Dynamic Programming

Written by Paul Schrimpf and modified by Hiro Kasahara
September 30, 2020
University of British Columbia
Economics 526

> "[Dynamic] also has a very interesting property as an adjective, and that is it's impossible to use the word, dynamic, in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. It's impossible. Thus, I thought dynamic programming was a good name." - Richard Bellman

Most of our applications of optimal control involved choosing something as a function of time. Dynamic programming is another approach to solving optimization problems that involve time. Dynamic programming can be especially useful for problems that involve uncertainty. Dynamic programming has the advantage that it lets us focus on one period at a time, which can often be easier to think about than the whole sequence. Because it only requires maximizing over a few variables at a time, dynamic programming can be a much more efficient way to calculate solutions. The computational advantage of dynamic programming is especially pronounced when some of the variables being maximized over are discrete.

1. References

These notes are about dynamic programming. References from our text books are chapter 11 of Dixit (1990), chapter 12 of De la Fuente (2000) and parts of Carter (2001). Two other useful references are Adda and Cooper (2003) and Acemoglu (2008). Adda and Cooper (2003) is very nice and available online from UBC library. Acemoglu (2008) focuses on economic growth, but includes two very nice chapters on dynamic programming and optimal control.

Stokey, Lucas Jr, and Prescott (1989) is the classic economics reference for dynamic programming. *Applied dynamic programming* by Bellman and Dreyfus (1962) and *Dynamic programming and the calculus of variations* by Dreyfus (1965) provide a good introduction to the main idea of dynamic programming, and are especially useful for contrasting the dynamic programming and optimal control approaches. Dreyfus (2002) has some amusing anecdotes from Bellman about the initial development of dynamic programming. Bertsekas (1976) is the classic reference for dynamic programming with uncertainty.

---

2. Introduction

Dynamic programming deals with similar problems as optimal control. To begin with consider a discrete time version of a generic optimal control problem.

$$\max_{x_t, y_t} \sum_{t=0}^{T} f(x_t, y_t, t) \tag{1}$$

$$\text{s.t.} \, y_{t+1} - y_t = g(y_t, x_t, t)$$
$$h(x_t, y_t, t) \le 0$$
$$y_0 \text{ given} \tag{2}$$

Dynamic programming can also be used for continuous time problems, but we will focus on discrete time.

Example 2.1. [Consumption-savings] An infinite horizon consumption-savings problem,

$$\max_{\{c_t\}_{t=0}^{\infty}, \{s_t\}_{t=1}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \text{ s.t. } s_{t+1} = (1 + r_t)(s_t - c_t),$$

involves maximizing over a countably infinite sequence of c_t and s_t. The interpretation of this problem is that $u(c)$ is the per-period utility from consumption. c_t is consumption at time t. s_t is the savings you have at time t. r_t is the return to savings at time t in period $t + 1$, and β is the discount factor.

Problems like (1) and example 2.1 can be solved using the Lagrangian. This approach focuses on characterizing x^* and y^* through the first order conditions. That is, optimal control focus on characterizing the maximizer. An alternative approach is to focus on the value of the maximized function. This value will depend on the entire problem, but in particular it depends on the initial condition y_0. Thus, we can think of the value as function of the initial state. Dynamic programming focuses on characterizing the value function.

The basic idea of dynamic programming can be illustrated in a familiar finite dimensional optimization problem. Consider a finite horizon discrete time consumption savings choice.

$$\max_{c_t, s_t} \sum_{t=0}^{T} \beta^t u(c_t) \text{ s.t. } s_{t+1} = (1 + r_t)(s_t - c_t)$$

with s_0 given, and the constraint that $s_{T+1} = 0$. We could just write down the first order conditions and try to solve them for c_t. However, if T is large, this might be very difficult. It can be especially difficult to calculate a solution numerically. The easiest maximization problems to solve numerically are ones where the objective function is linear or quadratic. In either of these cases, the amount work needed is proportional to the number of variables cubed. If T is large, T^3 can be so large that computing a solution takes prohibitively long.

We can divide this T dimensional problem to a series of smaller ones by first thinking about what happens at time T. At time T we have some savings s_T and want to choose c_T

to solve

$$\max_{c_T} u(c_T) \text{ s.t. } s_{T+1} = (1 + r_T)(s_T - c_T) = 0$$

As long as u is increasing, it must be that $c_T^*(s_T) = s_T$. If we define the value of savings at time T as

$$V_T(s) = u(s),$$

then at time $T - 1$ given s_{T-1}, we can choose c_{T-1} to solve

$$\max_{c_{T-1}, s'} u(c_{T-1}) + \beta V_T(s') \text{ s.t. } s' = (1 + r_{T-1})(s_{T-1} - c_{T-1}).$$

This is a relatively simple maximization problem with just two variables, so we can solve it without too much difficulty. Repeating in this way, for each t we can define the value of savings at time t as

$$V_t(s) = \max_{c_t, s'} u(c_t) + \beta V_{t+1}(s') \text{ s.t. } s' = (1 + r_t)(s - c_t). \tag{3}$$

This approach to sequential optimization was first proposed by Richard Bellman, so (3) is called a Bellman equation. Notice that if $(c_t^*(s_t), s_{t+1}^*(s_t))$ is a maximizer of (3) for each t, then the sequence of $c_0^*(s_0), s_1^*(s_0), c_1^*(s_1), ..., c_T^*$ must be a maximizer of the original problem. Bellman called this observation the **principle of optimality**. He described it as, "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." (Bellman and Dreyfus (1962))

2.1. **Infinite horizon.** In finite horizon problems, it easy to see that the Bellman equations will exist. However, if we have an infinite horizon problem,

$$\max_{c_t, s_t} \sum_{t=0}^{\infty} \beta^t u(c_t) \text{ s.t. } s_{t+1} = (1 + r_t)(s_t - c_t)$$

then we cannot start from the last period to define the value function. However, if the problem is stationary, that is if the problem at time t and at time $t + 1$ look the same, then it seems reasonable to think that the value function would not depend on t and we could just write

$$V(s) = \max_{c, s'} u(c) + \beta V(s') \text{ s.t. } s' = (1 + r)(s - c).$$

Stokey, Lucas Jr, and Prescott (1989) provide a fairly comprehensive analysis of various conditions when this is possible. We will just look at one case.

Consider a problem that is slightly more general than the consumption savings choice problem with fixed interest rate.

$$\max_{c_t, s_t} \sum_{t=0}^{\infty} \beta^t u(c_t, s_t)$$

$$\text{s.t.} g_0(c_t, s_t) \le s_{t+1} \le g_1(c_t, s_t), \qquad \underline{c} \le c_t \le \overline{c}$$

where $c \in \mathbb{R}$, $s \in \mathbb{R}$, $0 < \beta < 1$, and $u, g : \mathbb{R}^2 \rightarrow \mathbb{R}$. We want to show that the value function exists. Suppose we start with some guess at the value function $v_0 : \mathbb{R} \rightarrow \mathbb{R}$. Then we refine that guess by setting

$$v_1(s) = \max_{c,s'} u(c,s) + \beta v_0(s')$$
$$\text{s.t.} g_0(c,s) \leq s' \leq g_1(c,s),$$
$$\underline{c} \leq c \leq \overline{c}$$

] We could do this repeatedly. Hopefully eventually the value function guesses would stop changing and we would have the value function. Iteratively refining the value function guess in this manner is called value function iteration. It does converge to the true value function under fairly general conditions.

Theorem 2.1. *Consider*

$$\max_{c_t, s_t} \sum_{t=0}^{\infty} \beta^t u(c_t, s_t)$$
$$s.t. g_0(c_t, s_t) \leq s_{t+1} \leq g_1(c_t, s_t),$$
$$\underline{c} \leq c_t \leq \overline{c}$$

Assume

(1) *u is continuous and bounded (i.e. $u(c,s) \leq M$ for all c and s)*
(2) *g_0 and g_1 are continuous, $\underline{c}$ and $\overline{c}$ are finite*
(3) *$0 \leq \beta < 1$*

Then there exists a unique value function, v such that

$$v(s) = \max_{c,s'} u(c,s) + \beta v(s')$$
$$s.t. g_0(c,s) \leq s' \leq g_1(c,s),$$
$$\underline{c} \leq c \leq \overline{c}$$

Moreover, v is the limit of a sequence of value function iteration.

Proof. Let T be the operator defined by value function iteration.

$$T(v)(s) = \max_{c,s'} u(c,s) + \beta v(s')$$
$$\text{s.t. } g_0(c,s) \leq s' \leq g_1(c,s),$$
$$\underline{c} \leq c \leq \overline{c}$$

In order for T to be well defined, the above maximum must exist. Our assumptions that u is continuous, and that c and s' have upper and lower bounds ensures that the maximum exists. We will prove this later. These assumptions could be replaced by others that ensure the maximum exists.

Value function iteration generates a sequence with $v_{i+1} = T(v_i)$. Recall that for sequences of real numbers, $\{x_i\}$, $x_i \rightarrow x$ means that for any $\epsilon > 0$ there exists an N such that for all $i \geq N$, $|x_i - x| < \epsilon$. Similarly, for functions v_i, v_i converges to v if for all $\epsilon > 0$ there exists an N such that for all $i \geq N$, $\sup_s |v_i(s) - v(s)| < \epsilon$. In other words, $v_i \rightarrow v$ means that as

i increases the maximum of the absolute value of the difference between v_i and v shrinks to 0.

Consider

$$T(v_0)(s) - T(v_1)(s) = \big(u(c_0, s) + \beta v_0(s_0')\big) - \big(u(c_1, s) + \beta v_1(s_1')\big)$$

where c_i, s_i' is the maximizer to

$$\max_{c_i, s_i'} u(c_i, s) + \beta v_i(s_i')$$

$$\text{s.t. } g_0(c_i, s) \leq s_i' \leq g_1(c_i, s),$$

$$\underline{c} \leq c_i \leq \overline{c}$$

Notice that

$$Tv_0(s) = u(c_0, s) + \beta v_0(s_0') \geq u(c_1, s) + \beta v_0(s_1').$$

Therefore,

$$T(v_0)(s) - T(v_1)(s) \geq \beta(v_0(s_1') - v_1(s_1')).$$

Similarly,

$$T(v_0)(s) - T(v_1)(s) \leq \beta(v_0(s_0') - v_1(s_0')).$$

It follows that

$$\sup_s |T(v_0)(s) - T(v_1)(s)| \leq \sup_s \big|\beta(v_0(s) - v_1(s))\big|$$

We assumed that $\beta < 1$. Thus, we have shown that $T(v_0)$ and $T(v_1)$ are closer together than v_0 and v_1. A mapping with this property is called a contraction mapping. To show that the value function is unique, suppose $v = T(v)$ and $\tilde{v} = T(\tilde{v})$. Then,

$$\sup_s |v(s) - \tilde{v}(s)| = \sup_s |T(v)(s) - T(\tilde{v})(s)| \leq \beta \sup_s |v(s) - \tilde{v}(s)|$$

which is only possible if $\sup_s |v(s) - \tilde{v}(s)| = 0$, i.e. $v = \tilde{v}$.

To show that v exists we must show that the sequence of v_i converges. Since T contracts and makes the v_i closer and closer together, it makes sense that v_i should converge. Showing this formally is tedious. [1]

A defining assumption about real numbers is that every set of numbers bounded above has a least upper bound. See the notes on sets for more information. An immediate consequence of this assumption is that any increasing sequence of real numbers that is bounded above, must converge. The argument below will use this fact to show that the sequence of v_i converge.

Choose a sequence $\{\epsilon_i\}$ with $\epsilon_i \to 0$. Let N_i be such that

$$\sup_s \big|\beta(v_n(s) - v_{N_i}(s))\big| < \epsilon_i$$

---

[1]If you are familiar with the fact that Cauchy sequences converge in complete metric spaces, then it is easier. We have not yet covered Cauchy sequences or metric spaces, so we will not use this fact.

for all $n \geq N_i$. Such an N_i exists because $v_n = \underbrace{T(T(...T(v_0)...)}_{n \text{ times}} = T^n v_0$, and by repeating the same argument as above,

$$\sup_s \left| T^N(T^{n-N}(v_0))(s) - T^N(v_0)(s) \right| \leq \beta^N \sup_s \left| (T^{n-N} v_0(s) - v_0(s)) \right|.$$

Also, since u is bounded by say M, $|v_i(s)|$ is at most $M/(1-\beta)$, so $\sup_s \left| (T^{n-N} v_0(s) - v_0(s)) \right| \leq 2M/(1-\beta)$. Thus,

$$\sup_s \left| T^N(T^{n-N}(v_0))(s) - T^N(v_0)(s) \right| \leq \beta^N 2M/(1-\beta)$$

so we can always choose e.g. $N_i = \log(\epsilon_i(1-\beta)/(2M))/\log(\beta) + 1$.

Define a new sequence of functions U_i with

$$U_1(s) = v_{N_1}(s) + \epsilon_1$$

and

$$U_{i+1}(s) = \max\{U_i(s), v_{N_{i+1}}(s) + \epsilon_{i+1}\}.$$

Similarly define L_i but subtracting ϵ_i instead of adding it. Note that by definition for all $n \geq N_i$ and all s, $L_i(s) \leq v_n(s) \leq U_i(s)$. Also, $\sup_s |L_i(s) - U_i(s)| \leq 2\epsilon_i$, and $U_i(s)$ decreases and $L_i(s)$ increases with i for each s.

For each s, $U_i(s)$ is decreasing sequence of numbers bounded below by $L_1(s)$. Therefore, it must converge. Similarly, $L_i(s)$ is an increasing sequence bounded above, so it must converge.[2] Also, since $|L_i(s) - U_i(s)| \leq 2\epsilon_i \to 0$, it must be that $\lim_{i \to \infty} L_i(s) = \lim_{i \to \infty} U_i(s)$. Define $v(s)$ as this limit. Then, for $n \geq N_i$,

$$\sup_s |v_n(s) - v(s)| \leq \sup_s |v_n(s) - L_i(s)| + \sup_s |L_i(s) - v(s)|$$

$$\leq \epsilon_i + 2\epsilon_i$$

We can then conclude that $v_n \to v$. □

3. Solving dynamic programs

There are three basic ways to solve a dynamic program. They are:

(1) Guess and verify the form of the value function
(2) Iterate the Bellman equation analytically
(3) Iterate (or use some other alogirthm to solve) the Bellman equation numerically

If you guess correctly, the first method is fairly straightforward. However, guessing correctly is difficult and often is not possible at all. The second method will always work, but may not lead to a closed form expression, and can be tedious. The third method is the main way dynamic programs are solved in practice, but we will not go into the details.

Example 3.1 (Optimal growth by guessing and verifying). Consider an economy with a single infinitely lived representative consumer with per-period log utility from

---

[2]We will prove that bounded monotonic sequences converge later.

consumption and a discount factor of δ. The economy's production function is Cobb-Douglas with capital as the only input. Anything not consumed at time t becomes capital at time $t + 1$. The optimal growth problem is

$$\max_{\{c_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \delta^t \log(c_t)$$

$$\text{s.t. } c_t + k_{t+1} = k_t^{\alpha}.$$

If we use the constraint to solve for c_t and substitute into the objective, then we have

$$\max_{\{k_t\}_{t=1}^{\infty}} \sum_{t=0}^{\infty} \delta^t \log(k_t^{\alpha} - k_{t+1})$$

$$\text{s.t.} 0 \leq k_{t+1} \leq k_t^{\alpha}$$

The Bellman equation for this problem is

$$v(k) = \max_{k' \in [0, k^{\alpha}]} \log(k^{\alpha} - k') + \delta v(k')$$

Now, we guess the functional form of v. Since the per-period utility function is logarithmic and production is Cobb-Douglas, it is sensible to guess that $v(k) = c_0 + c_1 \log(k^a)$ where c_0, c_1, and a are each constant for which we solve. Now, since $c_1 \log(k^a) = c_1 a \log(k)$, a and c_1 are redundant, so we can get rid of a, and just guess that $v(k) = c_0 + c_1 \log(k)$.

We now use the Bellman equation to solve for c_0 and c_1. First we solve for the optimal k' for a given c_0 and c_1. The Bellman equation is:

$$c_0 + c_1 \log k = \max_{k' \in [0, k^{\alpha}]} \log(k^{\alpha} - k') + \delta \left(c_0 + c_1 \log k' \right).$$

We could write the Lagrangean with the constraints that $k' \geq 0$ and $k' \leq k^{\alpha}$. If we were not sure whether these constraints would bind we would include them in the Lagrangean and check the complementary slackness conditions. However, it is slightly easier to just notice that these constraints cannot bind because utility approaches $-\infty$ as k approaches k^{α} and the next period's value approaches $-\infty$ as k' approaches 0, so neither constraint will bind. Without the constraints, the first order condition is:

$$-\frac{1}{k^{\alpha} - k'} + \delta c_1 \frac{1}{k'} = 0$$

$$-k' + \delta c_1 (k^{\alpha} - k') = 0$$

$$k' = \frac{\delta c_1}{1 + \delta c_1} k^{\alpha}$$

Now, we plug this back into the Bellman equation and solve for c_0 and c_1 by varying k.

$$c_0 + c_1 \log k = \max_{k' \in [0, k^\alpha]} \log(k^\alpha - k') + \delta \left(c_0 + c_1 \log k' \right)$$

$$= \log \left(k^\alpha - \frac{\delta c_1}{1 + \delta c_1} k^\alpha \right) + \delta \left(c_0 + c_1 \log \left(\frac{\delta c_1}{1 + \delta c_1} k^\alpha \right) \right)$$

$$= \log \left(\frac{1}{1 + \delta c_1} \right) + \alpha \log k + \delta \left(c_0 + c_1 \log \left(\frac{\delta c_1}{1 + \delta c_1} \right) + \alpha \log k \right)$$

$$= \underbrace{\log \left(\frac{1}{1 + \delta c_1} \right) + \delta c_0 + \delta c_1 \log \left(\frac{\delta c_1}{1 + \delta c_1} \right)}_{=c_0} + \underbrace{(\alpha + \delta c_1 \alpha)}_{=c_1} \log k$$

Both the left and right sides of this equation are affince function of $\log k$. They can only be equal for all k if the coefficients are equal. Thus,

$$c_1 = \alpha + \delta c_1 \alpha$$

$$c_1 = \frac{\alpha}{1 - \delta \alpha}$$

and

$$(1 - \delta)c_0 = \log \left(\frac{1}{1 + \delta c_1} \right) + \delta c_1 \log \left(\frac{\delta c_1}{1 + \delta c_1} \right)$$

$$= - \left(1 + \delta \frac{\alpha}{1 - \delta \alpha} \right) \log \left(1 + \delta \frac{\alpha}{1 - \delta \alpha} \right) + \delta \frac{\alpha}{1 - \delta \alpha} \log \left(\delta \frac{\alpha}{1 - \delta \alpha} \right)$$

$$= \log(1 - \delta \alpha) + \frac{\delta \alpha}{1 - \delta \alpha} \log(\delta \alpha).$$

Finally, we should make sure that this solution doesn't violate the constraint. We have

$$k' = \frac{\delta c_1}{1 + \delta c_1} k^\alpha = \delta \alpha k^\alpha,$$

so the constraints are satisfied as long as $\delta \alpha \in (0, 1)$.

If we cannot guess the form of the value function, we can try to find it by repeatedly applying the Bellman operator. The Bellman operator is the T operator we defined above,

$$T(v)(s) = \max_{c, s'} u(c, s) + \beta v_0(s') \text{ s.t. } s' = g(c, s).$$

We already showed that T is a contraction (provided u is bounded and $|\beta| < 1$). Among other things, this means that if we start with an arbitrary guess of the value function, v_0, and then construct a sequence by repeatedly applying T, i.e.,

$$v_i = T(v_{i-1}),$$

then the sequence v_i will converge to a unique fixed point, v, that satisfies the Bellman equation.

Example 3.2 (Optimal growth by iterating). The same optimal growth problem as in the previous example can also be solved by iterating the Bellman operator. We start with *any* guess of the value function for v_0. A common choice is the zero function, $v_0(k) = 0$ for all k. Then we find v_1 by solving

$$v_1(k) = \max_{k' \in [0, k^\alpha]} \log(k^\alpha - k') + \delta v_0(k)$$

$$= \max_{k' \in [0, k^\alpha]} \log(k^\alpha - k')$$

$$= \alpha \log k.$$

Then, we repeat to find v_2.

$$v_2(k) = \max_{k' \in [0, k^\alpha]} \log(k^\alpha - k') + \delta v_1(k)$$

$$= \max_{k' \in [0, k^\alpha]} \log(k^\alpha - k') + \delta \alpha \log(k')$$

$$= c_2 + (\alpha + \delta \alpha^2) \log k,$$

where c_2 is some constant that involves δ, α, and their logs. The third equality comes from writing the first order condition, solving for k', and subsituting back into the objective. We can explicitly solve for c_2, but it doesn't matter for the first order condition for v_3, so we don't need to know it exactly. We repeat again to get v_3

$$v_3(k) = \max_{k' \in [0, k^\alpha]} \log(k^\alpha - k') + \delta v_1(k)$$

$$= \max_{k' \in [0, k^\alpha]} \log(k^\alpha - k') + \delta \alpha \log(k')$$

$$= c_3 + (\alpha + \delta \alpha^2 + \delta^2 \alpha^3) \log k.$$

We could repeat again to get

$$v_4(k) = c_4 + (\alpha + \delta \alpha^2 + \delta^2 \alpha^3 + \delta^3 \alpha^4) \log k$$

$$v_5(k) = c_5 + (\alpha + \delta \alpha^2 + \delta^2 \alpha^3 + \delta^3 \alpha^4 + \delta^4 \alpha^5) \log k$$

$$\vdots$$

etc. Eventually, we hopefully notice a pattern. The more obvious pattern is that each v_i and will always be of the form $v_i(k) = c_i + m_i \log(k)$. Thus, we know that $v(k)$ will have that same form and we can go back to the guess and verify method. Better yet, we could notice that

$$v_i(k) = c_i + \alpha \sum_{j=0}^{i} (\alpha \delta)^j \log(k),$$

so

$$v(k) = C + \frac{\alpha}{1 - \alpha \delta} \log k.$$

If we care about C, we could find it by either explicitly writing c_i in terms of δ and α and taking the limit; or using the guess verify method just for C.

Solving for the value function, whether by guessing and verifying or iterating can be a bit tedious. Even worse, for most specifications of the per-period payoff u and constraints g, there will be no closed form solution for v. That makes it impossible to guess the form, and iterating the Bellman equation will not lead to a discernible pattern (although it will still give a convergent sequence). Using a computer to solve for the value function avoids both these problems. A computer does not care that Bellman operator iteration is tedious, and it can numerically compute $v(k)$ even if it has no closed form.

Example 3.3 (Optimal growth numerically). We can also solve for the value function numerically. The simplest (although usually not the fastest) method is to iterate the Bellman operator. This can be done as follows:

- Choose a grid of capital values k_g for $g = 1, ..., G$
- Set $\hat{v}_0(k) = 0$ for all k (or anything else)
- Repeatedly:
  (1) Maximize the Bellman equation for each k_g

$$v_{g,i} = \max_{c,k'} u(c) + \delta \hat{v}_{i-1}(k') \text{ s.t. } k' = f(k_g) - c$$

  (2) Set $\hat{v}_i(k)$ = linear interpolation of $\{k_g, v_{g,i}\}$
  Stop when the value function stops changing, e.g. when

$$\max_g \left| v_{g,i} - v_{g,i-1} \right| < \epsilon$$

Figure 1 shows the resulting value function approximations for various i. Figure 2 shows the optimal policies for c and k' given k.

These figures were created using this R code. Sargent and Stachurski (2013) has a similar examples in Python and Julia.
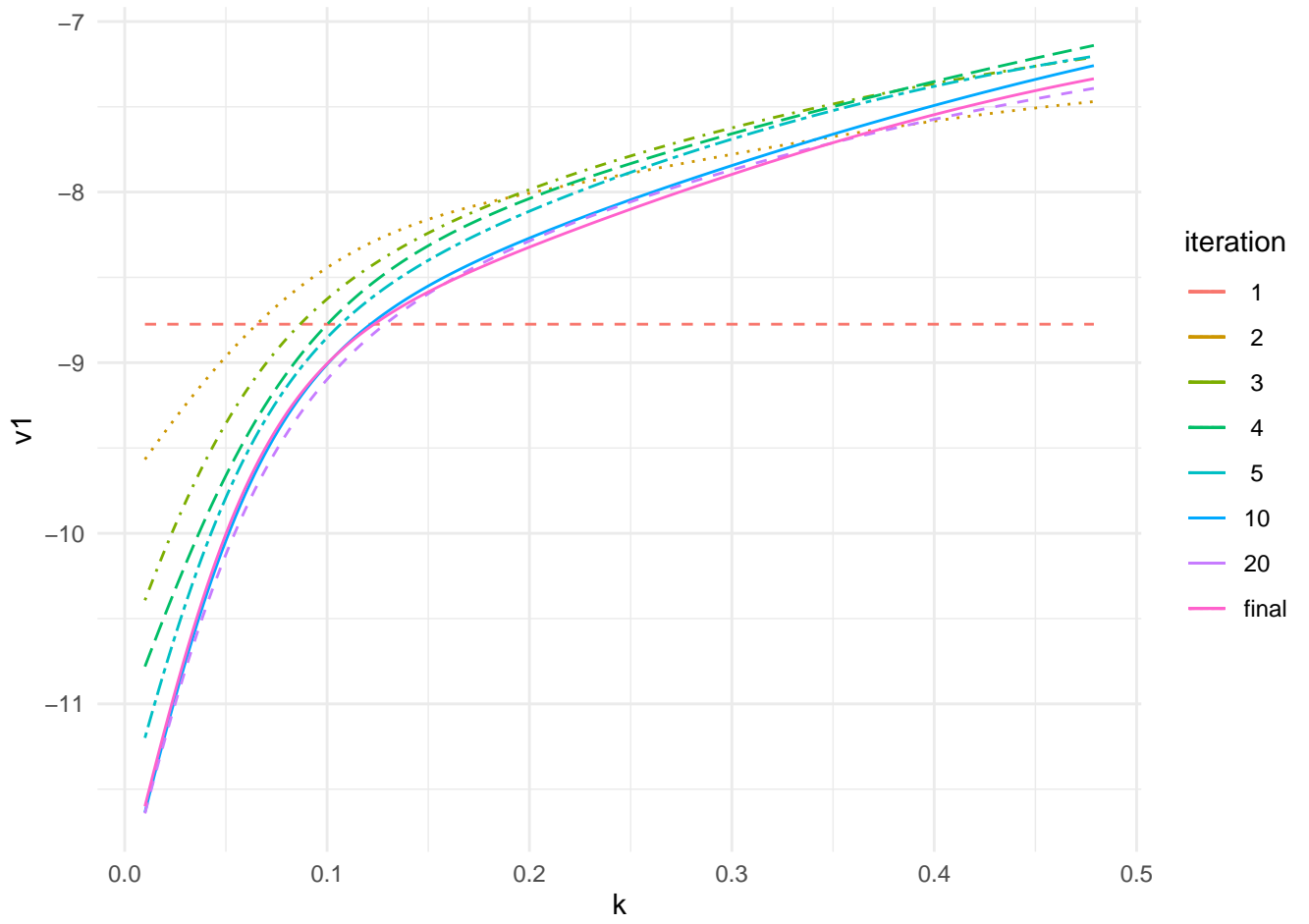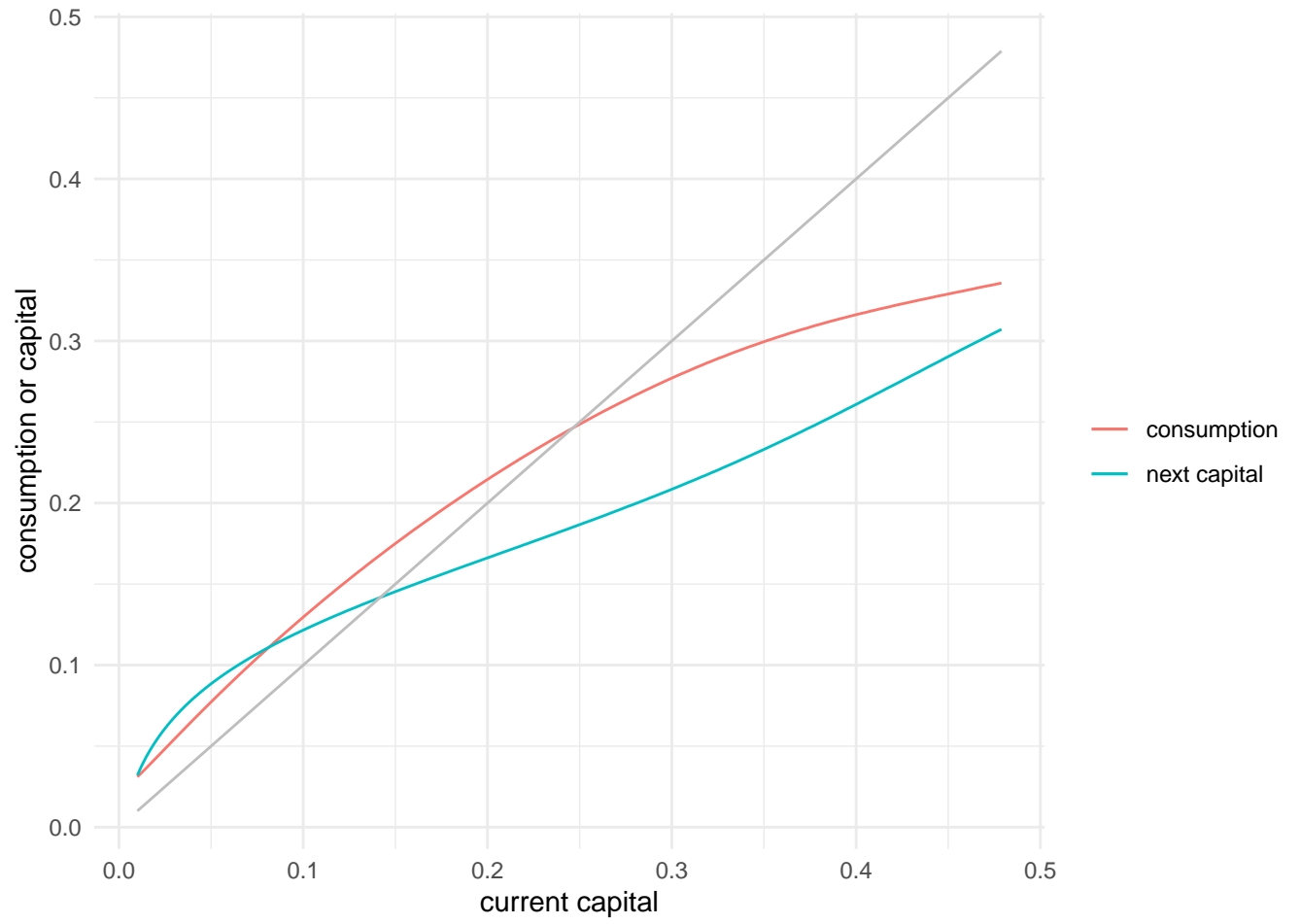

FIGURE 1. Value function iterations

FIGURE 2. Policy function

3.1. **Discrete control.** Another situation where dynamic programs can be solved analytically is when the control variable is discrete. For example, a person could be choosing to work or not each period, or a firm could be choosing to enter or exit a market.

Example 3.4 (Investment option). Each period an investor has an opportunity to invest in a project. Once the investment is made, the model ends. The investment costs I. At time t the project pays z_t. z_t is drawn independently from some distribution with pdf f, cdf F, and support $[0, B]$. The investor is risk neutral with discount rate β. At time t, the investor can choose to invest now or wait. Investing now pays $z_t - I$. Let $V(z)$ denote the value of this choice given the current z. The Bellman equation is

$$V(z) = \max\{z - I, \beta \int V(z')f(z')dz'\}.$$

The value of waiting, $\beta \int V(z')f(z')dz'$, does not depend on z, so the optimal decision is to invest if $z \geq z^*$ for some threshold z^*. At z^*, the investor is indifferent between waiting and investing, so

$$z^* - I = \beta \int_0^B V(z')f(z')dz'$$

$$= \beta \int_0^{z^*} \underbrace{\left[\beta \int_0^B V(z')f(z')dz'\right]}_{=z^*-I} f(\tilde{z})d\tilde{z} + \beta \int_{z^*}^B (z' - I)f(z')dz'$$

$$= \beta \int_0^{z^*} (z^* - I)f(\tilde{z})d\tilde{z} + \beta \int_{z^*}^B (z' - I)f(z')dz'$$

$$\left(\text{add and subtract } \int_{z^*}^B z^* f(z')dz' \right)$$

$$= \beta \int_0^B (z^* - I)f(\tilde{z})d\tilde{z} + \beta \int_{z^*}^B (z' - z^*)f(z')dz'$$

$$z^* - I = \frac{\beta}{1 - \beta} \int_{z^*}^B (z' - z^*)f(z')dz'$$

From this, we can see that $z^* > I$. The investor waits for a project with a strictly positive return. Also, we can see that as β increases, z^* increases, i.e. the investor will wait for a higher return. The probability of waiting t periods if $F(z^*)^{t-1}(1 - F(z^*))$. The expected waiting time is then $\frac{1}{1-F(z^*)}$. This too increases with z^*.

We can also think about what happens as the riskiness of the investment changes. Changing the riskiness of investment is some change to the distribution of z. Not all changes in f are increases in riskiness. One way to consider a change in f that is an increase in risk, is to look at change that holds $E[z]$ constant but increases the variance of z. Such a change is called a mean-preserving spread. We can always create a mean-preserving spread by replacing z with $\tilde{z} = z + \epsilon$ where $E[\epsilon|z] = 0$ and $V(\epsilon) \geq 0$. A function g is concave if for all z there exists $d(z)$ such that $g(z + \epsilon) \geq g(z) + d(z)\epsilon$ for

---

13

all ϵ. If g is differentiable, then d is just the derivative of g. For any concave function if $\tilde{z}$ is a mean preserving spread of z, then

$$E[g(\tilde{z})] = E[g(z + \epsilon)] \geq E[g(z) + d(z)\epsilon]$$
$$\geq E[g(z) + d(z)E[\epsilon|z]]$$
$$E[g(\tilde{z})] \geq E[g(z)].$$

[a] Now, let

$$g_{z^*}(z) = \max\{0, z - z^*\}.$$

This function is concave and decreasing in z^*. Also, from the above,

$$z^* - I = \frac{\beta}{1 - \beta}E[g_{z^*}(z)]$$

Thus, if we add a mean-preserving spread to z, then $E[g_{z^*}(z)]$ will increase, and z^* must increase to compensate.

---

[a]More generally, we say that the distribution of $\tilde{z}$ second-order stochastic dominates the distribution of z if this inequality holds for all concave functions g.

3.2. **Uncertainty.** Dynamic programming is often especially useful for dealing with models with uncertainty. The investment option model above is one example. Let's look at another.

Example 3.5 (Growth with random productivity). Let's consider the same growth model as before, but now production is $A_t k_t^\alpha$, where A_t is productivity. We will assume that productivity follows a Markov process. That is, the distribution of A_{t+1} connditional on all information at time t, only depends on A_t. The problem is to choose c_t given the current k_t and A_t. The Bellman equation is

$$V(k_t, A_t) = \max_{c_t, k_{t+1}} \log(c_t) + \delta E[V(k_{t+1}, A_{t+1})|A_t]$$
$$\text{s.t. } c_t + k_{t+1} = A_t k_t^\alpha$$

Let's guess and verify that the value function will again be log-linear with a constant that depends on A_t, i.e.

$$V(k, A) = b_1 \log(k) + b_0(A)$$

Putting in this guess for V, and substituting in the constraint to eliminate c, the problem becomes

$$V(k, A) = \max_{k'} \log(A k^\alpha - k') + \delta E[b_1 \log(k') + b_0(A')|A]$$

Solving the first order condition gives

$$k' = k^\alpha \frac{A \delta b_1}{1 + \delta b_1}$$

Putting this k' back into our guess for V and the Bellman equation gives

$$b_1 \log(k) + b_0(A) = \log\left(k^\alpha \frac{A}{1 + \delta b_1}\right) + \delta E\left[b_1 \log\left(Ak^\alpha \frac{A\delta b_1}{1 + \delta b_1}\right) + b_0(A')|A\right]$$

$$b_1 \log(k) + b_0(A) = \alpha \log(k)(1 + \delta b_1) + (1 + \delta b_1)\log(A) + \delta E[b_0(A')|A] + D$$

where $D = \delta b_1 \log(\delta b_1) - (1 + \delta b_1)\log(1 + \delta b_1)$. For this equation to hold for any k, it must be that

$$b_1 = \alpha(1 + \delta b_1)$$

so

$$b_1 = \frac{\alpha}{1 - \alpha\delta}.$$

We must also have

$$b_0(A) = (1 + \delta b_1)\log(A) + \delta E[b_0(A')|A] + D.$$

The exact value of $b_0(A)$ depends on the distribution of A' given A. The simplest case is if A is independent over time. Then $E[b_0(A')|A] = E[b_0(A')]$. In that case, we get

$$E[b_0(A)] = E\left[(1 + \delta b_1)\log(A) + \delta E[b_0(A)] + D\right]$$

$$E[b_0(A)] = \frac{1}{1 - \delta}\left((1 + \delta b_1)E[\log(A)] + D\right)$$

Hence,

$$b_0(A) = \frac{\log(A)}{1 - \alpha\delta} + \frac{\delta}{1 - \delta}\frac{E[\log(A')]}{1 - \alpha\delta} + \frac{D}{1 - \delta}$$

and the value function is

$$V(k, A) = \frac{\alpha}{1 - \alpha\delta}\log k + \frac{\log(A)}{1 - \alpha\delta} + \frac{\delta}{1 - \delta}\frac{E[\log(A')]}{1 - \alpha\delta} + \frac{D}{1 - \delta}$$

REFERENCES

Acemoglu, Daron. 2008. *Introduction to modern economic growth*. Princeton University Press.

Adda, Jerome and Russell W Cooper. 2003. *Dynamic economics: quantitative methods and applications*. MIT press. URL http://site.ebrary.com/lib/ubc/reader.action?docID=10225265.

Bellman, Richard Ernest and Stuart E Dreyfus. 1962. *Applied dynamic programming*. Rand Corporation.

Bertsekas, Dimitri P. 1976. *Dynamic programming and stochastic control*. Academic Press, Inc.

Carter, Michael. 2001. *Foundations of mathematical economics*. MIT Press.

De la Fuente, Angel. 2000. *Mathematical methods and models for economists*. Cambridge University Press.

Dixit, Avinash K. 1990. *Optimization in economic theory*. Oxford University Press Oxford.

Dreyfus, Stuart. 2002. "Richard Bellman on the birth of dynamic programming." *Operations Research* 50 (1):48–51.

Dreyfus, Stuart E. 1965. *Dynamic programming and the calculus of variations*. Academic Press New York.

Sargent, Thomas and John Stachurski. 2013. "Quantitative Economics." URL http://quant-econ.net/.

Stokey, Nancy L, Robert E Lucas Jr, and Edward C Prescott. 1989. *Recursive Methods in Economic Dynamics*. Harvard University Press.