OSM Lab - Dynamic Programming

Jason DeBacker
June 18-22, 2018
Notes for Lecture #2

Today:

- Sketch of proof of solution to $V(\cdot)$
- Computational Methods
  - Value Function Iteration
  - Interpolation
  - Policy Function Iteration
- The Stochastic Cake Eating Problem
  - Approximating AR(1) processes

General Dynamic Programming Problem:

- consider: $V(s) = \max_{c \in C(s)} \underbrace{\sigma(c,s)}_{\text{the payoff of period one}} + \beta V(s'),\ \forall s \in S$
- state: s
- control: c
- transition equation: $s' = \tau(s,c) \rightarrow$ what you have now (s) and what you choose (c) determines what you have tomorrow (s')
- or you could write:
- (**) $V(s) = \max_{s' \in \Gamma(s)} \tilde{\sigma}(s,s') + \beta V(s'),\ \forall s \in S$
- policy function: $s' = \phi(s)$
- The above is the general structure for the non-stochastic dynamic programming problem
- Existence of a solution:
  - Stokey and Lucas (1996), Theorem 4.6
  - Adda and Cooper (2003), page 24
  - If:
    - $\underbrace{\sigma(s,s')}_{\text{the payoff}}$ is continuous, real valued, and bounded (i.e., can be kept in a box)
    - $0 < \beta < 1$ (i.e., there's discounting)
    - $\Gamma(s)$ is non-empty, compact-valued ($\Gamma(s)$ maps into itself), continuous
  - Then $\exists$ a unique solution $V(s)$ solving $V(s) = \max_{s' \in \Gamma(s)} \tilde{\sigma}(s,s') + \beta V(s'),\ \forall s \in S$
    - Prove by applying a contraction mapping theorem
    - There exists a fixed point and the fixed point can be reached by iterating on an initial guess
    - $V_{i+1}(s) = \max_{s' \in \Gamma(s)} \tilde{\sigma}(s,s') + \beta V_i(s'),\ \forall s \in S$
    - $\lim_{i \to \infty} V_i \rightarrow V^*$

The Computational Solution

- There are several ways to solve this dynamic programming problem on a computer.
- Because these unknowns are functional equations, all of these approaches involve a fix point algorithm.
- We'll start with the most common approach: Value Function Iteration (VFI)

Value Function Iteration

- The algorithm:
  1. Create a discrete grid of the state variable, w
  2. Make an initial guess at the value function, $V_0(w)$
     - This will be a value for each point in the state space
  3. Perform the operation:
  $$TV(w) = \max_{w'} u(w - w') + \beta V_0(w') \tag{1}$$
  4. Update the guess at the value function: $V_1(w) = TV(w)$
  5. Repeat this process:
  $$V_{i+1}(w) = TV(w) = \max_{w'} u(w - w') + \beta V_i(w) \tag{2}$$
  6. Stop when, $|V_{i+1}(w) - V_i(w)| < \varepsilon$, where ε is a small number.
- See the "brute force" approach in this notebook
- Some notes on this:
  - We are approximating the value function over a discrete grid
    - We will have some approximation error
    - We can make the approximation error arbitrarily small by expanding the size of our grid
    - But we face a trade off - more grid points increase the computational cost
    - With multiple state variables, the costs of increasing the fineness of the grid space increases exponentially (this is the "curse of dimensionality")
  - There are somethings you can do to improve accuracy (reduce approximation error) without increasing the size of the grid:
    - Interpolation
      - See the VFI solution method using interpolation in this notebook
    - Wisely choosing grid points
      - E.g., more grid points where the value function has more curvature
      - E.g., Use a random grid - see Rust (1997)
    - Try a different solution method that uses more of the economic theory

Policy Function Iteration (PFI)

- Here, we'll consider Coleman policy iteration, but there are others - such as Howard's improvement algorithm

- With policy function iteration, we'll iterate on a guess at the policy function, as opposed to the value function
- To do this, we are going to work with the FOC(s) from the problem
  - One limitation of PFI is that you need the economic problem to have a first order condition (i.e., you need a continuous choice variable)
- The Coleman policy iteration algorithm:
  1. Create a discrete grid the state variable, w
  2. Make an initial guess at the policy function, $c = \phi_0(w)$
     - This will be a value of the control variable for each point in the state space
  3. Let K be an operator on the policy function. Use a root finder to find the $K\phi(w)$ such that:
  $$u(K\phi(w)) = \beta u(\phi_0(w - K\phi(w))) \tag{3}$$
  4. Update the guess at the policy function: $\phi_1(w) = K\phi(w)$
  5. Repeat this process, finding $K\phi(w)$ such that:
  $$u(K\phi(w)) = \beta u(\phi_i(w - K\phi(w))) \tag{4}$$
  6. Stop when, $|\phi_{i+1}(w) - \phi_i(w)| < \varepsilon$, where ε is a small number.
- An application of Coleman's PFI to the cake eating problem can be found in this notebook.
- Some notes on PFI:
  - Theoretically, save rate of convergence as VFI
  - In practice, PFI is usually faster and more accurate
    * Policy functions generally have less curvature than value functions
    * Policy functions iteration use more information from the economic problem - they embody the FOCs and envelope conditions
    * The root finding algorithm is usually quite robust (relative to the optimization used in VFI)

Endogenous Grid Method

- Coleman PFI can have some advantages over VFI, but still the root finding can be slow
- Carroll (2006) had an ingenious insight:
  - What if we recast the problem so that instead of the grid of the state today (e.g. w), we create a grid of the state tomorrow (e.g., w')?
  - Then we can invert the first order condition and find the grid for w that corresponds to the grid of w'
- This is known as the endogenous grid method (EGM) - it's a variable of PFI, but much faster
- The EGM algorithm:
  1. Create a discrete grid of the one-period-ahead state variable, w'
  2. Guess a policy function, $c = \phi_0(w)$
  3. Use the Euler equation to solve for the current period consumption that corresponds to each point j in the grid of w':
  $$c_j = u'^{-1}\left(\beta u'(\phi_0(w'_j))\right) \tag{5}$$

4. Use the law of motion for the state variable to find the endogenously determined grid for w:

$$w_j = w'_j + c_j \tag{6}$$

5. Update the guess at the policy function: $\phi_1(w) = K\phi(w) = c_j$
6. Repeat this process until convergence as before

- An example applying EGM to the cake eating problem can be found in this notebook
- Some notes on EGM:
  - Can be much faster than Coleman PFI or VFI since there is no root finding or optimization involved
  - Need the function in the FOC to be invertible.
  - Can be done with multiple control variables, but needs to be modified
    - See Barillas and Fernandez-Villaverde (2007)

Stochastic Cake Eating:

- Where to add uncertainty: (tildes indicate random variables)
  - tastes (i.e., in $u(\cdot)$): $\tilde{u}(c, \tilde{\varepsilon})$
  - storage technology: $w' = (w - c)\tilde{\rho}$
  - discount factor (e.g. probability of death/attrition): $\tilde{\beta}$
  - we'll work on just the first of these for now
- Choice under uncertainty
- Now we'll start talking about preferences over lotteries
- Use expected utility theory
- Lottery
  - $wp =$ with probability
  - Assume c_H wp π
  - Assume c_L wp $(1 - \pi)$
  - Expected Utility
    - $EU = \pi u(c_H) + (1 - \pi)u(c_L)$
    - this does not equal $u(\pi c_H + (1 - \pi)c_L)$
    - for risk averse agents (i.e., those with a concave utility function ($u'' < 0$)), the utility of the expected value is higher than the expected utility of the lottery
    - i.e., the risk averse would rather take the average of the two with certainty than the prob of one or the other (b/c u concave, $u(\alpha x + (1 - \alpha)y) \geq \alpha u(x) + (1 - \alpha)u(y)$)

Cake eating with taste shock:

- Preferences: $u(c, \underbrace{\tilde{\varepsilon}}_{\text{taste shock}}) = \varepsilon u(c)$
  - normal assumptions on $u : u' > 0, u'' < 0$

  - Note that ε affects u and u'

- Storage Technology: $w' = w - c$
- Information: ε known when consumption choice is made (Important to specify timing of knowledge!!)
- Representing the shock: (distribution of the shock)
  - $\varepsilon = \{\varepsilon_L, \varepsilon_H\}$: high or low, ε has two values
  - First-Order Markov Process for the transition (this means that the probability of going to state X one period ahead only depends upon what state in today)
    - $Prob(\varepsilon_{t+1} = \varepsilon_j | \varepsilon_t = \varepsilon_i) = \pi_{ij}$ (Probability tomorrow = state j given today = state i)
    - Draw a 2x2 matrix with the transition probabilities between the high and low states
    - Called a first order process because what happens tomorrow only depends on where at today
  - IID process (independently and identically distributed)
    - Past doesn't matter (like drawing a random # each new day)
    - π_{ij} independent of i (constant $\implies$ prob of state tomorrow does not depend on state today (Note: rows of matrix will still sum to one)

Stochastic Cake Eating Problem

- $V(w, \varepsilon) = \max_{w'} \varepsilon u(w - w') + \beta E_{\varepsilon'|\varepsilon} V(w', \varepsilon'),\ \forall (w, \varepsilon)$
- State variables: w=size of cake; ε=taste shock, $\varepsilon \in \{\varepsilon_H, \varepsilon_L\}$
- Control variables $= w'$ (or c)
- Transition equations: $w' = w - c$, $\underbrace{\Pi}_{2x2}$: transition matrix
  - $\pi_{ij} = Pr(\varepsilon' = \varepsilon_j | \varepsilon = \varepsilon_i), i = L, H, j = L, H$
  - $\sum_j \pi_{ij} = 1$ for $i = L, H$ (sum over j)
  - This is a first order Markov process - only one period in the past matters for conditional probe today
- Policy function: $w' = \varphi(w, \varepsilon)$
  - consumption a function of how much cake and the taste shock $\implies$ eat more when ε_H, less when ε_L
- in terms of empirical research, the policy function is what you care about $\implies$ it's the stepping stone between the transition equation and the value function
- π_{LL} near 1 $\implies$ eat cake now before get low and stuck there
- π_{LL} near 0 $\implies$ π_{LH} near 1 $\implies$ wait and eat cake when ε_H
- decision is based on the present ε and also what that ε means for the probability of different future ε's and expected future utility
- FOC:
  - $\frac{\partial V}{\partial w'}$: $\underbrace{\varepsilon}_{\text{For a particular value of } \varepsilon}$ $u'(w - w') = \beta E_{\varepsilon'|\varepsilon}$ $\underbrace{V_{w'}(w', \varepsilon')}_{\text{Random variable b/c } \varepsilon' \text{ random}}$, $\forall (w, \varepsilon)$
  - e.g., $\varepsilon = \varepsilon_H$:
  - $\varepsilon_H u'(w - w') = \beta\{\pi_{HL} \underbrace{V_{w'}(w', \varepsilon_L)}_{\text{value of low next period}} + \pi_{HH} \underbrace{V_{w'}(w', \varepsilon_H)}_{\text{value of high next period}} \}$

- Agent knows ε when choose w', so w' fixed, but w' a function of ε; $w' = \varphi(w, \varepsilon)$
- FOC for low state: $\varepsilon_L u'(w - w') = \beta\{\pi_{LL} V_{w'}(w', \varepsilon_L) + \pi_{LH} V_{w'}(w', \varepsilon_H)\}$
- Note: w' here different than w' above b/c chose to consume more at ε_H than ε_L

- Euler equation
  - Using the envelope theorem, we know:
    - $V_w(w, \varepsilon) = \varepsilon u'(c)$ (students can work this out on their own)
    - $\implies V_{w'}(w', \varepsilon') = \varepsilon' u'(c')$
  - $\implies$ Euler is: $\varepsilon u'(c) = \underbrace{\beta E_{\varepsilon'|\varepsilon} \varepsilon' u'(c')}_{\text{Discounted expected MU}}$
  - can write: $c = w - w', c' = w' - w''$ (where "'" is two periods ahead)
  - Substituting in the policy function ($w' = \varphi(w, \varepsilon)$) and $c = w - w'$ get:
  - $\varepsilon u'(w - \varphi(w, \varepsilon)) = \beta E_{\varepsilon'|\varepsilon} \varepsilon' u'[\varphi(w, \varepsilon) - \varphi(\varphi(w, \varepsilon), \varepsilon')]$
  - $c = w' - w = w - \varphi(w, \varepsilon) \equiv \phi(w, \varepsilon) \implies$ consumption depends on $\varepsilon \implies c' = \phi(w', \varepsilon')$

- Comparative statics with the stochastic dynamic programming problem:
  - Let $G(w', \varepsilon_H) = \varepsilon_H u'(w - w') - \beta\left[\pi_{HL}\varepsilon_L u'(w' - w'') + \pi_{HH}\varepsilon_H u'(w' - w'')\right] = 0$
  - Applying the IFT: $\frac{\partial w'}{\partial \varepsilon_H} = \frac{-G_2}{G_1} = -\frac{u'(w-w') - \beta\pi_{HH}u'(w'-w'')}{-\varepsilon_H u''(w-w') - \beta[\pi_{HL}\varepsilon_L u''(w'-w'') + \pi_{HH}\varepsilon_H u''(w'-w'')]}$
  - Now try to sign the derivative...
    - Intuitively, we think that the agent should save less if he is in the high state today and the value of the high state increases. This means the marginal utility of consumption increases today and so to keep equality across periods we need increase consumption this period and lower consumption next period as that will lower the MU of cons today and increase the MU cons tomorrow.
    - By the FOC, we know that: $\varepsilon_H u'(w - w') = \beta\left[\pi_{HL}\varepsilon_L u'(w' - w'') + \pi_{HH}\varepsilon_H u'(w' - w'')\right]$
    - $\implies \varepsilon_H u'(w - w') - \beta\pi_{HH}\varepsilon_H u'(w' - w'') = \pi_{HL}\varepsilon_L u'(w' - w'')$
    - $\implies u'(w - w') - \beta\pi_{HH} u'(w' - w'') = \underbrace{\frac{\pi_{HL}\varepsilon_L u'(w' - w'')}{\varepsilon_H}}_{>0}$
    - Thus we know that the numerator in the partial derivative is positive.
    - And since $u'' < 0$, we know the denominator is negative.
    - Thus we know that $\frac{\partial w'}{\partial \varepsilon_H} < 0$. That is, w' falls as ε_H increases

- But ε doesn't always have to be in the policy function
- Whether or not it is, depends upon how expectations are made
- Two alternative are:
- 1) Make it additive: $V(w, \varepsilon) = \max_{w'} \varepsilon + u(w - w') + \beta E_{\varepsilon'|\varepsilon} V(w', \varepsilon')$
  - make it additive rather than multiplicative so that ε has no effect at the margin, though it does affect overall utility
  - no ε in Euler (on left)(when take $\frac{\partial V}{\partial w'}$, ε left out)
- 2) Agent doesn't know shock when makes choice of control:
  - $V(w, \varepsilon_{-1}) = \max_{w'} E_{\varepsilon|\varepsilon_{-1}} \underbrace{\{\varepsilon u(w - w') + \beta V(w', \varepsilon)\}}_{\text{both depends on } E_{\varepsilon|\varepsilon_{-1}}}, \ \forall(w, \varepsilon_{-1})$
    - The FOC for this problem: $E_{\varepsilon|\varepsilon_{-1}}\left[\varepsilon u'(w - w') - \beta V(w', \varepsilon)\right] = 0$

* This implies $E_{\varepsilon|\varepsilon_{-1}}\varepsilon u'(c) = \beta E_{\varepsilon|\varepsilon_{-1}}u'(c')$
* Neither sides of the above equation are functions of ε (though they are functions of ε_{-1})
* the above is a model of making a decision today base on ε from a previous period
* since utility today and continuation value both uncertain in same way, the decision will only depend on the expected value
* always need to specify who knows what, when, why

References

Adda, Jerome and Russell W. Cooper, *Dynamic Economics: Quantitative Methods and Applications* Mit Press, MIT Press, 2003.

Barillas, Francisco and Jesus Fernandez-Villaverde, "A generalization of the endogenous grid method," *Journal of Economic Dynamics and Control*, August 2007, *31* (8), 2698–2712.

Carroll, Christopher D., "The method of endogenous gridpoints for solving dynamic stochastic optimization problems," *Economics Letters*, June 2006, *91* (3), 312–320.

Rust, John, "Using Randomization to Break the Curse of Dimensionality," *Econometrica*, May 1997, *65* (3), 487–516.

Stokey, Nancy L. and Robert E. Lucas, *Recursive methods in economic dynamics*, 4. print ed., Cambridge, Mass.: Harvard Univ. Press, 1996.