Dynamic Programming - II

Judd Chapter 12

David Childers (thanks to Y. Kryukov, K. Judd, and U. Doraszelski)

CMU, Tepper School of Business

April 10 - 12, 2023

Agenda

- We can solve any combination of features:
  - Time: Discrete ($\sum_{t=0}^{T} \beta^t$) or continuous ($\int_0^T e^{-\rho t}$)
  - State: discrete or continuous
  - State transition: deterministic or stochastic

1. Theory: Discrete time
2. Computation with discrete states: Value & Policy iterations
3. now: **Computation with continuous state**
   1. Discretization: reduce to previous model
   2. Value function iteration
   3. Projection
4. **Continuous time: mostly theory**
   1. Deterministic transition
   2. Stochastic transition: Jumps and Brownian motion

Continuous state: discretization

Write a discrete-state problem that's "similar" to the continuous-state one

- Support of x : grid of equally spaced points
  - Multidimensional $x \Rightarrow$ large grid
- Law of motion:
  - Easiest: assume determininstic transitions, but this can lead to weird solution
  - Alternative: choose $q_{ij}\left(u\right)$ so $\sum_{j=1}^{n} q_{ij}(u)V_j \approx V\left(x^{+}\left(x_i, u\right)\right)$. For sparsity: make $q_{ij}\left(u\right) = 0$ if j is far from i

Discretization example - optimal growth

$$
\begin{aligned}
V(k) &= \max_{c} u(c) + \beta V(k^{+}) \\
\text{s.t.} &: \quad k^{+} = f(k) - c
\end{aligned}
$$

- Discretize state: $k \in K = \{k_1, k_2, \ldots, k_n\}$
- With deterministic state transition
  - Equivalent to using next period's state as control

$$V(k) = \max_{k^{+} \in K} u(f(k) - k^{+}) + \beta V(k^{+})$$

- Stochastic state transition:

$$V(k) = \max_{c} u(c) + \beta \textstyle\sum_{j=1}^{n} V(k_j) q_{ij}(c)$$

Optimal Growth + deterministic transition

- Discrete time:

$$\begin{array}{rc} \max_{\{c_t\}} & \sum_{t=0}^{\infty} \beta^t u\left(c_t\right) \\ \text{subject to:} & k_{t+1} - k_t = f\left(k_t\right) - c_t \end{array}$$

- ... and state: $k \in K = \{k_1, k_2, \ldots, k_n\}$
- Constraint implies that: $c_t = k_t + f\left(k_t\right) - k_{t+1}$, so

$$\max_{\{k_t\}} \sum_{t=0}^{\infty} \beta^t u\left(k_t + f\left(k_t\right) - k_{t+1}\right)$$

  - I.e. we limit $c\left(k\right)$ to a discrete set of values
- Formulate Bellman equation:

$$V(k) = \max_{k^+ \in K} u(k + f(k) - k^+) + \beta V(k^+)$$

Cont. state: Approximation + Value iteration

- Approximate the value function as $\hat{V}(x;a)$,
  - Can use polynomials or splines
  - $a =$ vector of coefficients
- Keep iterating on Bellman equation
  - i.e. solving it at nodes x_i, $i = 1 : n$
- Use approximation to evaluate $\beta \mathbf{E} V\left(x^{+}\right)$
  - $\hat{V}(x;a)$ must be computed for any x (not just nodes)
- Updating V means updating the coefficients
  - Use solutions to Bellman at nodes

Cont. state: Approx + Value/Policy iteration

1. Pick nodes $\{x_i\}_{i=1}^n$ (e.g. poly. roots) & initial guess a^0.
2. Solve Bellman at nodes. For each $i = 1, \dots, n$:

$$V_i := \max_u \pi(x_i, u) + \beta \int \hat{V}(x^+; a^k) dF(x^+, x_i, u)$$

3. Update approximation:
   - Choose a^{k+1} to minimize $\left\{\hat{V}(x_i; a^{k+1}) - V_i\right\}_{i=1}^n$,
   - e.g. using Chebyshev formula or OLS
4. (Optional) $W_i := \pi(x_i, u_i^*) + \beta \int \hat{V}(x^+; a^{k+1}) dF(x^+, x_i, u_i^*)$
   - Return to 3 to approximate and repeat 3-4 for m steps
   - Or solve for $\{V_i\}_{i=1}^n$ given u^* directly for Policy Iteration
   - If interpolation and integration linear, this is linear system
5. If $||\hat{V}(x; a^{k+1}) - \hat{V}(x; a^k)|| < \epsilon$, stop; o/w, step 2.
   - Alternative criterion: $||a^{k+1} - a^k|| < \epsilon$

Cont. state: Approximation + Projection

- Original problem:

$$V(x) = \max_u \pi(x, u) + \beta \int V(x^+) dF(x^+, x, u)$$

- Compute FOC:

$$0 = \pi_u(x, u) + \beta \int V(x^+) dF_u(x^+, x, u)$$

- If FOC is sufficient, optimal $V(x)$ and $U(x)$ must satisfy:

$$V(x) = \pi(x, U(x)) + \beta \int V(x^+) dF(x^+, x, U(x)),$$
$$0 = \pi_u(x, U(x)) + \beta \int V(x^+) dF_u(x^+, x, U(x)).$$

Cont. state: Approximation + Projection

Solving for unknown $V(\cdot)$ and $U(\cdot)$

- Set up polynomial approximation $\hat{V}(x;a)$ and $\hat{U}(x;b)$
    - Degree N approximations $\Rightarrow$ $\{a,b\} \in \mathbb{R}^{2(N+1)}$
- Pick $N+1$ nodes x_i – e.g. polynomial roots
- System of equations, $i = 1 : N+1$:

$$\begin{aligned}
-\hat{V}(x_i) + \pi(x_i, \hat{U}(x_i)) + \beta \int \hat{V}(x^+) dF(x^+, x_i, \hat{U}(x_i)) &= 0 \\
\pi_u(x_i, \hat{U}(x_i)) + \beta \int \hat{V}(x^+) dF_u(x^+, x_i, \hat{U}(x_i)) &= 0
\end{aligned}$$

- where $\hat{V}(x_i) \equiv \hat{V}(x;a)$, $\hat{U}(x_i) \equiv \hat{U}(x;b)$
- Variables: $\{a,b\} \in \mathbb{R}^{2(N+1)}$

Continuous-time notation

- All variables are function of time: $x \equiv x(t)$, $u \equiv u(t)$
- Payoff $\pi(x, u)$ is a flow / rate, in \$/period:
    - Profit during the first year is $\int_0^1 \pi(x, u)\, dt$
    - Analogy: distance travelled in one hour is $\int_0^1 speed(t)\, dt$
- Dot = derivative w.r.t. time: $\dot{x} = \frac{\partial}{\partial t} x(t)$
    - Law of motion: $\dot{x} = f(x, u)$
    - $f(x, u)$ = rate of change in x, units per period.

Continuous time: deterministic transition

- Time, state & law of motion are all continuous:

$$\begin{array}{rl} \max_u & \int_0^\infty e^{-\rho t}\pi(x,u)\,dt \\ \text{subject to:} & \dot{x} = f(x,u), \qquad x(0) = x_0 \end{array}$$

- Bellman becomes Hamilton-Jacobi-Bellman:

$$\rho V(x) = \max_u \pi(x,u) + V'(x) f(x,u)$$

  - Note that equation is in terms of flows
- Intuition: $V(x)$ should remain constant if x does not change
  - $\pi(x,u)$ = flow of current profits
  - $V'(x)f(x,u)$ = flow of value from change in state
  - $\rho V(x)$ = decline in value over time

Continuous-time Bellman – derivation

- Discrete-time Bellman over time period $h \to 0$:

$$V(x) = \max_u h\pi(x,u) + e^{-h\rho} V(x + hf(x,u))$$

- Linear approximation to $e^{-h\rho}$ and $V(x + hf(x,u))$ at $h = 0$

$$V(x) = \max_u h\pi(x,u) + (1 - h\rho)\left[V(x) + hV'(x)f(x,u)\right]$$

- Open brackets and transform:

$$h\rho V(x) = \max_u h\pi(x,u) + (1 - h\rho)\, hV'(x)f(x,u)$$

- Divide by h: it cancels out in all but one term.
- Take limit at $h \to 0$:

$$\rho V(x) = \max_u \pi(x,u) + V'(x)f(x,u)$$

Continuous time + jumps in state

- Every once in a while, hurricane wipes out s capital
- *Poisson arrival process*: time between hurricanes follows Exponential distribution with mean $1/\mu$
- Bellman equation gets an extra term:

$$\rho V(x) = \max_u \pi(x,u) + V'(x) f(x,u) + \mu \left[V(x-s) - V(x) \right]$$

  - $\mu =$ arrival rate of event (events per year)
  - $V(x-s) - V(x) =$ change in value caused by the event
- Jumps allow for continuous time with discrete states
  - Independent event arrival $\implies$ only one at a given time
  - In discrete time, multiple events occur simultaneously $\implies$ expectation gets messy

Solving the above models

- If state is discrete, use discrete-state methods
  - Remember that l.h.s. of Bellman is $\rho V(x)$, not $V(x)$
  - Try to carry $V(x)$ out of objective
  - Otherwise, might need large dampening
- Continuous state - same methods as in discrete time
  - Value iteration with polynomial approximation
  - Projection using Bellman and FOC

Continuous & stochastic state

Based on lecture notes by Benjamin Moll of Princeton

- Discrete time = state follows a Markov chain process
  - e.g. random walk (AR(1) with $\rho = 1$):

$$x_{t+1} = x_t + \varepsilon_t, \varepsilon_t \sim N(0,1)$$

- Cont. time: want $x(t)$ to be continuous in t, and random
- Standard Brownian motion (Wiener) process $W(t)$, $t \in \mathbb{R}^+$:
  - $W(t + \Delta t) - W(t) = \left(\sqrt{\Delta t}\right)\varepsilon$, $\varepsilon \sim N(0,1)$
  - $W(0) = 0$
- Observations:
  - $W(t) \sim N(0,t)$
  - $t \in \{0,1,2,...\} \Rightarrow W(t)$ is a random walk

General diffusion process

- Introduce drift μ and variance σ:

$$x\left(t\right)=x\left(0\right)+\mu t+\sigma W\left(t\right)$$

  - Differential form:

$$dx=\mu dt+\sigma dW$$

  - From here on, we drop "$\left(t\right)$" as per convention
- Diffusion process:

$$dx=\mu\left(x\right)dt+\sigma\left(x\right)dW$$

$$x(t)=\int_0^t\mu\left(x\right)dt+\int_0^t\sigma\left(x\right)dW$$

- Choice of $\mu\left(x\right)$ and $\sigma\left(x\right)$ can allow highly general continuous x process:
  - Stationary: $\mu\left(x\right)=\theta\left(\bar{x}-x\right)$ (Ornstein-Uhlenbeck)
  - Bounded, e.g. to $\left[0,1\right]$: $\sigma\left(x\right)=\sigma x\left(1-x\right)$

Dynamic stochastic problem

- State x is now a diffusion:

$$\begin{aligned} \max_u \quad & \mathbf{E}_0 \int_0^\infty e^{-\rho t} \pi\left(x,u\right) dt \\ \text{s.t.:} \quad & dx = f\left(x,u\right) dt + \sigma\left(x,u\right) dW, \qquad x(0) = x_0 \end{aligned}$$

- HJB equation:

$$\rho V(x) = \max_u \pi\left(x,u\right) + V'(x) f(x,u) + \frac{1}{2}\sigma^2\left(x,u\right) V''\left(x\right)$$

  - Reverts to deterministic case if $\sigma = 0$
- It is nice to have a stationary state process
- Modeling asset price p: $x = \ln p \Rightarrow p > 0$

Theory behind HJB

Itô's lemma:

- x is a diffusion: $dx = \mu(x)\,dt + \sigma(x)\,dW$
- $f(x)$ is twice continuously differentiable
- Then:

$$df(x) = \left[\mu(x) f'(x) + \frac{1}{2}\sigma^2(x) f''(x)\right] dt + \sigma(x) f'(x)\,dW$$

- In HJB:
  - $f(x) = V(x)$
  - dt term $[...]$ goes into HJB objective
  - dW integrates out (since $\mathbf{E}dW = 0$)

Ito's lemma: Intuition

- For dt small, $(dW_t)^2 = dt$ since $W_t \sim O(\sqrt{t})$
- So $(dx_t)^2 = (\mu(x_t)dt + \sigma(x_t)dW_t)^2 = \sigma^2(x_t)dt$
- Let $f(x,t) \in C^2$, then Taylor expansion gives

$$
\begin{aligned}
df_t &= \frac{\partial f(x_t,t)}{\partial x}dx_t + \frac{\partial f(x_t,t)}{\partial t}dt + \frac{1}{2}\frac{\partial^2 f(x_t,t)}{\partial x^2}(dx_t)^2 \\
&= \frac{\partial f(x_t,t)}{\partial x}\left(\mu(x_t)dt + \sigma(x_t)dW_t\right) + \frac{\partial f(x_t,t)}{\partial t}dt \\
&\quad + \frac{1}{2}\frac{\partial^2 f(x_t,t)}{\partial x^2}\sigma^2(x_t)dt \\
&= \left(\frac{\partial f(x_t,t)}{\partial x}\mu(x_t) + \frac{\partial f(x_t,t)}{\partial t} + \frac{1}{2}\frac{\partial^2 f(x_t,t)}{\partial x^2}\sigma^2(x_t)\right)dt \\
&\quad + \frac{\partial f(x_t,t)}{\partial x}\sigma(x_t)dW_t
\end{aligned}
$$

- Recover above result in time-independent case

HJB Equation: Intuition

- Discrete-time Bellman over time period $h \to 0$:

$$V(x) = \max_u h\pi(x,u) + e^{-\rho h} E[V(x')|x]$$

- Taylor approx $e^{-\rho h} = \frac{1}{1+\rho h}$ and multiply

$$(1+\rho h)V(x) = \max_u h(1+\rho h)\pi(x,u) + E[V(x')|x]$$

- Subtract V and divide by h

$$\rho V(x) = \max_u (1+\rho h)\pi(x,u) + \frac{1}{h} E[V(x') - V(x)|x]$$

- Take h to 0

$$\rho V(x) = \max_u \pi(x,u) + \frac{E[dV(x)]}{dt}$$

HJB Equation

- Diffusion Limit of x and Ito's lemma for V give

$$dV(x) = (V'(x)f(x,u) + \frac{1}{2}\sigma^2(x,u)V''(x))dt + V'(x)\sigma(x,u)dW$$

- $E[dW] = 0$, giving

$$\rho V(x) = \max_u \pi(x,u) + V'(x)f(x,u) + \frac{1}{2}\sigma^2(x,u)V''(x)$$

- Optimal Policy is $u^*(x)$ solving FOC and Bellman equation

$$\pi_u(x,u^*(x)) + V'(x)f_u(x,u^*(x)) + \sigma(x,u)\sigma_u(x,u^*(x))V''(x)$$

$$\rho V(x) = \pi(x,u^*(x)) + V'(x)f(x,u^*(x)) + \frac{1}{2}\sigma^2(x,u^*(x))V''(x)$$

General: d-Dimensional Jump Diffusion

- $dX_t = \mu(X_t, u_t, t)dt + \sigma(X_t, u_t, t)dW_t + \sum_{k=1}^{p} \lambda_k(X_t, u_t, t)dN_{k,t}$
  - W_t $m \times 1$ iid Brownian motions, $\sigma(X_t, u_t, t)$ $d \times m$ volatility
  - $\{N_{k,t}\}_{k=1}^{p}$ independent Poisson, $\lambda_k(X_t, u_t, t)$ intensity
  - Jumps are Markov $X_{t+}|X_t \sim f_k(X'|X, u, t)$
- HJB equation takes form $\rho V(X, t) =$
  - $\max_u \pi(x, u) + \partial_t V(X, t) + \sum_{i=1}^{d} \mu_i(X, u, t) \frac{\partial}{\partial x_i} V(X, t)$
  - $+\frac{1}{2} \sum_{i,j} (\sigma(X, u, t)\sigma(X, u, t)^T)_{i,j} \frac{\partial^2}{\partial x_i \partial x_j} V(X, t)$
  - $+\sum_{k=1}^{p} \lambda_k(X, u, t)(\int V(X^+, t) f_k(X^+|X, u, t)dX^+ - V(X, t))$

Solving diffusion-based problems

- Value iteration with approximation to $V(\cdot)$
- Projection methods
- Existence/uniqueness issues for nonlinear PDE: 2^{nd} derivative may not exist everywhere (kink in value function)
  - $\rightarrow$ *Viscosity* solution is one generating optimum
  - Generalizes subgradient approach from convex optimization
  - Ensure this by using *upwind* variant of finite difference approach
  - Idea: use forward FD when drift > 0, backward when drift < 0
- See codes at Moll's webpage: `https://benjaminmoll.com/codes/`