Dynamic Programming

Takeki Sunakawa

Quantitative Methods for Monetary Economics

University of Mannheim

May 2018

Introduction

- A dynamic optimization problem is often *recursive*, i.e., the problem does not depend on the period in which individuals make decisions.
- Dynamic programming is a very powerful tool to solve recursive problems.
- The first half of the lecture note is based on Adda and Cooper chapter 2. The second half looks at the examples in McCandless chs. 4 and 5.

Setup

- Suppose that you have a cake of size W_1.
- At each point of time, $t = 1, 2, 3, ..., T$, you can consume some of the cake and save the remainder.
- Let c_t be your consumption in period t and let $u(c_t)$ represent the flow of utility from this consumption.
- Assume $u(\cdot)$ is real-valued, differentiable, strictly increasing and strictly concave. Assume that $\lim_{c \to 0} u'(c) = \infty$.

Cake eating problem defined

- Represent lifetime utility by

$$\sum_{t=1}^{T} \beta^{t-1} u(c_t)$$

where $\beta \in (0,1)$ is called discount factor.

- For now, assume that the cake does not depreciate (melt) or grow. Hence, the evolution of the cake over time is governed by:

$$W_{t+1} = W_t - c_t$$

for $t = 1, 2, ..., T$.

- How would you find the optimal path of consumption, $\{c_t\}_{t=1}^{T}$?

Direct attack

- One approach is to solve the constrained optimization problem directly. This is called the sequence problem. Consider the problem of

$$\max_{\{c_t, W_{t+1}\}_{t=1}^{T}} \sum_{t=1}^{T} \beta^{t-1} u(c_t)$$

  subject to

$$W_{t+1} = W_t - c_t$$

  for $t = 1, 2, 3, ..., T$. Also, $c_t \geq 0$ and $W_t \geq 0$. W_1 is given.

- Alternatively, the constraint is

$$\sum_{t=1}^{T} c_t + W_{T+1} = W_1.$$

Direct attack, cont'd

- Let λ be the multiplier on the constraint, the FOCs are

$$\beta^{t-1}u'(c_t) = \lambda$$

for $t = 1, 2, ..., T$ and $\phi = \lambda$, which is the multiplier on $W_{T+1} \geq 0$.

- Combining equations,

$$u'(c_t) = \beta u'(c_{t+1}),$$

holds for $t = 1, 2, ..., T-1$, which is an Euler equation.

Direct attack, cont'd

- Note that having the Euler equation is necessary but not sufficient.
- Suppose $W_{T+1} > 0$ so that there is cake left over. This is clearly an inefficient plan. This shows $W_{T+1} = 0$ and $\lambda = \phi > 0$ must hold.
- So, in effect, the problem is pinned down by an initial and terminal condition.

Value function

- Let
$$V_T(W_1) = \max_{\{c_t, W_{t+1}\}_{t=1}^T} \sum_{t=1}^{T} \beta^{t-1} u(c_t),$$
which is called a value function.

- Note that
$$V_T'(W_1) = \lambda = \beta^{t-1} u'(c_t)$$
$t = 1, 2, ..., T$. A slight increase in the size of the cake leads to an increase in lifetime utility equal to the marginal utility in any period. It doesn't matter when the extra cake is eaten given that the consumer is acting optimally.

Adding period 0

- We add a period 0 and give an initial cake of size W_0.
- Dynamic programming converts a T period problem into a 2 period problem with rewriting of the objective function. It uses the value function obtained from solving a shorter horizon problem.

Dynamic programming

- We take advantage of having $V_T(W_1)$. Given W_0, consider the problem of

$$\max_{c_0} u(c_0) + \beta V_T(W_1)$$

  where

$$W_1 = W_0 - c_0.$$

- The principle of optimality: It doesn't matter how the cake will be consumed after the initial period. The agent will be acting optimally and thus generating $V_T(W_1)$.

Dynamic programming, cont'd

- Note that
$$u'(c_0) = \beta V_T'(W_1).$$
- Also from the earlier discussion of the sequence problem,
$$V_T'(W_1) = u'(c_1) = \beta^t u'(c_{t+1})$$
for $t = 1, 2, ..., T-1$.
- These two conditions yields
$$u'(c_t) = \beta u'(c_{t+1})$$
for $t = 0, 1, 2, ..., T-1$.

Building the value function

- The problem looks simple by pretending that we have $V_T(W_1)$.
- Of course, we need to have $V_T(W_1)$ either by solving a sequence problem directly or by building it recursively starting from an initial single period problem (this is called backward induction).

Three period example

- Assume $u(c) = \ln(c)$.
- For $t = T = 3$, $c_3 = W_3$ and $V_3(W_3) = \ln(W_3)$.
- For $t = 2$, the Euler equation and resource constraint are

$$1/c_2 = \beta/c_3,$$
$$W_2 = c_2 + c_3.$$

Then we have

$$c_2 = \tfrac{W_2}{1+\beta}, \quad c_3 = \frac{\beta W_2}{1+\beta}.$$

Three period example, cont'd

- From this, we can solve for

$$\begin{aligned} V_2(W_2) &= \ln(c_2) + \beta \ln(c_3), \\ &= A_2 + B_2 \ln(W_2), \end{aligned}$$

where $A_2 = \ln(1/(1+\beta)) + \beta \ln(\beta/(1+\beta))$ and $B_2 = 1 + \beta$.

Three period example, cont'd

- For $t = 1$, the value function is

$$V_1(W_1) = \max_{W_2} \ln(W_1 - W_2) + \beta V_2(W_2).$$

- The first order condition is

$$1/c_1 = \beta V_2'(W_2) = \beta B_2/W_2.$$

- Note that from the $t = 2$ problem, we know $W_2 = (1+\beta)c_2$, then

$$1/c_1 = \beta/c_2.$$

Also we know

$$1/c_2 = \beta/c_3.$$

Three period example, cont'd

- Then we have

$$c_1 = \frac{W_1}{1+\beta+\beta^2}, \quad c_2 = \tfrac{\beta W_1}{1+\beta+\beta^2}, \quad c_3 = \frac{\beta^2 W_1}{1+\beta+\beta^2}.$$

- Subs into $V_1(W_1)$ yields

$$V_1(W_1) = A_1 + B_1 \ln(W_1)$$

where
$A_1 = \ln(1/(1+\beta+\beta^2)) + \beta\ln(\beta/(1+\beta+\beta^2)) + \beta^2\ln(\beta^2/(1+\beta+\beta^2))$
and $B_1 = 1+\beta+\beta^2$.

Infinite horizon

- Suppose the following infinite horizon sequence problem

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

subject to

$$W_{t+1} = W_t - c_t$$

for $t = 0, 1, 2, ..., T$, given W_0.

Infinite horizon dynamic programming

- Specifying this as a dynamic programming problem

$$V(W) = \max_{c \in [0,W]} u(c) + \beta V(W - c).$$

- The state variable is the size of the cake W at the beginning of the period. The control variable is being chosen, in this case c.

- The dependence of the state tomorrow on the state and control today, given by

$$W' = W - c$$

  is called the transition equation.

Bellman equation

- Alternatively, we specify the problem as choosing tomorrow's state

$$V(W) = \max_{W' \in [0,W]} u(W - W') + \beta V(W').$$

  Either specification yields the same result.

- This is a functional equation called as a Bellman equation.

Some notes on the Bellman equation

- Note that the unknown in the Bellman equation is the value function itself.
  - In the finite horizon problem, the terminal period is used to derive the value function by backward induction.
  - In the infinite horizon problem, the fixed point restriction of having $V(W)$ on both sides of the equation will provide us with a means of solving the functional equation.
- All relations is expressed without indicating time. This is the essence of stationarity.
- All information about the past is summarized by W, the size of the cake at the start of the period.

Solving the Bellman equation

- The first order condition is given by

$$u'(c) = \beta V'(W').$$

- What is the derivative of $V(W)$? The Envelope theorem yields

$$V'(W) = u'(c).$$

- Substitution leads to the Euler equation:

$$u'(c) = \beta u'(c').$$

Policy function

- The link between the state and control variables are called the policy function

$$c = \phi(W), \quad W' = \varphi(W) = W - \phi(W),$$

- When the state and control variables are observable, these policy functions will be useful for estimation of the underlying parameters.
- In general, finding closed form solutions for the value and poicy functions is not possible. In those cases, we solve these problems numerically.

Analytical example

- Sometimes there is the analytical solution. We use a guess and verify method.
- Given the previous results, we conjecture that the value function takes the form of:

$$V(W) = A + B \ln W.$$

- Taking this guess, the functional equation becomes

$$A + B \ln W = \max_{W'} \ln(W - W') + \beta(A + B \ln W')$$

for all W.

Analytical example, cont'd

- After some algebra,

$$W' = \frac{\beta B}{1+\beta B} W$$

which implies

$$A + B \ln W = \ln \frac{W}{1+\beta B} + \beta \left[A + B \ln \left(\frac{\beta B W}{1+\beta B} \right) \right]$$

for all W.

- Collecting a constant and the terms related to $\ln(W)$ in both sides,

$$\begin{aligned} A &= \beta A + \beta B \ln(\beta B) - (1+\beta B) \ln(1+\beta B), \\ B \ln(W) &= \ln(W) + \beta B \ln(W), \end{aligned}$$

which can be solved for A and B.

Analytical example, cont'd

- Then we have

$$
\begin{aligned}
A &= \beta B \ln(\beta B) - (1+\beta B)\ln(1+\beta B), \\
B &= 1/(1-\beta).
\end{aligned}
$$

- With this solution, we also have

$$c = (1-\beta)W, \quad W' = \beta W.$$

The optimal policy is to save a constant fraction of the cake and eat the remaining fraction. [Confirm this result by yourself.]

Value function iteration

- The above example can be solved numerically too. We use the value function iteration method with a successive approximation.

- We write the Bellman equation as

$$V^{(i+1)}(W) = \max_{W'} \left\{ \ln(W - W') + \beta V^{(i)}(W') \right\},$$

given the function $V^{(i)}(W)$.

Value function iteration, cont'd

- We update the value function iteratively until the function converges, i.e., $\left\| V^{(i+1)}(W) - V^{(i)}(W) \right\| \leq \epsilon$.
- We need to approximate $V^{(i)}(W')$, and numerically solve the maximization problem with regard to W'.

Successive approximation

- Let $W \in \mathcal{W} = \{W_1, W_2, ..., W_n\}$ where $W_1 < W_2 < ... < W_n$. $V(W)$ is approximated by a $(n \times 1)$ vector of $\{V(W_i)\}_{i=1}^{n}$. We can successively update the approximation as follows:

1. Given $V(W)$. Fix index i s.t. $W_i \in \mathcal{W}$. There exists the largest index $j > i$ s.t. $c = W_i - W_j > 0$.
2. Let $\mathbf{w}_i = [W_i - W_1, ..., W_i - W_j]$ and $\mathbf{v}_i = [V(W_1), ..., V(W_j)]$. The maximization problem is written as a vector form

$$V^*(W_i) = \max \{\ln \mathbf{w}_i + \beta \mathbf{v}_i\},$$

   for $i = 1, 2, ..., n$.
3. Update $V(W)$ by $V^*(W)$. Iterate 1-3 until convergence.

Mapping

- The DP problems can be formulated as a general form:

$$T(W)(s) = \max_{s' \in \Gamma(s)} \sigma(s, s') + \beta W(s') \tag{1}$$

  for all $s \in S$. S is the set of the state variable and $\Gamma(s)$ is the set of the control variable. T is an operator mapping continuous bound functions into continuous bound functions.

- This mapping takes a guess on the value function $W(s)$ for all s and produces another value function $T(W)(s)$.

Contraction

- Assumption I (from Cooper's lecture notes)

1. S is a convex subset of R^l and $\Gamma(s)$ is non-empty, compact-valued and continuous,
2. $\sigma(\cdot)$ is real-valued, continuous and bounded,
3. $0 < \beta < 1$.

Theorem

If Assumption I holds, then there exists a unique value function $V(s)$ that solves (1) and for any v_0, $\|T^n v_0 - V\| \leq \beta^n \|v_0 - V\|$.

- Note that the operator T is a contraction. See Stokey, Lucas and Prescott (1989) for more details.

Applying dynamic programming to RCK models

- Deterministic models (McCandless ch. 4)
- Stochastic models (McCandless ch. 5)

Deterministic model

- The individual solves the problem of

$$\max_{\{c_t, k_{t+1}\}} \sum_{t=0}^{\infty} \beta^t \log c_t$$

subject to

$$c_t + k_{t+1} - (1-\delta)k_t = f(k_t),$$

given k_0.

- k_t is the state variable. What is a control variable?

Bellman equation

- The value function is given by

$$V(k_0) = \max \sum_{t=0}^{\infty} \beta^t u(f(k_t) + (1-\delta)k_t - k_{t+1})$$

- It can be written recursively

$$V(k) = \max_{k'} u(f(k) + (1-\delta)k - k') + \beta V(k').$$

This is the Bellman equation.

Solving the Bellman equation

- The first order condition is given by

$$u'(f(k) + (1-\delta)k - k') = \beta V'(k').$$

- The Envelope theorem yields

$$V'(k) = u'(f(k) + (1-\delta)k - k')\left(f'(k) + 1 - \delta\right).$$

- Substitution leads to the Euler equation:

$$u'(c) = \beta u'(c')\left(f'(k') + 1 - \delta\right).$$

Value function iteration

- We use the value function iteration method to find an approximation of the value and policy functions.

- Assume $u(c) = \ln(c)$ and $f(k) = k^\theta$. We write the Bellman equation as

$$V^{(i+1)}(k) = \max_{k'} \left\{ \ln(k^\theta + (1-\delta)k - k') + \beta V^{(i)}(k') \right\},$$

given the function $V^{(i)}(k)$.

Value function iteration, cont'd

- We update the value function iteratively until the function converges, i.e., $\left\| V^{(i+1)}(k) - V^{(i)}(k) \right\| \leq \epsilon$.
- We need to approximate $V^{(i)}(k')$, and numerically solve the maximization problem with regard to k'.
- The sample codes in McCandless uses methods of linear interpolation and bisection search, which are implemented in Matlab.

Linear interpolation

- Suppose we have the values of $f(x)$ at grid points $x \in X = \{x_1, ..., x_n\}$.
- Then, the value of $f(x)$ at $x \in [x_i, x_{i+1}]$ is approximated by

$$\hat{f}(x) = wf(x_i) + (1-w)f(x_{i+1}),$$

where $w = \frac{x_{i+1}-x}{x_{i+1}-x_i}$.

- The command `interp1` in Matlab does the linear interpolation.

Minimization problem

- Let $f(x)$ is a continuous function. We will find $x^* \in [a,b]$ such that $f(x) \geq f(x^*)$ for any $x \in [x^* - \varepsilon, x^* + \varepsilon]$,

$$x^* = \text{arg} \min_{x \in [a,b]} f(x).$$

- Note that the method can be used even when f is not differentiable. However, it may find local minima when f is not quasi-convex.

- In Matlab, `fminbnd` is a minimization function based on the golden section search.

Golden section search

- We search iteratively for x, at each step tightening the bounds which bracket it. Set r to the golden ratio, $(3-\sqrt{5})/2$ and let

$$\begin{aligned} c &= (1-r)a + rb. \\ d &= ra + (1-r)b. \end{aligned}$$

1. If $f(c) \geq f(d)$ then we know the minumum will be in $[c, b]$.
2. If $f(c) < f(d)$ then we know the minumum will be in $[a, d]$.

In case 1, shift the new bounds $[a', b']$ onto $[c, b]$. In case 2, shift the new bounds $[a', b']$ onto $[a, d]$.

Golden section search

- [Demonstrate the golden section search]

A two state process

- The production function is given by $y_t = A_t f(k_t)$ where

$$A_t = \begin{cases} A_1 & \text{with prob. } p_1, \\ A_2 & \text{with prob. } p_2, \end{cases}$$

  and $p_1 + p_2 = 1$.

- We assume A_1 and A_2 are the only values that technology can take. The realization of A_t in each period is independent. Further we assume $A_1 > A_2$.

Stochastic model

- The individual solves the problem of

$$\max_{\{c_t, k_{t+1}\}} E_0 \sum_{t=0}^{\infty} \beta^t \log c_t$$

subject to

$$c_t + k_{t+1} - (1-\delta)k_t = A_t f(k_t),$$

given k_0.

where $A_t \in \{A^1, A^2\}$.

- k_t is called endogenous state variable, whereas A_t is called exogenous state variable.

Bellman equation

- The value function is given by

$$V(k_0, A_0) = \max \sum_{t=0}^{\infty} \beta^t u(A_t f(k_t) + (1-\delta)k_t - k_{t+1})$$

- It can be written recursively

$$V(k, A) = \max_{k'} u(Af(k) + (1-\delta)k - k') + \beta EV(k', A').$$

- For any particular choice of k', the expectation is equal to

$$EV(k', A') = p_1 V(k', A_1) + p_2 V(k', A_2).$$

Value function iteration

- We can write a pair of the Bellman equation as

$$\begin{aligned} V^{(i+1)}(k, A_1) &= \max_{k'} \left\{ \ln(A_1 k^\theta + (1-\delta)k - k') \right. \\ &\quad \left. + \beta p_1 V^{(i)}(k', A_1) + \beta p_2 V^{(i)}(k', A_2) \right\}, \\ V^{(i+1)}(k, A_2) &= \max_{k'} \left\{ \ln(A_2 k^\theta + (1-\delta)k - k') \right. \\ &\quad \left. + \beta p_1 V^{(i)}(k', A_1) + \beta p_2 V^{(i)}(k', A_2) \right\}, \end{aligned}$$

given the function $V^{(i)}(k, A_1)$ and $V^{(i)}(k, A_2)$.

Assignment #5

- Let $\beta = .96$ and $\theta = .36$. Consider the basic RCK model with full depreciation.

$$\max_{\{c_t, k_t\}} \sum_{t=0}^{\infty} \beta^t \ln c_t$$

subject to

$$c_t + k_t \leq A k_{t-1}^{\theta},$$

given k_{-1}.

1. Write down the Bellman equation.
2. Let $V(k)$ be the value function and conjecture $V(k) = A + B \ln k$. Use the guess-and-verify method to solve for the value function analytically (i.e., the parameters A and B).
3. Assume $k \in [0.3k^*, 2.0k^*]$. Use an approximation method (either successive approximation or linear interpolation or whatever) to solve for the value function numerically.
4. Compare the results in 2 and 3.