14.451 Recitation, February 18th, 2005 -- Todd Gormley

----------Dynamic Programming Handout ----------

What is Dynamic Programming (DP)? ----------

DP is when we us to change a problem from:

$$\max_{\{u_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t) \quad s.t. \quad x_{t+1} = g(x_t, u_t), \quad x_0 \text{ given}$$

Into something of the form,

$$V(x) = \max_u \{ r(x,u) + \beta V[g(x,u)] \}$$

Which is the state variable? Which is the control?

Maximization on the RHS gives us a policy function, $h(x)$, such that,

$$V(x) = r(x, h(x)) + \beta V[g(x, h(x))]$$

The goal of this handout is show you how to solve for $h(x)$ and $V(x)$

Conditions for the DP Solution ----------

In order to solve this problem, we need certain conditions to be true.
One set typically used is:

1. r concave and bounded
2. constraint set generated by g is convex and compact

But, for the purpose of 451, you should just assume that the necessary conditions for solving with the Bellman equation are satisfied. [For greater details on dynamic programming and the necessary conditions, see Stokey and Lucas (1989) or Ljungqvist and Sargent (2001). Ivan's 14.128 course also covers this in greater detail.]

General Results of Dynamic Programming ----------

1. $V(x)$ unique and strictly concave
2. $u_t = h(x_t)$ is time invariant and unique
3. Off corners, $V(x)$ is differentiable
4. Solution can be found by iteration: $j \to \infty$,

$$V_{j+1}(x) = \max\left\{r(x,u) + \beta V_j(\tilde{x})\right\} \quad \text{s.t. } \tilde{x} = g(x,u),\ x \text{ given}$$

Two Methods for Solving DP Problem ----------

1. Iteration over the Value Function
2. Guess and Verify

This handout will now work you through an example of how to solve a DP problem in the context of a basic growth model with no leisure. It will first solve the model in the usual way with Optimal Control. Then it will show you how to solve it using DP.

A Simple Model of Economic Growth ----------

The basic setup of the problem for the social planner is:

$$\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} U_0 = \sum_{t=0}^{\infty} \beta^t U(c_t)$$
$$s.t. \quad c_t + k_{t+1} \leq (1-\delta)k_t + f(k_t) \quad \forall t \geq 0$$
$$c_t \geq 0, \; k_{t+1} \geq 0 \quad \forall t \geq 0$$
$$k_0 > 0 \text{ given}$$

As usual, assume all the other typical growth model assumptions apply. e.g. inada conditions

Why can we ignore the conditions $c_t \geq 0, \; k_{t+1} \geq 0 \quad \forall t \geq 0$?

Finally, to be explicit, assume log preferences and a Cobb-Douglas production function.

$$U(c) = \ln c$$
$$f(k) = k^{\alpha}$$

Solving Using Optimal Control (i.e the typical Lagrangian) ----------

The Lagrangian looks as follows:

$$L = \sum_{t=0}^{\infty} b^t U(c_t) + \sum_{t=0}^{\infty} m_t \left[(1-d)k_t + f(k_t) - c_t - k_{t+1}\right]$$

- Note that we have an infinite number of constraints. One for each period.
- We can now solve this just like a normal Lagrangian

The FOCs are as follows:

$$c_t: \quad \frac{b^t}{c_t} = m_t \quad \forall t$$
$$k_{t+1}: \quad m_{t+1}\left(1-d+a k_{t+1}^{a-1}\right) = m_t \quad \forall t$$

And our budget constraint is:

$$c_t + k_{t+1} = (1-d)k_t + k_t^a$$

Combining our FOCs, we have:

$$\frac{c_{t+1}}{c_t} = b\left[1 + a k_{t+1}^{a-1} - d\right]$$

So, we can find the growth rate in the economy, which is nice. However, it is a bit difficult to see how much individuals consume and invest in each period of time. This is one way Dynamic Programming can help us.

Solving Using Dynamic Programming ----------

First, let's rewrite the problem in the DP form. This is done as follows:

$$V(k_t) = \max_{c_t, k_{t+1}} \{\ln c_t + \beta V(k_{t+1})\}$$
$$s.t. \quad c_t + k_{t+1} = k_t^{\alpha}$$

Note: I've set $\delta = 1$ in order to simplify the math.

How should we interpret $V(k)$?

We can actually now drop the time subscripts of the problem, and plug in for the constraint. After doing this, we have the following:

$$V(k) = \max_{0 \le k' \le k^{\alpha}} \left\{\ln\left(k^{\alpha} - k'\right) + \beta V(k')\right\}$$

There are now two ways to solve the problem… via "Guess and Verify" or via "Iteration". I'll first start with "Guess and Verify".

Guess and Verify Method: The Policy Functions ----------

Let's guess that the $V(k)$ is of the form $E + F\ln k$ where E and F are unknown constants. Using this guess, we solve the maximization problem on the RHS of the Bellman equation.

$$\max_{k'}\left\{\ln\left(k^{a}-k'\right)+\beta E+\beta F\ln k'\right\}$$

The FOC is:

$$-\frac{1}{k^{a}-k'}+\frac{\beta F}{k'}=0$$

Solving this, we have:

$$k'=\frac{\beta F}{1+\beta F}k^{a}$$

How can we interpret this?

Using the budget constraint, we see that it is optimal for the individual to consume

$$c=\frac{1}{1+\beta F}k^{a}$$

Both of these solutions are the policy functions. i.e. Capital tomorrow, k', and consumption today, c, are only functions of capital today, k! We express these policy functions as $c(k)$ and $k'(k)$. This is clearly a much cleaner result than when we solved with optimal control.

Guess and Verify Method: Finding & Proving the Value Function ----------

To finish the problem, we actually need to finish solving for the constants E and F, and prove that our guess satisfies the Bellman equation.

To do this, we plug our policy functions back into the Bellman equation, and solve for our constants. This is shown below:

$$V(k)=\max\left\{\ln\left(k^{a}-k'\right)+\beta E+\beta F\ln k'\right\}$$

$$E+F\ln k=\ln\left(k^{a}-k'(k)\right)+\beta E+\beta F\ln k'(k)$$

$$E+F\ln k=\ln\left(k^{a}-\frac{\beta F}{1+\beta F}k^{a}\right)+\beta E+\beta F\ln\left(\frac{\beta F}{1+\beta F}k^{a}\right)$$

With a fair amount of algebra, you can solve for the constants. I'll leave that math to you.

Solving DP via Iteration: Why Does it Work? ----------

Suppose we have the following type of Bellman Equation:

$$V(k) = \max_{c,k'} \{U(c) + \beta V(k')\}$$
$$s.t. \quad c + k' = f(k)$$
$$c, k' \geq 0$$

This can then be rewritten as:

$$V(k) = \max_{0 \leq k' \leq f(k)} \{U(f(k) - k') + \beta V(k')\}$$

Under the proper assumptions, this Bellman equation is a contraction mapping and has a unique fixed point. More specifically: Let B be a set of continuous and bound functions $v : [0, f(k)] \to R$ and consider the mapping $T : B \to B$ defined as follows:

$$Tv(k) = \max_{c,k'} \{U(c) + \beta v(k')\}$$
$$s.t. \quad c + k' = f(k)$$
$$c, k' \geq 0$$

Because T is a contraction mapping, and has a unique fixed point $V = TV$, we can use iteration to solve the problem. Specifically, we use $V_{j+1}(k) = \max\{U(c) + \beta V_j(k')\}$, and do the following:

1. Make an initial guess at the value function, $V(k)$. Call this $V_0(k)$.
2. Perform the maximization of the Bellman equation, using your guess $V_0(k)$
3. This yields you a new value function, $V_1(k)$
4. Replace $V_0(k)$ with $V_1(k)$, and repeat Step 2.
5. Continue this iteration until convergence to the fixed point $V(k)$

Solving via Iteration on the Value Function (in practice) ----------

We first need to make a guess at the value function. We'll call the first guess $V_0(k)$. And to keep it simple, I'll guess that $V_0(k)=0$. Now let's solve the following problem:

$$V_1(k)=\max_{0\le k'\le k^a}\left\{\ln\left(k^a-k'\right)+b\,V_0(k')\right\}$$

Plugging in for our guess, we have:

$$V_1(k)=\max_{0\le k'\le k^a}\left\{\ln\left(k^a-k'\right)\right\}$$

What is k' and $V_1(k)$?

Now, we use the same method to find $V_2(k)$, and we continue doing this until convergence.

Unfortunately, this method is quite time and algebra intensive, so I won't work out any more steps here. Thankfully, there is a quicker way to do this, and this is done by using a mathematical program such as MATLAB.

This handout will now provide a rather detailed sketch on how to numerically solve a dynamic programming using a mathematical program, such as MATLAB. For help with MATLAB syntax, please see the handout written by Francesco Franco.

A Rough Outline on How to Numerically Solve a DP Problem

1. **Create a vector of discrete values for your state variable,** k

   a. This will be your vector of potential state variables to choose from. You might want to create a vector of values that spans the steady state value of the economy.
   b. For example. Suppose the steady state is $k^* = 3$. Then, you might create the following vector of state values:

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix}$$

2. **For each value of the state variable in your vector, calculate the potential utility possible from each choice over your vector of possible states and store these values.**

   a. For example, using the above 5 possible states: $k = 1,2,3,4,5$

      i. Calculate the consumption and utility of the agent if $k = 1$ and she chooses $k' = 1$. Suppose this yields $U(\cdot) = 2.4$. Store this value.

      ii. Now calculate and store the utility of the agent when $k = 1$ and $k' = 2$. Suppose this yields $U(\cdot) = 2.7$

      iii. In this example, you will have five utilities values for each of the five different states, depending on your five potential choices of k'. Thus, you should have 25 values saved in total.

      ***Note**: You have to be a bit careful here, some of the values for k' in your vector of potential states may imply a $c < 0$. I.e. You are violating the constraints of the problem. You need to watch for these cases, and make sure they are not considered a viable choice for the individual.*

   b. Save these values to a matrix as follows:

$$\begin{matrix} 2.4 & 1 & 3 & 2.6 & 0 \\ 2.7 & 2.1 & 4.1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 3 & 1 & 2 \\ 1 & 1 & 0 & 1 & 3.3 \end{matrix}$$

      i. Columns represent k, rows represent k'

3. Make an initial guess at the value function. Call this $V_{old}(k)$.

a. One way to do this is to create an appropriately long vector of zeros. I.e. Your initial guess is that for any of the states you are considering, the value function returns zero. $V(k)=0\ \forall k$

i. In the example I've been using so far, I would need to create a vector of length five since I have five possible states: $k=1,2,3,4,5$.

ii. This would look like:

$$\begin{matrix}0\\0\\0\\0\\0\end{matrix}$$

4. Now iterate over the value function until it converges… i.e. Use your initial guess at the value function, $V_{old}(k)$, to calculate a new guess at the value function, $V_{new}(k)$.

a. First, think of your Bellman equation as follows: $V_{new}(k)=\max\{U(c)+\beta V_{old}(k')\}$

b. Second, choose the maximum value for each potential state variable by using your initial guess at the value function, $V_{old}(k)$ and the utilities you calculated in part 2. i.e. calculate $U(c)+\beta V_{old}(k')$ for each k and k' combo and choose the maximum value for each k. This is your $V_{new}(k)$.

i. In the above example of $k=1$, it is easy to see from my matrix of utility values in part 2B that the maximum utility the agent can achieve is 2.7, and this occurs when she chooses $k'=2$.

ii. So, using $U(c)=2.7$ and $\beta V_{old}(2)=0$ (because my initial guess is $V_{old}(k)=0\ \forall k$), we see that $V_{new}(1)=2.7$

c. Third, store these values into a vector, $V_{new}(k)$. This is your new guess at the value function for each potential state.

i. In the above example, I would have the following after the first iteration:

$$\begin{matrix}2.7\\2.1\\4.1\\2.6\\3.3\end{matrix}$$

d. Use some test to see if your new guess at the value function, $V_{new}(k)$ is arbitrarily close to your initial guess, $V_{old}(k)$.

   i. If it fails the test, replace your old guess at the value function with the new one, and start at 4A again. I.e. $V_{old}(k) = V_{new}(k)$

   ii. If it passes the test, you're done! Move on to Step 5.

5. Plot your value function

6. Plot your policy functions: $c(k)$ and $k'(k)$

a. In the above example when $k = 1$, we saw that it was optimal to choose $k' = 2$. Suppose this remains true after we achieved convergence. Then, $k'(1) = 2$. You want to store this in a vector, along with the $k'(k)$ for all other possible values of k

   i. You can do something similar to find $c(k)$

Note: *If you wanted to, you could calculate the policy function after each iteration on the value function.*