More on Dynamic Programming

- We have concentrated our attention mainly on finite horizon problems, especially in terms of the applications we have been looking at. However, some important research literature embraces and almost never relaxes the infinite horizon setting.
- In some cases you can obtain approximate solutions by increasing the number of periods. However, this can be quite inefficient, especially if there are more suitable techniques than the usual backward induction approach in which we have relied quite heavily.
- Some weeks ago we did see how you would solve, for example, the original Phelps' problem, which was in infinite horizon, but we did not spend a lot of time emphasizing the techniques used.
- By now we are quite familiar with the notation so we can dive into the less familiar details.
- Think of our usual problem of maximizing the discounted sum of a function of some state and controls, problem we can link through the familiar Bellman equation that will allow us under some conditions to solve for the optimal value depending on the state, and for the optimal path of the control.
- We can jointly solve for $V(x)$, and $\delta(x)$, which are linked by the Bellman Equation

$$V(x) = \max_u [r(x,u) + \beta V[g(x,u)]], \tag{1}$$

where $g(x,u)$ represents the law of motion of the state variable. The maximizer on the right hand side of (1) is a policy function $h(x)$ that satisfies

$$V(x) = r[x,h(x)] + \beta V[g[x,h(x)]]. \tag{2}$$

Here (1) or (2) is the functional equation to be solved to obtain $V(x)$ and $h(x)$. Methods to solve for these unknown functions depend on the nature of r and g. Under various assumptions for these we can show that:

1. The functional equation (1) has a unique strictly concave solution.

2. This solution is approached in the limit as the number of iterations goes to ∞ on

$$V_{j+1}(x) = \max_u [r(x,u) + \beta V_j(\bar{x})], \tag{3}$$

subject to $\bar{x} = g(x,u)$, x given, starting from any bounded and continuous initial V_o.

3. There is a unique and time invariant optimal policy of the form $u_t = h(x_t)$, where h is chosen to maximize the right hand side of (1)

4. For interior solutions the limiting value function V is differentiable with

$$V'(x) = \frac{\delta r}{\delta x}[x, h(x)] + \beta \frac{\delta g}{\delta x}[x, h(x)] V'[g[x, h(x)]]. \tag{4}$$

Notice that if we formulate the transition law as independent of the state variable, only the first part of the RHS matters.

Computational Methods

- *Value Function Iteration*: We proceed by constructing a sequence of value functions and associated policy functions. The sequence is created by iterating on the following equation, starting from $V_0 = 0$, and continuing until V_j has converged:

$$V_{j+1}(x) = \max_u [r(x,u) + \beta V_j(\bar{x})], \tag{5}$$

subject to $\bar{x} = g(x,u)$, and x given. This method is called *value function iteration* or *iterating on the Bellman equation*.

- *Guess and verify*: A second method involves guessing and verifying a solution V to equation (1). This relies on the uniqueness of the solution to the equation, but because it relies on luck or intuition is rarely available.

- *Howard's improvement algorithm*: This method is also known as policy function iteration. It consists of the following steps

1. Pick a feasible policy, $u = h_0(x)$, and compute the value associated with operating forever with that policy

$$V_{h_j}(x) = \sum_{t=0}^{\infty} \beta^t r[x_t, h_j(x_t)], \tag{6}$$

where $x_{t+1} = g[x_t, h_j(x_t)]$, with $j = 0$.

2. Generate a new policy $u = h_{j+1}(x)$ that solves the two-period problem

$$\max_u [r(x,u) + \beta V_{h_j}[g(x,u)]], \tag{7}$$

for each x.

3. Iterate over j to convergence of steps 1 and 2.

It can be shown that this algorithm converges to the true solution of the Bellman equation. This method often converges faster than value function iteration.

- In many cases we cannot solve analytically some problem that can be framed in the setup explained above. In that case we can use several methods. Some of them we have already described, some others we will cover in the next classes.

  Think again of Phelps (1962) Life-Cycle Consumption-Saving Problem. As we saw some weeks ago the Bellman Equation for this problem is given by:

$$V(w) = \max_{0 \leq c \leq w} \left[log(c) + \beta \int_0^\infty V(r(w-c)) f(r) dr \right]. \tag{8}$$

  This problem has an analytical solution for V:

$$V(w) = \frac{log(w)}{(1-\beta)} + \frac{log(1-\beta)}{(1-\beta)} + \frac{\beta log(\beta)}{(1-\beta)^2}, \tag{9}$$

  and a linear optimal consumption rule $c(w)$:

$$c(w) = (1-\beta)w. \tag{10}$$

- Let's try to solve this problem using Howard's Policy Iteration Method via Discretization:

1. **Truncate and partition the state space**: Divide $[0, \overline{w}]$ using possibly unequally spaced points $(w_1, w_2, \cdots, w_n)$, satisfying $w_1 = \varepsilon > 0$ and

$$w_i = w_{i-1} + \frac{(\overline{w} - w_{i-1})}{(n-i+1)^\lambda}, \tag{11}$$

where $\lambda = 1$ yields evenly spaced points and $\lambda > 1$ yields points which are more closely spaced when w is small.

2. **Approximate the Conditional Expectation Operator**:
$$EV(w,c) = \int_0^\infty V(r(w-c))f(r)dr, \tag{12}$$
$$\hat{EV}_t(w,c) = \sum_{i=1}^{N} V_t(w_i)\omega_i(w,c), \tag{13}$$
where
$$\omega_i(w,c) = \int_{\frac{m_i-1}{w-c}}^{\frac{m_i}{w-c}} f(r)dr \quad i = 1,\cdots,n-1 \tag{14}$$
$$\omega_n(w,c) = \int_{\frac{m_n}{w-c}}^{\infty} f(r)dr \tag{15}$$
where $m_1 = 0$ and $m_i = \frac{w_i+w_{i-1}}{2}$ $i = 2,\cdots,n$, and V_t is an approximation to V at the n points $(w_1,\cdots,w_n)$.
3. **Initialize Algorithm:** using $V_t(w_i) = 0$ in the approximate conditional expectation operator.
4. **Policy Improvement Step**: For each $i = 1,\cdots,N$
$$c_t(w_i) = arg \max_{0\leq c\leq w_i} [log(c) + \beta\hat{EV}_t(w_i,c)]. \tag{16}$$
5. **Policy Evaluation Step**: Construct and n by n transition probability matrix P_i with (i,j) element given by
$$P_t(i,j) = \omega_j(w_i,\hat{c}_t(w_i)), \tag{17}$$
and compute updated value function V_{t+1} as the n by 1 vector given by
$$V_{t+1} = [I-\beta P_t]^{-1}u_t, \tag{18}$$
where u_t is the n by 1 vector given by:
$$u_t(i) = log(c_t(w_i)). \tag{19}$$
6. **Convergence Test**: If $\| V_t - V_{t-1} \| \leq \delta$, stop, otherwise set t to be $t+1$ and go back to step 4.

- Policy Iteration always generates an improved policy, in fact always generates a strict improvement in the value function. Since there are only a finite number of Markov states we are solving over it follows that policy iteration always converges to the optimal decision rule in a finite number of iterations.

- In Rust (1996) he reports that in his experience for a wide array of problems this method converges in under 20 iterations. The reason for the fast convergence is closely connected to the very rapid quadratic convergence rates of Newton's method for nonlinear equations.

- Remember that we can approximate the Conditional Expectation Operator via the Probability Integral Transform, that we already covered:

$$\hat{EV}(w,c) = \frac{1}{S}\sum_{s=1}^{S} V(F^{-1}(u_s)(w-c)), \tag{20}$$

  where $F(r) = \int_{-\infty}^{r} f(x)dx$, and $(u_1,\cdots,u_s)$ are independent and identically distributed draws from a $U(0,1)$, or alternatively, draws from a low discrepancy sequence such as a Generalized Faure sequence.