Week 4

Matthew Fitzgerald

UCSB

January 31, 2020

Agenda

- Last week we discussed how to use backward induction to solve problems
- Today we will continue to discuss solution methods
- We will also talk about how to solve these problem numerically

Guess and Verify

- Guess and Verify (also known as the *method of undetermined coefficients*) requires that solution to the Bellman Equation is unique, and the guess is correct (hence it is generally not available)
- Steps:
  1. Guess the functional form of the value or policy function with stand in coefficients
  2. Verify that the guess is consistent with optimization
  3. Solve for the coefficients of the guess
- General Idea: if our guess is correct, then when "operated on" it should recover that same form and we can back out the coefficients previously left undetermined

Guessing the Policy Function

Consider the following problem solved by the household:

$$\max_{\{c_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t ln(c_t) \quad s.t. \quad c_t + k_{t+1} = k_t^{\alpha}$$

The problem can be written recursively as follows:

$$V(k) = \max_{k'} \left\{ ln(k^{\alpha} - k') + \beta V(k') \right\}$$

Now let's guess that the policy function takes the form $k' = \eta k^{\alpha}$ where η is the undetermined coefficient. Now we solve for η and find the policy function for capital and consumption.

Take FOC:

$$\frac{1}{k^\alpha - k'} = \beta V'(k')$$

Now plug in guess for k':

$$\frac{1}{k^\alpha - (\eta k^\alpha)} = \beta V'(k')$$

$$\frac{1}{(1-\eta)k^\alpha} = \beta V'(k')$$

Note:

$$V'(k) = \frac{1}{k^\alpha - k'}(\alpha k^{\alpha-1})$$

Again plugging in our guess for k' we have

$$V'(k) = \frac{1}{(1-\eta)k^\alpha}(\alpha k^{\alpha-1})$$

$$= \frac{\alpha}{(1-\eta)k}$$

Pushing $V'(k)$ forward one period we have:

$$\begin{aligned} V'(k') &= \frac{\alpha}{(1-\eta)k'} \\ &= \frac{\alpha}{(1-\eta)\eta k^{\alpha}} \quad \text{(plugging in guess)} \end{aligned}$$

We can plug this in to our FOC and solve for the undetermined coefficient η:

$$\begin{aligned} \frac{1}{(1-\eta)k^{\alpha}} &= \beta V'(k') \\ \frac{1}{(1-\eta)k^{\alpha}} &= \beta \frac{\alpha}{(1-\eta)\eta k^{\alpha}} \end{aligned}$$

Solving this yields:

$$\eta = \alpha\beta$$

which is indeed a constant. Finally, the policy function for k' and c are

$$\begin{aligned} k' &= \alpha\beta k^{\alpha} \\ c &= (1-\alpha\beta)k^{\alpha} \end{aligned}$$

Guessing the Value Function

Now we'll try guessing the value function. Consider the same problem:

$$V(k) = \max_{k'} \left\{ ln(k^\alpha - k') + \beta V(k') \right\}$$

Now guess that $V(k) = A + Bln(k)$ where A and B are undetermined coefficients. We can take the FOC of the maximization problem with respect to k' and plug in our guess:

$$\frac{d}{dk}\left\{ ln(k^\alpha - k') + \beta[A + Bln(k')] \right\} = 0$$
$$\implies k'^* = \frac{\beta B k^\alpha}{1 + \beta B}$$

Now evaluate the right hand side of the Bellman Equation at the optimum:

$$RHS(k'^*) = ln\left(k^\alpha - \frac{\beta B k^\alpha}{1 + \beta B}\right) + \beta\left[A + Bln\left(\frac{\beta B k^\alpha}{1 + \beta B}\right)\right]$$
$$= ln\left(\frac{k^\alpha}{1 + \beta B}\right) + \beta A + \beta B ln\left(\frac{\beta B k^\alpha}{1 + \beta B}\right)$$

Now we group the constants together and separate out the k terms:

$$RHS(k'^{*}) = \underbrace{-ln(1+\beta B) + \beta A + \beta ln\Big(\frac{\beta B}{1+\beta B}\Big)}_{constant} + \underbrace{\alpha(1+\beta B)}_{ln(k)\text{-term coeff.}} ln(k)$$

Now let's think about the LHS of our Bellman Equation, V. We know that if our guess is correct, the LHS V will have the same form as our guess, therefore the $RHS(k'^{\alpha})$ we found above will be *equal to* $A + Bln(k)$.

$$A = \text{constant} = -ln(1+\beta B) + \beta A + \beta ln\Big(\frac{\beta B}{1+\beta B}\Big)$$

$$B = ln(k)\text{-term coeff.} = \alpha(1+\beta B)$$

Now we can solve for B using the second equation, and then A from the first:

$$B = ln(k)\text{-term coeff.} = \frac{\alpha}{1-\alpha\beta}$$

$$A = \frac{1}{1-\beta}\Big[ln(1-\alpha\beta) + \frac{\alpha\beta}{1-\alpha\beta} ln(\alpha\beta)\Big]$$

From here we might want to get the policy function. We solved for this earlier and found

$$k' = \frac{\beta B k^{\alpha}}{1 + \beta B}$$

So now we can plug in B and we get:

$$k' = \alpha \beta k^{\alpha}$$

Value Function Iteration

- This method relies on the ideas of a contraction / fixed point that we discussed last time
- Recall: (*paraphrasing*) "starting from any possible V_0 in the space of potential solutions, when iterating on V_0 we will get closer and closer to the true V"
- That is, we are going to make an initial guess at V (a good place to start is $V_0 = 0$), and then operate for a while until we are comfortable that the value function has converged

- The above is true if we are assuming an infinite horizon; what if there is some terminal period T (for example, a lifecycle model)?
- Then you can think of this process like you would backwards induction...
  - V_0 is the value at $T + 1$ $(= 0)$
  - V_1 is the value you get in T when optimizing knowing that $V_0 = 0$
  - etc.
- You would continue this process until you found the value function for the first date through the last date: $V_T, \ldots, V_1, V_0$, and your answer would effectively be the sequence of value (or policy) functions

First Iteration

Let's try our hand at an easy example; consider the problem we've been working with and initialize the iterations with $V_0(k') = 0$

$$V_1(k) = \max_{k'} \left\{ ln(k^\alpha - k') + \beta \underbrace{0}_{V_0(k')} \right\}$$

Optimization requires that $k' = g_1(k) = 0$ (calculus doesn't work here, we are at a "corner"). Knowing this, we can plug in this optimal policy to find $V_1(k)$.

$$V_1(k) = ln(k^\alpha - 0) = \alpha ln(k)$$

Now for a second round ...

Second Iteration

$$V_2(k) = \max_{k'} \left\{ ln(k^\alpha - k') + \beta \underbrace{\alpha ln(k')}_{V_1(k')} \right\}$$

Let's optimize.

$$\frac{d}{dk'} = 0 \qquad \Longrightarrow \qquad k' = g_2(k) = \frac{\alpha\beta}{1+\alpha\beta} k^\alpha$$

Knowing this, we can plug in this optimal policy to find $V_2(k)$.

$$\begin{aligned} V_2(k) &= ln\left(\frac{k^\alpha}{1+\alpha\beta}\right) + \alpha\beta ln\left(\frac{\alpha\beta}{1+\alpha\beta} k^\alpha\right) \\ &= ln\left(\frac{1}{1+\alpha\beta}\right) + \alpha\beta ln\left(\frac{\alpha\beta}{1+\alpha\beta}\right) + \alpha(1+\alpha\beta) ln(k) \end{aligned}$$

Let's go for a third round, paying attention to an emerging pattern.

Third Iteration

$$V_3(k) = \max_{k'} \left\{ ln(k^\alpha - k') + \beta \underbrace{\left[ln\left(\frac{1}{1+\alpha\beta}\right) + \alpha\beta ln\left(\frac{\alpha\beta}{1+\alpha\beta}\right) + \alpha(1+\alpha\beta)ln(k') \right]}_{V_2(k')} \right\}$$

Optimize.

$$\frac{d}{dk'} = 0 \qquad \Longrightarrow \qquad k' = g_3(k) = \frac{\alpha\beta + (\alpha\beta)^2}{1 + \alpha\beta + (\alpha\beta)^2} k^\alpha$$

Plug the optimal policy back in to find $V_3(k)$.

$$V_3(k) = \beta ln\left(\frac{1}{1+\alpha\beta}\right) + \alpha\beta^2 ln\left(\frac{\alpha\beta}{1+\alpha\beta}\right) + (\alpha\beta + (\alpha\beta)^2) ln\left(\frac{\alpha\beta + (\alpha\beta)^2}{1 + \alpha\beta + (\alpha\beta)^2}\right)$$

$$+\alpha(1 + \alpha\beta + (\alpha\beta)^2) ln(k)$$

We could continue (or make our computer do it), but for this problem we can see a pattern emerge. Perchance most nobaly on the policy function, we can see that as we let the iteration $s \to \infty$, we'll have

$$g^*(k) = \lim_{s \to \infty} g_s(k) = \alpha\beta k^\alpha.$$

See earlier slides for how to "derive" the above. The value function itself can be shown to converge to

$$V^*(k) = \lim_{s \to \infty} V_s(k) = \frac{1}{1-\beta}\left[ln(1-\alpha\beta) + \frac{\alpha\beta}{1-\alpha\beta} ln(\alpha\beta)\right] + \frac{\alpha}{1-\alpha\beta} ln(k)$$

which is, recall, what we found earlier with *Guess and Verify*.

Functional Euler Equation

- This last "method" doesn't actually solve for the value / policy functions directly (though you can back them out)
- Sometimes you might see this called the Euler-Lagrange Equation
- It involves the construction of a "Lagrangian" using the Bellman Operator; here, though, we are mapping functions to functions (hence "functional equation")
- The procedure should feel very familiar to you, as you've sort of seen it in previous sections; set up the "Lagrangian" and take F.O.C.s, find the EE and use constraints / conditions (e.g. market clearing) to solve for whatever you desire

Just to get an idea, let's try it out on our running example (we've actually done this last time but I just plugged the constraint in for c).

$$\mathcal{L} = ln(c) + \beta V(k') + \lambda\left[k^{\alpha} - c - k'\right]$$

$$\frac{\partial \mathcal{L}}{\partial c} = 0: \qquad \lambda = \frac{1}{c} \tag{1}$$

$$\frac{\partial \mathcal{L}}{\partial k'} = 0: \qquad \lambda = \beta \frac{dV(k')}{dk'} \tag{2}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0: \qquad c + k' = k^{\alpha} \tag{3}$$

Combine (1) and (2).

$$\frac{1}{c} = \beta \frac{dV(k')}{dk'} \tag{4}$$

Now let's use the envelope theorem.

$$\frac{\partial \mathcal{L}}{\partial k} = \frac{dV(k)}{dk} \qquad \Longrightarrow \qquad \frac{dV(k)}{dk} = \lambda[\alpha k^{\alpha-1}]$$

Pushing this forward, and plugging in for $\lambda' = 1/c' \ldots$

$$\frac{dV(k')}{dk'} = \frac{1}{c'}[\alpha k'^{\alpha-1}] \tag{5}$$

noting that $\alpha k^{\alpha-1} = 1 + r$ when $\delta = 1$ (which is what we have in this example). Now, plug (5) back into (4) to reunite with an old friend.

$$\frac{1}{c} = \beta[\alpha k'^{\alpha-1}]\frac{1}{c'} \tag{EE}$$

- From here we can do many things...
- You might want to find steady state values: plug in to budget constraints / market clearing conditions, do comparative statics, etc.
- Alternatively, you may want to recover the policy function: one way would be to iterate on the EE like we did in section 2 (you would find $k' = \alpha\beta k^{\alpha}$)
- This technique may seem more roundabout than last time (recall I just plugged the constraint right into the objective), but just note that this method is slightly more general insofar as it handles situations where you can't easily substitute in all constraints

Solving Problems Numerically

- While we've seen a "nice" problem that we can solve by hand, most problems do not have a nice analytical solution

Solving Problems Numerically

- While we've seen a "nice" problem that we can solve by hand, most problems do not have a nice analytical solution

- For these problems we need to solve with a computer

Solving Problems Numerically

- While we've seen a "nice" problem that we can solve by hand, most problems do not have a nice analytical solution
- For these problems we need to solve with a computer
- How do we solve with a computer?

Solving Problems Numerically

- While we've seen a "nice" problem that we can solve by hand, most problems do not have a nice analytical solution
- For these problems we need to solve with a computer
- How do we solve with a computer?
- Let's take a look at an example

Numerical Value Function Iteration

- We're going to look at a very simple way of solving these problems on a computer which entails discretizing the state space
  - This means approximating a continuous k with many discrete values
  - We do this by creating a grid of k values

- There are better ways of doing this, but creating a grid for the state space is an easy way to see what's going on in the code, therefore we will do this to build our intuition

Numerical Value Function Iteration

$$V(k) = \max_{k'} \left\{ ln(k^{\alpha} - k') + \beta V(k') \right\}$$

- Up to this point we have taken a derivative and set it equal to zero to solve the max, but how can we do this on a computer?

Numerical Value Function Iteration

$$V(k) = \max_{k'} \left\{ ln(k^{\alpha} - k') + \beta V(k') \right\}$$

- Up to this point we have taken a derivative and set it equal to zero to solve the max, but how can we do this on a computer?
- Note that if we have a grid of $k's$, then we can simply evaluate the RHS at each value of k and k' and find the k' that maximizes the expression for each value of k

- Let $V_0 = \mathbf{0}$ be our initial guess for V_0, and let's compute $ln(k^\alpha - k') + \beta V(k')$ for each value of k and k'
- For a simple example, assume k can take the values $[0.1, 0.2, 0.3]$
- Then we can find the value of k' that maximizes the RHS of our Bellman equation for each value of k
- Let's start with $k = 0.1$ and let $\alpha = 0.4$

$$ln((0.1)^{0.4} - 0.1) + \beta 0 = -1.21$$

$$ln((0.1)^{0.4} - 0.2) + \beta 0 = -1.61$$

$$ln((0.1)^{0.4} - 0.3) + \beta 0 = -2.32$$

- Clearly if we start the period with $k = 0.1$ then the value that maximizes the expression is $k' = 0.1$

- We can do this for the remaining values of k
- for $k = 0.2$ we have:

$$ln((0.2)^{0.4} - 0.1) + \beta 0 = -0.85$$

$$ln((0.2)^{0.4} - 0.2) + \beta 0 = -1.12$$

$$ln((0.2)^{0.4} - 0.3) + \beta 0 = -1.49$$

- Cleary if we start the period with $k = 0.2$ then we want to choose $k' = 0.1$
- for $k = 0.3$ we have:

$$ln((0.3)^{0.4} - 0.1) + \beta 0 = -0.65$$

$$ln((0.3)^{0.4} - 0.2) + \beta 0 = -0.87$$

$$ln((0.3)^{0.4} - 0.3) + \beta 0 = -1.14$$

- Cleary if we start the period with $k = 0.3$ then we want to choose $k' = 0.1$

- Therefore our new value function is:

$$V_1 = \begin{bmatrix} -1.21 \\ -0.85 \\ -0.65 \end{bmatrix} \tag{1}$$

- Then note:

$$V_0 - V_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -1.21 \\ -0.85 \\ -0.65 \end{bmatrix} = \begin{bmatrix} 1.21 \\ 0.85 \\ 0.65 \end{bmatrix}$$

- Need to define a metric to measure how close our new value function is to our old value function so we know when to stop iterating

- Thinking about this more generally we can use matrices to find $u(f(k) - k') + \beta V(k')$ all in one step:

$$u\left(f\left(\begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.2 & 0.2 & 0.2 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}\right) - \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.1 & 0.2 & 0.3 \\ 0.1 & 0.2 & 0.3 \end{bmatrix}\right) - \beta \begin{bmatrix} V(0.1) & V(0.2) & V(0.3) \\ V(0.1) & V(0.2) & V(0.3) \\ V(0.1) & V(0.2) & V(0.3) \end{bmatrix} \tag{2}$$

- Note that since our initial guess was $V_0 = \mathbf{0}$, in the first iteration we would have $V(0.1) = V(0.2) = V(0.3) = 0$ (i.e. the last matrix would be a matrix of zeros)

- Then note that the first row of the resulting matrix would correspond to when we're starting the period with $k = 0.1$ and the first column of the first row (i.e. position (1,1) in the matrix) corresponds to the value of starting with $k = 0.1$ this period and choosing $k' = 0.1$

- The next step is to maximize

- We want to find the maximum value in each **row**
  - The maximum value in the first row will correspond to the optimal value of k' given that we started with $k = 0.1$
  - Since we worked this out by hand, we know that the first column of the first row is the maximum value (row one column one corresponds to choosing $k' = 0.1$ given that we started the period with $k = 0.1$, row 1 column 2 corresponds to choosing $k' = 0.2$ given that we started the period with $k = 0.1$, etc.)

- Once we have the maximum of each row, we will end up with a 3×1 vector which will be exactly (1)

- Next we compute $V_0 - V_1$, and if $\rho(V_0 - V_1) <$ Tolerance (where ρ is our norm), we're done

- If $\rho(V_0 - V_1) \geq$ Tolerance, we go back and recompute (2) and compute a new value function V_2

- Continue this process until the condition $\rho(V_t - V_{t+1}) <$ Tolerance is met, or we have met some maximum allowable number of iterations

Summary and things for next time

- Today we went over different solution methods for our problems
- We also discussed the intuition for implementing value function iteration on a computer
  - You will get a chance to practice this on the next homework
- Next time we will discuss how to add uncertainty into our problems