Notes 2: Dynamic Programming

Julio Garín
Advanced Macroeconomics
Spring 2019
Outline

1. Lagrangian Methods and Intertemporal Optimization
   1.1 Endowment Economy
      1.1.1 Model I: No assets, and no storage
      1.1.2 Model II: Riskless interest-bearing asset
   1.2 Production Economy
      1.2.1 Model III: A Robinson Crusoe Economy
      1.2.2 Model IV: Productive Capital and Riskless Asset

2. Dynamic Programming
   2.1 Basics
   2.2 Finite Horizon
      2.2.1 A simple three-period deterministic model
      2.2.2 Generalized $T$
   2.3 Infinite Horizon
      2.3.1 Principle of Optimality
      2.3.2 Operators and Blackwell’s sufficient conditions
   2.4 Solution Techniques
      2.4.1 Value function iteration
      2.4.2 Guess and verify
Two-Period Intertemporal Optimization
Model Overview

▶ Agents:
  ▶ *Forward looking.*
  ▶ Rational and their behavior is optimizing.

▶ Preferences:
  ▶ Lifetime utility:
    \[ U = u(c) + \beta u(c'). \]

▶ Two periods is unrealistic but useful.
▶ If \( u_{cc} < 0: \)
  ▶ Marginal utility is decreasing in consumption.
  ▶ Quasi-concave utility implies convex preferences.
  ▶ Agent would like to smooth consumption.

▶ Four technology scenarios:
  1. Endowment economy, no assets.
  2. Endowment economy, riskless, interest-bearing asset.
  3. “Robinson Crusoe” economy.
  4. Productive capital and riskless asset.
Assume no assets and no storage.
- What is consumption equal to?

Technology.
- Production.
  - For now, we will look at an endowment economy.
  - Stream of income given by $y$ and $y'$.

How can output be transferred between periods?
- We start with a one-period riskless bond.
Model II: Endowment Economy with Riskless Asset

- Now households have access to a riskless asset, $a$, that pays $r$.
  - Amount borrowed limited by $a$.
- Household’s problem given by:

$$\max_{\{c, c', a'\}} U = u(c) + \beta u(c')$$

subject to:

$$a' = y - c \quad (1)$$
$$c' = (1 + r)a' + y' \quad (2)$$
$$a' \geq a. \quad (3)$$

- We can solve using Lagrangian methods.
  - Euler equation.
Model III: A Robinson Crusoe Economy

- Now, instead of a riskless asset, households can move resources forward in time by buying productive capital.
- Increasing and concave production function: \( y = f(k) \).
- Household’s problem becomes:

\[
\max_{\{c,c',k\}} \mathcal{U} = u(c) + \beta u(c')
\]

subject to:

\[
c + k' = f(k) + (1 - \delta)k \quad (4)
\]
\[
c' = f(k') + (1 - \delta)k' \quad (5)
\]
\[
k \text{ is given.} \quad (6)
\]

- Lagrangian methods give us the Euler.
- Now the rate of return is \( f_k' + 1 - \delta \).
The agent faces an asset allocation problem. The problem is given by:

\[
\max_{\{c,c',a',k\}} \mathcal{U} = u(c) + \beta u(c')
\]

subject to:

\[
c + k' + a' = f(k) + (1 - \delta)k \quad (7)
\]

\[
c' = f(k') + (1 - \delta)k' + (1 + r)a' \quad (8)
\]

\[k \text{ is given.} \quad (9)\]

In addition to the standard Euler we obtain:

\[f_{k'} + 1 - \delta = 1 + r.\]
Dynamic Programming
Why “Dynamic Programming”?

[...]
What title, what name, could I choose? In the first place I was interested in planning, in decision making, in thinking. But planning, is not a good word for various reasons. I decided therefore to use the word ‘programming’. I wanted to get across the idea that this was dynamic, this was multistage, this was time-varying—I thought, let’s kill two birds with one stone. Let’s take a word that has an absolutely precise meaning, namely ‘dynamic’, in the classical physical sense. It also has a very interesting property as an adjective, and that is it’s impossible to use the word ‘dynamic’ in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. It’s impossible. Thus, I thought “dynamic programming” was a good name. It was something not even a Congressman could object to. So I used it as an umbrella for my activities.
– Richard Bellman
Ultimately we would like to think about multi-period models... or even infinite horizon models. Why?

Suppose we have the following problem for a household:

\[
\max_{\{c_t, b_t\}_{t=0}^{\infty}} \mathcal{U}_0 = \sum_{t=0}^{\infty} \beta^t u(c_t)
\]

subject to:

\[
c_t + \frac{b_{t+1}}{1+r} = b_t + y_t \quad \text{for } t = 0, 1, 2, \ldots
\]

\[
b_0 \geq 0 \text{ given.}
\]

Now bonds are discounted bonds.
Why Do We Need Another Tool?

- For a rational, forward-looking household, consumption/saving decisions are intertemporally related.
- We could use Lagrangian methods to solve this.
  - System of many non-linear equations in many unknowns.
    - Hard, but feasible.
- Dynamic programming helps us to transform this problem into something more tractable.
  - Especially useful for stochastic models.
- We will start with a three-period problem.
  - Think about problems recursively.
  - Introduce the lingo.
Three-Period Consumption/Saving Problem

The problem is:

\[
\max \quad \mathcal{U}_0 = u(c_t) + \beta u(c_{t+1}) + \beta^2 u(c_{t+2})
\]

subject to:

\[
\begin{align*}
    c_t + \frac{b_{t+1}}{1 + r} &= b_t + y_t \\
    c_{t+1} + \frac{b_{t+2}}{1 + r} &= b_{t+1} + y_{t+1} \\
    c_{t+2} + \frac{b_{t+3}}{1 + r} &= b_{t+2} + y_{t+2} \\
    b_t &\geq 0 \text{ given.}
\end{align*}
\]

Let’s assume for now that \( y_{t+j} = 0 \) for \( j = 0, 1, 2 \).
Solving the Three-Period Problem: $t + 2$

- We solve the problem *backwards*.
  - Today’s decisions are not independent of future outcomes.
  - The *Principle of Optimality*.

- At period $t + 2$ the problem is:
  
  \[
  \max_{\{c_{t+2}, b_{t+3}\}} \mathcal{U}_0 = u(c_{t+2})
  \]

  subject to:
  
  \[
  c_{t+2} + \frac{b_{t+3}}{1+r} = b_{t+2}.
  \]

- Since $u' > 0 \Rightarrow b_{t+3}^* = 0$ and
  
  \[
  c_{t+2}^* \equiv f_{t+2}(b_{t+2}) = b_{t+2}.
  \]

- Given the *state variable*, the *policy function*, $f$, tells the optimal value for the *choice variable*. 

Solving the Three-Period Problem: Value Function

► We can define \( V_{t+2}(b_{t+2}) \equiv U^*_{t+2} \).

► The value function, \( V_{t+2}(b_{t+2}) \), is defined by the following functional equation:

\[
V_{t+2}(b_{t+2}) = \max \{ c_{t+2}, b_{t+3} \} u(c_{t+2})
\]

subject to:

\[
c_{t+2} + \frac{b_{t+3}}{1 + r} = b_{t+2}.
\]

► The transition equation maps, for a given value of the choice variable, the state variable in one period into the next period’s state variable.

► Note what \( V'(b_{t+2}) \) is equal to.
  ► We will come back to this.
Solving the Three-Period Problem: $t + 1$

Let’s keep working backwards:

$$\max_{\{c_{t+1}, b_{t+2}\}} U_{t+1} = u(c_{t+1}) + \beta V_{t+2}(b_{t+2})$$

subject to:

$$c_{t+1} + \frac{b_{t+2}}{1 + r} = b_{t+1}.$$ 

The *continuation value* is $V_{t+2}(b_{t+2})$ and is taken as given.

- Decisions today affect future via changes in state variables, not via changes in future policy functions.
  - This *time consistency* of decision rules is known as the Principle of Optimality.

We can transform this problem into an unconstrained one.

- What is the *control* variable? What is the state variable?
The F.O.C. for the unconstrained problem is:

\[ u'(b_{t+1} - \frac{b_{t+2}}{1+r}) = \beta(1+r)V'_t(b_{t+2}). \]

Intuition?

Given \( b_{t+1} \) we have

\[ b_{t+2}^* = g_{t+1}(b_{t+1}) \]

and from there we can solve for the optimal consumption from the budget constraint:

\[ c_{t+1}^* = f_{t+1}(b_{t+1}). \]

Note that these policy functions are time-dependent.

Write down the value function representation.
Solving the Three-Period Problem: \( t \)

\[
\max_{c_t, b_{t+1}} U_0 = u(c_t) + \beta V_{t+1}(b_{t+1})
\]

subject to:

\[
c_t + \frac{b_{t+1}}{1 + r} = b_t.
\]

▶ The F.O.C. looks familiar.
  ▶ From there we obtain \( b_{t+1}^* \) and \( c_t^* \).
  ▶ While we solve backwards in time, we now can use the optimal policy functions to move forward, starting from an initial \( b_t \).
Method Comparison: Lagrange vs. Dynamic Programming

We will compare the methods with a very simple utility model.
Simple Model: Lagrange Method

Let’s solve:

\[
\max_{\{c_0, c_1, c_2, b_1, b_2\}} U_0 = \ln(c_0) + \beta \ln(c_1) + \beta^2 \ln(c_2)
\]

subject to:

\[
\begin{align*}
c_0 + \frac{b_1}{1+r} &= b_0 \\
c_1 + \frac{b_2}{1+r} &= b_1 \\
c_2 + \frac{b_3}{1+r} &= b_2
\end{align*}
\]

\[b_0 \geq 0 \text{ given.}\]

▶ Note that the solution is in recursive form.
As before, we use backward induction.
- We know what will happen in period 2.
- We start with
  \[
  \max_{\{c_1, b_2\}} \ln(c_1) + \beta V_2(b_2)
  \]
subject to:
  \[
  c_1 + \frac{b_2}{1 + r} = b_1
  \]
- From there, let’s go to period 1 and 0.
- Clearly, \( \tilde{f}_i = f_i \) and \( \tilde{g}_i = g_i \) for \( i = 0, 1 \).
- Consistent with Principle of Optimality.
From Period $t$ to $t + T$

If $T \to \infty$, the problem

$$V_t(b_t) = \sum_{i=0}^{T} \beta^i u(c_{t+i}^*)$$

$$= \max_{c_t, b_{t+1}} u(c_t) + \beta V_{t+1}(b_{t+1})$$

subject to:

$$c_t + \frac{b_{t+1}}{1 + r} = b_t$$

can be expressed as

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

subject to:

$$c_t + \frac{b_{t+1}}{1 + r} = b_t.$$

- Value functions and policy functions will be *time-autonomous.*
Transversality Condition

- In the finite horizon we implicitly ruled out dying with debt.
  - Now we have a similar condition: *transversality condition*.
- Let’s put the income process back into the problem.
- After some work, we find that the condition is given by
  \[ \lim_{n \to \infty} \left( \frac{1}{1 + r} \right)^n b_{t+n} = 0. \]
  - A relatively weak condition.
- If we let \( n \to \infty \) and we impose the transversality condition we have
  \[ \sum_{i=0}^{\infty} \left( \frac{1}{1 + r} \right)^i c_{t+i} = b_t + \sum_{i=0}^{\infty} \left( \frac{1}{1 + r} \right)^i y_{t+i}. \]
A General Version

- We have the problem

\[
\max_{\{u_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)
\]

subject to:

\[
x_{t+1} = g(x_t, u_t)
\]

\[x_0 \text{ given.}\]

- \(r(x_t, u_t)\) is a concave return or payoff function.
- What are the state and the control variables?
- Condition on transition equation:
  - It must represent a convex and compact set.
- Mapping a known model into this problem.
If we didn’t know dynamic programming we would have to find the *optimal infinite sequence* of choice variables.

What about a stochastic setting?

If we could solve for that sequence, we could obtain $V(x_0)$.

Dynamic programming offers an alternative:

We seek the optimal *time-invariant* policy function.

With the transition equation, this gives us the sequence $\{u_t^*\}_{t=0}^\infty$.
If we knew $V(x)$, we could find $h(x)$ by solving:

$$\max_{u,x'} u \cdot r(x,u) + \beta V(x')$$

subject to:

$$x' = g(x,u).$$

For each $x$, compute $h(x)$.

No time subscripts.

If we knew $h(x)$ we could solve the following for $V(x)$:

$$V(x) = r(x,h(x)) + \beta V(g(x,h(x))).$$

For each $x$, compute $V(x)$.

However...

We don’t know $h(x)$ nor $V(x)$!

Have to solve for them simultaneously.
We will solve for $h(x)$ and $V(x)$ simultaneously, by solving the Bellman equation:

$$V(x) = \max_u r(x,u) + \beta V(g(x,u))$$

How do we know that the recursive solution to this problem is the same as the solution to the sequential problem?

Under appropriate assumption the operator in the right side is a contraction mapping.

The $V$ that solves the Bellman equation is a fixed point of the operator.

When is an operator a contraction mapping?

Blackwell’s sufficient conditions.

Why do we care?

When operators are contractions, repeated application of the operator will converge to the fixed point.
Properties of the Bellman Equation

Under ‘standard’ assumptions:

1. There is a unique solution, \( V \), to the Bellman equation and it is strictly concave.
2. As \( j \to \infty \), \( V \) is approached in the limit by iterating on

\[
V_{j+1}(x) = \max_u r(x, u) + \beta V_j(g(x, u)).
\]

3. There is a unique and time-invariant policy function.
   - \( V(x) \) is differentiable.
Stochastic Dynamic Programming

- Not too much changes.
  - Add expectation operator.
  - Might add another state variable.
    - The stochastic element may become a state variable.

- Example: stochastic growth model with persistent productivity shocks:

\[
V(k, z) = \max_{\{c, k'\}} \left\{ u(c) + \beta \mathbb{E}_{z'|z} V(k', z') \right\}
\]

subject to:

\[
k' + c = zf(k).
\]

- \( z \) would be a state variable even if it is i.i.d.
Inspired on 2.

1. Make an initial guess about $V$, call it $V_i$.
2. Use this guess on the r.h.s. of the Bellman equation.
3. Maximizing on the r.h.s. yields new $V$, call it $V_{i+1}$.
4. Use this new $V_{i+1}$ as the $V_i$ on the r.h.s. in the next iteration.
5. Iterate until $V_i$ and the resulting $V_{i+1}$ are nearly equal.

This rarely works in closed form.

When solving numerically, the key is how to represent functions like $V$. 

Advanced Macroeconomics Dynamic Programming
Solution Techniques: Guess and Verify

- Guess functional form and verify that it solves the Bellman equation.
- Only works with problems that have a closed-form solution.
Solution Techniques: Policy Function Iteration

Also known as “Howard’s Improvement Algorithm”.

- ‘Opposite’ to VFI.

1. Start with a guess for the policy function, $h(x)$.
2. Calculate the value of following $h(x)$ forever.
3. Calculate new policy function by maximizing r.h.s. of the Bellman with the $V$ from the previous step.
4. Iterate until convergence.