})$ and a discounted continuation payoff $\beta V(x_{t+1})$.
- We call $V(\cdot)$ solution to Bellman equation.[1]

[1] We haven't yet demonstrated that a solution $V(\cdot)$ exists.

Dynamic Programming

We briefly show that

- value function defined by sequence problem is also solution to Bellman equation and
- vice versa (with an additional condition $\lim_{n\to\infty} \beta^n V(x_n) = 0$ for any feasible x sequences).

11.C. Solving Bellman equation

There are in general three methods to solve Bellman equation:

- Guess and verify
- Iterate functional operator analytically
- Iterate functional operator numerically (We will not cover this method in this course.)

11.C.1. Guess and verify

- Let us reconsider infinite-horizon life-cycle saving.
- Bellman equation:

$$V(k_t) = \max_{k_{t+1} \in [0,k_t]} \{\ln(k_t - k_{t+1}) + \beta V(k_{t+1})\}.$$

- Solution must be interior.
- Two conditions:
  1. FOC: $-\frac{1}{k_t - k_{t+1}} + \beta V'(k_{t+1}) = 0$.
  2. Envelope theorem: $V'(k_t) = \frac{1}{k_t - k_{t+1}}$.

Guess the value function

- Guess value function takes the form:

$$V(k) = a + b\ln(k),$$

  where a and b are constants to be determined.

- We try this form because utility function is of log form.

Guess the policy function

- Alternatively, we could also guess the form of the policy function.
- Guess $k_{t+1} = \theta k_t$, where θ is a constant to be determined.

11.C.2. Iterate functional operator analytically

How to do it

- Start with any initial guess, for example, $V_0(k) = 0$
- First iteration: $V_1(k_t) = \max_{k_{t+1} \in [0,k_t]} \{\ln(k_t - k_{t+1})\}$
  - Solution is $V_1(k_t) = \ln(k_t)$
- Second iteration: $V_2(k_t) = \max_{k_{t+1}} \{\ln(k_t - k_{t+1}) + \beta \ln(k_{t+1})\}$
  - Solution is $V_2(k_t) = \text{some constant} + (1 + \beta) \ln(k_t)$
- Third iteration: $V_3(k_t) = \max_{k_{t+1}} \left\{ \ln(k_t - k_{t+1}) + \beta V_2(k_t) \right\}$
  - Solution is $V_3(k_t) = \text{some constant } + (1 + \beta + \beta^2) \ln(k_t)$

Iterate functional operator analytically

- Continuing iteration, eventually, we will obtain

$$V(k_t) = \text{some constant } + \frac{1}{1-\beta}\ln(k_t)$$

- Since

$$V(k_t) = \max_{k_{t+1}\in[0,k_t]} \Big\{ \ln(k_t - k_{t+1}) + \beta V(k_{t+1}) \Big\}$$

we get $k_{t+1} = \beta k_t$.

- After obtaining policy function, we could get value function.

Iterate functional operator analytically

- In this example, we have shown that $\lim_{n \to \infty} V_n \to V$ when $V_0(k) = 0$.
- In fact, we will always get convergence independent of choice of V_0.
- Theory will be briefly discussed later.

Iterate functional operator analytically

- Above iteration method could be described in a more convenient way.
- For any function $w : \mathbb{R}_+ \to \mathbb{R}$, we can define a new function $Bw : \mathbb{R}_+ \to \mathbb{R}$ by

$$(Bw)(k_t) = \max_{k_{t+1} \in [0, k_t]} \Big\{ \ln(k_t - k_{t+1}) + \beta w(k_{t+1}) \Big\}.$$

Iterate functional operator analytically

- When we use this notation, previous method is equivalent to choosing a function V_0 and studying sequence $\{V_n\}$ defined by $V_{n+1} = BV_n$ for $n = 0, 1, 2, ...$
- Goal is to show that this sequence of functions converge to limit function V that satisfies

$$V(k_t) = \max_{k_{t+1} \in [0,k_t]} \Big\{ \ln(k_t - k_{t+1}) + \beta V(k_{t+1}) \Big\}.$$

Iterate functional operator analytically

- Or equivalently, we could view B as a mapping from some set of functions into itself.
- Then, what we are looking for is a fixed point of mapping B, that is, a function V that satisfies $V = BV$.
- Operator B is called Bellman operator.

Iterate functional operator analytically

- In a general setting, Bellman operator:

$$(Bw)(x_t) = \sup_{x_{t+1} \in \Gamma(x_t)} \{F(x_t, x_{t+1}) + \beta w(x_{t+1})\} \text{ for all } x_t \in X$$

- What we do is to pick some w and iterate $B^n w$ until convergence.
  - (Uniform) convergence of a sequence of functions is defined by convergence in sup-norm.

Why it works

Short answer is: B is a contraction mapping.

Definition 11.C.1 (Contraction mapping)**.** Let (S, ρ) be a metric space and $T : S \to S$ be a function mapping S into itself. T is a contraction mapping (with modulus β) if for some $\beta \in (0, 1)$, $\rho(Tx, Ty) \leq \beta\rho(x, y)$, for all $x, y \in S$.

In plain words, T is a contraction mapping if operating T on any two elements in S moves them strictly closer to each other.

Why it works

For our result, we need following two results:

1. Contraction Mapping Theorem (Theorem 11.C.2): a fixed point theorem
2. Blackwell's sufficient conditions (Theorem 11.C.2): sufficient conditions for an operator to be a contraction mapping

Why it works

Theorem (Contraction Mapping Theorem (Stokey, Lucas & Prescott Theorem 3.2))**.** If (S, ρ) is a complete metric space and $T : S \to S$ is a contraction mapping with modulus β, then

a. T has exactly one fixed point v in S, and

b. for any $v_0 \in S$, $\rho(T^n v_0, v) \leq \beta^n \rho(v_0, v)$, $n = 0, 1, 2, \ldots$.

Why it works

Theorem (Blackwell's sufficient conditions for a contraction (SLP Theorem 3.3))**.** Let $X \subseteq \mathbb{R}^l$, and let $B(X)$ be a space of bounded functions $f : X \to \mathbb{R}$, with the sup norm. Let $T : B(X) \to B(X)$ be an operator satisfying

a. (monotonicity) $f, g \in B(X)$ and $f(x) \leq g(x)$, for all $x \in X$, implies $(Tf)(x) \leq (Tg)(x)$, for all $x \in X$;

b. (discounting) there exists some $\beta \in (0, 1)$ such that $[T(f+a)](x) \leq (Tf)(x)+\beta a$, all $f \in B(X), a \geq 0, x \in X$.

Then T is a contraction with modulus β.

Why it works

Remark. Blackwell's sufficient conditions are only sufficient but not necessary: some contraction mappings do not satisfy these sufficient conditions.

Why it works

Example. Check Blackwell sufficient conditions for life-cycle saving problem:

$$(Bw)(k_t) = \max_{k_{t+1} \in [0, k_t]} \Big\{ \ln(k_t - k_{t+1}) + \beta w(k_{t+1}) \Big\}$$

Why it works

- By Blackwell's sufficient conditions (Theorem 11.C.2), B is a contraction mapping.
- By Contraction Mapping Theorem (Theorem 11.C.2), B has a unique fixed point, which could be reached from any initial point.

Remark. This result implies that the Bellman equation has a unique solution.

11.D. Examples

11.D.1. Example 1: Optimal growth model

Finite-horizon, backward induction

Consider social planner's problem:

$$\max_{\substack{\{c_t\}_{t=0}^T \\ \{k_t\}_{t=1}^T}} \sum_{t=0}^{T} \beta^t \ln(c_t)$$

$$\text{s.t. } c_t + k_{t+1} = k_t^\alpha \text{ for all } t = 0, ..., T$$

$k_0 > 0$ is given and the terminal capital $k_{T+1} = 0$.

Finite-horizon, backward induction

- We will apply dynamic programming to solve $T = 2$.
- Method of solving the problem extends to all finite T.
- We solve the problem by Backward Induction.

Infinite-horizon

Consider infinite-horizon version:

$$\max_{\substack{\{c_t\}_{t=0}^{\infty} \\ \{k_t\}_{t=1}^{\infty}}} \sum_{t=0}^{\infty} \beta^t \ln(c_t)$$

$$\text{s.t. } c_t + k_{t+1} = k_t^{\alpha} \text{ for all } t = 0, 1, 2, ...$$

$k_0 > 0$ given.

Infinite-horizon

- Bellman equation is

$$V(k_t) = \max_{k_{t+1} \in [0, k_t^\alpha]} \{\ln(k_t^\alpha - k_{t+1})) + \beta V(k_{t+1})\}.$$

1. FOC:

$$-\frac{1}{k_t^\alpha - k_{t+1}} + \beta V'(k_{t+1}) = 0.$$

2. Envelope theorem:

$$V'(k_t) = \frac{\alpha k_t^{\alpha-1}}{k_t^\alpha - k_{t+1}}.$$

Infinite-horizon

We apply guess and verify method.

- Guess value function: $V(k) = a + b\ln(k)$
- Guess policy function: $k_{t+1} = \theta f(k_t) = \theta k_t^{\alpha}$

Infinite-horizon

- We could obtain same result by iterating functional operator analytically.
- For example, try initial guess $V_0(k_t) = 0$.

Stochastic growth

- Dynamic programming is also applicable to stochastic problems.
- Social planner's problem is modified:

$$\max_{\substack{\{c_t(z_t)\}_{t=0}^{\infty} \\ \{k_t(z_t)\}_{t=1}^{\infty}}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \ln(c_t) \tag{11.1}$$

$$\text{s.t. } c_t + k_{t+1} = z_t k_t^{\alpha} \text{ for all } t = 0, 1, 2, ...$$

$k_0 > 0$ given.

Stochastic growth

- $\{z_t\}$ is a sequence of i.i.d. r.v. with $\mathbb{E}_0(\ln(z_t)) = \mu$.
- At the beginning of period t, exogenous shock z_t is realized.
- Thus when making period t decision, ocial planner knows (k_t, z_t) and accordingly current output $z_t k_t^\alpha$.
- (k_t, z_t) is called state of the economy.
- Note that now olution is expressed in terms of contingency plans, that is, c_t and k_{t+1} are functions of z_t.

Stochastic growth

- Problem could still be equivalently expressed using recursive formulation.
- Bellman equation is:

$$V(k_t, z_t) = \max_{k_{t+1} \in [0, z_t k_t^\alpha]} \{\ln(z_t k_t^\alpha - k_{t+1}) + \beta \mathbb{E}_t V(k_{t+1}, z_{t+1})\}.$$

Stochastic growth

1. FOC:

$$-\frac{1}{z_t k_t^\alpha - k_{t+1}} + \beta \mathbb{E}_t \frac{\partial V(k_{t+1}, z_{t+1})}{\partial k_{t+1}} = 0.$$

2. Envelope theorem:

$$\frac{\partial V(k_t, z_t)}{\partial k_t} = \frac{\alpha z_t k_t^{\alpha-1}}{z_t k_t^\alpha - k_{t+1}}.$$

Stochastic growth

Similar to deterministic model,

- Guess value function: $V(k) = a + b\ln(k) + c\ln(z)$
- Guess policy function: $k_{t+1} = \theta f(k_t) = \theta z_t k_t^{\alpha}$

11.D.2. Example 2: Job market search

(Dixit Example 1 + unemployment compensation)

- There is a whole spectrum of jobs paying different wages.
- CDF is $\Phi(w)$.
- Corresponding PDF is $\phi(w) = \Phi'(w)$.

Job market search

- A worker must engage in search to find out how much a particular job pays.
- Each period, an unemployed worker draws w.
- He could either accept or reject.
- If reject, then worker stays unemployed and waits until next period to draw another wage offer.
- Worker receives unemployment compensation c for each of unemployed period.
- Discount factor is β.

11.D.3. Example 3: Saving under uncertainty (Dixit Example 2)

- Consider a consumer with wealth W that earns a random total return (principal plus interest) of r per period, and no other income.
  - Starting period t with wealth W_t, if consumer consumes C_t and saves $W_t - C_t$, his random wealth at start of next period will be $W_{t+1} = r_{t+1}(W_t - C_t)$.
- Note that r_{t+1} is not realized when making consumption decision.

Saving under uncertainty

- Consumption of C_t in any period gives him utility

$$U(C) = \frac{C^{1-\varepsilon}}{(1-\varepsilon)}, \text{ with } \varepsilon > 0.$$

- Discount factor is β.