Note on Practical Dynamic Programming

Makoto Nakajima, UIUC
January 2007

1 Introduction

We quickly review the basics of the dynamic programming. We look at two classes of problems separately: infinite horizon and finite horizon.

The classic reference on the dynamic programming is Bellman (1957) and Bertsekas (1976). More recent one is Bertsekas (1995). Stokey et al. (1989) is the basic reference for economists.

2 Finite Horizon: A Simple Example

Consider the following life-cycle consumption-savings problem of an agent who lives for I periods. An agent is endowed with k_1 when he is born (age 1), earns e_i in age i, and chooses how much to save and consume in each period. The interest rate associated with savings is r.

Problem 1 (Life-cycle model: sequential formulation)

$$\max_{\{c_i,k_{i+1}\}_1^I} \sum_{i=1}^{I} \beta^{i-1} u(c_i)$$

subject to

$$\begin{aligned} & k_1 \quad \textit{given} \\ & c_i + k_{i+1} = (1+r)k_i + e_i \quad \forall i \\ & c_i \geq 0 \quad \forall i \\ & k_{i+1} \geq 0 \quad \forall i \end{aligned}$$

The solution to the problem is a sequence $\{c_i, k_{i+1}\}_1^I$ which maximizes the discounted sum of period utility of the agent. Therefore, potentially, this is a hard problem. If different agents have different k_1, the optimal sequence of consumption and savings must be found for each k_1.

The beauty of the dynamic programming is to convert the sequential problem into a collection of two-period problems, each of which is easy to solve. Let's see how we can transform the sequential problem into a collection of two-period problems.

Since we convert the original sequential problem into a collection of small problems, it is important to make sure that the solution to each of the small problems is actually optimal in the original problem. In other words, the optimal choice in each of the two-period problem must be globally optimal to justify the transformation. Richard Bellman, the inventor of the dynamic programming method, calls it the *Principle of Optimality*.

When we convert a sequential problem into a dynamic programming problem, we need to detect *state variables* and *control variables*. State variables are the variables which are pre-determined when decision is made, and relevant for pay-offs. They can be exogenous or endogenous. For the current example, the capital stock holding k and age i are the state variables. Control variables are the ones that the agent choose in each period, conditional on the state variables. In the current example, c and k' are the control variables. Actually, however, the smallest set of control variables is either one of k' or c, because once you have one of the two, you can compute the other using the budget constraint and the state variables. We use c as the control variable below, but you can proceed with k' instead and can get exactly the same results.

Notice that it is important that the prices are constant. In other words, the problem is *stationary*. If the prices are changing over time, agents need to know the prices to make a decision. In other words, prices have to be included as a part of the state variables. If prices are functions of some aggregate states of the world, we can include aggregate states of the world instead of the prices in the set of state variables. State variables which are agent specific are called *individual state variable*, and the state variables which are not individual specific are called the *aggregate state variables*. In the current problem, there is no aggregate state variable.

Let's define a *value function* for age i. It gives the sum of discounted utility from the current period on conditional on the state variables. The value function for the current problem is $V(i,k)$. Since the life of an agent starts from age 1, we have:

$$V(1,k_1) = \max_{\{c_i,k_{i+1}\}_1^I} \sum_{i=1}^{I} \beta^{i-1} u(c_i)$$

subject to

$$\begin{aligned} &c_i + k_{i+1} = (1+r)k_i + e_i \quad \forall i \\ &c_i \geq 0 \quad \forall i \\ &k_{i+1} \geq 0 \quad \forall i \end{aligned}$$

In age 2, given k_2, the agent solves the following problem:

$$V(2,k_2) = \max_{\{c_i,k_{i+1}\}_2^I} \sum_{i=2}^{I} \beta^{i-2} u(c_i)$$

subject to

$$\begin{aligned} &c_i + k_{i+1} = (1+r)k_i + e_i \quad \forall i \\ &c_i \geq 0 \quad \forall i \\ &k_{i+1} \geq 0 \quad \forall i \end{aligned}$$

Notice that, in age 2, the agent does not discount the utility in age 2. That's why the power to β is $i-2$ instead of $i-1$. From the point of view from age 1 agent, the utility in age 2 has to be discounted at β.

For a easier notation, let's define the constraint set, which is a mapping from the set of state variables, $C(i,k_i)$ as follows:

$$C(i,k_i) = \{(c_i,k_{i+1}) | c_i \geq 0, k_{i+1} \geq 0, c_i + k_{i+1} = (1+r)k_i + e_i\}$$

Now, go back to the problem in age 1.

$$
\begin{aligned}
V(1,k_1) &= \max_{\{(c_i,k_{i+1})\in C(i,k_i)\}_1^I} \sum_{i=1}^{I} \beta^{i-1}u(c_i) \\
&= \max_{(c_1,k_2)\in C(1,k_1)} \left\{ u(c_1) + \max_{\{(c_i,k_{i+1})\in C(i,k_i)\}_2^I} \sum_{i=2}^{I} \beta^{i-1}u(c_i) \right\} \\
&= \max_{(c_1,k_2)\in C(1,k_1)} \left\{ u(c_1) + \beta \max_{\{(c_i,k_{i+1})\in C(i,k_i)\}_2^I} \sum_{i=2}^{I} \beta^{i-2}u(c_i) \right\} \\
&= \max_{(c_1,k_2)\in C(1,k_1)} \{u(c_1) + \beta V(2,k_2)\}
\end{aligned}
$$

The last equation holds for any $i = 1, 2, ..., I$. Another way of saying this is that the problem has a *recursive* structure. In addition, the last equation contains only the variables in age 1 and age 2. Or, in general, the equation for age i contains only variables in i and $i+1$. Following the standard notation, let's denote the variable in age i and $i+1$ as those without and with primes. Then we get the following representation of the problem, which is called the *Bellman equation*.

Problem 2 (Life-cycle model: recursive formulation)

$$V(i,k) = \max_{(c,k')\in C(i,k)} \{u(c) + \beta V(i+1,k')\}$$

The *optimal decision rules* are the functions from the state variables to the choice variables. The optimal decision rules associated with the recursive problem are $k' = d_k(i,k)$ and $c = d_c(i,k)$.

Since this is a finite horizon problem, the problem can be solved using backward induction. Notice $V(I+1,k) = 0$ for all k (there's no utility after the death of the agent). It implies that $d_k(I,k) = 0$, and $d_c(I,k) = (1+r)k + e_i$. With these optimal decisions, we can compute $V(I,k) = u(d_c(I,k))$. With $V(I,k)$, and the Bellman equation, we can solve for $V(I-1,k)$. And we can continue until we solve for $V(1,k)$.

3 Infinite Horizon: A Simple Example

Consider the following problem:

Problem 3 (Neoclassical growth model: sequential formulation)

$$\max_{\{C_t,K_{t+1}\}_0^\infty} \sum_{t=0}^{\infty} \beta^t u(C_t)$$

subject to

$$
\begin{aligned}
&K_0 \quad \textit{given} \\
&C_t + K_{t+1} = F(K_t) + (1-\delta)K_t \quad \forall t \\
&C_t \geq 0 \quad \forall t \\
&K_{t+1} \geq 0 \quad \forall t
\end{aligned}
$$

With a neoclassical production function $F(K)$, this is the standard neoclassical growth model. Apparently, an solution to the problem is an infinite sequence $\{C_t, K_{t+1}\}_0^\infty$. This is a hard problem.

The beauty of dynamic programming is to convert a sequential problem like this into a collection of two-period problems, which is easier to handle. Let's see how the example above can be converted.

Notice that, for the infinite horizon problem, the problem that the agent faces in the next period is exactly the same if the same amount of capital is endowed. This is because the time horizon is infinite. The future from today and the future from tomorrow is of the same length (which is infinity). Therefore, the value function for the problem, which is the sum of discounted utility that the agent in the problem gains optimally, is not a function of the time period, but only of the capital stock endowed. Formally, the value function can be defined as follows:

$$V(K_0) = \max_{\{(C_t,K_{t+1})\in C(K_t)\}_0^\infty} \sum_{t=0}^{\infty} \beta^t u(C_t)$$

such that

$$C(K_t) = \{(C_t, K_(t+1))|C_t \geq 0, K_{t+1} \geq 0, C_t + K_{t+1} = F(K_t) + (1-\delta)K_t\}$$

Now, notice that, in period 0, K_0 is pre-determined, and C_0 and K_1 are chosen. In period 1, K_1 is pre-determined, and C_1 and K_2 are chosen. We can separate the problem in period 0 and the problem after period 1 as follows:

$$\begin{aligned}
V(K_0) &= \max_{\{(C_t,K_{t+1})\in C(K_t)\}_0^\infty} \sum_{t=0}^{\infty} \beta^t u(C_t) \\
&= \max_{\{(C_t,K_{t+1})\in C(K_t)\}_0^\infty} \left\{ u(C_0) + \sum_{t=1}^{\infty} \beta^t u(C_t) \right\} \\
&= \max_{(C_0,K_1)\in C(K_0)} \left\{ u(C_0) + \max_{\{(C_t,K_{t+1})\in C(K_t)\}_1^\infty} \sum_{t=1}^{\infty} \beta^t u(C_t) \right\} \\
&= \max_{(C_0,K_1)\in C(K_0)} \left\{ u(C_0) + \beta \max_{\{(C_t,K_{t+1})\in C(K_t)\}_0^\infty} \sum_{t=0}^{\infty} \beta^t u(C_{t+1}) \right\}
\end{aligned}$$

The important assumption behind this transformation is that the optimal decision made in $t = 1$ is also optimal even if the choice is made in $t = 0$. That's why we can separate the optimal decision problem from period 1 on from the optimal decision problem in period 0. In other words, *Principle of Optimality* is (correctly) assumed.

Notice that, the problem inside the bracket looks identical to the original problem, except for the starting period which is 1. In other words, the problem has a *recursive structure*. Thanks to the recursive structure, we can replace the problem inside the bracket by the same value function, conditional on K_1 instead of K_0. Now we have:

$$V(K_0) = \max_{(C_0,K_1)\in C(K_0)} \{u(C_0) + \beta V(K_1)\}$$

Since we have only period 0 and period 1 in the equation above, we can drop the time script, and denote that the variables in period 0 as those without a prime, and the variables in period 1 as those with primes. Then we have:

Problem 4 (Neoclassical growth model: recursive formulation)

$$V(K) = \max_{(C,K')\in C(K)} \{u(C) + \beta V(K')\}$$

This equation is called the *Bellman equation*. Notice that, since the problem has a recursive structure, the optimal choice in the model is characterized by functions $K' = d_K(K)$ and $C = d_C(K)$. Instead of finding an optimal sequence $\{C_t, K_{t+1}\}_0^\infty$, we only need to find the optimal choice of K' and C given K.

A problem is that, it doesn't seem that we can apply the same solution method (backward induction) as for the finite horizon model, since there is no *last period*. Moreover, we are not sure if the value function which satisfies the Bellman equation exits. Actually, for a large class of models that we use, we have nice properties of the model, as shown below.

4 Infinite Horizon: General Formulation

Notice that the Bellman equation can be interpreted as a *functional equation*, or an *operator*. The Bellman equation maps a value function $V(K)$ into another value function $V(K)$. The two are not necessarily the same. But the value function that we are looking for is the fixed point of the Bellman operator. In this sense, the Bellman equation implicitly characterizes the value function $V(K)$. We know that, under a set of conditions which are satisfied by many models used in macroeconomics, the Bellman operator the one above has a fixed point $V^*(K)$. Moreover, it's unique.

In order to use the results of Stokey et al. (1989) (SLP hereinafter), let us formulate a general dynamic programming problem:

Problem 5 (General formulation of a dynamic programming problem)

$$V(s) = \max_{c\in C(s)} \{u(s,c) + \beta V(s')\}$$

s is the *state variable*, which is pre-determined when the choice is the current period is made. c is the choice variable. $C(s)$ is the constraint set, which is conditional on the current state s. For our problem $C(s)$ is characterized by the budget constraint, nonnegativity constraint of consumption, and nonnegativity constraint for the capital stock. The associated optimal decision rule is denoted as $c = d(s)$. Here's a theorem from SLP:

Theorem 1 (SLP 4.6)
If (i) $u(s,c)$ *is real-valued, continuous, and bounded,* (ii) $\beta \in (0,1)$, (iii) $C(s)$ *is non-empty, compact-valued, and continuous, there exists a unique value function that solves Problem 5, and the value function is the limit of the Bellman operator.*

I omit the proof but let me mention that the Contraction Mapping Theorem plays a crucial role in the proof. The nice thing is that the value function is proved to be unique, and it is obtained by continuously applying the Bellman operator to a guess. The value function iteration, which is a popular method to solve dynamic problem, method is based on this result.

In addition, there are two theorems which prove useful properties of the value function $V(s)$ and the associated optimal decision rule $d(s)$ under some additional conditions.

Theorem 2 (SLP 4.7)
If, in addition to conditions in Theorem 1, (i) $u(s, c)$ *is strictly increasing, and* (ii) $C(s)$ *is monotone, then* $V(s)$*, the unique solution to Problem 5 is strictly increasing.*

Theorem 3 (SLP 4.8)
If, in addition to conditions in Theorem 1, (i) $u(s, c)$ *is strictly concave, and* (ii) $C(s)$ *is convex, then* $V(s)$*, the unique solution to Problem 5 is strictly concave, and the associated* $d(s)$ *is single-valued and continuous.*

We are not going into the models with shocks, but similar theorems can be applied. Interested readers are encouraged to read chapter 9 of SLP.

5 Remarks

- It's always nice to have theorem 3. If we have the continuity of the optimal decision rule and the strict concavity of the value function, it's easy to search for the optimal decision. However, many interesting models do not satisfy the property. Examples are:
  1. Models with convex cost of adjustment of capital. An interesting example is the fixed cost of changing the size of the house. Non-convex adjustment cost is fine.
  2. Models with non-trivial tax schedule function.
- Whether theorem 3 holds or not (i.e. the optimal decision rule is continuous or not) affects the numerical methods that can be applied to solve the problem crucially. If theorem 3 holds, you can interpolate the optimal decision rule function using some kind of continuous function. It means that you don't need to solve the optimal decision rule at a lot of points. On the other hand, in case theorem 3 does not hold, there is no guarantee that the optimal decision rule can be approximated by a continuous function. Then the only reasonable way to approximate to optimal decision rule is to use discretization. It means that it takes a long time to solve the problem.
- You cannot get nice results for models where agents have hyperbolic or quasi-geometric discounting. Since the problem solved by the agent tomorrow is different from the problem that the agent today is solving. In other words, the preference is *time inconsistent*. For a finite horizon case, you can still solve the model using backward induction, but it's not trivial to solve the model with infinite horizon.
- You need some tricks to be able to formulate models without commitment recursively. The classic example is the time inconsistent policy by **?**. Proposed tricks are to include some state variable which works as record keeping.

References

Bellman, Richard, *Dynamic Programming*, Princeton, NJ: Princeton University Press, 1957.

Bertsekas, Dimitri P., *Dynamic Programming and Stochastic Control*, New York, NY: Academic Press, 1976.

_ , *Dynamic Programming and Optimal Control*, Belmont, MA: Athena Scientific, 1995.

Stokey, Nancy, Robert E. Lucas, and Edward C. Prescott, *Recursive Methods in Economic Dynamics*, Cambridge, MA: Harvard University Press, 1989.