Discrete State, Discrete Control Dynamic Programming

Kenneth L. Judd, Hoover Institution

April 20, 2020

Discrete-Time Dynamic Programming

- Objective:

$$E\left\{\sum_{t=1}^{T}\pi(x_t,\,u_t,\,t)+W(x_{T+1})\right\}, \tag{12.1.1}$$

  - X: set of states
  - $\mathcal{D}$: the set of controls
  - $\pi(x,\,u,\,t)$ payoffs in period t, for $x\in X$ at the beginning of period t, and control $u\in\mathcal{D}$ is applied in period t.
  - $D(x,t)\subseteq\mathcal{D}$: controls which are feasible in state x at time t.
  - $F(A;x,u,t)$: probability that $x_{t+1}\in A\subset X$ conditional on time t control and state

- Value function

$$V(x,t)\equiv\sup_{\mathcal{U}(x,t)}E\left\{\sum_{s=t}^{T}\pi(x_s,\,u_s,\,s)+W(x_{T+1})|x_t=x\right\}. \tag{12.1.2}$$

- Bellman equation

$$V(x,t)=\sup_{u\in D(x,t)}\pi(x,\,u,\,t)+E\left\{V(x_{t+1},\,t+1)|x_t=x,u_t=u\right\} \tag{12.1.3}$$

- Existence: boundedness of π is sufficient

Autonomous, Infinite-Horizon Problem:

- Objective:

$$\max_{u_t} E\left\{\sum_{t=1}^{\infty} \beta^t \pi(x_t,\, u_t)\right\} \tag{12.1.1}$$

  - X: set of states
  - $\mathcal{D}$: the set of controls
  - $D(x) \subseteq \mathcal{D}$: controls which are feasible in state x.
  - $\pi(x,\, u)$ payoff in period t if $x \in X$ at the beginning of period t, and control $u \in \mathcal{D}$ is applied in period t.
  - $F(A; x, u)$: probability that $x^+ \in A \subset X$ conditional on current control u and current state x.

- Value function definition: if $\mathcal{U}(x)$ is set of all feasible strategies starting at x.

$$V(x) \equiv \sup_{\mathcal{U}(x)} E\left\{\sum_{t=0}^{\infty} \beta^t \pi(x_t,\, u_t) \middle| x_0 = x\right\}, \tag{12.1.8}$$

- Bellman equation for $V(x)$

$$V(x) = \sup_{u \in D(x)} \; \pi(x,\, u) + \beta\, E\, \left\{ V(x^+) | x, u \right\} \equiv (TV)(x), \tag{12.1.9}$$

- Optimal policy function, $U(x)$, if it exists, is defined by

$$U(x) \; \in \arg \max_{u \in D(x)} \; \pi(x,\, u) + \beta\, E \left\{ V(x^+) | x, u \right\}$$

- Standard existence theorem:

Theorem 1 *If X is compact, $\beta < 1$, and π is bounded above and below, then the map*

$$TV = \sup_{u \in D(x)} \; \pi(x, u) + \beta E \left\{ V(x^+) \mid x, u \right\} \tag{12.1.10}$$

is monotone in V, is a contraction mapping with modulus β in the space of bounded functions, and has a unique fixed point.

Applications

- Economics
  - Business investment
  - Life-cycle decisions on labor, consumption, education
  - Portfolio problems
  - Economic policy
- Operations Research
  - Scheduling, queueing
  - Blood bank
  - See new book by Powell - "Approximate Dynamic Programming"
- Climate change
  - Business response to climate policies
  - Optimal policy response to global warming problems

Deterministic Growth Example

- Problem:

$$
\begin{aligned}
V(k_0) = \max_{c_t} \textstyle\sum_{t=0}^{\infty} \beta^t u(c_t),& \\
k_{t+1} = F(k_t) - c_t& \\
k_0 \text{ given}&
\end{aligned}
\tag{12.1.12}
$$

  - Euler equation:

$$u'(c_t) = \beta u'(c_{t+1}) F'(k_{t+1})$$

  - Bellman equation

$$V(k) = \max_c \; u(c) + \beta V(F(k) - c). \tag{12.1.13}$$

  - Solution to (12.1.12) is a policy function $C(k)$ and a value function $V(k)$ satisfying

$$0 = u'(C(k))F'(k) - V'(k) \tag{12.1.15}$$

$$V(k) = u(C(k)) + \beta V(F(k) - C(k)) \tag{12.1.16}$$

- (12.1.16) defines the value of an arbitrary policy function $C(k)$, not just for the optimal $C(k)$.
- The pair (12.1.15) and (12.1.16)
  - expresses the value function given a policy, and
  - a first-order condition for optimality.

Stochastic Growth Accumulation

- Problem:

$$V(k,\theta) = \max_{c_t,\ell_t} \; E\left\{\sum_{t=0}^{\infty} \beta^t \; u(c_t)\right\}$$
$$k_{t+1} = F(k_t, \theta_t) - c_t$$
$$\theta_{t+1} = g(\theta_t, \varepsilon_t)$$
$$\varepsilon_t : \text{ i.i.d. random variable}$$
$$k_0 = k, \;\; \theta_0 = \theta.$$

- State variables:
  - k: productive capital stock, endogenous
  - θ: productivity state, exogenous
- The dynamic programming formulation is

$$V(k,\theta) = \max_{c} \;\; u(c) + \beta E\{V(F(k,\theta) - c, \theta^+)|\theta\} \tag{12.1.21}$$
$$\theta^+ = g(\theta, \varepsilon)$$

- The control law $c = C(k, \theta)$ satisfies the first-order conditions

$$0 = u_c\left(C(k,\theta)\right) - \beta\, E\left\{u_c(C(k^+,\theta^+))F_k(k^+,\theta^+) \mid \theta\right\}, \tag{12.1.23}$$

where

$$k^+ \equiv F(k, L(k,\theta), \theta) - C(k,\theta),$$

General Stochastic Accumulation

- Problem:

$$\begin{aligned} V(k,\theta) = \max_{c_t,\ell_t} \; & E\left\{\sum_{t=0}^{\infty} \beta^t \, u(c_t,\ell_t)\right\} \\ k_{t+1} &= F(k_t,\ell_t,\theta_t) - c_t \\ \theta_{t+1} &= g(\theta_t,\varepsilon_t) \\ k_0 &= k, \;\; \theta_0 = \theta. \end{aligned}$$

- State variables:
  - k: productive capital stock, endogenous
  - θ: productivity state, exogenous

- The dynamic programming formulation is

$$V(k,\theta) = \max_{c,\ell} \; u(c,\ell) + \beta E\{V(F(k,\ell,\theta) - c, \theta^+)|\theta\}, \tag{12.1.21}$$

where θ^+ is next period's θ realization.

- Control laws $c = C(k,\theta)$ and $\ell = L(k,\theta)$ satisfy foc's

$$
\begin{aligned}
0 &= u_c(C(k,\theta), L(k,\theta)) F_k(k, L(k,\theta), \theta) - V_k(k,\theta), \\
0 &= u_\ell(C(k,\theta), L(k,\theta)) + F_\ell(k,\theta) u_c(C(k,\theta), L(k,\theta)).
\end{aligned}
$$

- Euler equation implies

$$
0 = u_c\left(C(k,\theta), L(k,\theta)\right) - \beta\, E\left\{u_c(C(k^+,\theta^+), \ell^+) F_k(k^+, \ell^+, \theta^+) \mid \theta\right\}, \tag{12.1.23}
$$

where next period's capital stock and labor supply are

$$
\begin{aligned}
k^+ &\equiv F(k, L(k,\theta), \theta) - C(k,\theta), \\
\ell^+ &\equiv L(k^+, \theta^+),
\end{aligned}
$$

Discrete State Space, Discrete Control Problems

- Special structure
- Illustrate basic algorthmic ideas

Definition

- State space $X = \{x_i,\ i = 1, \cdots, n\}$
  - Wealth
  - Education, job experience
  - Capital
- Controls $\mathcal{D} = \{u_i | i = 1, ..., m\}$
  - Investment
  - Time for education, learning
- Choice of controls determines changes in state
  - $q_{ij}^t(u) = \Pr(x_{t+1} = x_j | x_t = x_i, u_t = u)$
  - $Q^t(u) = \left(q_{ij}^t(u)\right)_{i,j}$: Markov transition matrix at t if $u_t = u$.
- $\pi(x, u, t)$ Payoff at time at time t if state is $x \in X$ and control is $u \in \mathcal{D}$

Finite Horizon Problem

- Terminal value:

$$V_i^{T+1} = W(x_i), \ i = 1, \cdots, n.$$

- Value function, V_i^t, is the present value of payoffs if in state x_i at time t
  - We often implicitly assume that we use the optimal policy
  - This is really a vector of length n
- Bellman equation: time t value function is

$$V_i^t = \max_{u \in \mathcal{D}} [\pi(x_i, u, t) + \beta \sum_{j=1}^{n} q_{ij}^t(u) \, V_j^{t+1}], \ i = 1, \cdots, n$$

- Bellman equation can be directly computed by value function iteration:
  - max problem is a finite operation: unique value, but not unique solution u
  - Given V^{t+1} compute V^t, for $t = T, T-1, T-2, ...1$
  - Only choice for finite-horizon problems because the problem is not stationary.

Infinite Horizon Problems

- Infinite-horizon problems
- Bellman equation is now:

$$V_i = \max_{u \in \mathcal{D}} \left[\pi(x_i, u) + \beta \sum_{j=1}^{n} q_{ij}(u)\, V_j \right], \; i = 1, \cdots, n$$

- This is a finite system of equations for the unknowns $V_i, i = 1, ..., n$

Value Function Iteration

- VFIValue function iteration is now

$$V_i^{k+1} = \max_{u \in \mathcal{D}} \left[\pi(x_i, u) + \beta \sum_{j=1}^{n} q_{ij}(u)\, V_j^k \right], \quad i = 1, \cdots, n$$

  - Begin with an initial (and arbitrary) V_i^0 and iterate $k \to \infty$.
  - Convergence implied by contraction mapping property
  - Error is given by contraction mapping property:

$$\left\| V^k - V^* \right\| \leq \frac{1}{1-\beta} \left\| V^{k+1} - V^k \right\|$$

Algorithm 12.1: Value Function Iteration Algorithm

Objective:	Solve the Bellman equation
Step 0:	Make initial guess V^0; choose stopping criterion $\epsilon > 0$.
Step 1:	For $i = 1, ..., n$, compute $V_i^{\ell+1} = \max_{u \in D} \ \pi(x_i, u) + \beta \sum_{j=1}^{n} q_{ij}(u) V_j^{\ell}$.
Step 2:	If $\parallel V^{\ell+1} - V^{\ell} \parallel < \epsilon$, then go to step 3; else go to step 1.
Step 3:	Compute the final solution, setting $U^* = \mathcal{U} V^{\ell+1}$, $P_i^* = \pi(x_i, U_i^*), \qquad i = 1, \cdots, n$, $V^* = (I - \beta Q^{U^*})^{-1} P^*$, and STOP.
Output:	

Value of a Policy

- Value function idea can be applied to an arbitrary policy
- Let $U \in \mathcal{D}^n$ denote the policy of choosing $U_i \in \mathcal{D}$ when in state x_i
- The present value, V, of policy U is defined by

$$V_i = \pi\left(x_i, U_i\right) + \beta \sum_{j=1}^{n} q_{ij}(U_i)\, V_j, \;\; i = 1, \cdots, n,$$

Policy Iteration (a.k.a. Howard improvement)

- Value function iteration is slow
  - Linear convergence at rate β
  - Convergence is particularly slow if β is close to 1.
- Policy iteration is
  - Current guess:
$$V_i^k, \ i = 1, \cdots, n.$$
  - Iteration: compute optimal policy today if V^k is value tomorrow:
$$U_i^{k+1} = \arg\max_{u \in \mathcal{D}} \left[\pi(x_i, u) + \beta \sum_{j=1}^{n} q_{ij}(u) \, V_j^k \right], \ i = 1, \cdots, n,$$
  - Compute the value function if the policy U^{k+1} is used forever, which is solution to the linear system
$$V_i^{k+1} = \pi\left(x_i, U_i^{k+1}\right) + \beta \sum_{j=1}^{n} q_{ij}(U_i^{k+1}) \, V_j^{k+1}, \ i = 1, \cdots, n,$$

- Comments:
- Policy iteration depends on only monotonicity
  - Policy iteration is faster than value function iteration
    - If initial guess is above or below solution then policy iteration is between truth and value function iterate
    - Works well even for β close to 1.

Algorithm 12.2: Policy Function Algorithm

Objective:	Solve the Bellman equation, (12.3.4).
Step 0:	Choose stopping criterion $\epsilon > 0$.
	EITHER make initial guess, V^0, for the value function and go to step 1,
	OR make initial guess, U^1, for the policy function and go to step 2.
Step 1:	$U^{\ell+1} = \mathcal{U} V^{\ell}$
Step 2:	$P_i^{\ell+1} = \pi\left(x_i, U_i^{\ell+1}\right), \qquad i = 1, \cdots, n$
Step 3:	$V^{\ell+1} = \left(I - \beta Q^{U^{\ell+1}}\right)^{-1} P^{\ell+1}$
Step 4:	If $\parallel V^{\ell+1} - V^{\ell} \parallel < \epsilon$, STOP; else go to step 1.

- Modified policy iteration
- If n is large, difficult to solve policy iteration step
  - Alternative approximation: Assume policy $U^{\ell+1}$ is used for k periods:

$$V^{\ell+1} = \sum_{t=0}^{k} \beta^t \left(Q^{U^{\ell+1}}\right)^t P^{\ell+1} + \beta^{k+1} \left(Q^{U^{\ell+1}}\right)^{k+1} V^{\ell}$$

  - Theorem 4.1 points out that as the policy function gets close to U^*, the linear rate of convergence approaches β^{k+1}. Hence convergence accelerates as the iterates converge.

(Putterman and Shin) The successive iterates of modified policy iteration with k steps, (12.4.1), satisfy the error bound

$$\frac{\left\| V^{*} - V^{\ell+1} \right\|}{\left\| V^{*} - V^{\ell} \right\|} \leq \min \left[\beta,\ \frac{\beta(1-\beta^{k})}{1-\beta} \parallel U^{\ell} - U^{*} \parallel + \beta^{k+1} \right]$$

Gaussian acceleration methods for infinite-horizon models

- Key observation: Bellman equation is a simultaneous set of equations

$$V_i = \max_{u \in \mathcal{D}} \left[\pi(x_i, u) + \beta \sum_{j=1}^{n} q_{ij}(u)\, V_j \right],\ \ i = 1, \cdots, n$$

- Idea: Treat problem as a large system of nonlinear equations
- Value function iteration is the *pre-Gauss-Jacobi* iteration

$$V_i^{k+1} = \max_{u \in \mathcal{D}} \left[\pi(x_i, u) + \beta \sum_{j=1}^{n} q_{ij}(u)\, V_j^k \right],\ \ i = 1, \cdots, n$$

- True Gauss-Jacobi is

$$V_i^{k+1} = \max_{u \in \mathcal{D}} \left[\frac{\pi(x_i, u) + \beta \sum_{j \neq i} q_{ij}(u)\, V_j^k}{1 - \beta q_{ii}(u)} \right],\ \ i = 1, \cdots, n$$

- pre-Gauss-Seidel iteration
  - Value function iteration is a pre-Gauss-Jacobi scheme.
  - Gauss-Seidel alternatives use new information immediately
    - Suppose we have V_i^ℓ
    - At each x_i, given $V_j^{\ell+1}$ for $j < i$, compute $V_i^{\ell+1}$ in a pre-Gauss-Seidel

- Gauss-Seidel iteration
- Suppose we have V_i^ℓ
  - If optimal control at state i is u, then Gauss-Seidel iterate would be

$$V_i^{\ell+1} = \pi(x_i, u) + \beta \frac{\sum_{j<i} q_{ij}(u) V_j^{\ell+1} + \sum_{j>i} q_{ij}(u) V_j^\ell}{1 - \beta q_{ii}(u)}$$

  - Gauss-Seidel: At each x_i, given $V_j^{\ell+1}$ for $j < i$, compute $V_i^{\ell+1}$

$$V_i^{\ell+1} = \max_{u \in \mathcal{D}} \frac{\pi(x_i, u) + \beta \sum_{j<i} q_{ij}(u) V_j^{\ell+1} + \beta \sum_{j>i} q_{ij}(u) V_j^\ell}{1 - \beta q_{ii}(u)}$$

  - Iterate this for $i = 1, .., n$
- Gauss-Seidel iteration: better notation
  - No reason to keep track of ℓ, number of iterations
  - At each x_i,

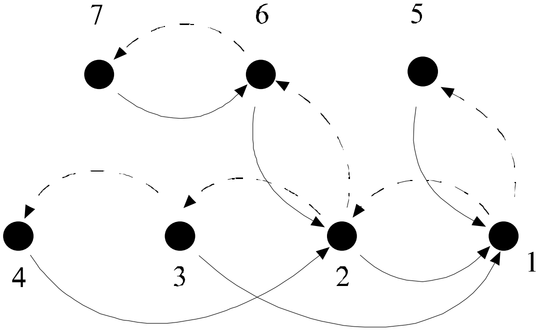
$$V_i \longleftarrow \max_{u \in \mathcal{D}} \frac{\pi(x_i, u) + \beta \sum_{j<i} q_{ij}(u) V_j + \beta \sum_{j>i} q_{ij}(u) V_j}{1 - \beta q_{ij}(u)}$$

  - Iterate this for $i = 1, .., n, 1, ..., $ etc.

State versus Information Flows

Consider the following graph:

- Solid arrows are permissible state transitions
- Broken arrows represent information flow



Upwind Gauss-Seidel

- Gauss-Seidel methods in (12.4.7) and (12.4.8)
- Sensitive to ordering of the states.
  - Need to find good ordering schemes to enhance convergence.
- Example:
  - Two states, x_1 and x_2, and two controls, u_1 and u_2
    - u_i causes state to move to x_i, $i = 1, 2$
    - Payoffs:

$$\begin{aligned} \pi(x_1, u_1) &= -1,\ \pi(x_1, u_2) = 0, \\ \pi(x_2, u_1) &= 0,\ \pi(x_2, u_2) = 1. \end{aligned}$$

    - $\beta = 0.9$.
  - Solution:
    - Optimal policy: always choose u_2, moving to x_2
    - Value function:

$$V(x_1) = 9,\ V(x_2) = 10.$$

    - x_2 is the unique steady state, and is stable

- Converges linearly:

$$V^1(x_1) = 0,\ V^1(x_2) = 1,\ U^1(x_1) = 2,\ U^1(x_2) = 2,$$
$$V^2(x_1) = 0.9,\ V^2(x_2) = 1.9,\ U^2(x_1) = 2,\ U^2(x_2) = 2,$$
$$V^3(x_1) = 1.71,\ V^3(x_2) = 2.71,\ U^3(x_1) = 2,\ U^3(x_2) = 2,$$

- Policy iteration converges after two iterations

$$V^1(x_1) = 0,\ V^1(x_2) = 1,\ U^1(x_1) = 2,\ U^1(x_2) = 2,$$
$$V^2(x_1) = 9,\ V^2(x_2) = 10,\ U^2(x_1) = 2,\ U^2(x_2) = 2,$$

- Upwind Gauss-Seidel
- Value function at absorbing states is trivial to compute
  - Suppose s is absorbing state with control u
    - $V(s) = \pi(s, u)/(1 - \beta)$.
  - With absorbing state $V(s)$ we compute $V(s')$ of any s' that sends system to s.
$$V\left(s'\right) = \pi\left(s', u\right) + \beta V\left(s\right)$$
  - With $V(s')$, we can compute values of states s'' that send system to s'; etc.

Alternative Orderings

It may be difficult to find proper order.

- Alternating Sweep
  - Idea: alternate between two approaches with different directions.

$$
\begin{aligned}
W &= V^k, \\
W_i &= \max_{u \in \mathcal{D}} \; \pi(x_i, u) + \beta \textstyle\sum_{j=1}^{n} q_{ij}(u) W_j, \;\; i = 1, 2, 3, ..., n \\
W_i &= \max_{u \in \mathcal{D}} \; \pi(x_i, u) + \beta \textstyle\sum_{j=1}^{n} q_{ij}(u) W_j, \;\; i = n, n-1, ..., 1 \\
V^{k+1} &= W
\end{aligned}
$$

  - Will always work well in one-dimensional problems since state moves either right or left, and alternating sweep will exploit this half of the time.
  - In two dimensions, there may still be a natural ordering to be exploited.
- Simulated Upwind Gauss-Seidel
  - It may be difficult to find proper order in higher dimensions
  - Idea: simulate using latest policy function to find downwind direction
    - Simulate to get an example path, $x_1, x_2, x_3, x_4, ..., x_m$
    - Execute Gauss-Seidel with states $x_m, x_{m-1}, x_{m-2}, ..., x_1$

Linear Programming Approach

- If $\mathcal{D}$ is finite, we can reformulate dynamic programming as a linear programming problem.
- (12.3.4) is equivalent to the linear program

$$\begin{array}{ll} \min_{V_i} \sum_{i=1}^{n} V_i & \\ s.t. \quad V_i \geq \pi(x_i, u) + \beta \sum_{j=1}^{n} q_{ij}(u) V_j, \ \forall i, u \in \mathcal{D}, \end{array} \tag{12.4.10}$$

- Computational considerations
  - (12.4.10) may be a large problem
  - Trick and Zin (1997) pursued an acceleration approach with success.
  - OR literature did not favor this approach, but recent work by Daniela Pucci de Farias and Ben van Roy has revived interest.