15 Controlled Markov Jump Processes

We conclude with models for controlled optimization problems in a continuous-time stochastic setting. This lecture is about controlled Markov jump processes, which are relevant when the state space is discrete.

15.1 The dynamic programming equation

The DP equation in incremental form is

$$F(x,t) = \inf_u \{c(x,u)\delta t + E[F(x(t+\delta t), t+\delta t) \mid x(t) = x\,, u(t) = u)]\}.$$

If appropriate limits exist then this can be written in the limit $\delta t \downarrow 0$ as

$$\inf_u [c(x,u) + F_t(x,t) + \Lambda(u)F(x,t)] = 0\,.$$

Here $\Lambda(u)$ is the operator defined by

$$\Lambda(u)\phi(x) = \lim_{\delta t \downarrow 0} \left[\frac{E[\phi(x(t+\delta t)) \mid x(t) = x, u(t) = u] - \phi(x)}{\delta t} \right] \tag{15.1}$$

or

$$\Lambda(u)\phi(x) = \lim_{\delta t \downarrow 0} E\left[\frac{\phi(x(t+\delta t)) - \phi(x)}{\delta t} \,\middle|\, x(t) = x, u(t) = u \right]$$

the conditional expectation of the 'rate of change' of $\phi(x)$ along the path. The operator Λ converts a scalar function of state, $\phi(x)$, to another such function, $\Lambda\phi(x)$. However, its action depends upon the control u, so we write it as $\Lambda(u)$. It is called the **infinitesimal generator** of the controlled Markov process. Equation (15.1) is equivalent to

$$E[\phi(x(t+\delta t) \mid x(t) = x, u(t) = u] = \phi(x) + \Lambda(u)\phi(x)\delta t + o(\delta t)\,.$$

This equation takes radically different forms depending upon whether the state space is discrete or continuous. Both are important, and we examine their forms in turn, beginning with a discrete state space.

15.2 The case of a discrete state space

Suppose that x can take only values in a discrete set, labelled by an integer j, say, and that the **transition intensity**

$$\lambda_{jk}(u) = \lim_{\delta t \downarrow 0} \frac{1}{\delta t} P(x(t+\delta t) = k \mid x(t) = j,\, u(t) = u)$$

is defined for all u and $j \neq k$. Then

$$\begin{aligned} &E\big[\phi(x(t+\delta t)) \mid x(t) = j, u(t) = u\big] \\ &\qquad = \sum_{k \neq j} \lambda_{jk}(u)\phi(k)\delta t + \left(1 - \sum_{k \neq j} \lambda_{jk}(u)\delta t\right)\phi(j) + o(\delta t)\,, \end{aligned}$$

whence it follows that

$$\Lambda(u)\phi(j) = \sum_k \lambda_{jk}(u)[\phi(k) - \phi(j)]$$

and the DP equation becomes

$$\inf_u \left[c(j,u) + F_t(j,t) + \sum_k \lambda_{jk}(u)[F(k,t) - F(j,t)] \right] = 0\,. \tag{15.2}$$

This is the optimality equation for a **Markov jump process**.

15.3 Uniformization in the infinite horizon case

In this section we explain how (in the infinite horizon case) the continuous time DP equation (15.2) can be rewritten to look like a discrete time DP equation. Once this is done then all the ideas of Lectures 1–6 can be applied. In the discounted cost case (15.2) undergoes the usual modification to

$$\inf_u \left[c(j,u) - \alpha F(j,t) + F_t(j,t) + \sum_k \lambda_{jk}(u)[F(k,t) - F(j,t)] \right] = 0.$$

In the infinite horizon case, everything becomes independent of time and we have

$$\inf_u \left[c(j,u) - \alpha F(j) + \sum_k \lambda_{jk}(u)[F(k) - F(j)] \right] = 0. \tag{15.3}$$

Suppose we can choose a B large enough that it is possible to define

$$\lambda_{jj}(u) = B - \sum_{k \neq j} \lambda_{jk}(u) \geq 0\,,$$

for all j and u. By adding $(B+\alpha)F(j)$ to both sides of (15.3), the DP equation can be written

$$(B+\alpha)F(j) = \inf_u \left[c(j,u) + \sum_k \lambda_{jk}(u)F(k) \right]\,,$$

Finally, dividing by $B+\alpha$, this can be written as

$$F(j) = \inf_u \left[\bar{c}(j,u) + \beta \sum_k p_{jk}(u)F(k) \right]\,, \tag{15.4}$$

where

$$\bar{c}(j,u) = \frac{c(j,u)}{B+\alpha}\,, \quad \beta = \frac{B}{B+\alpha}\,, \quad p_{jk}(u) = \frac{\lambda_{jk}(u)}{B} \quad \text{and} \quad \sum_k p_{jk}(u) = 1\,.$$

This makes the dynamic programming equation look like a case of discounted dynamic programming in discrete time, or of negative programming if $\alpha = 0$. All the results we have for those cases can now be used (e.g., value iteration, OSLA rules, etc.) The trick of using a large B to make the reduction from a continuous to a discrete time formulation is called **uniformization**.

In the undiscounted case we could try a solution to (15.2) of the form $F(j,t) = -\gamma t + \phi(j)$. Substituting this in (15.2), we see that this gives a solution provided,

$$0 = \inf_u \left[c(j,u) - \gamma + \sum_k \lambda_{jk}(u)[\phi(k) - \phi(j)] \right] .$$

By adding $B\phi(j)$ to both sides of the above, then dividing by B, setting $\bar{\gamma} = \gamma/B$, and making the other substitutions above (but with $\alpha = 0$), this is equivalent to

$$\phi(j) + \bar{\gamma} = \inf_u \left[\bar{c}(j,u) + \sum_k p_{jk}(u)\phi(k) \right] , \tag{15.5}$$

which has the same form as the discrete-time average-cost optimality equation of Lecture 6. The theorems and techniques of that lecture can now be applied.

15.4 Example: admission control at a queue

Consider a queue of varying size $0, 1, \ldots$, with constant service rate μ and arrival rate u, where u is controllable between 0 and a maximum value λ. Let $c(x,u) = ax - Ru$. This corresponds to paying a cost a per unit time for each customer in the queue and receiving a reward R at the point that each new customer is admitted (and therefore incurring reward at rate Ru when the arrival rate is u). Let us take $B = \lambda + \mu$, and without loss of generality assume $B = 1$. The average cost optimality equation from (15.5) is

$$\begin{aligned} \phi(0) + \gamma &= \inf_u [-Ru + u\phi(1) + (\mu + \lambda - u)\phi(0)] , \\ &= \inf_u [u\{-R + \phi(1) - \phi(0)\} + (\mu + \lambda)\phi(0)] , \end{aligned}$$

$$\begin{aligned} \phi(x) + \gamma &= \inf_u [ax - Ru + u\phi(x+1) + \mu\phi(x-1) + (\lambda - u)\phi(x)] , \\ &= \inf_u [ax + u\{-R + \phi(x+1) - \phi(x)\} + \mu\phi(x-1) + \lambda\phi(x)] , \quad x > 0 . \end{aligned}$$

Thus u should be chosen to be 0 or 1 as $-R + \phi(x+1) - \phi(x)$ is positive or negative.

Let us consider what happens under the policy that take $u = \lambda$ for all x. The relative costs for this policy, say f, are given by

$$f(x) + \gamma = ax - R\lambda + \lambda f(x+1) + \mu f(x-1) , \quad x > 0 .$$

The solution to the homogeneous part of this recursion is of the form $f(x) = d_1 1^x + d_2(\mu/\lambda)^x$. Assuming $\lambda < \mu$ and we desire a solution for f that does not grow exponentially, we take $d_2 = 0$ and so the solution is effectively the solution to the inhomogeneous part, i.e.,

$$f(x) = \frac{ax(x+1)}{2(\mu - \lambda)} , \qquad \gamma = \frac{a\lambda}{\mu - \lambda} - \lambda R ,$$

Applying the idea of policy improvement, we conclude that a better policy is to take $u = 0$ (i.e., don't admit a customer) if $-R + f(x+1) - f(x) > 0$, i.e., if

$$\frac{(x+1)a}{\mu - \lambda} - R > 0 .$$

Further policy improvement would probably be needed to reach the optimal policy. However, this policy already exhibits an interesting property: it rejects customers for smaller queue length x than does a policy which rejects a customer if and only if

$$\frac{(x+1)a}{\mu} - R > 0 .$$

This second policy is optimal if one is purely concerned with whether or not an individual customer that joins when there are x customers in front of him will show a profit on the basis of the difference between the reward R and his expected holding cost $(x+1)a/\mu$. This example exhibits the difference between **individual optimality** (which is myopic) and **social optimality**. The socially optimal policy is more reluctant to admit customers because it anticipates that more customers on the way; thus it feels less badly about forgoing the profit on a customer that presents himself now, recognizing that admitting such a customer can cause customers who are admitted after him to suffer greater delay. As expected, the policies are nearly the same if the arrival rate λ is small.