Macro - 7020: TA Session 8

Paul Bousquet

University of Virginia

October 2023

Dynamic Programming: Why?

Beauty of DP: take a function with an infinite dimensional vector of choice variables $\{y_t \in \mathbb{R}^k\}_{t\geq 0}$ and represent it as a function with one vector of choice variables $\{y \in \mathbb{R}^k\}$

But we need to make sure such a representation *exists*

- Largely, what that means is that the Bellman setup has a *fixed point* (like $f(x) = x$)
- Every first year Macro class: proof that, under palatable assumptions, Bellman representation of lifetime utility maximization problem has a fixed point
- The distinction is the Bellman operator maps functions to functions

Dynamic Programming

Bellman operator:

$$(Tv)(x) = \max_y \left\{F(x,y) + \beta v(x')\right\} \text{ s.t } x' = g(x,y)$$

A fixed point w.r.t this operator is $Tv = v$.

If we don't have a fixed point, the Bellman operator just returns some function

In practice, we want to find the fixed point. To add onto the beauty, if we give the Bellman operator any initial guess (function), we will find it

$$Tv^0 = v^1 \implies Tv^1 = v^2 \implies \cdots \implies \lim_n Tv^n = v$$

This is the contraction map property

Dynamic Programming: Iteration Example

Let's see this in action with basic asset accumulation model w/ inelastic utility

$c = a - s$; $a' = Rs$. Let $F(a,s) = \frac{(a-s)^{1-\sigma}}{1-\sigma}$. Plug $z(x) = 0$ in Bellman operator

$$\implies (Tz)(a) = \max_s \{F(a,s)\} = F(a,s^*)$$

In this case $s^* = 0$.

Dynamic Programming: Iteration Example

Let's see this in action with basic asset accumulation model w/ inelastic utility

$c = a - s;\ a' = Rs$. Let $F(a,s) = \frac{(a-s)^{1-\sigma}}{1-\sigma}$. Plug $z(x) = 0$ in Bellman operator

$$\implies (Tz)(a) = \max_{s} \{F(a,s)\} = F(a,s^*)$$

In this case $s^* = 0$. So we have a *new* function, call it $z^1 = \frac{a^{1-\sigma}}{1-\sigma}$. Let $r = R^{-1}$

$$\implies (Tz^1)(a) = \max_{a'} \left\{ \frac{(a - ra')^{1-\sigma}}{1-\sigma} + \beta \frac{(a')^{1-\sigma}}{1-\sigma} \right\} = z^2(a)$$

Dynamic Programming: Iteration Example

Let's see this in action with basic asset accumulation model w/ inelastic utility

$c = a - s$; $a' = Rs$. Let $F(a,s) = \frac{(a-s)^{1-\sigma}}{1-\sigma}$. Plug $z(x) = 0$ in Bellman operator

$$\implies (Tz)(a) = \max_s \{F(a,s)\} = F(a,s^*)$$

In this case $s^* = 0$. So we have a *new* function, call it $z^1 = \frac{a^{1-\sigma}}{1-\sigma}$. Let $r = R^{-1}$

$$\implies (Tz^1)(a) = \max_{a'} \left\{ \frac{(a - ra')^{1-\sigma}}{1-\sigma} + \beta \frac{(a')^{1-\sigma}}{1-\sigma} \right\} = z^2(a)$$

Iteration: For all $\varepsilon > 0$, $\exists N$ s.t $n > N \implies ||z^n - z^{n-1}|| < \varepsilon$

Dynamic Programming: Iteration w/ Policy Function

We can also take a look at the previous example through the lens of a *policy function*

Policy function: state $\rightarrow$ ($\underline{\text{optimal choice}}$ of) control

Dynamic Programming: Iteration w/ Policy Function

We can also take a look at the previous example through the lens of a *policy function*

Policy function: state $\to$ (optimal choice of) control

In other words, let $\pi(x) = y^*$ be the policy function. Then

$$(Tv)(x) = \max_{y} \{F(x,y) + \beta v(g(x,y))\} = F(x,\pi(x)) + \beta v\big(g(x,\pi(x))\big)$$

Dynamic Programming: Iteration w/ Policy Function

We can also take a look at the previous example through the lens of a *policy function*

Policy function: state $\rightarrow$ (optimal choice of) control

In other words, let $\pi(x) = y^*$ be the policy function. Then

$$(Tv)(x) = \max_y \{F(x,y) + \beta v(g(x,y))\} = F(x,\pi(x)) + \beta v\big(g(x,\pi(x))\big)$$

In our previous example we had $z^2(a) = \max_{a'} \left\{ \frac{(a-ra')^{1-\sigma}}{1-\sigma} + \beta \frac{(a')^{1-\sigma}}{1-\sigma} \right\}$. We could also write

$$\textbf{FOC:}\ (a-ra')^{-\sigma} = \beta(a')^{-\sigma} \implies \pi^1(a) = \frac{a}{1+r}$$

Dynamic Programming: Iteration w/ Policy Function

We can also take a look at the previous example through the lens of a *policy function*

Policy function: state $\to$ (optimal choice of) control

In other words, let $\pi(x) = y^*$ be the policy function. Then

$$(Tv)(x) = \max_y \{F(x,y) + \beta v(g(x,y))\} = F(x,\pi(x)) + \beta v\big(g(x,\pi(x))\big)$$

In our previous example we had $z^2(a) = \max_{a'} \left\{ \frac{(a-ra')^{1-\sigma}}{1-\sigma} + \beta \frac{(a')^{1-\sigma}}{1-\sigma} \right\}$. We could also write

$$\textbf{FOC: } (a - ra')^{-\sigma} = \beta (a')^{-\sigma} \implies \pi^1(a) = \frac{a}{1+r}$$

$$\implies z^2(a) = \frac{(a - r\pi^1(a))^{1-\sigma}}{1-\sigma} + \beta \frac{\pi^1(a)^{1-\sigma}}{1-\sigma}$$

Dynamic Programming: Iteration w/ Policy Function

We can also take a look at the previous example through the lens of a *policy function*

Policy function: state $\to$ (optimal choice of) control

In other words, let $\pi(x) = y^*$ be the policy function. Then

$$(Tv)(x) = \max_y \{F(x,y) + \beta v(g(x,y))\} = F(x,\pi(x)) + \beta v\big(g(x,\pi(x))\big)$$

In our previous example we had $z^2(a) = \max_{a'} \left\{ \frac{(a-ra')^{1-\sigma}}{1-\sigma} + \beta \frac{(a')^{1-\sigma}}{1-\sigma} \right\}$. We could also write

$$\textbf{FOC: } (a - ra')^{-\sigma} = \beta (a')^{-\sigma} \implies \pi^1(a) = \frac{a}{1+r}$$

$$\implies z^2(a) = \frac{(a - r\pi^1(a))^{1-\sigma}}{1-\sigma} + \beta \frac{\pi^1(a)^{1-\sigma}}{1-\sigma}$$

Which leads to

$$z^3(a) = (Tv^2)(a) = \max_{a'} \left\{ \frac{(a-ra')^{1-\sigma}}{1-\sigma} + \beta \left[\frac{(a' - r\pi^1(a'))^{1-\sigma}}{1-\sigma} + \beta \frac{\pi^1(a')^{1-\sigma}}{1-\sigma} \right] \right\}$$

Dynamic Programming: Proof Guide

- We can use Blackwell's Theorem because of Theorem of the Max (key result: T maps continuous functions to continuous functions)
- Blackwell's Theorem tells use T is a contraction map
    - Key Assumptions: Monotonicity and Discounting
- Contraction Mapping Theorem: given any continuous, function v_0, we can generate a sequence using $v_n = Tv_{b-1} = T^n v_0$, which limits to a fixed point v
- If X convex, $\Gamma : X \to X$ well-defined (non-empty, compact, and continuous), and objective function bounded and continuous, this fixed point is unique.

Dynamic Programming: Summarizing/Bellman Equation

All the hard work we've done is to earn the right to just write the following

$$v(x) = \max_{y} \left\{ F(x,y) + \beta v(x') \right\} \text{ s.t } x' = g(x,y)$$

This is the Bellman equation. From now on you can start here, though you may have to say why you're able to (does problem satisfy assumptions from last slide)

Something to unpack in Econ context: we don't always control the evolution of all states

Endogenous state variables are contained within y.

For instance, household treats aggregate capital as exogenous, and both HH and social planner have no control over exogenous, stochastic processes

Dynamic Programming: Full Example

Model: Habit Persistence. The motivation is people generally dislike change and unlike some models, the comovement of L_t, C_t is not restricted (Barro-King Critique)

Let $A_{t+1} = R(A_t - c_t)$. Say agents want to maximize

$$\sum_{t=0}^{\infty} \beta^t u(c_t, c_{t-1})$$

Dynamic Programming: Full Example

There are two state variables

- yesterday's consumption (c_{t-1})
- today's assets (A_t)

We cannot change anything about these values.

They are the result of decisions made in the previous period.

Dynamic Programming: Full Example

There are two control variables:

- today's consumption (c_t)
- tomorrow's assets (A_{t+1})

Essentially, we have to decide how much we want to spend vs. how much we will save

However, once we've decided how much to spend, that automatically/implicitly determines how much we will save (everything left over from today's assets).

And vice versa.

Dynamic Programming: Full Example

Another way of putting it: say you have A_t in your bank account.

- Then you decide to spend c_t.
- Then you have $A_t - c_t$ left in your bank account, which earns interest R
- So tomorrow you will have $R(A_t - c_t)$
  - (Here $R > 1$ –an interest rate of 3% implies $R = 1.03$, for instance)

Dynamic Programming: Full Example

My approach is usually to substitute things out when given the chance.

So let $r = R^{-1}$. Then $c_t = A_t - rA_{t+1}$.

This means our new problem is

$$\sum_{t=0}^{\infty} \beta^t u(A_t - rA_{t+1}, c_{t-1})$$

Dynamic Programming: Full Example

To put this in terms of dynamic programming, we formulate the **Bellman equation**

Two parts of Bellman: utility we get today and utility we will get from the future.

Specifically, if we choose A_{t+1} today

- we get utility $u(A_t - rA_{t+1}, c_{t-1})$
- Then tomorrow, A_{t+1} is now a state variable (so is c_t), so we make a decision for A_{t+2} based on the new states.
- Then the period after that, A_{t+2} becomes a state variable and so on.

The bellman equation says that as we make this choice over and over again, we must be using the same rule or logic to make these decisions.

Dynamic Programming: Full Example (arriving at Bellman)

Formally, let $A^t_> = \{A_{t+1}, A_{t+2}, \dots\}$ be the set of all assets beyond period t.

The value function returns the value of utility (expected value in stochastic setups) when the optimal path is followed given initial state variables.

So here

$$v(A_t, c_{t-1}) = \max_{A^t_>} \left\{ \sum_{i=0}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, c_{t-1+i}) \right\}$$

Note that the indexing is flexible; the "argmax" can remain the same, for instance:

$$\operatorname*{argmax}_{A^t_>} \left\{ \sum_{i=0}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, c_{t-1+i}) \right\} = \operatorname*{argmax}_{A^t_>} \left\{ \sum_{i=0}^{\infty} \beta^{t+k} u(A_{t+i} - rA_{t+1+i}, c_{t-1+i}) \right\}$$

Dynamic Programming: Full Example (arriving at Bellman)

Therefore, substituting for c_{t-1+i} when $i > 0$, we have

$$v(A_t, c_{t-1}) = \max_{A^t_>} \left\{ \sum_{i=0}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, c_{t-1+i}) \right\}$$

Dynamic Programming: Full Example (arriving at Bellman)

Therefore, substituting for c_{t-1+i} when $i > 0$, we have

$$
\begin{aligned}
v(A_t, c_{t-1}) &= \max_{A^t_>} \left\{ \sum_{i=0}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, c_{t-1+i}) \right\} \\
&= \max_{A_{t+1}} \left\{ u(A_t - rA_{t+1}, c_{t-1}) + \max_{A^{t+1}_>} \left\{ \sum_{i=1}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, A_{t-1+i} - rA_{t+i}) \right\} \right\}
\end{aligned}
$$

Dynamic Programming: Full Example (arriving at Bellman)

Therefore, substituting for c_{t-1+i} when $i > 0$, we have

$$v(A_t, c_{t-1}) = \max_{A_>^t} \left\{ \sum_{i=0}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, c_{t-1+i}) \right\}$$

$$= \max_{A_{t+1}} \left\{ u(A_t - rA_{t+1}, c_{t-1}) + \max_{A_>^{t+1}} \left\{ \sum_{i=1}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, A_{t-1+i} - rA_{t+i}) \right\} \right\}$$

This last line implies that for any fixed integer $k \geq 0$

$$v(A_{t+k}, c_{t+k-1}) = \max_{A_{t+1+k}} \left\{ u(A_{t+k} - rA_{t+1+k}, c_{t+k-1}) + \max_{A_>^{t+1+k}} \left\{ \sum_{i=1+k}^{\infty} \beta^i u(A_{t+i+k} - rA_{t+1+i+k}, A_{t-1+i+k} - rA_{t+i+k}) \right\} \right\}$$

Dynamic Programming: Full Example (arriving at Bellman)

Therefore, substituting for c_{t-1+i} when $i > 0$, we have

$$v(A_t, c_{t-1}) = \max_{A^t_>} \left\{ \sum_{i=0}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, c_{t-1+i}) \right\}$$

$$= \max_{A_{t+1}} \left\{ u(A_t - rA_{t+1}, c_{t-1}) + \max_{A^{t+1}_>} \left\{ \sum_{i=1}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, A_{t-1+i} - rA_{t+i}) \right\} \right\}$$

This last line implies that for any fixed integer $k \geq 0$

$$v(A_{t+k}, c_{t+k-1}) = \max_{A_{t+1+k}} \left\{ u(A_{t+k} - rA_{t+1+k}, c_{t+k-1}) + \max_{A^{t+1+k}_>} \left\{ \sum_{i=1+k}^{\infty} \beta^i u(A_{t+i+k} - rA_{t+1+i+k}, A_{t-1+i+k} - rA_{t+i+k}) \right\} \right\}$$

Moreover, for $k = 1$ this simplifies to

$$v(A_{t+1}, c_t) = \max_{A_{t+2}} \left\{ u(A_{t+1} - rA_{t+2}, A_t - rA_{t+1}) + \max_{A^{t+2}_>} \left\{ \sum_{i=2}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, A_{t-1+i} - rA_{t+i}) \right\} \right\}$$

Dynamic Programming: Full Example (arriving at Bellman)

Continuing our substitution process, we have (let $v(\cdot) = v(A_t, c_{t-1})$)

$$
\begin{aligned}
v(\cdot) &= \max_{A^t_>} \left\{ \sum_{i=0}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, c_{t-1+i}) \right\} \\
&= \max_{A_{t+1}} \left\{ u(A_t - rA_{t+1}, c_{t-1}) + \max_{A^{t+1}_>} \left\{ \sum_{i=1}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, A_{t-1+i} - rA_{t+i}) \right\} \right\}
\end{aligned}
$$

Dynamic Programming: Full Example (arriving at Bellman)

Continuing our substitution process, we have (let $v(\cdot) = v(A_t, c_{t-1})$)

$$
\begin{aligned}
v(\cdot) &= \max_{A^t_>} \left\{ \sum_{i=0}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, c_{t-1+i}) \right\} \\
&= \max_{A_{t+1}} \left\{ u(A_t - rA_{t+1}, c_{t-1}) + \max_{A^{t+1}_>} \left\{ \sum_{i=1}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, A_{t-1+i} - rA_{t+i}) \right\} \right\} \\
&= \max_{A_{t+1}} \left\{ u(A_t - rA_{t+1}, c_{t-1}) + \beta \underbrace{\max_{A_{t+2}} \left\{ u(A_{t+1} - rA_{t+2}, A_t - rA_{t+1}) + \max_{A^{t+2}_>} \left\{ \sum_{i=2}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, A_{t-1+i} - rA_{t+i}) \right\} \right\} }_{= v(A_{t+1}, c_t) \, = \, v(A_{t+1}, A_t - rA_{t+1})} \right\}
\end{aligned}
$$

Dynamic Programming: Full Example (arriving at Bellman)

Continuing our substitution process, we have (let $v(\cdot) = v(A_t, c_{t-1})$)

$$
\begin{aligned}
v(\cdot) &= \max_{A^t_>} \left\{ \sum_{i=0}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, c_{t-1+i}) \right\} \\
&= \max_{A_{t+1}} \left\{ u(A_t - rA_{t+1}, c_{t-1}) + \max_{A^{t+1}_>} \left\{ \sum_{i=1}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, A_{t-1+i} - rA_{t+i}) \right\} \right\} \\
&= \max_{A_{t+1}} \left\{ u(A_t - rA_{t+1}, c_{t-1}) + \underbrace{\beta \max_{A_{t+2}} \left\{ u(A_{t+1} - rA_{t+2}, A_t - rA_{t+1}) + \max_{A^{t+2}_>} \left\{ \sum_{i=2}^{\infty} \beta^i u(A_{t+i} - rA_{t+1+i}, A_{t-1+i} - rA_{t+i}) \right\} \right\}}_{= v(A_{t+1}, c_t) \,=\, v(A_{t+1}, A_t - rA_{t+1})} \right\} \\
&= \max_{A_{t+1}} \left\{ u(A_t - rA_{t+1}, c_{t-1}) + \beta v(A_{t+1}, A_t - rA_{t+1}) \right\}
\end{aligned}
$$

Important note: I'm assuming existence here (you will rigorously prove that in class)

<u>On exam</u>: when asked to write down Bellman, start here!

Dynamic Programming: Full Example

Returning to the derivation for our earlier problem, we ended with

$$v(A_t, c_{t-1}) = \max_{A_{t+1}} \left\{ u(A_t - rA_{t+1}, c_{t-1}) + \beta v(A_{t+1}, A_t - rA_{t+1}) \right\}$$

Taking FOC

$$ru_1(A_t - rA_{t+1}, c_{t-1}) = \beta \Big(v_1(A_{t+1}, A_t - rA_{t+1}) - rv_2(A_{t+1}, A_t - rA_{t+1}) \Big)$$

Every time you see a v_i (a derivative of the value function), you need to replace that.

Let $v(\boldsymbol{t+1}) = v(A_{t+1}, A_t - rA_{t+1})$, meaning $v(\boldsymbol{t}) = v(A_t, c_{t-1})$.

Dynamic Programming: Full Example

Take the derivative of $v(\boldsymbol{t})$ with respect to the variables in $\boldsymbol{t}$.

For example, the derivative with respect to the first variable is $v_1(\boldsymbol{t})$.

If you take all the variables in $v_1(\boldsymbol{t})$

- move them forward one period (e.g, change $t-1$ to t in your expression)
- the envelope theorem says you will have an expression for $v_1(\boldsymbol{t+1})$.

Dynamic Programming: Full Example (Envelope Theorem)

Here is a more general way of phrasing that.

Let t_i be the ith variable in $\boldsymbol{t}$. Let $\boldsymbol{x}_t$ denote all the variables in a given derivative of a value function. Define $G(\boldsymbol{x}_t) = \frac{\partial v(\boldsymbol{t})}{\partial t_i}$. Then

$$v_i(\boldsymbol{t}) = G(\boldsymbol{x}_t) \implies v_i(\boldsymbol{t}+\boldsymbol{1}) = G(\boldsymbol{x}_{t+1})$$

General Envelope Theorem: Derivative of the value function $H(x, \pi(x))$ w.r.t x_i is equal to the derivative of the Lagrange w.r.t x_i evaluated at the optimum

Dynamic Programming: Full Example (Envelope Theorem)

Returning to our problem, this implies

$$v_1(\boldsymbol{t}) = u_1(A_t - rA_{t+1}, c_{t-1}) + \beta v_2(\boldsymbol{t+1}) \;\; \underline{\textbf{and}} \;\; v_2(\boldsymbol{t}) = u_2(A_t - rA_{t+1}, c_{t-1})$$

These are our two envelope conditions. Notice that you can update the first one by using the second. The first envelope condition becomes

$$v_1(\boldsymbol{t}) = u_1(A_t - rA_{t+1}, c_{t-1}) + \beta u_2(A_{t+1} - rA_{t+2}, A_t - rA_{t+1})$$

Dynamic Programming: Full Example

Plugging this into the FOC and subbing back in consumption (and $R = r^{-1}$)

$$u_1(c_t, c_{t-1}) + \beta u_2(c_{t+1}, c_t) = R\beta[u_1(c_{t+1}, c_t) + \beta u_2(c_{t+2}, c_{t_1})]$$

Or if you prefer to leave assets in, you get a longer expression

$$\begin{aligned} ru_1(A_t - rA_{t+1}, c_{t-1}) = \beta[&u_1(A_{t+1} - rA_{t+2}, A_t - rA_{t+1}) \\ &\cdots + \beta u_2(A_{t+2} - rA_{t+3}, A_{t+1} - rA_{t+2}) - ru_2(A_{t+1} - rA_{t+2}, A_t - rA_{t+1})] \end{aligned}$$

Aside: this is a case where Lagrange multiplier isn't just marginal utility