EC316a: Advanced Scientific Computation, Fall 2003

Notes Section 2

Discrete time, discrete state dynamic models

Dynamic economic models often present three complications typically absent from dynamic models in the physical sciences. Humans are forward–looking beings, able to evaluate how their actions will affect them in the future as well as the present, so that the most useful dynamic models are forward–looking. Second, human behavior is innately unpredictable, requiring a stochastic model. Third, the predictable component of human behavior is often complex, requiring an inherently nonlinear model. Thus anything beyond a trivially simple model will lack an explicit analytical solution, and numerical methods must be employed. In this section of the module, we consider a simple class of models in discrete time—that is, the decisions made, and resulting outcomes, appear at discrete intervals of time–that arise from a discrete–state model, in which a finite number of states of the world may occur at each discrete time. The simplest of these frameworks, which we study here, is the DTDS Markov decision model, where the state of the economic system is a controlled Markov process.

A Markov process is a sequence of random variables, $\{X_t \mid t = 0, 1, 2, \ldots\}$ defined on a state space S whose distributions depend only on the immediately prior value of the sequence: that is, X_{t+1} depends only upon X_t. Such a process is said to be memoryless since the distribution conditioned on the entire history of the process is completely determined by the most recent element of that history. A Markov chain is a Markov process with a finite state space $S = \{1, 2, 3, \ldots, n\}$ which is completely characterized by its transition probabilities $P_{tij} = Pr\{X_{t+1)} = j | X_t = i\},\ i, j \in S$. It is said to be stationary if the transition probabilities are not dependent on time, with transition probability matrix P. A stationary Markov chain will have a steady–state distribution π which, if it exists, will completely characterize the long–run behavior of the chain. Existence depends on the Markov chain being irreducible: starting from every state, there is a positive probability of eventually visiting every other state. Given transition probability matrix P, if there exists an n-vector π such that

$$P'\pi = \pi, \sum_i \pi_i = 1$$

then the steady–state distribution exists, and can be computed from a system of linear equations. This distribution indicates what percentage of the time the system will spend in each of the states. A very simple example of a Markov process: the macroeconomy may have two states, (1) expansion and (2) recession. Consider that an expansion has a 70% probability of being sustained next quarter, while a recession has a 40% probability of being sustained. Then the Markov transition probability matrix is

$$\begin{pmatrix} 0.7 & 0.3 \\ 0.6 & 0.4 \end{pmatrix}$$

where each row sums to one (an expansion has an 30% probability of ending in a particular period; a recession has a 60% probability of ending). The steady–state distribution exists, and may be calculated from the linear equation system

$$\begin{pmatrix} I - P' \\ 1 \end{pmatrix} \pi = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

in which one row of P' may be dropped, due to linear dependencies, to arrive at a unique solution. In the example above, the steady-state distribution has $\pi_1 = 2/3$ and $\pi_2 = 1/3$: that is, the economy spends 2/3 of the time in expansion in the long run.

The DTDS Markov model has the following structure: in every period t, an economic agent observes the state of the system s_t, takes an action x_t, and earns a reward $f(s_t, x_t)$ which depends both on the state of the system and the action taken: that is, the reward is said to be state–contingent. The state space S and action space X are both finite, and the state is a controlled Markov process, depending only on the prior state and the agent's action at this point in time. The solution to this model is a sequence of policies $\{x_t^*\}$ that prescribe the action $x_t = x_t^*(s_t)$ that should be taken in any given state and period so as to maximize the present value of current and expected future rewards over a time horizon T, discounted at a per–period factor δ.

A discrete Markov model may be defined over an infinite horizon ($T = \infty$) or a finite horizon, and may be either deterministic or stochastic. In the prior case, next period's state is known with certainty once the current state and action are known, and we may define the deterministic state transition function $s_{t+1} = g(s_t, x_t)$.

Discrete Markov decision models may be analyzed using dynamic programming, and Bellman's principle of optimality: the notion that an optimal policy for this period is formed given that all future decisions are optimal, as expressed in the Bellman equation defining the value function for each point in time. The agent must optimally trade off an immediate reward $f(s_t, x_t)$ against expected future rewards $\delta E_t V_{t+1}(s_{t+1})$, where $V()$ is the value function.

In a finite–horizon problem, the optimizing agent faces decisions in periods 1 through T, and may earn a final reward $V_{T+1}(s_{T+1})$ depending on the realization of the state in that period (for instance, an investment project may have a scrap or salvage value at the end of its useful life, with a value dependent on the current market for used capital goods). Given the terminal value function V_{T+1}, the problem may be solved recursively from the last period back to the first period. In an infinite–horizon problem, the value functions will not depend on time, but may be derived by a contraction mapping as long as the discount factor δ is less than one.

Let us now consider some concrete examples of this type of model, and how solutions to these models might be computed. As a first example, we consider management of a mine. The mine will be shut down and abandoned after T years of operation. The price of extracted ore is \$$p$ per ton, and the total cost of extracting x tons of ore in a year, given that the mine contains s tons at the beginning of the year, is \$$c(s,x)$, reflecting the fact that extracting the deeper ore may be more costly. The mine currently contains $\bar{s}$ tons of ore. If we restrict output to an integer number of tons, what extraction schedule maximizes profits of operating the mine over this finite horizon?

The state variable s, $s \in \{0,1,2,\ldots,\bar{s}\}$ is the amount of ore remaining at the beginning of each year. The action variable x, $x \in \{0,1,2,\ldots,s\}$ is the amount extracted over the year. The state transition function keeps track of the ore: $g(s,x) = s - x$, while the reward function is $f(s,x) = p\ x - c(s,x)$. The value of the mine, given that it contains s tons of ore at the beginning of year t, satisfies the Bellman equation

$$V_t(s) = \max_{s \in \{0,1,2,\ldots,s\}} \{p\ x - c(s,x) + \delta V_{t+1}(s-x)\}$$

subject to the terminal condition $V_{T+1}(s) = 0$.

How might we set up a computational solution for such a problem? Assume that the states $S = \{1, 2, \ldots, n\}$ and actions $X = 1, 2, \ldots, m$ are given by the first n and m integers. $v_i \in \Re$ will denote an arbitrary value vector, for the value in state i, while $x_i \in X$ will denote the action in state i. For each policy x, let $f(x) \in \Re^n$ denote the n-vector of rewards earned in each state when one follows the prescribed policy: $f_i(x)$ is the reward in state i, given action x_i taken. For problems with constrained actions (e.g. mining more ore than exists in the mine) we set $f_i(x) = -\infty$ to indicate that a certain action is not admissible. Finally, let $P(x) \in \Re^{n\ x\ n}$ denote the $n\ x\ n$ state transition probabilities when one follows the prescribed policy: that is, $P_{ij}(x)$ is the probability of a jump from state i to j, given that action x_i is taken. Given this notation, the Bellman equation for a finite–horizon problem may be expressed as a recursive vector equation. If $v_t \in \Re^n$ denotes the value function in period t, then

$$v_t = \max_x [\ f(x) + \delta P(x) v_{t+1}\ ]$$

where max is the vector operation of taking the maximum element of each row individually. The Bellman equation for an infinite–horizon problem may be written similarly as a vector fixed–point equation.

Dynamic programming problems are subject to the "curse of dimensionality": any solution algorithm may be expressed as a set of three nested loops, where for each time, we must consider each possible state, and within each state, each possible action. The computational effort is roughly proportional to the product of the number of times each loop must be executed: for n states and m actions, then

$n\ m$ total actions must be evaluated for each outer iteration (time period). The dimensionality of the state and action variables is particularly important. In the mine management example, there is only one state variable, but in many DTDS dynamic models, there are multiple state variables. The computational effort grows exponentially with the dimensionality of the state space, since for a k–dimensional state variable with l levels, the number of states is $n = l^k$. The same will be true for the dimensionality of the action space (e.g., if the economic agent must choose two or more policies each period). This "curse" is the binding constraint on computational solution of these problems.

Let us consider a numerical solution for the mine management example (`demddp01`). Let market price p=1, initial stock of ore $\bar{s}$=100, the cost of extraction $c(s, x) = x^2/(1 + s)$ and the annual discount factor δ=0.9. We define state and action spaces S and X as integer vectors $(0:\bar{s})'$, and scalars n and m as the length of those spaces, respectively. The reward matrix is then defined as the profit to be realized for each period's activity, in current dollars, with the reward set to $-\infty$ for extraction beyond the feasible scale. This is a deterministic model, so that the state transition rule is just the accounting identity defining the g matrix as the amount of ore remaining, given the previous state and current action. The CompEcon routine `ddpsolve` generates a solution, returning the n–vector of values v, the n–vector of optimal actions x, and the n x n controlled state transition probability matrix P^*. The `ddpsimul` routine may be used to forecast the optimal path of the state variable over a finite horizon.

A second example: a manufacturer who must decide whether an existing capital good should be replaced or kept in service. Assume that these goods have a maximum life of n years, after which they must be replaced by law. An a–year–old good yields a profit contribution of $p(a)$, and a new capital good costs c. What is the profit–maximizing asset replacement policy?

The state variable $a \in \{1, 2, 3, \ldots, n\}$ is the age of the asset in years. The action variable x for each year takes on a value $keep$ or $replace$. The state transition function is thus

$$g(a, x) = \begin{cases} a + 1, & x = keep \\ 1, & x = replace \end{cases}$$

with reward function

$$f(a, x) = \begin{cases} p(a), & x = keep \\ p(0) - c, & x = replace \end{cases}$$

The value of an asset of age a satisfies the Bellman equation

$$V(a) = \max\{p(a) + \delta V(a+1), p(0) - c + \delta V(1)\}$$

where we set $p(n) = -\infty$ to enforce retirement of an n–year–old asset. If the decision is $keep$, the manufacturer earns $p(a)$ over the coming year and begins the next year with an asset one year older and worth $V(a+1)$. If she replaces the asset, she earns $p(0) - c$ over the year, and begins the next year with a one–year–old asset worth $V(1)$.

To set up a numerical solution to this problem (`demddp02`), we must define the profit function (e.g., $p(a) = 50 - 2.5a - 2.5a^2$), which is concave in a: that is, an older asset is costlier to operate. If we assume that the maximum asset life is 5 years, a new asset costs \$75, and a discount factor of $\delta = 0.9$, we may solve the problem to find that the asset is replaced every four years, when its profit contribution becomes zero.

A variation on this model may be developed which allows for the possibility of annual maintenance which increases the asset's productivity. Then the action variable takes on one of three values: *no action*, *service* or *replace*. If *no action* is chosen, then the asset is retained, but not serviced. There are now two state variables: a, as before, the age of the asset in years, and s, the number of servicings it has received (each of which costs \$$k$). The state transition and reward functions are then

$$g(a,s,x) = \begin{cases} (a+1,s) & x = no\ action \\ (1,0) & x = replace \\ (a+1,s+1) & x = service \end{cases}$$

and

$$f(a,s,x) = \begin{cases} p(a,s), & x = no\ action \\ p(0,0) - c, & x = replace \\ p(a,s+1) - k, & x = service \end{cases}$$

The Bellman equation for this two–state–variable problem now expresses the asset's value as

$$V(a,s) = \max\{p(a,s) + \delta V(a+1,s),$$

$$p(a, s+1) - k + \delta V(a+1, s+1),$$
$$p(0,0) - c + \delta V(1,0)\}$$

where $p(n, s) = -\infty$ to enforce replacement of a n–year–old asset. The impact of servicings on the productivity of an asset with a five–year lifetime is given by the function

$$p(a, s) = (1 - \frac{(a-s)}{5})(50 - 2.5a - 2.5a^2)$$

so that if unserviced, the profit contribution of an asset is reduced by 20% per year of age. Servicing can undo this reduction entirely (so that the first term is always unity) or stay its course. It is likely, then (depending on the cost of servicing) that the optimal path will involve periodic maintenance of the asset.

The Bellman equation indicates that if an asset of age a with s servicings is replaced, the firm earns $p(0,0) - c$ over the coming year and then has an asset worth $V(1,0)$. If the existing asset is serviced, the firm earns $p(a, s+1) - k$ over the coming year and then has an asset worth $V(a+1, s+1)$. If no action is taken, earnings are $p(a, s)$, and the asset at year's end is worth $V(a+1, s)$. The δ in each term translates that end–of–year value into present terms.

A numerical solution to this problem (`demddp03`) is more complicated than that without maintenance, since there are now two state variables. A grid of values must be defined over a and s that considers all possible combinations of age and number of servicings. Likewise, the reward matrix now must consider the rewards for three possible outcomes at each point in time, so that the dimension of the problem is greater. In a numerical solution to this problem, with a reward function defined to make a serviced asset more productive, the asset is serviced during its four–year expected useful life, at the beginning of the second and third years of operation.

As a last example of the DTDS model, consider binomial asset pricing. An American put option gives the right (but not the obligation) to sell a specified quantity of a commodity at a specified price on or before a specified expiration date. In a binomial asset–pricing model, the underlying commodity price is assumed to follow a two–state jump process. If the price is p at time t, then the price at time $t + \Delta t$ will be pu with probability q and p/u with probability $(1 - q)$, where $u = exp(\sigma\sqrt{\Delta t})$, $q = 1/2 + \frac{\sqrt{\Delta t}}{2\sigma}(r - 1/2\sigma^2)$ and $\delta = exp(-r\Delta t)$. In this framework, r is the annualized continuously compounded interest rate, σ is the annualized volatility of the commodity price, and Δ is the time step, in years. If the price of the commodity is p_0 at time 0, what is the value of an American put option with strike price K expiring T years hence?

This is a finite horizon stochastic model (the first stochastic model we have considered) with time measured in $\Delta t = T/N$ years. The state $p \in \{p_0 u^i | i = -N, -N+1, \ldots, N-1, N\}$ is the commodity price that could occur in each future period, while the action variable x for each point in time is either $hold$ or $exercise$. Once the option is exercised, it is dead. The state transition probabilities are

$$Pr(p'|p,x) = \begin{cases} q, & p' = pu \\ 1-q, & p' = p/u \\ 0, & otherwise \end{cases}$$

with reward function

$$f(p,x) = \begin{cases} 0, & x = hold \\ K-p, & x = exercise \end{cases}$$

That is, the option either expires worthless (if the price of the good is higher than K) or is worth its "moneyness" at that point in time: if the good's price is p, one can buy it at that price, exercise the option, and earn $K-p$ (the amount the option is said to be "in the money") as a riskless profit. If $K-p$ is never positive, the option is worthless at that point in time (and perhaps over its entire lifetime). Buying a put option is a “bearish” bet that the price will decline during the life of the option; the maximum value of the option is thus K if the price of the good falls to (near) zero. Given that the current price is p, the value of the option at the beginning of period i satisfies the Bellman equation

$$V_i(p) = max\{K-p, q\delta V_{t+1}(pu)+(1-q)\delta V_{t+1}(p/u)\}$$

subject to the terminal condition $V_{N+1}(p) = 0$, that is, that the option is worthless if not exercised by period N. At each decision point, the owner must decide whether the option should be exercised (earning $K - p$, given that this amount is positive) or held for another period. The latter choice has the stochastic value given by the second term above: the discounted expected value of the option over the two possible (up/down) states for the price process.

Thus the decision at each point in time is a judgment of whether the option is worth more "dead or alive." At the time of expiration, the option is worth the amount by which it is "in the money"; prior to that time, it is worth more than that (e.g., an "out of the money" option will still have positive value, and trade at a positive price, or option premium) given the likelihood that it will be worth more at a later point. As the time to expiration declines, the "time value" of the option is reduced; thus we consider an option (either put or call) to be a "wasting asset", since its time value is a monotone declining function of its remaining lifetime.

A numerical implementation of the binomial option pricing problem is given by `demddp04`.