Lecture notes

Control of Stochastic Processes

Adam Shwartz, Electrical Engineering, Technion, Israel

Version of November 1, 2003
For the latest see
http://www.ee.technion.ac.il/˜adam/GRADUATES/048913

Abstract

These are lecture notes for the course: written in LaTeX2_ε since I need much mathematics, and in English since there is no standard for Hebrew (yet).

1 Introduction

This took place in class no. 1, using notes I posted on the WEB (word-7).

2 Markov chains and Controlled Markov chains

2.1 Markov chains

Definition 2.1 *A discrete time stochastic process* $\{x_0, x_1, \ldots\}$ *is called a Markov chain if, for all* n*, all sets* A *and points* $b_0, b_1, \ldots$,

$$\mathbb{P}\{x_{n+1} \in A \mid x_i = b_i,\ i \leq n\} = \mathbb{P}\{x_{n+1} \in A \mid x_n = b_n\}\ . \tag{2.1}$$

The state *of the Markov chain at time* n *is the value of* x_n*. The* state space $\mathcal{S}$ *of the chain is the space of all possible states.*

Fine points: for non countable state spaces, need to define the conditional probability in the right way.

We shall deal with *discrete state spaces*, e.g. integers. In this case we shall denote

$$p_{ij} \stackrel{def}{=} \mathbb{P}\{x_{n+1} = j \mid x_n = i\}\ . \tag{2.2}$$

Definition 2.2 *If the rhs of* (2.2) *does not depend on* n*, we call the Markov chain* homogeneous*, and the notation of* (2.2) *is valid. We denote* $P \stackrel{def}{=} \{p_{ij}\}$ *and call it the transition matrix. We define the row-vector*

$$\mu_n(j) \stackrel{def}{=} \mathbb{P}\{x_n = j\}\ . \tag{2.3}$$

Obviosly, μ_n depends on the distribution of x_0.

Homogeneous MC are sometimes called stationary: a poor terminology.

Example 2.3 *The sequence of coin tosses (get $+1$ for heads, -1 for tails) is a Markov chain. Write the distribution of x_1 as a function of x_0, and as a function of μ_0. What is the structure of the transition matrix?*

Theorem 2.4 *Given an initial state x and a transition matrix P, there exists a probability measure $\mathbb{P}_x$ and a Markov proceses $x_0, x_1, \ldots$ so that*

$$\mathbb{P}_x\{x_0 = x\} = 1, \tag{2.4}$$

$$\mathbb{P}_x\{x_n = y\} = \mu_n(y), \tag{2.5}$$

$$\mu_{n+1} = \mu_n P. \tag{2.6}$$

Exercise 2.5 *Let $\xi_0, \xi_1, \ldots$ be a sequence of i.i.d. random variables. Consider the general recursion*

$$x_{n+1} = f\left(x_n, \xi_n\right) \ , \quad x_0 = x. \tag{2.7}$$

Show that $x_0, x_1, \ldots$ is a Markov process. Under what conditions is the state space equal $\mathbb{N}$? In this case write an expression for p_{ij}.

Exercise 2.6 *This complements the previous exercise. Let $y_0, y_1, \ldots$ be a homogeneous Markov process with state space $\mathbb{N}$ and transition probabilities p_{ij}. Let $\xi_0, \xi_1, \ldots$ be a sequence of Uniform [0,1], i.i.d. random variables. Show that it is possible to construct a function f so that the Markov chain* *given by*

$$x_{n+1} = f(x_n, \xi_n) \ , \quad x_0 = y_0, \tag{2.8}$$

has the same distribution as the process y.

2.2 Controlled Markov chains

The latest text on the subject is [10].

A controlled Markov chain differs from a Markov chain by the inclusion of a "control" or "action" variable a. This variable may change the transition probabilities. We restrict for the moment to the homogeneous case. The process is now defined (recursively as before) through the following.

Definition 2.7 *A* Controlled Markov chain *is a process where, given $a_n = a$, we have*

$$p_{ij}(a) \stackrel{def}{=} \mathbb{P}\{x_{n+1} = j \mid x_n = i, \ a_n = a\} \tag{2.9}$$

$$= \mathbb{P}\{x_{n+1} \in A \mid x_n = i, \ a_n = a, \ x_k = b_k, \ a_k = \alpha_k, \ k < n\} \ . \tag{2.10}$$

Example 2.8 *Suppose I have two coins in my pocket: a normal (fair) coin, and a coin that gives probability $3/4$ to heads. I throw the fair coin. If I get tails, I throw the same coin 2 more times. If I get heads, I throw the biased coin two more times. Is the sequence of coin tosees (get $+1$ for heads, -1 for tails) a Markov chain? Is it a controlled chain?*

The available actions are defined through an action space. There may be additional restrictions on available actions, depending on the state x. In choosing an action a at time n (while in state x), what information do we have? and what information are we allowed to use? Obviously, we should not use information about the future (which we usually do not have). But it is reasonable to use the information we do have, namely everything that happened since we started the process, at time 0.

Definition 2.9 *The available actions are defined through the action space $\mathcal{A}$. Actions available when at state x are from the action space $\mathcal{A}(x)$. The history of the process at time n is the sequence*

$$h_n \stackrel{def}{=} \{x_0, a_0, x_1, a_1, \ldots, x_{n-1}, a_{n-1}, x_n\}. \tag{2.11}$$

A decision rule is an assignment of an action to each history. A policy is a collection of decision rules, one for each time n.
For example, a function $g(x)$ with values in $\mathcal{A}(x)$ defines a decision rule. A policy π that, at each n, uses some decision rule (which may depend on n!) is called a Markov policy. If the same decision rule is used for each n, than it is a Stationary policy.
A randomized policy defines a probability for using each action, rather than specifying a single action. Thus, in general,

$$\pi = \{\pi_0, \pi_1, \ldots\} \tag{2.12}$$

where

$$\pi_n = \pi_n(a|h_n) \tag{2.13}$$

is the probability to use action a at time n, when the observed history is h_n.

Denote by Π the collection of all admissible policies. It can be shown that given an initial state $x_0 = x$ and a policy π, there exists a Probability $\mathbb{P}_x^\pi$ and of a controlled stochastic process that correspond to x, π.

So, our process behaves as follows. At time n we have observed the history h_n. Based on our chosen policy, we have a probability $\pi_n = \pi_n(a|h_n)$ that our choice of action will be a. If indeed the action turns our to be a, then the next state will be j with probability $p_{x_n j}(a)$.

Exercise 2.10 *Show that a Markov policy makes the process into a (possibly non homogeneous) Markov process, and a Stationary policy makes the process into an homogeneous Markov process.*

Definition 2.11 *Extending the notation p_{ij}, $p_{ij}(a)$ we define for a deterministic (non randomized) decision rule g*

$$p_{ij}(g) \stackrel{def}{=} \mathbb{P}_x(x_{n+1} = j | x_n = i, g(x_n)) = p_{x_n j}(g(x_n)). \tag{2.14}$$

For a randomized decision rule g the notation $p_{ij}(g)$ means

$$p_{ij}(g) \stackrel{def}{=} \mathbb{P}_x(x_{n+1} = j | x_n = i, g(x_n)) = \sum_{a \in \mathcal{A}(x_n)} p_{x_n j}(a) g(a|x_n). \tag{2.15}$$

Exercise 2.12 *Suppose $p_{ij}(g; n)$ depends explicitly on time. Embed this chain in a new homogeneous Controlled chain. Give the new state and action spaces as well as transitions explicitely.*

Exercise 2.13 *Extend exercises 2.5–2.6 to the controlled case, where*

$$x_{n+1} = f\left(x_n, a_n, \xi_n\right) \ . \tag{2.16}$$

Suppose we fix an initial state x and a policy π. We then have a probability $\mathbb{P}_x^\pi$ and expectation (operator) $\mathbb{E}_x^\pi$.

Example 2.14 *Let μ be a row vector and think of a function f as a column vector (its easy if f is defined for a finite number of arguments). Give an interpretation of the following objects and write them in terms of $\mathbb{P}_x^a$, $\mathbb{E}_x^a$ and x_0: $P(a)$, $\mu P(a)$, μf, $P(a)f$, $\mu P(a)f$. Suppose we choose a feedback control, so that when in state x, we choose action $g(x)$. What is the form of the transition matrix? Suppose we use g_0 at time 0 and g_1 at time 1, so that $\pi = \{g_0, g_1\}$. Express $\mathbb{E}_x^\pi f(x_2)$ in terms of transition matrices.*

The theorem below is stated and proved for the case where $\mathcal{S}$ and $\mathcal{A}$ are both finite (just a bit more care would extend to the countable case). It is, however, far more general. See [8, Theorem 13.2].

Theorem 2.15 *Let π be any admissible policy and fix an initial state x. Define a randomized Markov (non stationary!) policy g through its decision rules*

$$g_n(i;a) \stackrel{def}{=} \frac{\mathbb{P}_x^\pi\left(x_n = i, a_n = a\right)}{\mathbb{P}_x^\pi\left(x_n = i\right)} \ . \tag{2.17}$$

Define the policy $g = \{g_0, g_1, \dots\}$. Then

$$\mathbb{P}_x^\pi\left(x_n = i, a_n = a\right) = \mathbb{P}_x^g\left(x_n = i, a_n = a\right) \qquad \textit{for all } n, i, a. \tag{2.18}$$

Proof. By induction. For $n = 0$, we only need to check for $i = x$, and then

$$\mathbb{P}_x^\pi(x_0 = x, a_0 = a) = \frac{\mathbb{P}_x^\pi(x_0 = x, a_0 = a)}{\mathbb{P}_x^\pi(x_0 = x)} \tag{2.19}$$

$$\stackrel{def}{=} g_0(x; a) \tag{2.20}$$

$$= \mathbb{P}_x^g(a_0 = a) \tag{2.21}$$

$$= \frac{\mathbb{P}_x^g(x_0 = x, a_0 = a)}{\mathbb{P}_x^g(x_0 = x)} \tag{2.22}$$

$$= \mathbb{P}_x^g(x_0 = x, a_0 = a) \ . \tag{2.23}$$

Now suppose the theorem is true up to n. Then

$$\mathbb{P}_x^g(x_{n+1} = j) = \sum_i \mathbb{P}_x^g(x_{n+1} = j, x_n = i) \tag{2.24}$$

$$= \sum_i \mathbb{P}_x^g(x_{n+1} = j \mid x_n = i)\, \mathbb{P}_x^g(x_n = i) \tag{2.25}$$

$$= \sum_i \sum_a p_{ij}(a) g_n(i; a)\, \mathbb{P}_x^g(x_n = i) \tag{2.26}$$

$$= \sum_i \sum_a p_{ij}(a) \frac{\mathbb{P}_x^\pi(x_n = i, a_n = a)}{\mathbb{P}_x^\pi(x_n = i)}\, \mathbb{P}_x^g(x_n = i) \tag{2.27}$$

$$= \sum_i \sum_a p_{ij}(a)\, \mathbb{P}_x^\pi(x_n = i, a_n = a) \tag{2.28}$$

$$= \mathbb{P}_x^\pi(x_{n+1} = j) \ . \tag{2.29}$$

The rest is by definition of g and of a controlled Markov process. ∎

3 Markov decision processes

To complete the picture we need a way to distinguish between good performance and bad performance of our system. The way this is done is, as usual, chosen for two reasons. First, it is quite general and flexible. Second, it facilitates analysis, leads to methods for computing performance and synthesizing good policies, and to various algorithms. We shall comment on other types of performace measures later on.

The starting point is a function which penalizes (or specifies a cost for) an action, depending on which action is taken, and at which state. This is called the immediate cost function, and we denote it by $c(x, a)$. From this we define a value $V(x; \pi)$ for each policy, via the immediate cost. The standard Markov decision problem is to maximize (or minimize) some V: different problems may differ in the structure of V. The standard structures of V are defined and investigated in subsequent chapters. They are all the expectation of a weighed sum of immediate cost functions, that is, for some sequence $\alpha_n,\ n \geq 0$ they have the form

$$V(x; \pi) = \mathbb{E}_x^\pi \sum_{n=0}^{\infty} \alpha_n c(x_n, a_n) \tag{3.1}$$

or a similar expression (with the usual stipulation that the expression is well defined!). The less standard Markov decision problems are, for example, constrained (through several cost functionals), multicriteria etc.

To make things concrete, let us describe a few criteria. If $\alpha_n = 0$ for all $n > n_0$ then we are interested only in a finite number of steps, and the problem and criterion are called "finite horizon."

If $\alpha_n = \beta^n$ for some $\beta < 1$ then this is the discounted cost problem. It has a ready economic interpretation, but more generally it emphasizes short-term costs, and discounts long term effects.

The third, standard type of criterion is mathematically more challanging: it is the "average cost," defined roughly as

$$V_{av}(x;\pi) \stackrel{def}{=} \lim_{T\to\infty} \frac{1}{T}\, \mathbb{E}_x^\pi \sum_{n=0}^{T} c(x_n, a_n). \tag{3.2}$$

This is not a precise definition, for technical reasons, and we shall return to the definition in later chapters. Note, however, that this criterion completely ignores short term costs, and values only the remote future.

Definition 3.1 *We denote the immediate cost function by $c(x,a)$, the value of a policy π starting at x by $V(x;\pi)$. When we wish to specify the type of criterion we shall use V_f to denote finite cost, V_β to denote discounted cost, and V_{av} to denote average cost.*

Definition 3.2 *The formal standard definition of MDP. A Markov decision process is a 5-tuple $\{\mathcal{S}, \mathcal{A}, P, c, V\}$ where*

- *$\mathcal{S}$ is the state space,*
- *$\mathcal{A}$ is the action space,*
- *$P(a) = \{p_{ij}(a)\}$ is the transition matrix, when action a is taken,*
- *$c = c(x,a)$ is the immediate cost when at state x using action a, and*

- $V = V(x;\pi)$ *is the value of the criterion when starting at* x *and using policy* π*.*

More complicated models allow for the avaliable actions to depend on the state as $\mathcal{A}(x)$. Allowing explicit time dependence is conceptually trivial although could be practically annoying. Slight complications arise when we wish to include events in our model, since often there are events (or pairs of events) that lead to no change in the state.

Corollary 3.3 *If* V *is a weighted sum of immediate cost functions, then Markov policies suffice.*

This follows immediately from Theorem 2.15. ∎

Note that, in the most abstract form, we can view our variable as a policy (the argument of V) or, more abstractly, as a probability measure in the space of measures which can be generated by policies.

We now turn to modelling.

Exercise 3.4 *Consider a buffer where, at each unit of time, one of the following happens: a new request for service arrives w.p.* p_r*, and if the buffer is not empty, a service is completed w.p.* p_s *where* $p_r + p_s < 1$*. Is there a state such that the processes is a Markov chain? Now suppose the available action is to accept an arrival or not. Give the parameters of the CMC. Write an immediate cost function that gives a fixed reward for accepting a request, and a fixed penalty for each request in the buffer for each unit of time.*

Exercise 3.5 *Repeat Exercise 3.4 in the case that, if an arrival occurs, it carries requests according to an integer valued random variable (with finite number of values): this is called batch arrivals. Now allow the control action to accept any subset of the arrival batch. Now allow batch departures.*

Exercise 3.6 *Let* $d_n; n \geq 0$ *be a sequence of iid, positive, integer valued random variables representing a demand process. Suppose the available action is to order any (positive) number into stock; but the order arrives at the next time unit. Write the CMC in the case that unfulfilled demand is lost, and in the case that it is "backlogged." Write an immediate cost which is negative and affine in the order size, gives a fixed reward for each fulfilled unit of demand, and a penalty for a parially completed request.*

Exercise 3.7 *Let* f *be an arbitrary function and let consider a CMC* x_n*. Fix* $\lambda > 0$ *and suppose we want to maximize*

$$\mathbb{P}_x^\pi \left(\max_{0 \leq n \leq N} f(x_n \geq \lambda \right). \tag{3.3}$$

Write this in a standard form. Hint: define a variable y_n *which keeps the last largest value of* $f(x_n)$*.*

Let $V^i(x;\pi),\ i = 1, 2, \ldots, K$ be some cost functions. The two main problems we shall consider below are

$$\begin{aligned}
&\text{Optimization:} && \\
&\text{Maximize} && V^1(x;\pi) \qquad (3.4)\\
&\text{Constrained Optimization:} && \\
&\text{Maximize} && V^1(x;\pi) \qquad (3.5)\\
&\text{Subject to} && V^i(x;\pi) \geq v_i,\ i = 2, \dots, K. \qquad (3.6)
\end{aligned}$$

and we shall denote the value of the problem by $V^1(x)$: that is, this is the maximal value obtained in (3.4), or the maximal value obtained in (3.5) under the constraints (3.6). A policy π^* satisfying the constraints in (3.6) is called feasible at x. A (feasible) policy π^* satisfying $V^1(x;\pi) = V^1(x)$ will be called optimal at x. If it is optimal for all x, it will be called optimal.

Note that, in general, this value depends on the initial state x:

Example 3.8 *Let $\mathcal{S} = \{0,1\}$ with a single action a. Let $p_{10}(a) = p_{00}(a) = 1$. Let $c(1,a) = 1$ and $c(0,a) = 0$. Consider the ocnstrained optimization problem where the immediate cost associated with the constraint satisfies $c_2(1,a) = 1$, $c_2(0,a) = 0$. Then for $v_2 = \alpha_0/2$ the optimization problem is not even feasible starting at state 1.*

4 The finite horizon MDP.

The finite-horizon problem is defined by setting

$$V_{fh}(x;\pi) = \mathbb{E}_x^\pi \left(\left[\sum_{n=0}^{N-1} c(x_n, a_n) \right] + c_0(x_N) \right) . \tag{4.1}$$

Definition 4.1 *The finite horizon MDP is to Maximize $V_{fh}(x;\pi)$ over all policies π.*

Example 4.2 *Cheapest path on a graph. Consider a graph with points which we denote by $i\alpha$. Here i denotes the distance to our destination: the "end" of the graph, and α takes the values U (up) or D (down). At each step we can choose the action U (up), in which case our immediate cost is $c(i\alpha, U)$ and we move to $(i-1)U$ with probability $p_{\alpha u}$, and to $i-1, D$ with probability $1 - p_{\alpha u}$. If on the other hand we choose action D (down), our immediate cost is $c(i\alpha, D)$ and we move to $(i-1)U$ with probability $p_{\alpha d}$, and to $i-1, D$ with probability $1 - p_{\alpha d}$.*

Example 4.3 ([11, I.2]) . *This is a gambling example: at any stage you can bet any amount up to your present holding. If you win, with probability p, then you double your bet. Otherwise you lose it. You are allowed n plays. The objective is to maximize the (expectation of the) logarithm of your final holdings.*

The natural state here is your current holding, which is a real nonnegative number. The action can be taken to be the fraction of your holdings that you bet: so, it is a number between 0 and 1. This is more general than our *formal model, but we shall apply the same tools (although we do not prove them valid!).*

Corollary 4.4 *For the finite horizon MDP (optimization and constrained optimization), Markov policies suffice.*

Once we have restricted to Markov policies, we can establish the principle of optimality.

Definition 4.5 *For a policy π we use the notation*

$$\pi = \{\pi_0^k, \pi_{k+1}^N\} \quad \textit{where} \quad \pi_0^k = \{\pi_0, \dots, \pi_k\} \quad \textit{and} \quad \pi_{k+1}^N = \{\pi_{k+1}, \dots, \pi_N\} \ . \tag{4.2}$$

We also recall that

$$\mathbb{P}_x(\cdot) = \mathbb{P}(\cdot \mid x_0 = x) \tag{4.3}$$

so that, due to assumed homogeneity we can write

$$\mathbb{P}\left(x_N = y \mid x_k = x\right) = \mathbb{P}_x\left(x_{N-k} = y\right) \ . \tag{4.4}$$

Note that the two sides have a different meaning, but the values agree.

Theorem 4.6 *Let π be an optimal Markov policy for a finite horizon MDP. Then, for each $k < N$, the policy π_{k+1}^N is optimal for the finite horizon MDP with objective*

$$V_{fh;k}(x;\pi) = \mathbb{E}_x^\pi\left(\left[\sum_{n=0}^{N-k-2} c(x_n, a_n)\right] + c_0(x_{N-k-1})\right) \tag{4.5}$$

for each x such that $\mathbb{P}_x^{\pi_0^k}(x_{k+1} = x) > 0$. Conversly, suppose π_{k+1}^N is optimal for the $N-k-1$-step finite horizon MDP with the above objective, for all initial states x. Then given any optimal policy σ, the policy $\{\sigma_0^k, \pi_{k+1}^N\}$ is optimal.

Note: we use the convention that $\sum_{i=0}^{-1} a_i = 0$. Note also that in order to calculate the distribution of x_{k+1}, we only need to know x and π_0^k.

Proof. By contradiction. Let σ be an optimal Markov policy for $V_{fh;k}$. Consider the policy

$$\tilde{\pi} = \{\pi_0^k, \sigma\} \tag{4.6}$$

and note that this is a Markov policy. Write

$$V_{fh}(x;\pi) = \mathbb{E}_x^\pi \left[\sum_{n=0}^{k} c(x_n, a_n) + \mathbb{E}_x^\pi \left(\left[\sum_{n=k+1}^{N-1} c(x_n, a_n) \right] + c_0(x_N) \,\middle|\, x_{k+1} \right) \right] . \tag{4.7}$$

Since by assumption σ is optimal and π_{k+1}^N is not, we have for all x, (in obvious notation)

$$\left(\mathbb{E}_x^\sigma - \mathbb{E}_x^{\pi_{k+1}^N} \right) \left(\left[\sum_{n=0}^{N-k-2} c(x_n, a_n) \right] + c_0(x_{N-k-1}) \right) > 0 \tag{4.8}$$

where we have shifted the indices as in definition 4.5. But this is exactly the second term in our evaluation of V_{fh} above. Therefore, $\tilde{\pi}$ performs strictly better than π, a contradiction. The proof of the converse is left as an exercise.

■

Definition 4.7 *The optimality equation for the finite horizon problem is*

$$V_n(x) = \max_a \left(c(x,a) + \mathbb{E}_x^a V_{n-1}(x_1) \right) \ ; \quad V_0(x) = c_0(x) \ . \qquad (4.9)$$

Theorem 4.8 *Consider a finite MDP with finite horizon. Then there exists an optimal policy which is Markov and deterministic. The value function can be computed by backward induction using the optimality equation, any Markov policy which satisfies the optimality equation is optimal, and any optimal Markov policy satisfies the equation.*

Proof. Note that the index n in the optimality equation counts steps to go, and not running time index.

By the principle of optimality, we should do the last step in an optimal way; this is the case $n = 1$. This shows that we can choose a Markov decision rule at time $N - 1$, so that the action depends only on the state x at that time, and is not randomized. Moreover, this decision rule is optimal for all x. Now apply the principle of optimality twice: the optimal policy must do the last two steps in an optimal way, and our just-computed last step can be used as the last part of the two-step problem. Thus, again from the optimality equation, the last two steps use a non-randomized Markov policy. Continue $N - 1$ times. ∎

Note that there may be more than one optimal policy, or more specifically, at any given time and state, more than one action may be optimal. According to the theorem, all possible combinations of actions than satisfy the optimality equation will result in optimal policy. This includes combinations created by randomization. Randomization may not be necessary, but it can be used.

Going back to example 4.2, let us use the optimality equation in order to compute the optimal policy and cost. The final cost is, by definition, 0. With one step to go we obtain

$$V_1(1U) = \max_a \{c(1U, U) + p_{uu}V_0(0U) + (1 - p_{uu})V_0(0D), \tag{4.10}$$

$$c(1U, D) + p_{ud}V_0(0U) + (1 - p_{ud})V_0(0D)\} \tag{4.11}$$

$$= \max_a \{c(1U, U), c(1U, D)\} \tag{4.12}$$

$$V_1(1D) = \max_a \{c(1D, U) + p_{du}V_0(0U) + (1 - p_{du})V_0(0D), \tag{4.13}$$

$$c(1D, D) + p_{dd}V_0(0U) + (1 - p_{dd})V_0(0D)\} \tag{4.14}$$

$$= \max_a \{c(1D, U), c(1D, D)\} \ . \tag{4.15}$$

This equation gives both the value, as well as the optimal policy. The action to take is the one achieving the maximum, and we obtain an action which depends on the state.

The next step is identical, except that the probabilistic terms do not disappear, since the "remaining steps" are no longer of zero cost:

$$V_2(2U) = \max_a \{c(2U, U) + p_{uu}V_1(1U) + (1 - p_{uu})V_1(1D), \tag{4.16}$$

$$c(2U, D) + p_{ud}V_1(1U) + (1 - p_{ud})V_1(1D)\} \tag{4.17}$$

$$V_2(2D) = \max_a \{c(2D, U) + p_{du}V_1(1U) + (1 - p_{du})V_1(1D), \tag{4.18}$$

$$c(2D, D) + p_{dd}V_1(1U) + (1 - p_{dd})V_1(1D)\} \ . \tag{4.19}$$

Note that the resulting policy is Markov, deterministic, and time dependent. We need to convert from "steps to go" to the real time index. The complexity of this computation is linear in the number of stages: at each step we have 8 multiplications, 12 additions and two max operations, each between two numbers.

Finally, note that making the transition probabilities state-dependent, adding the possibility of staying where we are, or moving from iU to iD etc. would not complicate things in principle—only the notation will become more complex.

Example 4.3 can be solved exactly, using the backward induction algorithm. The condition at the last step is, by definition

$$V_0(x) = \log x \,. \tag{4.20}$$

The optimality equation reads

$$V_n(x) = \max_{0\le\alpha\le 1} \left[pV_{n-1}(x+\alpha x) + (1-p)V_{n-1}(x-\alpha x) \right]. \tag{4.21}$$

Assume $p > 0.5$. For the first step we obtain

$$\begin{aligned} V_1(x) &= \max_{0\le\alpha\le 1} \left[p\log(x+\alpha x) + (1-p)\log(x-\alpha x) \right] && (4.22) \\ &= \max_{0\le\alpha\le 1} \left[p\log(1+\alpha) + (1-p)\log(1-\alpha) + \log x \right]. && (4.23) \end{aligned}$$

The maximum is attained at $\alpha = 2p-1 > 0$, and so

$$V_1(x) = C + \log x \,, \quad C = \log 2 + p\log p + (1-p)\log(1-p). \tag{4.24}$$

But then, the optimality equation with $n = 2$ yields

$$\begin{aligned} V_2(x) &= \max_{0\le\alpha\le 1} \left[p\log(x+\alpha x) + (1-p)\log(x-\alpha x) + C \right] && (4.25) \\ &= 2C + \log x \,. && (4.26) \end{aligned}$$

Repeating the calculation we conclude that

$$V_n(x) = nC + \log x\,. \tag{4.27}$$

Moreover, the optimal action is to be, at each stage, a fraction $2p - 1$.

Example 4.9 *One of the most famous problems in optimization (and in computer science) is the travelling salesman problem. In this problem a travelling salesman is required to travel between N given cities. There is a cost for each possible leg of the trip, and all legs are possible, that is, for any two cities, it is possible to obtain a flight between these two. The objective is to visit each city, without returning to any city more than once, and all that at minimal cost. So, our fist mission is to model this problem as a finite MDP A more complicated version of this problem allows for random routing: choosing to go from city a to b may, with some probability, bring us actually to city c.*

References

[1] E. Altman *Constrained Markov decision processes* To appear, 1998.

New book, advanced.

[2] D. P. Bertsekas *Dynamic programming and stochastic control* Academic Press 1976

Engineering approach, basics.

[3] D. P. Bertsekas and S. E. Shreve *Stochastic Optimal Control* Academic Press New York 1978

Very mathematical.

[4] Cyrus Derman *Finite state Markovian decision processes* Academic Press New York 1970

Standard reference for the finite case.

[5] Eugene A. Feinberg and Adam Shwartz *Handbook of Markov Decision Proccesses: methods and applications* Kluwer, 2002.

Collection of advanced articles on the state of the art in Markov Decision processes, including applications to communication networks and other applications. Some articles are mathematically challenging, some are not.

[6] D. P. Heyman and M. J. Sobel *Stochastic Methods in Operations Research I: Stochastic Processes and Operating Characteristics* McGraw-Hill New York 1982

[7] D. P. Heyman and M. J. Sobel *Stochastic Methods in Operations Research II: Stochastic Optimization* McGraw-Hill New York 1984

A standard reference; covers much more.

[8] A. Hordijk Dynamic programming and Markov potential theory, Mathematical Centre Tracts 51, Mathematisch Centrum, Amsterdam, 1977.

Deep, theoretical classic, hard to read and somewhat old style.

[9] H.J. Kushner Introduction to stochastic control, Hold Reinhart Winston 1971.

Sample path approach, quite different from the rest and worthwhile. Watch out for the many typoes!

[10] M. L. Puterman *Markov decision processes* Elsevier Science Publishers 1990

Very detailed, very strong on computational techniques.

[11] S. M. Ross *Introduction to Stochastic Dynamic Programming* Academic Press 1984

Elementary, very good intuition, but be careful with proofs!

[12] H. C. Tijms *Stochastic Modelling and Analysis: a computational approach* John Wiley New York 1986

Good introduction to the modelling aspect.

[13] P. Whittle *Optimization over time; dynamic programming and stochastic control* Wiley 1983

Comprehensive, good intuition, sometimes sloppy proofs.