An Introduction to Dynamic Programming

Jin Cao

Macroeconomics (Research, WS10/11)

November, 2010

Outline

Motivation
- Why Dynamic Programming
- Basic Idea

Optimality Conditions
- The First Order Condition
- The Envelope Condition
- An Example: Brock-Mirman Model

Value Function and Policy Function
- Guess and Verify
- Value Function Iteration
- Numerical Simulation

Why dynamic programming?

- Lagrangian and optimal control are able to deal with most of the dynamic optimization problems, even for the cases where dynamic programming fails.
- However, dynamic programming has become widely used because of its appealing characteristics:
  - Recursive feature: flexible, and significantly reducing the complexity of the problems;
  - Convergence in the value function: quantitative analysis, especially numerical simulation;
  - Although based on profound theories, numerical computation is rather simple as well as full-fledged. — At least one can get numerical results.
- In this presentation: How to USE dynamic programming methods.

The prototype problem

- Consider a general discrete-time optimization problem

$$\begin{aligned} \max_{\{c_t, k_{t+1}\}_{t=0}^{+\infty}} \quad & \sum_{t=0}^{+\infty} \beta^t u(c_t) \\ s.t. \quad & k_{t+1} = f(c_t, k_t). \end{aligned}$$

- Now define a function (mostly, bounded in value)

$$\begin{aligned} V(k_t) &= \max_{c_t, k_{t+1}} \sum_{i=0}^{+\infty} \beta^i u(c_{t+i}) = \max_{c_t, k_{t+1}} \left\{ u(c_t) + \beta \sum_{i=0}^{+\infty} \beta^i u(c_{t+i+1}) \right\} \\ &= \max_{c_t, k_{t+1}} \{u(c_t) + \beta V(k_{t+1})\} \\ s.t. \quad & k_{t+1} = f(c_t, k_t). \end{aligned}$$

Basic idea: recursive structure

- Recursive nature of the problem — same problem for all t!
  Bellman equation

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

- More jargons, similar as before: State variable k_t, control variable c_t, transition equation (law of motion), **value function** $V(k_t)$, **policy function** $c_t = h(k_t)$.

- Now the problem turns out to be a one-shot optimization problem, given the transition equation!

The first order condition

- The next step: finding the optimality conditions!

- Trivial to see: FOC of the maximization problem.

$$V(k_t) \;=\; \max_{c_t, k_{t+1}} \{u(c_t) + \beta V(k_{t+1})\} \longrightarrow \frac{\partial u(c_t)}{\partial k_{t+1}} + \beta \frac{\partial V(k_{t+1})}{\partial k_{t+1}} = 0.$$

- Reason: Decision problem at period t is to allocate resources between c_t and k_{t+1}. $V(k_t)$ is an optimized value for each period t — FOC with respect to k_{t+1} should hold.
- To get Euler equation, still need a second equation to eliminate $V(\cdot)$ term.

The envelope condition

- The Envelope Theorem

Theorem

Suppose that value function $m(a)$ is defined as following:

$$m(a) = \max_x f(x(a), a).$$

Then the total derivative of $m(a)$ with respect to a equals the partial derivative of $f(x(a), a)$ with respect to a, if $f(x(a), a)$ is evaluated at $x = x(a)$ that maximizes $f(x(a), a)$, i.e.

$$\frac{dm(a)}{da} = \left.\frac{\partial f(x(a), a)}{\partial a}\right|_{x=x(a)}.$$

The envelope condition

- Then the envelope condition of the maximization problem.

$$V(k_t) = \max_{c_t, k_{t+1}} \{u(c_t) + \beta V(k_{t+1})\} \longrightarrow \frac{\partial V(k_t)}{\partial k_t} = \frac{\partial u(c_t)}{\partial k_t}.$$

- Combine with the FOC: update and eliminate $V(\cdot)$

$$\frac{\partial u(c_t)}{\partial k_{t+1}} + \beta \frac{\partial V(k_{t+1})}{\partial k_{t+1}} = 0.$$

- To see how it works? An example.

Example: deterministic Brock-Mirman model

- Consider the following social planner's problem:

$$\begin{aligned} \max_{\{c_t, k_t\}_{t=0}^{+\infty}} \quad & \sum_{t=0}^{+\infty} \beta^t \ln c_t \\ s.t. \quad & k_{t+1} = k_t^\alpha - c_t. \end{aligned}$$

- Bellman equation:

$$\begin{aligned} V(k) \quad &= \quad \max_{k'} \left\{ \ln c + \beta V(k') \right\} \\ s.t. \quad & \quad c + k' = k^\alpha. \end{aligned}$$

Example: optimality conditions

- Use the transition equation to replace c

$$V(k) \quad = \quad \max_{k'} \left\{ \ln(k^\alpha - k') + \beta V(k') \right\}.$$

- The first order condition and the envelope condition

$$-\frac{1}{c} + \beta V'(k') = 0$$
$$V'(k) = \frac{1}{c}\alpha k^{\alpha-1} \rightarrow V'(k') = \frac{1}{c'}\alpha k'^{\alpha-1}$$

- Euler equation, same as one can get from Hamiltonian: $\frac{c'}{c} = \alpha\beta k'^{\alpha-1}$.

What's new: making more senses

- If dynamic programming simply arrives at the same outcome as Hamiltonian, then one doesn't have to bother with it.

- However, the marginal return from dynamic programming becomes higher if one explores deeper. Take a closer look:
  - Value function? Tells you how different paths may affect your value on the entire time horizon. Policy evaluation!
  - Policy function? Tells you explicitly how you make optimal choice in each period, given the state!
- Strategy: Determine $V(k)$, and optimize to get $c_t = h(k_t)$. Not easy...
  - Analytical — not always tractable, or
  - Numerical — in principle, always works.

Mickey Mouse models: guess and verify

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

- Brock-Mirman model:

$$\begin{aligned} V(k) &= \max_{k'} \left\{ \ln c + \beta V(k') \right\} \\ s.t. &\quad c + k' = k^{\alpha}. \end{aligned}$$

- Guess: $V(k) = A + B \ln k$. Verify with the first order condition

$$-\frac{1}{c} + \beta V'(k') = -\frac{1}{k^{\alpha} - k'} + \frac{\beta B}{k'} = 0.$$

- Solve to get $k' = \frac{\beta B}{1+\beta B} k^{\alpha}$, as well as $c = \frac{1}{1+\beta B} k^{\alpha}$. Then apply these equations back to Bellman.

Mickey Mouse models: guess and verify

- Compare with our conjecture $V(k) = A + B \ln k$

$$V(k) = \ln \beta B + \beta A - (1 + \beta B)\ln(1 + \beta B) + \alpha(1 + \beta B)\ln k.$$

- Solve to get the value of the parameters and the policy function

$$B = \frac{\alpha}{1 - \alpha\beta},$$

$$A = \frac{1}{1 - \beta}\left[\ln(1 - \alpha\beta) + \frac{\alpha\beta}{1 - \alpha\beta}\ln \alpha\beta\right],$$

$$c_t = \frac{1}{1 + \beta B}k_t^{\alpha} = (1 - \alpha\beta)k_t^{\alpha}.$$

Value function at the end of the world

- More general approach: To find the value function in the limit. Suppose that the world ends after some finite peroid T. Then surely $V(k_{T+1}) = 0$ as well as $c_T = k_T^{\alpha}$, and $k_{T+1} = 0$.

- Apply these in the Bellman equation

$$V(k_T) = \ln k_T^{\alpha} + \beta V(k_{T+1}) = \ln k_T^{\alpha}.$$

- Then take one period backward, the agent has to solve

$$V(k_{T-1}) = \max_{c_{T-1}, k_T} \left\{ \ln(c_{T-1}) + \beta V(k_T) \right\}.$$

Value function before the end of the world

- Insert $V(k_T)$ and solve for $V(k_{T-1})$ in terms of k_{T-1}

$$V(k_{T-1}) = \alpha\beta \ln(\alpha\beta) - (1+\alpha\beta)\ln(1+\alpha\beta) + (1+\alpha\beta)\ln k_{T-1}^{\alpha}.$$

- In the limit $T \to +\infty$ one can show that the value function converges to

$$\begin{aligned} V(k_t) &= \max_{c_t, k_{t+1}} \left\{ \ln c_t + \beta \left[\frac{1}{1-\beta} \left(\ln(1-\alpha\beta) + \frac{\alpha\beta}{1-\alpha\beta} \ln \alpha\beta \right) \right.\right. \\ &\qquad \left.\left. + \frac{\alpha}{1-\alpha\beta} \ln k_{t+1} \right] \right\}. \end{aligned}$$

- Then solve this static maximization problem to get the policy function

$$c_t = (1-\alpha\beta)k_t^{\alpha}.$$

Numerical simulation

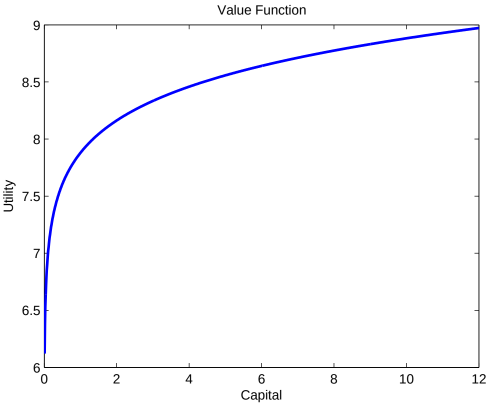
- How if the problem gets more complicated? Consider

$$\begin{aligned} V(k) &= \max_{c,k'} \left\{ u(c) + \beta V(k') \right\} \\ s.t. \quad & c + k' = Ak^{\alpha} - \delta k. \end{aligned}$$

- No open form solution! But... let the computer do the value function iteration.
  - Discretize the state space, and determine its range;
  - Start from the end of the world, and do the backward induction
  - Until the change in value function meets the convergence criterion.

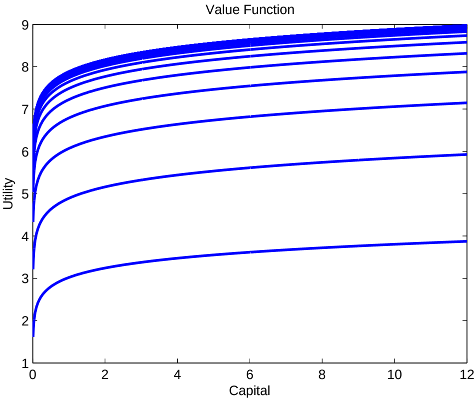
Numerical simulation: value function

- Take $\beta = 0.6$, $A = 20$, $\alpha = 0.3$, $\delta = 0.5$ and run MATLAB



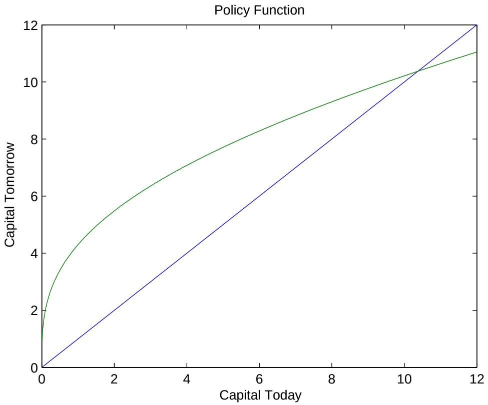
Numerical simulation: convergence

- ... and see how it converges ...



Numerical simulation: steady state

- … and find the steady state.



Summary

- If one only needs the Euler equation and qualitative reasoning, dynamic programming is no better than Hamiltonian.
- To use dynamic programming, more issues to worry: Recursive? Existence of equilibrium (Blackwell sufficient conditions for contraction mapping, and fixed point theorem)? Stochastic? ...
- But rewarding if one wants to know more
  - Flexibility in modelling;
  - Well developed numerical methods.
- Especially, some interesting new research
  - Dynamic contract theory, e.g. Marcet & Marimon (1998), Werning (2008), Doepke (2008);
  - Computating dynamic equilibrium, e.g. Heer & Maussner (2005), and dynamic equilibrium econometrics, e.g. Canova (2007);
  - Stochastic models, e.g. Stachurski (2009), Dixit & Pindyck (1994).

For further reading...

Klaus Wälde.
Applied Intertemporal Optimization.
Mimeo, University of Mainz, 2010.

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