Advanced Macroeconomics

(Department of Industrial Engineering and Economics, Spring 2Q, FY2016)

An Introduction to Dynamic Optimization

Ryoji Ohdoi

Dept. of Ind. Eng and Econ., Tokyo Tech

Jun 14, 2016

Plan of Lectures in the Part of "Dynamic Optimization"

- June, 14 (Tue) (Today): An introduction to dynamic optimization
- June, 17 (Fri): Infinite-horizon dynamic programming in discrete time
- June, 21 (Tue): Continuous-time optimal control
- June, 24 (Fri): Dynamical system

Introduction: A Cake-eating Problem

- Suppose that you are presented with a cake of size $W > 0$.
- In each period $t \ (= 1, 2, 3, \ldots, T)$, you can eat some of the cake, and save the rest.
- Let
  - $c_t \geq 0$ be amount of your consumption in period t;
  - $u : \mathbb{R}_+ \to \mathbb{R}$ be your one-period utility function from c_t;
  - $\boldsymbol{c} = (c_1, c_2, \ldots, c_T) \in \mathbb{R}_+^T$.

Your Preferences

- Your preferences are assumed to be given by the function $U : \mathbb{R}_+^T \to \mathbb{R}$:

$$\begin{aligned} U(\boldsymbol{c}) &= u(c_1) + \beta u(c_2) + \ldots + \beta^{T-1} u(c_T) \\ &= \sum_{t=1}^{T} \beta^{t-1} u(c_t), \end{aligned}$$

  where $\beta \in (0, 1)$ is called the discount factor (割引因子).

- If one defines $\rho > 0$ such that $\beta \equiv \dfrac{1}{1+\rho}$, then ρ is called the discount rate (割引率).

Optimization Problem

- Question: How do you decide your optimal plans of eating the cake?

- From the view point of microeconomic theory, the problem is formulated as the utility maximization problem as follows:

$$\begin{aligned} \max_{\boldsymbol{c}} \quad & \sum_{t=1}^{T} \beta^{t-1} u(c_t), \\ \text{s.t.} \quad & \sum_{t=1}^{T} c_t \leq W, \\ & c_t \geq 0 \quad t = 1, 2, \ldots, T. \end{aligned} \tag{P}$$

($*$) From the above constraints, $c_t \leq W$ automatically implies for all t.

Optimization Problem

- Thus, in this simple example, there is no
  - Trade in markets: you do not buy or sell the cake; or
  - Strategic interactions between you and other people: you do not need to share the cake with any others.
- Instead, this example focus on your own *intertemporal choice* of consumption. This gives the benchmark for the analysis of an individual's saving-consumption decision in macroeconomics.

Optimization Problem

- Let $\mathcal{D}$ denote the constraint set (制約集合):

$$\mathcal{D} = \left\{ \boldsymbol{c} \in \mathbb{R}_+^T \;\middle|\; \sum_{t=1}^{T} c_t \leq W \right\}$$

$\Rightarrow$ $\mathcal{D}$ is compact.

Assumption

$U : \mathbb{R}_+^T \to \mathbb{R}$ is a continuous function on $\mathcal{D}$.

$\Downarrow$

- The Weierstrass Theorem:
  U attains a maximum (and a minimum) on $\mathcal{D}$.

Optimization Problem

- Exercise 1.1: Suppose that $u(c_t) = c_t$ for all $t = 1, 2, \dots, T$. Then, show that U is maximized at

$$c_1 = W, \quad c_2 = c_3 = \ldots = c_T = 0.$$

Inequality-constrained Optimization

- The above problem (P) is the inequality-constrained optimization problem.
- Hereafter, in addition to its continuity, we assume

Assumption

1. $u'(c) > 0$;
2. $u''(c) < 0$;
3. $\lim_{c \to 0} u'(c) = +\infty$.

The third property is called the Inada condition (稲田条件).

Using the Theorem of Kuhn and Tucker: A "Cookbook" Procedure

- Construct the following Lagrangian:

$$L(\boldsymbol{c}, \lambda, \boldsymbol{\mu}) = U(\boldsymbol{c}) + \lambda \left(W - \sum_{t=1}^{T} c_t \right) + \sum_{t=1}^{T} \mu_t c_t,$$

  where
  - λ: The KT multiplier associated with the constraint: $\sum_{t=1}^{T} c_t \leq W$;
  - μ_t: That associated with the constraint $c_t \geq 0$, and $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_T)$.

Using the Theorem of Kuhn and Tucker: A "Cookbook" Procedure

- Then, derive the first order conditions (F.O.Cs):

$$\beta^{t-1}u'(c_t) + \mu_t = \lambda, \quad t = 1, 2, \ldots, T, \tag{1}$$

$$\sum_{t=1}^{T} c_t \leq W,\ \lambda \geq 0,\ \lambda\left(W - \sum_{t=1}^{T} c_t\right) = 0, \tag{2}$$

$$c_t \geq 0,\ \mu_t \geq 0, \mu_t c_t = 0. \tag{3}$$

- (2) and (3) are called the complementary slackness condition (相補性条件).
- Thanks to the concavity of U and the fact that $\mathcal{D}$ is a convex set, the above F.O.Cs provide necessary and sufficient conditions for the maximization problem.

Solution

- Let $\boldsymbol{c}^* = (c_1^*, c_2^*, \ldots)$ denote the solution of the problem. Hereafter, we call this the optimal consumption plan (最適消費計画).

- Since $u'(c) > 0$,

$$\sum_{t=1}^{T} c_t^* = W. \tag{4}$$

- In addition, thanks to the Inada condition $\lim_{c\to 0} u'(c) = +\infty$, it follows that

$$c_t^* > 0 \forall t.$$

Solution

- Since $c_t^* > 0$, $\mu_t = 0$ holds from (3).
- Then, substituting $\mu_t = 0$ into (1), we have

$$\beta^{t-1} u'(c_t^*) = \lambda, \quad t = 1, 2, \ldots, T.$$

$\Rightarrow$ c_t is obtained as $c_t^* = (u')^{-1}(\lambda/\beta^{t-1})$, where $(u')^{-1}$ is the inverse function of u.

$\Rightarrow$ Substituting this result into (4),

$$\sum_{t=1}^{T} \beta^{t-1} \underbrace{(u')^{-1}(\lambda/\beta^{t-1})}_{c_t^*} = W.$$

Thus, by specifying the functional form of u, we can solve the above equation for λ, which in turn determines the value of c_t^*.

Reformulation of the Problem: Optimal Control

- Consider the same problem, but now consider a "dynamic feature" of the problem explicitly.

- Let w_t denote the size of the leftover cake, which remains to be available for you in period t. The following two conditions are satisfied.

$$
\begin{aligned}
&w_1 = W, \\
&w_t \leq W \quad t = 2, 3, \ldots, T+1.
\end{aligned}
$$

- The value of w_t changes over time, according to the following law of motion:

$$w_{t+1} = w_t - c_t. \tag{5}$$

(5) is called the transition equation (推移方程式).

Reformulation of the Problem: Optimal Control

- Then, the cake-eating problem can be formulated also as the following optimal control problem (最適制御問題) in discrete time:

$$
\begin{aligned}
\max \quad & U(\boldsymbol{c}) = \sum_{t=1}^{T} \beta^{t-1} u(c_t) \\
\text{s.t.} \quad & w_{t+1} = w_t - c_t, \quad t = 1, 2, \ldots, T \\
& w_{T+1} \geq 0 \;\; \text{(terminal condition)}, \\
& w_1 = W \;\; \text{(initial condition)}.
\end{aligned}
$$

(∗) The inequality constraint, $c_t \geq 0$, is now omitted because we have already known that it never binds owing to the Inada condition.

- w_{T+1} is amount of a leftover piece of cake in period T. Thus, notice that

$$\sum_{t=1}^{T} c_t = W \Leftrightarrow w_{T+1} = 0.$$

Euler Equation and Transversality Condition

- Once reset the meanings of notations, λ and μ, defined above.

- Construct the following Lagrangian :

$$
\begin{aligned}
L &= \sum_{t=1}^{T} \beta^{t-1} u(c_t) + \sum_{t=1}^{T} \tilde{\lambda}_t (w_t - c_t - w_{t+1}) + \mu w_{T+1} \\
&= \sum_{t=1}^{T} \beta^{t-1} \left[u(c_t) + \lambda_t (w_t - c_t - w_{t+1}) \right] + \mu w_{T+1}
\end{aligned}
$$

where

- $\lambda_t (= \beta^{-(t-1)} \tilde{\lambda}_t)$: The multiplier associated with the transition equation;
- μ: The KT multiplier associated with the constraint $w_{t+1} \geq 0$.

$(*)$ λ_t is called the costate variable (共役変数) in the context of control.

Euler Equation and Transversality Condition

- The F.O.Cs are given by

$$u'(c_t) = \lambda_t, \tag{6}$$

$$\lambda_t = \beta\lambda_{t+1}, \tag{7}$$

$$\lambda_T = \mu, , \tag{8}$$

$$w_{T+1} \geq 0,\ \lambda_T \geq 0,\ \beta^{T-1}\lambda_T w_{T+1} = 0. \tag{9}$$

- From (6) and (7), we can have the following expression:

$$u'(c_t) = \beta u'(c_{t+1}). \tag{10}$$

(10) is called the Euler equation.

- On the other hand, from (6), (8) and (9), the third condition of (11) is rewritten as

$$\beta^{T-1}u'(c_T)w_{T+1} = 0. \tag{11}$$

This is called the transversality condition.

Optimal con

- In sum, the optimal consumption plan $\boldsymbol{c}^* = (c_1^*, c_2^*, \ldots c_T^*)$ and $\boldsymbol{w}^* \equiv (w_1^*, w_2^*, \ldots, w_{T+1}^*)$ are given by the following 2T+1 equations:

  (i) Transition equation: $w_{t+1}^* = w_t^* - c_t^* \quad t = 1, 2, \ldots, T,$

  (ii) Euler equation: $u'(c_t^*) = \beta u'(c_{t+1}^*) \quad t = 1, 2, \ldots, T-1,$

  (iii) Transversality condition: $\beta^{T-1} u'(c_T^*) w_{T+1}^* = 0,$

  $\Rightarrow w_{T+1}^* = 0$ in this case,

  (iv) Initial condition: $w_1^* = W.$

  Of course, the obtained consumption plan must be the same as that obtained under the original formulation (on pp. 13).

- So, what is advantage of this formulation?
  1. We can use this method even when the transition equation is non-linear.
  2. We can utilize the "recursive feature" of the problem.

Recursive Feature of the Problem

- So far, we formulate the cake-eating problem in two different ways.
- Note that, in either case, you solved the problem *in the initial period*.
- Suppose that you stop and reconsider the problem in period, say, t_0. Your problem from then on is

$$\begin{aligned} \max \quad & \sum_{t=t_0}^{T} \beta^{t-t_0} u(c_t) \\ \text{s.t.} \quad & w_{t+1} = w_t - c_t, \\ & w_{T+1} \geq 0, \\ & w_{t_0} \text{ given.} \end{aligned}$$

You will solve essentially the same problem as you did in the initial period!

Recursive Feature of the Problem

- The dynamic programming technique, based on the Bellman's principle of optimality, utilizes such a property that the problem is recursively defined.

- Let

$$V(w_1) = \max_{\boldsymbol{c}} \left\{ \sum_{t=1}^{T} \beta^{t-1} u(c_t) \mid w_{t+1} = w_t - c_t, t = 1, 2, \ldots T \right\},$$

where $V : \mathbb{R}_+ \to \mathbb{R}$ is called the value function (価値関数). In the context of economics, V is called the indirect utility function.

Bellman Equation

- Briefly speaking, the principle of optimality means

$$\begin{aligned} V(w_1) &= \max_{\boldsymbol{c}} \left\{ \sum_{t-1}^{T} \beta^{t-1} u(c_t) \;\middle|\; w_{t+1} = w_t - c_t, t = 1, 2, \ldots T \right\} \\ &= \max_{c_1} \left\{ u(c_1) + \beta V(w_2) \mid w_2 = w_1 - c_1 \right\}. \end{aligned} \tag{12}$$

(12) is called the Bellman equation (ベルマン方程式).

- The dynamic programming technique, now widely used in macroeconomics, solves the maximization problem by converting the original problem into a two-period problem characterized in the Bellman equation.

  ($*$) Note that the value function in (12) is *still to be determined.* Thus, the Bellman equation is a functional equation.

Next Week

- Infinite-horizon dynamic programming with more general functional forms