Chapter 2

Markov Decision Process Fundamentals

Any artisan who wishes to do his job well must first sharpen his tools[1].

Confucius, Chinese philosopher, 551-479 BCE.

To effectively apply Markov decision processes and reinforcement learning, one must establish a solid grounding in the fundamental concepts and tools introduced in this chapter.

The discussion begins with a description of the core components of a Markov decision process: decision epochs, states, actions, transition probabilities, and rewards. From these elements, decision rules, policies, stochastic processes and reward processes are derived. Although not formally part of the model definition, these constructs arise naturally and inherit structure from the model's core components. Building on them, the chapter then describes optimality criteria, which are the basis for comparing the quality of decisions and underpin the concepts of optimal policies, policy and optimal value functions, state-action value functions and the Bellman equation.

The book primarily focuses on discrete time models with finite discrete state and action spaces. It briefly discusses generalizations where appropriate such as the case of partially observable Markov decision processes (Chapter 8) in which continuous state spaces arise naturally.

2.1 Basic model components

A Markov decision process model describes the fundamental elements of a recurring decision problem in which today's decision impacts future options. Decisions take place over a *planning horizon*. *Decision epochs* represent specific points of time when a decision can be made (Figure 2.1). Time is divided into *periods* or *stages*; a period begins at one decision epoch and ends at the next decision epoch.

[1] Confucius [2003].

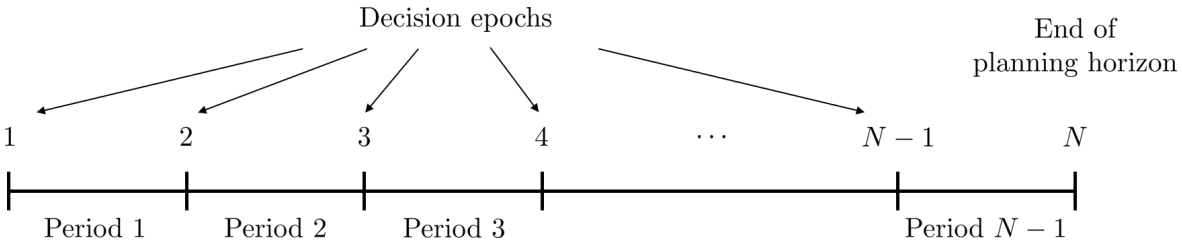

Figure 2.1: Decisions are made over a planning horizon divided into periods. The planning horizon may be finite (as shown) or infinite.

At a decision epoch, the decision maker[2] observes the state of the system, chooses an *action* and as a result of this choice, the system evolves to a new state according to a probability distribution that depends on the current state and the choice of action. After the system moves to the subsequent state, the decision maker receives a reward that depends on the starting state, action and possibly the subsequent state[3] These steps are repeated at each subsequent decision epoch. Figure 2.2 illustrates the timing of these events between two decision epochs. Such timelines are fundamental to developing a Markov decision process model. These concepts are formalized in the following subsections.

2.1.1 Planning horizons and decision epochs

The *planning horizon* is the time interval over which decisions are made. It may be *finite* or *infinite*. A *discrete time*[4] Markov decision process model is either:

- A *Finite Horizon Model*, in which the planning horizon is a bounded interval divided into $N-1$ periods, or
- An *Infinite Horizon Model*, in which the planning horizon is unbounded and divided into an infinite number of periods.

Each period begins at a *decision epoch*, a point in time at which the decision maker observes the system state and chooses an action. A period ends immediately before the subsequent decision epoch. For example, each day at midnight, a retailer's inventory management system observes the stock level of all products and decides how many additional units of each to order from suppliers. In this case, the period is a day, and the decision epoch corresponds to midnight of that day.

[2]This book primarily uses the expression *decision maker* to refer to a possibly animate entity who makes decisions. In the computer science literature, the decision maker is often referred to as an *agent* and in the engineering literature as a *controller* reflecting a reality in which an inanimate entity in some way chooses and executes decisions.

[3]The reinforcement learning literature uses the acronym SARSA to refer to the sequence: *state, action, reward, state, action*.

[4]This book will not describe continuous time models.

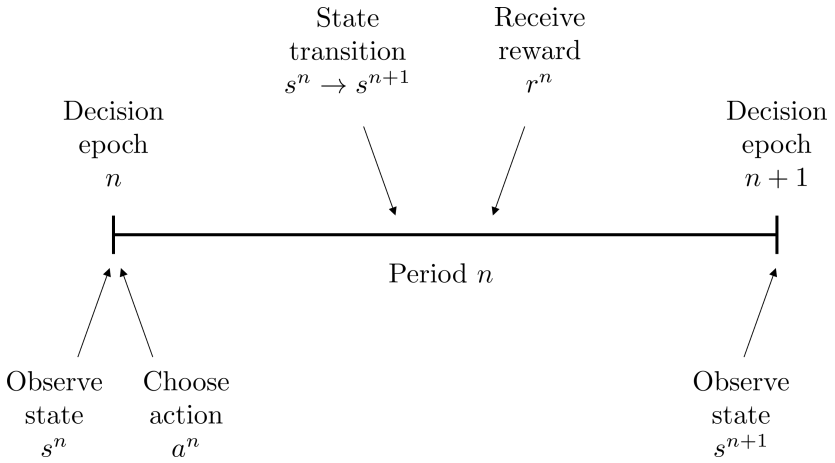

Figure 2.2: Timeline showing key events in period n. The decision maker observes state s^n at decision epoch n and chooses action a^n. The state transitions to state s^{n+1} according to the transition probabilities, the decision maker receives a reward r^n (that could depend on s^{n+1}), after which the decision maker is in the position to choose a new action at decision epoch $n+1$ after observing state s^{n+1}.

When the planning horizon is of finite length, it is divided it into $N-1$ periods with $N-1$ decision epochs. No decision is made at the end of the planning horizon, epoch N, which is referred to as the *terminal epoch*. It is included to evaluate the consequences of the decision at epoch $N-1$. Using $N-1$ also leads to cleaner formulas. Accordingly, the book uses the convention that a finite horizon problem is referred to as an $(N-1)$-period problem. In a limited number of finite horizon applications such as shortest path problems, decision epochs may index the order in which decisions are made, rather than pre-defined points in time. Chapter 4 focuses on finite horizon models, while Chapters 5 – 7 focus on infinite horizon models.

Throughout the book, "realizations", that is, states visited, actions implemented or rewards received following a decision or terminal epoch will be represented by **superscripts**. Moreover, transition probability functions and reward functions will be indexed by subscripts corresponding to the epoch to which they pertain.

In contrast, specific states and actions within the set of possible states and actions will be represented by **subscripts**[a].

[a]For example a^n will denote the action chosen at decision epoch n and a_m will denote the m-th action from a set of actions.

2.1.2 States

The state of the system contains all *relevant* information available to the decision maker at a decision epoch. This means that by knowing this information and choosing an action, the transition probabilities and rewards are fully specified. Let S denote the set of all states (the *state space*) and s denote a specific *state* in S. States are necessarily *exhaustive* and *mutually exclusive*. At each decision epoch, the system must occupy a unique state in S[5].

Elements of S may be scalar (e.g., the inventory level of a single product) or vector-valued (e.g., queue lengths of each priority class in a multi-class queuing system). They may be ordered (e.g., the number of people in a queue) or abstract (e.g., the configuration of a Tetris screen and the shape awaiting placement in the wall).

Although our primary focus in this book is on models with S discrete and finite, there are many possible generalizations, including choosing S to be countably infinite or a subset (either bounded or unbounded) of finite-dimensional Euclidean space. Also, specifically in network or decision analysis settings, S may vary with the decision epoch, in which case one may use an epoch-specific state space S_n at epoch n. An alternative is simply to define S as the union of S_n over all decision epochs n, though this approach may unnecessarily expand the state space at each decision epoch (because some states may not be reachable at certain epochs).

The book assumes that S is independent of the decision epoch.

2.1.3 Actions

Denote by A_s the set of all actions available to the decision maker in state $s \in S$. Refer to A_s as an *action set* and each $a \in A_s$ as an *action*. Actions are mutually exclusive; the decision maker cannot choose two actions at the same time, although, as seen later, the decision maker may specify a distribution over the set of actions.

Like the state space, each A_s may be a finite set, a countably infinite set, a subset of finite-dimensional Euclidean space, or an abstract set. The action set may be independent of s, as in an infinite-capacity queuing control model in which the action is to decide whether to admit or not admit a job. The book will focus on A_s being discrete and finite. Assuming the states and actions are ordered, $a_{s,j}$ represents the j-th action in the s-th state.

Note that in some states, A_s may contain a single element corresponding to "no action". Such states are *non-actionable*. These situations occur most often when the system reaches an absorbing state or set of states, Δ, from which it cannot exit. In that case, $A_\Delta = \{\text{Do nothing}\}$. This applies to episodic models (such as controlling a robot to move from an origin to a destination or in problems in which the decision maker can stop the system at any time).

[5]Unlike Schrödingers's cat, the system cannot occupy two different states at the same time.

2.1.4 Transition probabilities

Let $p_n(j|s,a)$ denote the probability that the system state becomes j at decision epoch $n+1$ when the decision maker chooses action a in state s at decision epoch n. Note that several random events may occur throughout the period between decision epochs; the transition probability does not explicitly represent each one, but instead represents the net change in state. The transition probability function[6], $p_n(j|s,a)$ has the following properties:

$$\begin{aligned} p_n(j|s,a) &\geq 0 \quad \text{for all } j \in S, a \in A_s, s \in S \\ \sum_{j \in S} p_n(j|s,a) &= 1 \quad \text{for all } a \in A_s, s \in S. \end{aligned}$$

Note that in a non-actionable state, δ, $p_n(\delta|\delta, \text{“Do nothing”}) = 1$.

When the transition probabilities do not vary over decision epochs, they are referred to as *stationary*. For infinite horizon models presented in this book, transition probabilities are assumed to be stationary, so that the subscript n is dropped.

2.1.5 Rewards

Consider two representations for a reward: $r_n(s,a,j)$ and $r_n(s,a)$. The quantity $r_n(s,a,j)$ denotes the reward received in period n when the decision maker chooses action a in state s and the system transitions to state j at decision epoch $n+1$. The latter quantity, $r_n(s,a)$, has a similar interpretation, but is independent of the subsequent state. The choice of reward function depends on the application – usually one of these two forms better describes a specific context. The examples in Chapter 3 will illustrate both.

Assume $r_n(s,a,j)$ and $r_n(s,a)$ are scalar and real-valued. Generalizations include vector-valued or abstract rewards. For example, as the result of an action in a fantasy game, the decision maker may receive a sword, a shield and a vial of magic potion. Instead of keeping track of this bundle of goods in the reward function, the decision maker might assign a numerical value to each item and be indifferent between collections of items with the same total value.

These quantities are *rewards* because the book formulates the problem of a decision maker seeking to *maximize* rewards. *Costs* can be represented as negative rewards to allow for a decision maker who seeks to minimize costs. Consequently, minimizing costs corresponds to maximizing rewards. To avoid awkward minus signs in expressions, on some occasions when minimizing costs, rewards will be replaced by $c_n(s,a)$ to represent the cost of being state s and choosing action a or $c_n(s,a,j)$ to include the possibility that the cost depends on the state at the next decision epoch j.

[6]When S represents an uncountable subset of finite-dimensional Euclidean space, the transition probability may be represented by a probability density function.

When using optimality criteria based on expected rewards[7], the quantity $r_n(s,a)$ may also represent the *expected* reward in period n where the expectation is taken over the possible states at decision epoch $n+1$:

$$r_n(s,a) = \sum_{j \in S} r_n(s,a,j) p_n(j|s,a). \tag{2.1}$$

For discrete time models, the model formulation does not account for how the reward is accumulated throughout a period between two decision epochs. For example, in an inventory control model with weekly decision epochs, the inventory levels may change during the week resulting in varying holding costs between decision epochs. A discrete time formulation accumulates all of these costs and summarizes them in the reward function for that one period.

In finite horizon models, a *terminal reward* or *scrap value* $r_N(s)$ is used to represent the consequence of ending the planning horizon in state s. In infinite horizon models, the terminal reward is omitted. Furthermore, the subscript n is deleted from the reward function in the infinite horizon setting, since the focus in that case is on *stationary* rewards.

> Rewards are an intrinsic part of the model that arise naturally from the application. However, in reinforcement learning models, the modeler may need to construct an *artificial* reward function that conforms with the objectives of the task.

2.1.6 Markov decision processes and decision trees

A decision tree provides a visual display of the basic model components. In Figure 2.3 square boxes represent states, arcs from boxes to circles represent possible actions in each state, arcs from circles to subsequent states denote transitions that occur according to a transition probability and result in a reward.

It should be evident from the decision tree representation that, in general, there is an exponential explosion in the number of possible trajectories through the tree as the length of the planning horizon is increased.

2.2 Derived objects

A Markov decision process is fully specified[8] by the five model components described in the previous section:

[7] In some applications, particularly in economic contexts, a decision maker seeks to maximize expected *utility* or some other risk sensitive criterion.

[8] Contrary to formulations in some disciplines, the discount rate is not specified as a model component. It is more appropriately associated with the optimality criterion.

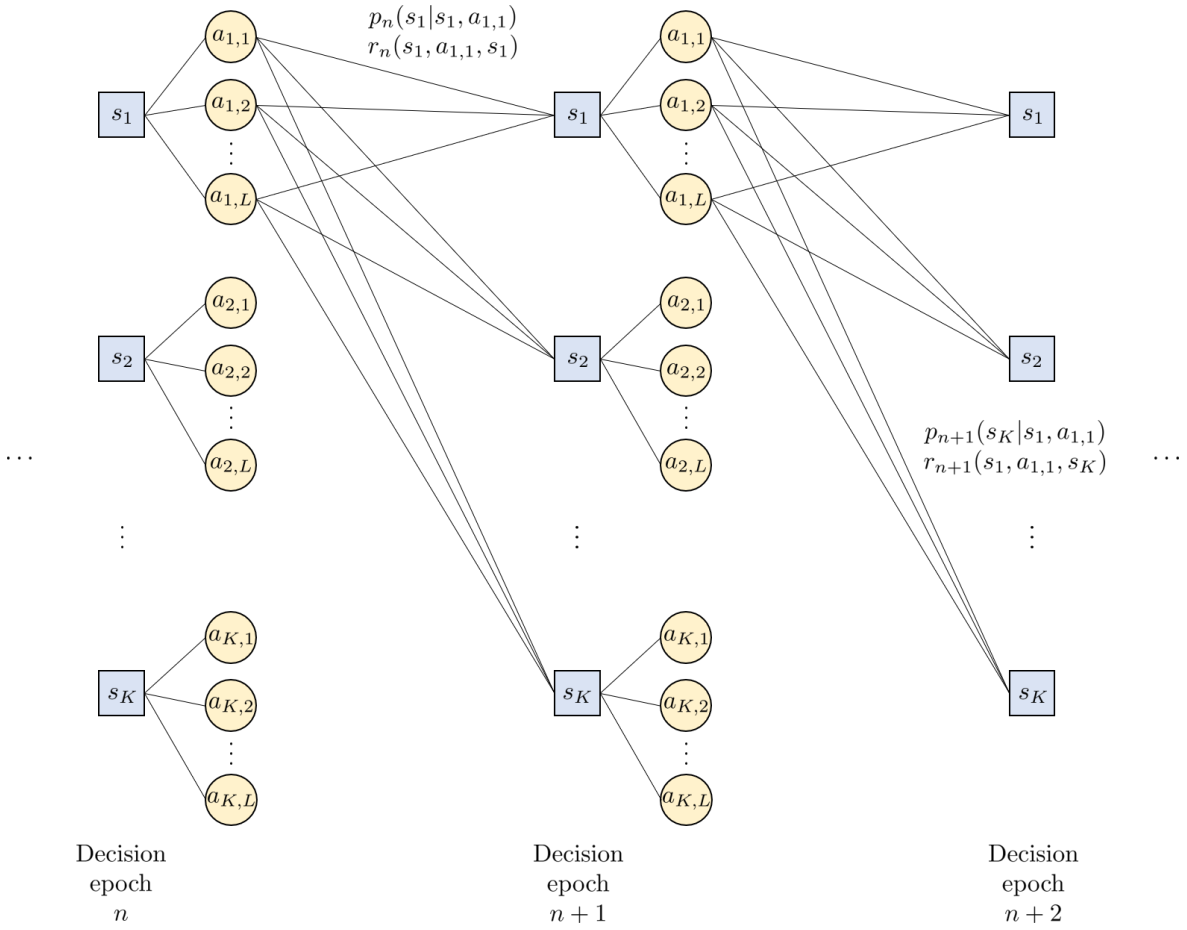

Figure 2.3: Graphical representation of a Markov decision process by a decision tree.

- the planning horizon N (which may be finite or infinite),
- the set of states S,
- the sets of actions A_s for each $s \in S$,
- the transition probabilities $p_n(j|s, a)$, and
- the rewards $r_n(s, a, j)$ or $r_n(s, a)$.

In this section, decision rules, policies, derived stochastic processes and reward processes are described. They are not part of the basic formulation, but are fundamental concepts derived from the basic model components.

2.2.1 Decision rules

A *decision rule* describes both the information and mechanism a decision maker uses to select an action in a given state at a single, specific decision epoch. Decision rules can be classified on the basis of two independent dimensions:

Information: Markovian or history-dependent

Mechanism: Deterministic or randomized

Information

The Markovian vs. history-dependent dichotomy describes the *information* used by the decision maker when choosing an action. A *Markovian* decision rule uses only the state at the current decision epoch to select actions, while a *history-dependent* decision rule considers some or all of the previous states and actions up to and including the current state when choosing an action. That is, at decision epoch n, a Markovian decision rule is a function of s^n and a history-dependent decision rule is a function of a *trajectory* $(s^1, a^1, s^2, \ldots, a^{n-1}, s^n)$[9]. Thus, a Markovian decision rule is a special case of a history-dependent decision rule in which the history is summarized in a single state. Write

$$H_n = \{h^n = (s^1, a^1, s^2, \ldots, a^{n-1}, s^n) \mid s^1 \in S, a^1 \in A_{s^1}, s^2 \in S, \ldots, a^{n-1} \in A_{s^{n-1}}, s^n \in S\} \tag{2.2}$$

as the set of all histories, h^n, leading up to decision epoch n. Note that $H_1 = S$. While the past sequence of rewards could also be explicitly included in the history, its inclusion is redundant in the Markov decision process model, since the history of states and actions alone allows one to reconstruct the past rewards through the reward functions[10]. Note that $h^1 = s^1$ and for $n = 2, 3, \ldots, N$,

$$h^n = (h^{n-1}, a^{n-1}, s^n). \tag{2.3}$$

This recursion is used explicitly in Section 4.1.3.

Mechanism

The deterministic vs. randomized dichotomy describes the *mechanism* used to select an action at a given decision epoch. A *deterministic* decision rule selects an action with certainty, while a *randomized* decision rule selects an action according to a specified probability distribution. A deterministic decision rule is a special case of a randomized decision rule corresponding to a degenerate probability distribution[11].

[9] Recall the convention that superscripts refer to the state and/or action chosen at decision epoch n. Also, history-dependent decision rules need not consider the entire history, but a subset of past actions and states.

[10] In model-free reinforcement learning, a functional form for $r_n(s, a, j)$ is not be known or specified. In this case the history may be augmented to include the reward.

[11] A degenerate probability distribution places all of the probability on one action.

Types of decision rules

The two dimensions described above lead to four classes of decision rules.

- A *Markovian deterministic* (MD) decision rule, d_n, is a function from states to actions. More formally, $d_n(s^n) = a^n$ denotes the decision rule that at decision epoch n chooses action $a^n \in A_{s^n}$ when the system is in state $s^n \in S$.

- A *history-dependent deterministic* (HD) decision rule, d_n, is a function from the set of histories to actions. More formally, $d_n(h^n) = a^n$ denotes the decision rule that chooses action $a^n \in A_{s^n}$ when the system history is $h^n \in H_n$ and s^n is the state at decision epoch n. The subtlety here is that although the decision rule's action choice may vary with history, it can only choose actions from the set A_{s^n} at epoch n, which depends only on the state at decision epoch n. This means that given two different histories h^n and $\hat{h}^n$ that both arrive at state s^n, $d_n(h^n)$ and $d_n(\hat{h}^n)$ may choose different actions, but each action will be chosen from the same action set A_{s^n}.

- A *Markovian randomized* (MR) decision rule, d_n, is a mapping from the set of states to the set of probability distributions over the action set. More formally, $w^n_{d_n}(a^n|s^n)$ denotes the probability that decision rule d_n chooses action a^n in state s^n.

- A *history-dependent randomized* (HR) decision rule, d_n, is a mapping from the set of histories to the set of probability distributions over the action set. More formally, $w^n_{d_n}(a^n|h^n)$ denotes the probability that decision rule d_n chooses action a^n given history h^n.

These classes of decision rules are denoted as D^{MD}, D^{HD}, D^{MR} and D^{HR}, respectively.

It is important to note that because of the ability to construct history-dependent decision rules, the Markov decision process formulation gives rise to a system that might not evolve in a Markovian fashion. The expression "Markov decision process" refers to the fact that rewards and transition probabilities depend on the past only through the state and action at the present decision epoch. It does not, however, mean that every stochastic process generated by this model is a Markov chain. See Section 2.2.3 below for more details on this point.

Subsequent chapters will establish the optimality of Markovian deterministic decision rules so it will be unnecessary to consider randomized decision rules when seeking optimal policies. However, randomized decision rules are fundamental in linear programming formulations of the infinite-horizon Markov decision process model (Chapters 5-7), exploration-based simulation algorithms (Chapter 10) and policy-based reinforcement learning algorithms (Chapter 11).

2.2.2 Policies

A *policy* π, sometimes referred to as a *contingency plan* or *strategy*, is a sequence of decision rules, one for each decision epoch. In finite horizon models, write $\pi = (d_1, d_2, \ldots, d_{N-1})$. In infinite horizon models, $\pi = (d_1, d_2, \ldots)$. In the classical Markov decision process setting, once an optimality criterion has been specified, the decision maker's goal is to find an optimal policy – one that maximizes or minimizes the designated criterion.

The four classes of decision rules defined in Section 2.2.1 result in four classes of policies, Π^{MD}, Π^{HD}, Π^{MR} and Π^{HR}. In addition, stationary deterministic (SD) and stationary randomized (SR) policies, denoted by the classes Π^{SD} and Π^{SR}, respectively, refer to sets of *stationary policies*, that is policies[12] that choose the same decision rule at every decision epoch. In infinite horizon models, stationary policies that use the decision rule d at every decision epoch will be denoted by d^∞, that is,

$$d^\infty := (d, d, \ldots). \tag{2.4}$$

Some comments about policies follow:

- The class Π^{HR} denotes the most general class of policies; all policies considered in the book are in this class. This level of generality is required to define optimality, but often not required to find optimal policies. For example, in finite horizon models it will be sufficient to restrict one's search for an optimal policy to the class of Markovian deterministic policies (see Section 2.4).

- Policies in Π^{HR} generally cannot be implemented in long finite-horizon or infinite horizon models because storage requirements increase exponentially with respect to the planning horizon length. Fortunately, in finite horizon models, there exist optimal policies in this class that are Markovian and deterministic, and in infinite horizon models, there exist optimal policies that are stationary and deterministic.

- Stationary policies are most relevant to infinite horizon models. Proofs and methods in Chapters 5 – 6 will use the result that there exist stationary deterministic policies that are optimal in the class of history-dependent randomized policies under several different optimality criteria.

- In finite horizon models, a Markovian deterministic policy may be characterized by a *lookup table*. A lookup table is an array in which rows correspond to states, columns correspond to decision epochs and an entry indicates which action the policy selects in that state at that decision epoch. Although a useful conceptual framework, a lookup table may be an impractical method of encoding a policy in models with a large number of states. Chapters 9 and 11 replace lookup tables

[12] In most cases, a stationary policy is necessarily Markovian.

by function approximation. Models in which lookup tables apply are referred to as *tabular models* in the reinforcement learning literature.

- Figure 2.4 summarizes the relationship between policy classes[13].

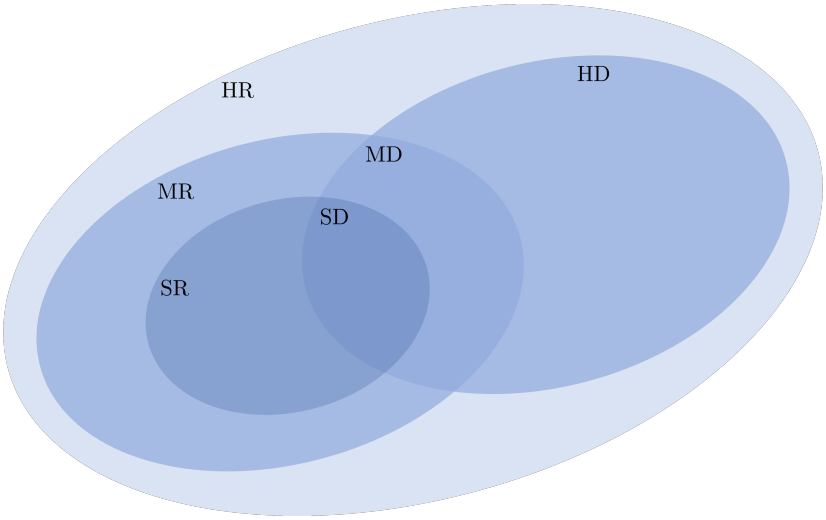

Figure 2.4: Relationships between policy classes. Labels near the edge of an oval refer to the entire oval, whereas labels inside the intersection of two ovals denote the entire intersection between them. For example, SD is the intersection of SR and HD. MD is the intersection of MR and HD, which includes SD.

2.2.3 Derived stochastic processes

Once a policy is chosen and a probability distribution over the starting state of the process is specified, the probabilistic evolution of a Markov decision process is completely determined. Let the random variable X_1 denote the initial state of the Markov decision process and let $\rho(s) := P[X_1 = s]$ denote the initial state probability distribution, which by assumption is independent of policy choice. When, as in most applications, the system starts in a specific state at decision epoch 1, say s^1, write $\rho(s) = 1$ for $s = s^1$ and $\rho(s) = 0$ for $s \neq s^1$.

Let π denote a policy for either a finite or infinite horizon problem and let the random variable Y_1 denote the action chosen by d_1 at decision epoch 1. There are two potential sources of variability that affect Y_1:

[13] Although not represented in the figure, there may exist policies that are stationary and history-dependent (i.e., non-Markovian). An example is a policy that makes decisions based on the current state as well as the starting state. However, these examples represent "edge cases" and are uncommon. Thus, in this book, only stationary policies are considered within the Markovian class.

1. Variability in X_1, which occurs when X_1 is random, and

2. Variability in action choice in X_1, which occurs if d_1 is a randomized decision rule.

Once Y_1 is chosen, the next state, X_2, is random with a distribution determined by the transition probabilities. This is true even if X_1 and Y_1 are not random, that is when the system starts in a specific state s^1 and d_1 is deterministic. Consequently, the second action Y_2 will be random as well, and so on. Thus, a policy and initial state distribution generate the stochastic process

$$(X_1, Y_1, X_2, Y_2, \ldots, X_{N-1}, Y_{N-1}, X_N) \tag{2.5}$$

in a finite horizon model and

$$(X_1, Y_1, X_2, Y_2, \ldots) \tag{2.6}$$

in an infinite horizon model. In these expressions, X_n represents the state of the stochastic process and Y_n denotes the action chosen by policy π at decision epoch n.

Some comments about policies with respect to derived stochastic processes follow:

1. When $\pi \in \Pi^{\mathrm{MR}}$, the stochastic process is a discrete time Markov chain. See Exercise 3.

2. When π is history-dependent, the stochastic process need not be a Markov chain. That is, at each decision epoch, given a state and action, the transition probabilities may depend on all or some of the past. See Exercise 3.

3. A policy $\pi = (d_1, d_2, \ldots)$ induces a probability distribution p^π on realizations of the stochastic process $(X_1, Y_1, X_2, Y_2, \ldots)$. These realizations are referred to as *trajectories*. To determine this distribution requires enumerating realizations and multiplying the transition probabilities and action randomization distributions corresponding to each realization as follows:

   (a) For $\pi \in \Pi^{\mathrm{HR}}$,

   $$\begin{aligned} &p^\pi(s^1, a^1, s^2, a^2, s^3, \ldots) \\ &\quad = \rho(s^1) w^1_{d_1}(a^1|s^1) p_1(s^2|s^1, a^1) w^2_{d_2}(a^2|h^2) p_2(s^3|s^2, a^2) w^3_{d_3}(a^3|h^3) \ldots \end{aligned} \tag{2.7}$$

   where $h^1 = s^1$, $h^2 = (s^1, a^1, s^2)$, and so on. Recall that $w^n_{d_n}(a|h)$, defined in Section 2.2.1, denotes the probability that decision rule d_n chooses action a in state s at epoch n given history h. That is $w^n_{d_n} = P^\pi[Y_n = a]$ for $\pi = (d_1, d_2, \ldots)$.

   Note that the entire history first enters into this expression through the randomization distribution at decision epoch 2 and subsequently, but not

through the transition probabilities which only depend on the state and action. If a deterministic policy was used, the history would also only enter through the decision rule as in (2.7), but the decision rule would explicitly choose the action in each transition probability.

(b) For $\pi \in \Pi^{\text{MR}}$,

$$\begin{aligned} &p^\pi(s^1, a^1, s^2, a^2, s^3, \ldots) \\ &\quad = \rho(s^1) w^1_{d_1}(a^1|s^1) p_1(s^2|s^1, a^1) w^2_{d_2}(a^2|s^2) p_2(s^3|s^2, a^2) w^3_{d_3}(a^3|s^3) \ldots \end{aligned} \tag{2.8}$$

This expression is similar to (2.7), except that histories are replaced with states.

(c) For $\pi \in \Pi^{\text{MD}}$

$$p^\pi(s^1, a^1, s^2, a^2, s^3, \ldots) = \rho(s^1) p_1(s^2|s^1, a^1) p_2(s^3|s^2, a^2) \ldots \tag{2.9}$$

since $d_1(s^1) = a^1$, $d_2(s^2) = a^2$, and so on. In this case, transition probabilities need only be multiplied with each other (and the initial state distribution) to find the probability distribution of the stochastic process generated by π.

2.2.4 Reward processes

Section 2.2.3 showed that a policy π generates a stochastic process of states and actions represented by (2.5) in finite horizon models and (2.6) for infinite horizon models with distribution $p^\pi(\cdot)$ defined in (2.7) in the most general case. These processes generate corresponding stochastic processes of rewards

$$(r_1(X_1, Y_1, X_2), r_2(X_2, Y_2, X_3) \ldots, r_{N-1}(X_{N-1}, Y_{N-1}, X_N), r_N(X_N)) \tag{2.10}$$

in finite horizon models, and

$$(r_1(X_1, Y_1, X_2), r_2(X_2, Y_2, X_3), \ldots) \tag{2.11}$$

in infinite horizon models.

In greater generality, let these stochastic processes be represented by either

$$\mathcal{R}_N := (R_1, R_2, \ldots, R_N) \tag{2.12}$$

or

$$\mathcal{R}_\infty := (R_1, R_2, \ldots) \tag{2.13}$$

where the random variable R_n represents the reward received in period n.

Letting r^n denote a realization of R_n, the expressions

$$P[\mathcal{R}_N = (r^1, r^2, \ldots, r^n)] \quad \text{and} \quad P[\mathcal{R}_\infty = (r^1, r^2, \ldots)]$$

equal the probability of the given finite or infinite sequence of rewards. When reward processes arise in a Markov decision process, the policy-dependent probability $p^\pi(\cdot)$ replaces the generic probability $P[\cdot]$.

Simulation or observing repeated experiments in the real world will generate many different sequences of rewards. The probability distribution of $\mathcal{R}_N$ and $\mathcal{R}^\infty$ describes the likelihood of observing each sequence. In this book, there will be little occasion to explicitly compute such probability distributions, but they will provide the basis for interpreting several expressions. Most calculations will avoid this onerous task by evaluating distributions and expectations sequentially. The example in Section 4.1.2 shows how to derive the distribution of $\mathcal{R}_N$ in an Markov decision process by determining which states and actions map into the same reward sequence.

In the discussion of values in the next section, it is sufficient to work directly with the reward process described above. However, in the simulation methods in Chapters 10 and 11, realizations (trajectories) that include states, actions and rewards of the form[14]

$$(X_1, Y_1, R_1, X_2, Y_2, \ldots, R_N) \quad \text{and} \quad (X_1, Y_1, R_1, X_2, Y_2, \ldots)$$

will be considered. Thus a realization of the process will be written as either

$$(s^1, a^1, r^1, s^2, \ldots, r^N) \quad \text{or} \quad (s^1, a^1, r^1, s^2, \ldots).$$

Following convention, a stochastic process together with a reward function will be referred to as a *reward process*. When the stochastic process is a Markov chain, it is a *Markov reward process*. Markov reward processes most often arise as the sequence of rewards of a Markovian policy in a Markov decision process.

2.2.5 Assigning a value to a reward process

This section shows how reward processes lead to decision making criteria. As noted in Section 2.1.5, assume that reward functions $r_n(s, a, j)$ are real-valued so that each R_n, as defined implicitly through (2.12) or (2.13), is real-valued. Thus, the random reward sequence $\mathcal{R}_N = (R_1, R_2, \ldots, R_N)$ takes values in $\mathbb{R}^N$[15] and $\mathcal{R}_\infty = (R_1, R_2, \ldots)$ takes values in $\mathbb{R}^\infty$.

Expected utility

To assign a value to a sequence of rewards, a decision maker may use *expected utility*. Let $u(\cdot)$ denote a real-valued function on $\mathbb{R}^N$ or $\mathbb{R}^\infty$, and let $E[\cdot]$ denote expectation with respect to the probability distribution of $\mathcal{R}_N$ or $\mathcal{R}_\infty$. The expected utility of these reward sequences equals $E[u(R_1, R_2, \ldots, R_N)]$ and $E[u(R_1, R_2, \ldots)]$, respectively.

[14] Technically speaking when $r_n(\cdot, \cdot, \cdot)$ is a function of the current state, the current action and the next state, we need to observe X_{n+1} before we can evaluate the reward. So technically, at least, we should write the stochastic process $(X_1, Y_1, X_2, R_2, Y_2, \ldots, X_N, R_N)$.

[15] Denotes N-dimensional Euclidean space.

Often utility functions are additive, so

$$u(R_1, R_2, \ldots, R_N) = \sum_{n=1}^{N} u_n(R_n) \tag{2.14}$$

for N finite or infinite, where $u_n(\cdot)$ is a real-valued function for each n. When N is finite, the expected utility becomes

$$E[u(\mathcal{R}_N)] = E[u(R_1, R_2, \ldots, R_N)] = E\left[\sum_{n=1}^{N} u_n(R_n)\right]. \tag{2.15}$$

When N is infinite, define[16]

$$E[u(\mathcal{R}_\infty)] = E[u(R_1, R_2, \ldots)] := \lim_{N\to\infty} E\left[\sum_{n=1}^{N} u_n(R_n)\right]. \tag{2.16}$$

Interpreting utilities

Utilities $u_n(\cdot)$ encode a decision maker's attitude towards both risk (variability) and timing of rewards. The most common choice for $u_n(\cdot)$ is

$$u_n(r) = \lambda^{n-1} r$$

where $0 \leq \lambda \leq 1$ so that

$$E[u_n(R_n)] = \lambda^{n-1} E[u_n(R_n)].$$

Refer to the quantity λ as the *discount factor* or *discount rate*. When $\lambda = 1$ the decision maker is indifferent to the timing of rewards. When $\lambda < 1$ the decision maker prefers receiving rewards sooner than later. For example, if $\lambda = 0.95$, a decision maker would be indifferent between receiving one unit of utility next period and 0.95 units now.

When using $E[R_n]$ to make decisions, a decision maker is said to be *risk-neutral*. For example, a risk-neutral decision maker would be indifferent between the following two gambles since they have the same expected value:

- Gamble A: win \$100 with probability 1, or
- Gamble B: win \$0 with probability 0.5 and \$200 with probability 0.5.

A *risk-seeking* decision maker would prefer Gamble B and a *risk-averse* decision maker would prefer Gamble A. Convex increasing utility functions such as r^2, correspond

[16] In (2.16), it would be preferable to take the limit inside the expectation but that limit need not exist. An example below explores this point further

to risk-seeking decision making while concave increasing utility functions such as $\sqrt{r}$ correspond to risk-averse decision making[17].

As an example, consider coaching decisions in sport. If a team is behind near the end of the game, its coach may be risk-seeking in an attempt to catch up. Alternatively, if a team is ahead, its coach may choose to make more conservative decisions, based on a risk-averse utility function, in order to hold on to the lead.

The Markov decision process models considered in this book will focus on decision making by risk-neutral decision makers, so utility is replaced by rewards or discounted rewards directly. However, it is important to be aware that other utility functions apply, and optimal policies with respect to one utility function will not necessarily be optimal for a different utility function.

2.3 Optimality criteria: Transforming a Markov decision process into a Markov decision problem

Up to this point, the Markov decision process has been formulated without considering the decision maker's preferences for reward sequences generated by different policies. The following sections define and comment on the following three optimality criteria that are commonly used to evaluate and compare policies:

- expected total reward,
- expected discounted reward, and
- long-run average reward.

The expected total reward criterion applies to both finite and infinite horizon models, while the latter two are most often used in infinite horizon models. A policy is said to be *optimal* when it maximizes the appropriate criterion over the set of all history-dependent randomized policies. To be precise, the expression *Markov decision problem* will correspond to a Markov decision process *together* with an optimality criterion[18]. However, as with most of the published literature, this book will use the general phrase *Markov decision process* to refer to both cases, whether an optimality criterion is included or not.

2.3.1 Expected total reward

This section defines the expected total reward of a reward sequence and policy. Technical considerations lead us to distinguish finite horizon from infinite horizon models.

[17] Which would better reflect your attitude towards gambles? Why?

[18] We emphasize that a Markov decision process is defined independent of the optimality criterion.

Finite horizon

Define the *expected total reward* of a finite sequence of random rewards $(R_1, R_2, \ldots, R_N)$ by

$$E\left[\sum_{n=1}^{N} R_n\right] \tag{2.17}$$

where the expectation is over the distribution of reward sequences. Since many reward sequences might map into the same total reward, in theory, when R_n is discrete, this quantity can be computed by first enumerating all possible values for the sum and and then accumulating the probabilities of the trajectories with the same sum.

As noted in Section 2.2.3, a policy π generates a random sequence

$$(X_1, Y_1, X_2, Y_2, \ldots, X_{N-1}, Y_{N-1}, X_N)$$

of states and actions and consequently generates a random sequence of rewards by setting $R_n = r_n(X_n, Y_n, X_{n+1})$ or $R_n = r_n(X_n, Y_n)$ for $n = 1, 2, \ldots, N-1$ and $R_N = r_N(X_N)$.

Thus, the expected total reward of a policy $\pi \in \Pi^{\text{HR}}$ can be defined as follows.

Definition 2.1. For all $s \in S$, the *finite horizon expected total reward* of policy π is

$$v^\pi(s) := E^\pi\left[\sum_{n=1}^{N} R_n \,\middle|\, X_1 = s\right]. \tag{2.18}$$

In equation (2.18), the expectation is with respect to the probability distribution of random rewards that result from different realizations of the stochastic process generated by π with $X_1 = s$ (Section 2.2.4).

Refer to the expected total reward of policy π, $v^\pi(s)$, as its *value*. Implicit in this notation for finite horizon models is that $v^\pi(s)$ gives the value for a model with $N-1$ decision epochs and terminal epoch N[19]. The function $v^\pi(\cdot)$ will be called a *policy value function*. Note that $v^\pi(s)$ will most often be seen written as either

$$v^\pi(s) = E^\pi\left[\sum_{n=1}^{N-1} r_n(X_n, Y_n, X_{n+1}) + r_N(X_n) \,\middle|\, X_1 = s\right] \tag{2.19}$$

or

$$v^\pi(s) = E^\pi\left[\sum_{n=1}^{N-1} r_n(X_n, Y_n) + r_N(X_n) \,\middle|\, X_1 = s\right] \tag{2.20}$$

depending on the form of r_n. To simplify notation, $E_s^\pi[\cdot]$ will often be used instead of $E^\pi[\cdot|X_1 = s]$.

[19] Some authors use the notation $v_N^\pi(s)$ where N denotes the planning horizon.

Simulating the expected total reward

The following algorithm describes how to use simulation to estimate $v^\pi(s)$. It generates a single replicate for a single starting state s^1. Since the decision maker often needs $v^\pi(s)$ for all $s \in S$, this algorithm would be repeated for all states. *Monte Carlo estimates* (Chapter 10) are obtained averaging $v^\pi(s)$ over many replicates.

Algorithm 2.1. Simulating a policy value function for a deterministic policy based on a single replicate for a finite horizon model.

1. **Initialize:**
   (a) Specify $\pi = (d_1, d_2, \ldots, d_{N-1})$.
   (b) Specify $s^1 \in S$.
   (c) $n \leftarrow 1$ and $v \leftarrow 0$.
2. **Iterate:** While $n \leq N - 1$:
   (a) $a^n \leftarrow d(s^n)$.
   (b) Sample s^{n+1} from $p_n(\cdot|s^n, a^n)$.
   (c) $r^n \leftarrow r_n(s^n, a^n, s^{n+1})$.
   (d) $v \leftarrow v + r^n$.
   (e) $n \leftarrow n + 1$.
3. $v \leftarrow v + r_N(s^N)$
4. **Terminate:** Return v.

Some comments about simulating the expected total reward follow:

1. Although you might not intend to simulate this process, this algorithm describes how the process would evolve in an implementation.
2. The functions $p_n(\cdot|\cdot,\cdot)$ and $r_n(\cdot,\cdot,\cdot)$ are explicitly used in the above procedure. When this is done, the approach is said to be *model-based*. Alternatively if s^{n+1} in step 2(b) and r^n in step 2(c) were outputs of a simulation or real-time process, the approach is said to be *model-free*. This distinction will be most significant in Chapters 10 and 11.
3. If one wished to instead simulate a randomized policy, step 2(a) would be replaced by "Sample a^n from $w^n_{d_n}(\cdot|s^n)$".
4. Obvious modifications would be required to evaluate a history-dependent policy. This would be impractical if N is large and the decision rules depended on the whole past.

Chapter 4 will develop a straightforward approach to compute $v^\pi(s)$ numerically or analytically for any $s \in S$ and any π. However, in problems with large state spaces, simulation and approximation may be preferable.

Infinite horizon models

The expected total reward (value) of a policy π in an infinite horizon model can be defined as follows.

Definition 2.2. For all $s \in S$, the *infinite horizon expected total reward* of policy π is

$$v^\pi(s) := \lim_{N\to\infty} E_s^\pi \left[\sum_{n=1}^{N} R_n \right]. \tag{2.21}$$

Note that the expected total reward of a policy is defined as the limit of the expected value of its finite sums. This is because

$$\lim_{n\to\infty} \sum_{n=1}^{N} R_n$$

need not exist so that its expectation may be undefined. Equation (2.21) can be written in terms of the reward function directly as

$$v^\pi(s) = \lim_{N\to\infty} E_s^\pi \left[\sum_{n=1}^{N} r_n(X_n, Y_n, X_{n+1}) \right] \text{ or } v^\pi(s) = \lim_{N\to\infty} E_s^\pi \left[\sum_{n=1}^{N} r_n(X_n, Y_n) \right]. \tag{2.22}$$

In contrast to the finite horizon setting, it is now of concern whether the limits in (2.22) exist. Some possible limiting behaviors for this (or any) sequence include:

Convergence: when $E[R_n]$ decreases sufficiently quickly or becomes zero eventually,

Divergence: when $E[R_n]$ remains sufficiently positive or negative,

Oscillation: when $E[R_n]$ alternates between positive and negative values and does not die out.

When using the expected total reward criterion, the infinite horizon Markov decision process literature has focused on models in which the limit in (2.21) exists. At first glance, it may not be apparent how these expectations can be finite when all rewards are positive. The reason is that most examples of this kind are either *episodic models* or *optimal stopping problems*, namely, models that terminate in *reward-free* absorbing

states[20]. Such models include policies that ensure the expected time to reach one of the absorbing states is finite so that effectively they behave like finite horizon problems but with policy-specific variable horizons. Often the random horizon is referred to as the *effective horizon*. .

In such problems, the decision maker attempts to delay reaching an absorbing state as long as possible when rewards are mostly positive. When rewards are mostly negative, the decision maker attempts to reach an absorbing state as quickly as possible. These models include *transient* models and *stochastic shortest path* models. See Sections 3.5, 3.2, and 3.8 for concrete examples of such problems.

An illustrative example*

The following example justifies the above definition of the expected total reward in infinite horizon models.

Example 2.1. Consider the Markov reward process depicted in Figure 2.5. It contains three states $S = \{s_1, s_2, s_3\}$. Rewards and transition probabilities are represented in brackets below the arcs. The first number in the bracket represents the reward received if that transition takes place and the second number represents the probability of that transition. Thus, from state s_1 the system transitions to state s_2 or s_3 with probability 0.5 and generates a reward of 0. From state s_2 the Markov chain transitions to state s_3 with certainty and generates a reward of 1 and from state s_3 the system transitions to state s_2 with certainty and generates a reward of -1. No other transitions are possible.

Next consider the calculation of $v(s_1)$. There are two possible realizations of the Markov chain describing the state at epoch n, namely

$$(s_1, s_2, s_3, s_2, s_3, \ldots) \quad \text{or} \quad (s_1, s_3, s_2, s_3, s_2, \ldots).$$

Each occurs with probability 0.5. The corresponding Markov reward process generates sequences

$$\xi_1 := (0, -1, 1, -1, 1, \ldots) \quad \text{and} \quad \xi_2 := (0, 1, -1, 1, -1, \ldots).$$

each occurring with probability 0.5. Thus

$$P[\mathcal{R}_\infty = \xi_1 | X_1 = s_1] = P[\mathcal{R}_\infty = \xi_2 | X_1 = s_1] = 0.5.$$

The sequence of **total rewards** corresponding to each realization are

$$\sigma_1 := (0, -1, 0, -1, 0, \ldots) \quad \text{and} \quad \sigma_2 := (0, 1, 0, 1, 0, \ldots).$$

[20] Appendix B defines basic Markov chain concepts.

For finite N even, the total reward

$$P\left[\sum_{n=1}^{N} R_n = -1 \,\middle|\, X_1 = s_1\right] = P\left[\sum_{n=1}^{N} R_n = 1 \,\middle|\, X_1 = s_1\right] = 0.5,$$

while for N odd,

$$P\left[\sum_{n=1}^{N} R_n = 0 \,\middle|\, X_1 = s_1\right] = 1.$$

Next let $N \to \infty$. Since both possible total reward series oscillate, neither σ_1 or σ_2 has a limit. Hence the quantity $\lim_{N\to\infty} \sum_{n=1}^{N} R_n$ does not exist, so that the expression

$$E\left[\lim_{N\to\infty} \sum_{n=1}^{N} R_n \,\middle|\, X_1 = s_1\right]$$

is not well defined. However for any N,

$$E\left[\sum_{n=1}^{N} R_n \,\middle|\, X_1 = s_1\right] = 0,$$

so

$$\lim_{N\to\infty} E\left[\sum_{n=1}^{N} R_n \,\middle|\, X_1 = s_1\right] = 0$$

so that the limit in (2.21) exists.

This is the reason the expected total reward in an infinite horizon model is defined as the "limit of expected partial sums" as opposed to the "expectation of the limit of partial sums".

Note the limit can be passed inside the expectation by considering the *lim inf* and *lim sup* of the sequence of partial sums. These quantities always exist and are well defined. Thus, in this example

$$P\left[\limsup_{N\to\infty} \sum_{n=1}^{N} R_n = 0 \,\middle|\, X_1 = s_1\right] = P\left[\limsup_{N\to\infty} \sum_{n=1}^{N} R_n = 1 \,\middle|\, X_1 = s_1\right] = 0.5$$

so that

$$E\left[\limsup_{N\to\infty} \sum_{n=1}^{N} R_n \,\middle|\, X_1 = s_1\right] = 0.5.$$

Moreover

$$E\left[\liminf_{N\to\infty} \sum_{n=1}^{N} R_n \,\middle|\, X_1 = s_1\right] = -0.5.$$

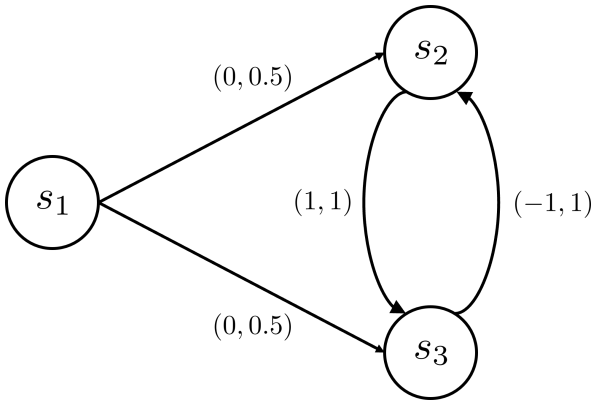

Figure 2.5: Markov reward process analyzed in Example 2.1. The labels (r, p) refer to the reward and transition probabilities, respectively.

Optimal policies

Definition 2.3. A policy π^* is *optimal* if

$$v^{\pi^*}(s) \geq v^{\pi}(s) \tag{2.23}$$

for all $s \in S$ and $\pi \in \Pi^{\text{HR}}$.

Chapter 4 analyzes finite horizon Markov decision process models and Chapter 6 considers infinite horizon models under this optimality criterion.

2.3.2 Expected discounted reward

In contrast to the expected total reward criterion, the expected discounted reward criterion accounts for the "time value of money" – that is, receiving a reward at some future epoch is worth less than receiving an identical reward now. This is the most widely used criterion in infinite horizon models.

Finite horizon models

Define the *expected discounted reward* of the reward sequence $\mathcal{R}_\infty = (R_1, R_2, \ldots)$ by

$$E\left[\sum_{n=1}^{N} \lambda^{n-1} R_n\right] \tag{2.24}$$

where the expectation is respect to the distribution of the sequence of rewards and the quantity $0 \leq \lambda < 1$ is the *discount factor*. In finite horizon models, discounting has no impact on theory or algorithms, but may affect a decision maker's preference for policies. For example, with a discount factor close to 0, a decision maker will prefer

actions that lead to larger immediate rewards, in contrast to a situation with a discount factor close to 1 when future rewards would be of greater significance.

Definition 2.4. For all $s \in S$, the *finite horizon expected discounted reward* of policy π is

$$v_\lambda^\pi(s) := E_s^\pi \left[\sum_{n=1}^{N} \lambda^{n-1} R_n \right]. \tag{2.25}$$

This definition is analogous to the expected total reward case. Above, the expectation is respect to the probability distribution of random rewards that result under different realizations of the stochastic process determined by π as described in Section 2.2.4.

Infinite horizon models

Discounting plays a fundamental role in the application and analysis of infinite horizon models and is the most studied optimality criterion.

Definition 2.5. For all $s \in S$, the *infinite horizon expected discounted reward* of policy π is

$$v_\lambda^\pi(s) := \lim_{N\to\infty} E_s^\pi \left[\sum_{n=1}^{N} \lambda^{n-1} R_n \right]. \tag{2.26}$$

Unlike in the expected total reward case, no special model structure is required for the infinite sum (2.26) to converge, only that $E[\lambda^{n-1} R_n]$ decays sufficiently quickly. To ensure this, it is assumed throughout the book that rewards are bounded. This means that there exists a finite M for which,

Bounded reward assumption:

$$|r(s,a,j)| \leq M \text{ for all } a \in A_s,\ s \in S \text{ and } j \in S. \tag{2.27}$$

Since the sum of a geometric series satisfies $\sum_{n=1}^{\infty} \lambda^{n-1} = 1/(1-\lambda)$, under the bounded reward assumption it follows that

$$\frac{-M}{1-\lambda} \leq v_\lambda^\pi(s) \leq \frac{M}{1-\lambda} \tag{2.28}$$

for all $\pi \in \Pi^{\text{HR}}$. Equation (2.28) shows the key role of the assumption that $\lambda < 1$. As noted previously when $\lambda = 1$ the series may not converge even when rewards are bounded.

Note that in contrast to the infinite horizon expected total reward, the quantity

$$\lim_{N\to\infty} \sum_{n=1}^{N} \lambda^{n-1} R_n$$

exists in a discounted model[21].

Hence it follows[22] that

$$\lim_{N\to\infty} E_s^\pi \left[\sum_{n=1}^{N} \lambda^{n-1} R_n \right] = E_s^\pi \left[\lim_{N\to\infty} \sum_{n=1}^{N} \lambda^{n-1} R_n \right] = E_s^\pi \left[\sum_{n=1}^{\infty} \lambda^{n-1} R_n \right]. \tag{2.29}$$

where the expression on the right is a shorthand for the middle expression.

[21] This is true because when rewards are bounded, the "tail" of the sum can be made arbitrarily small for every trajectory.

[22] The result follows from the bounded convergence theorem.

Example 2.1 (ctd.) Returning to Example 2.1, this example shows that when $\lambda < 1$, the limit of the sequence of partial sums for each of the discounted sequences of rewards exists and is finite.

For ξ_1, the sequence of its discounted partial sums σ_1' is

$$\sigma_1' = (0, -\lambda, -\lambda + \lambda^2, -\lambda + \lambda^2 - \lambda^3, \ldots)$$

and for ξ_2, the sequence of partial sums is

$$\sigma_1' = (0, \lambda, \lambda - \lambda^2, \lambda - \lambda^2 + \lambda^3, \ldots).$$

Thus the limit for ξ_1 is given by[a]

$$\sum_{n=1}^{\infty} \lambda^{n-1} R_n = -\frac{\lambda}{1+\lambda}.$$

Similarly, the limit of the sum of discounted rewards of sequence ξ_2 equals $\lambda/(1+\lambda)$. Thus when $\lambda < 1$

$$E\left[\lim_{N\to\infty} \sum_{n=1}^{N} \lambda^{n-1} R_n \,\middle|\, X_1 = s_1\right] = -0.5\frac{\lambda}{1+\lambda} + 0.5\frac{\lambda}{1+\lambda} = 0.$$

Since $\sigma_1' + \sigma_2' = (0, 0, 0, \ldots)$, it follows that

$$\lim_{N\to\infty} E\left[\sum_{n=1}^{N} \lambda^{n-1} R_n \,\middle|\, X_1 = s_1\right] = 0,$$

since the expectation equals 0 for any finite N. Hence (2.29) holds in this model.

[a] To see this, let $X = 1 - \lambda + \lambda^2 + \ldots$. Then $X = 1 - \lambda X$ so that $X = 1/(1+\lambda)$. Since the limit for $\xi_1 = X - 1$, the result follows.

An alternative and important interpretation of discounting

Discounting arises naturally in models where rewards are units of currency. Due to inflation, near-term rewards are preferable to future rewards. Discounting also arises in another, rather surprising way which extends its applicability.

Suppose the planning horizon length is not fixed but represented by a random variable, T, distributed according to a geometric distribution[23] with parameter λ, independent of $(R_1, R_2, \ldots)$ or $(X_1, Y_1, X_2, Y_2, \ldots)$.

[23] A random variable T follows a *geometric distribution* with parameter λ if $P[T = k] = (1-\lambda)\lambda^{k-1}$ for $k = 1, 2, \ldots$.

Let the expected total reward over a random horizon of length T be

$$E_T\left[E\left[\sum_{n=1}^{T} R_n\right]\right], \tag{2.30}$$

where E_T denotes the expectation with respect to T. Assuming R_n is bounded and $0 \leq \lambda < 1$,

$$\begin{aligned} E_T\left[E\left[\sum_{n=1}^{T} R_n\right]\right] &= E\left[\sum_{k=1}^{\infty}(1-\lambda)\lambda^{k-1}\sum_{n=1}^{k} R_n\right] \\ &= E\left[\sum_{n=1}^{\infty}\sum_{k=n}^{\infty}(1-\lambda)\lambda^{k-1}R_n\right] = E\left[\sum_{n=1}^{\infty}\lambda^{n-1}R_n\right] \end{aligned}$$

where the last inequality follows from the relationship $\sum_{k=n}^{\infty}\lambda^{k-1} = \lambda^{n-1}/(1-\lambda)$. The assumptions on R_n and λ allow interchange of the order of summation. Thus

$$E_T\left[E\left[\sum_{n=1}^{T} R_n\right]\right] = E\left[\sum_{n=1}^{\infty}\lambda^{n-1}R_n\right]. \tag{2.31}$$

When the rewards are generated by policy π starting from state s, this representation of the expected discounted reward follows:

Alternative representation for the expected discounted reward:

$$v_\lambda^\pi(s) = E_T\left[E_s^\pi\left[\sum_{n=1}^{T} R_n\right]\right]. \tag{2.32}$$

This result provides an alternative interpretation for discounting. Since the random variable T may be regarded as the time until the first "failure" in an independent, identically distributed series of Bernoulli trials with "success" probability λ, it means that when a decision maker uses expected total reward to evaluate policies in a system that terminates at the time of a random failure independent of the decision maker's policy it is equivalent to using the expected discounted reward. This is particularly appropriate in applications where the process can terminate suddenly for exogenous reasons. Examples include:

- **Animal behavior modeling**, when the animal might die of unanticipated causes (e.g., predation) independent of its decision making. See Section 3.5.
- **Clinical decision making**, in which a patient may die as a consequence of some event independent of the disease being treated. See Section 3.6.

Thus, discounting makes sense even in models that do not use economic values for rewards.

Note this provides an approach for estimating a discounted reward through simulation. The two steps are:

1. Generate T from a geometric distribution.
2. Apply Algorithm 2.1.

Optimal policies

Definition 2.6. A policy π^* is *discount optimal* if

$$v_\lambda^{\pi^*}(s) \geq v_\lambda^{\pi}(s) \tag{2.33}$$

for all $s \in S$ and $\pi \in \Pi^{\text{HR}}$.

Chapter 5 analyzes infinite horizon Markov decision process models under this optimality criterion.

2.3.3 Long-run average reward for an infinite horizon model

In contrast to discounting, which emphasizes short term behavior, the long-run average reward criterion focuses on steady state or limiting behavior of derived stochastic processes. For that reason, the long-run average reward is most appropriate for infinite horizon, non-terminating models with frequent decision epochs. Note that in finite horizon models, the average reward is equivalent to the expected total reward (why?).

As an example, consider a queuing system in which the decision maker inspects the system state very frequently, such as every second, and decides which service rate to use. Section 3.4 provides a rigorous formulation of such a problem. For such a problem:

- The **expected total reward** criterion would not be able to distinguish between policies because expected total costs or rewards would be unbounded.
- The **expected discounted reward** criterion would not be appropriate because the decision maker is interested in long-term system performance. Given the time scale of decision making, rewards at future decision epochs should **not** be less valuable than current rewards. However, if random failure of the system could occur discounting may be appropriate but it would require discount rates very close to 1.

The *average reward*[24] or more accurately the *long-run average reward* of the reward

[24] This quantity is interchangeably referred to as the *long-run average reward* or *gain*.

sequence $(R_1, R_2, \ldots)$ is given by

$$\lim_{N\to\infty} \frac{1}{N} E\left[\sum_{n=1}^{N} R_n\right].$$

When rewards are generated by a policy π, the average reward is defined as follows.

Definition 2.7. For all $s \in S$, the *average reward*, $g^\pi(s)$, of policy π is

$$g^\pi(s) := \lim_{N\to\infty} \frac{1}{N} E_s^\pi\left[\sum_{n=1}^{N} R_n\right]. \tag{2.34}$$

Note that the limit in (2.34) exists for all stationary policies when S is finite, as will be shown in Chapter 7. It need not exist when S is countable or policies are history-dependent. In such cases replace the limit above by the "lim inf" or the "lim sup" (see Chapter 7). The quantity g^π is sometimes referred to as the *gain* because the expected total reward in state s grows ("gains value") at rate $g^\pi(s)$ per epoch in the limit.

Similarly to the discounted model, when rewards are bounded,

$$\lim_{N\to\infty} \frac{1}{N} E_s^\pi\left[\sum_{n=1}^{N} R_n\right] = E_s^\pi\left[\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} R_n\right]. \tag{2.35}$$

Example 2.1 (ctd.) Returning to Example 2.1, this example shows that when rewards are bounded, the limit of the sequence of the average partial sums for each of the sequences of rewards exists and is finite.

For ξ_1, the sequence of average partial sums σ_1'' is

$$\sigma_1'' = \left(0, -\frac{1}{2}, 0, -\frac{1}{4}, \ldots\right)$$

so that its limit exists and equals 0. Similarly, the limit of average of partial sums for ξ_2 equals 0. Thus

$$E\left[\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} R_n \middle| X_1 = s_1\right] = 0.$$

since the expectation equals 0 for any finite N. Hence it is easy to see that (2.35) holds in this model.

Optimal policies

Definition 2.8. A policy π^* is *average reward optimal* or *gain optimal* if

$$g^{\pi^*}(s) \geq g^{\pi}(s) \tag{2.36}$$

for all $\pi \in \Pi^{\text{HR}}$ and $s \in S$.

Chapter 7 analyzes infinite horizon Markov decision process models under this optimality criterion.

2.4 The one-period problem: A fundamental building block

One-period models are the building blocks of Markov decision processes. Most of the algorithms presented later in the book are based on decomposing a multi-period model into a series of interlinked one-period models.

A one-period model begins with a decision epoch, evolves for one period of time and ends at the terminal non-decision epoch. Thus, it is the simplest representation of a finite horizon Markov decision process, namely the case of $N = 2$. In it, a policy is a single Markovian decision rule and the derived stochastic process is (X_1, Y_1, X_2). Note that in a one-period model, history-dependent policies and Markovian policies are equivalent because the only information available to the decision maker at the first (only) decision epoch is the state.

Figure 2.6 illustrates the timing of events and the nomenclature of the one-period problem.

Let S denote the state space and A_s denote the sets of actions defined for each $s \in S$. Assume that S and each A_s are discrete and finite. Let $r_1(s, a, j)$ denote the reward function in period 1, $p_1(j|s, a)$ denote the transition probability function in period 1, and $r_2(j)$ denote the terminal reward function. Analysis of this simple model provides intuition for more complex models.

Definition 2.9. Given a policy π in a one-period problem, the *policy value function* denoted $v^{\pi}(s)$, is the expected total reward when the process starts in state s at decision epoch 1. It is given by

$$v^{\pi}(s) := E^{\pi}[r_1(X_1, Y_1, X_2) + r_2(X_2)|X_1 = s], \tag{2.37}$$

where the expectation is with respect to the probability distribution of (X_1, Y_1, X_2) induced by policy π when the system starts in state $s \in S$.

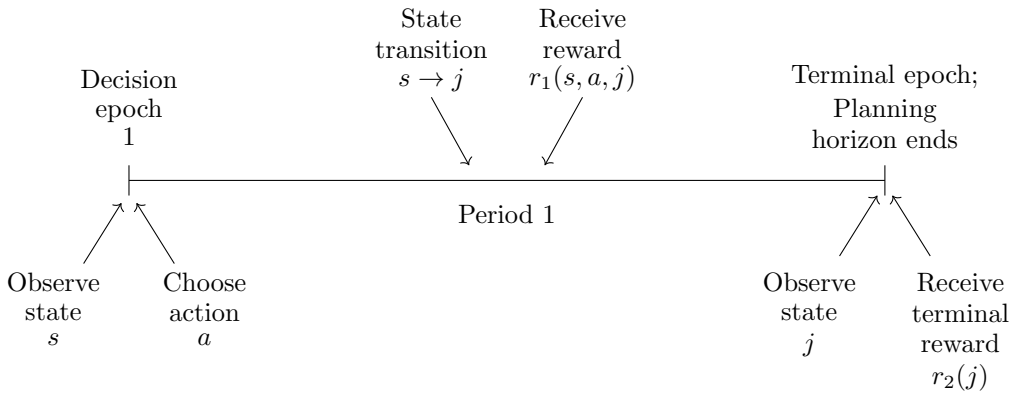

Figure 2.6: Illustration of timing of events and notation for the one-period problem. Action a may be chosen deterministically or probabilistically. Technically speaking, there is no need to observe state j since it is only used to evaluate the terminal reward $r_2(j)$.

Suppose $\pi = (d_1)$ is deterministic and $d_1(s) = a'$. Since s is fixed and $d_1(s) = a'$, the only random quantity in (2.37) is X_2. Under this assumption, (2.37) is equivalent to

$$v^\pi(s) = \sum_{j \in S} p_1(j|s, a')(r_1(s, a', j) + r_2(j)). \tag{2.38}$$

Therefore, computing $v^\pi(s)$ requires only $p_1(j|s, a') = P[X_2 = j|X_1 = s, Y_1 = a']$; the distribution of $r_1(X_1, Y_1, X_2) + r_2(X_2)$ does not need to be explicitly derived. Consequently, this avoids an enumeration of all of the latter's possible values as described in Section 2.2.4. Expressions of the form (2.38) appear frequently in the book, so it is important to understand why (2.38) is equivalent to (2.37).

When d_1 is randomized, Y_1 is also random with $P[Y_1 = a|X_1 = s] = w^1_{d_1}(a|s)$. In this case

$$v^\pi(s) = \sum_{a' \in A_s} w^1_{d_1}(a'|s) \sum_{j \in S} p_1(j|s, a')(r_1(s, a', j) + r_2(j)). \tag{2.39}$$

In both cases, $v^\pi(s)$ is referred to as the *value* of policy π.

2.4.1 Optimal value functions

The following definition is fundamental.

Definition 2.10. The *optimal value function* in a finite horizon Markov decision process, $v^*(s)$, satisfies

$$v^*(s) := \sup_{\pi \in \Pi^{\mathrm{HR}}} v^\pi(s) \tag{2.40}$$

for all $s \in S$.

Sometimes, the optimal value function is simply referred to as the *value function*. The adjective "optimal" distinguishes it from a policy value function.

In the one-period model, $\Pi^{\mathrm{HR}} = \Pi^{\mathrm{MR}}$. This identity holds because in a one-period model, the only decision is at the first decision epoch and the history at that decision epoch is the state s. Thus in the one-period model,

$$v^*(s) = \sup_{\pi \in \Pi^{\mathrm{MR}}} v^\pi(s), \tag{2.41}$$

where $v^\pi(s)$ is given by (2.39). As a result of (2.39) and (2.41)

$$v^*(s) = \sup_{w \in \mathscr{P}(A_s)} \left\{ \sum_{a \in A_s} w(a) \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\}, \tag{2.42}$$

where $\mathscr{P}(A_s)$ denotes the set of probability distributions on A_s. The reason that (2.42) holds is that there is a one-to-one relationship between the set $\mathscr{P}(A_s)$ and the set of Markovian randomized decision rules. Since $\mathscr{P}(A_s)$ is an uncountable infinite set[25], *sup*[26] instead of *max* appears in (2.42).

The following easily proved lemma is fundamental here and in latter chapters.

Lemma 2.1. Let U be an arbitrary finite set, let $f(\cdot)$ be a real-valued function on U, $\mathscr{P}(U)$ denote the set of probability distributions on U and $w \in \mathscr{P}(U)$. Then

$$\max_{u \in U} f(u) \geq \sum_{u \in U} w(u) f(u) \tag{2.43}$$

and

$$\max_{u \in U} f(u) = \sup_{w \in \mathscr{P}(U)} \sum_{u \in U} w(u) f(u). \tag{2.44}$$

Proof. To prove (2.43)

$$\sum_{u \in U} w(u) f(u) \leq \sum_{u \in U} w(u) \left(\max_{u \in U} f(u) \right) = \max_{u \in U} f(u).$$

[25] For each state, the set of randomized decision rules in that state can be represented by a unit simplex corresponding to *all* probability distributions on A_s. The set Π^{MR} is equivalent to the Cartesian product of these simplices.

[26] When maximizing over a non-finite set, "sup" is used even when it is attained to emphasize that the set is not finite. This point is explored further in the book appendix.

To prove (2.44), choose $u^* \in \arg\max_{u \in U} f(u)$ and define $w^* \in \mathscr{P}(U)$ by $w^*(u) = 1$ if $u = u^*$ and $w^*(u) = 0$, if $u \neq u^*$. Then, by the definition of w^* and (2.43),

$$\sum_{u \in U} w^*(u) f(u) = \max_{u \in U} f(u) \geq \sum_{u \in U} w(u) f(u)$$

for all $w \in \mathscr{P}(U)$ so the "sup" is attained by w^* and the result follows. □

Applying Lemma 2.1 to (2.42) gives the first key result for one-period models.

Theorem 2.1. The optimal value function in a one-period model satisfies

$$v^*(s) = \max_{a \in A_s} \left\{ \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\} \tag{2.45}$$

for all $s \in S$.

Proof. Choose $s \in S$, set $U = A_s$ and

$$f(s) = \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)),$$

and apply (2.44). □

The main consequence of this result is that in a one-period problem the value can be obtained by restricting attention to Markovian deterministic policies, which are equivalent to Markovian deterministic decision rules. The following corollary states this result more formally.

Corollary 2.1. In a one-period model,

$$v^*(s) = \max_{\pi \in \Pi^{\mathrm{MD}}} v^\pi(s) \tag{2.46}$$

for all $s \in S$.

The result is expressed this way so that it generalizes to multi-period models.

2.4.2 Optimal policies

Attention now turns to identifying optimal policies. The concept of an "arg max" is critical and will be used throughout the book.

Definition 2.11. Let U denote an arbitrary set and $f(u)$ denote a real-valued function on U that attains its maximum on U. The *arg max* of $f(u)$ is defined by

$$\arg\max_{u\in U} f(u) := \left\{ u^* \in U \mid f(u^*) = \max_{u\in U} f(u) \right\}. \tag{2.47}$$

The "arg max" of a function returns the set of arguments that maximize the function. Thus, in general, the arg max is a set with multiple elements. Only when there is a unique maximizer of a function does the arg max return a single value. Of course the definition assumes that the maximum if attained. If not, the arg max is empty.

The following theorem shows how to identify optimal policies in the one-state model.

Theorem 2.2. In a finite state and action one-period Markov decision problem, let $d_1^*(\cdot)$ denote a deterministic decision rule that for each $s \in S$ satisfies

$$d_1^*(s) \in \arg\max_{a\in A_s} \left\{ \sum_{j\in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\}. \tag{2.48}$$

Then the policy $\pi^* = (d_1^*)$ is an optimal policy.

Proof. From (2.48) and (2.38),

$$v^{\pi^*}(s) = \sum_{j\in S} p_1(j|s,d_1^*(s))\big(r_1(s,d_1^*(s),j) + r_2(j)\big) \tag{2.49}$$

$$= \max_{a\in A_s} \left\{ \sum_{j\in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\} = v^*(s) \tag{2.50}$$

for all $s \in S$. Hence from (2.41), for all $s \in S$, $v^{\pi^*}(s) \geq v^{\pi}(s)$ for all $\pi \in \Pi^{\mathrm{MR}}$ so that it is an optimal policy. □

Some observations follow:

1. **Greedy actions:** Any action that achieves the maximum in the right hand side of (2.48) is referred to as a *greedy* action.

2. **Independent problems:** To find an optimal policy in this model, one solves an independent problem (2.48) for each state $s \in S$.

3. **Inter-temporal trade-offs:** Expressions like

$$\max_{a\in A_s} \left\{ \sum_{j\in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\}$$

appear frequently in Markov decision process theory and algorithms. This maximization highlights the trade-off between the immediate reward $r_1(s, a, j)$ and the future reward $r_2(j)$. This notion of balancing a current or *myopic* reward with future rewards is fundamental to Markov decision processes.

4. **Future values:** The future reward is encapsulated in the model component $r_2(X_2)$ in the one-period model. An important question in multi-period finite or infinite horizon models is: *Can the future reward be replaced with a single function that captures the cumulative reward associated with system evolution beyond the current decision epoch*? The answer is "yes". Subsequent chapters elaborate on this key concept.

5. **Rollout and approximations:** In modern applications with very large state spaces, the terminal reward might represent an approximation to the "value" of being in state s that has been determined "offline". Then to determine the next action in state s "online", the decision maker solves the one-period problem and uses the greedy action. Such an approach has been used to determine good strategies in checkers, chess, go and backgammon.

2.4.3 State-action value functions

A significant development in computer-based model-free reinforcement learning is to shift the focus from value functions to *state-action value functions*. While not essential here and in Chapters 4-8, they play a key role in Chapters 10 and 11.

In general, a state-action value function gives the expected total reward (or discounted reward) when choosing action a in state s. For problems with planning horizons exceeding $N = 2$, the state-action value function must be distinguished according to whether the decision maker follows a specific policy or the optimal policy after action specification. This distinction is irrelevant in one-period problems in which there are no decisions after the first decision epoch.

Definition 2.12. In a one-period problem, the *state-action value function*, $q(s, a)$, is defined for all $a \in A_s$ and $s \in S$ by

$$q(s, a) := \sum_{j \in S} p_1(j|s, a)(r_1(s, a, j) + r_2(j)). \tag{2.51}$$

State-action value functions often referred to as q-functions.

The state-action value function provides the decision maker with all the information necessary to find policy value functions, optimal policies and optimal value functions. Let $\pi = (d_1)$ denote an arbitrary deterministic policy. Then from (2.38), its value is given by

$$v^\pi(s) = q(s, d_1(s)). \tag{2.52}$$

Moreover, as a direct result of Theorem 2.1,

$$v^*(s) = \max_{a \in A_s} q(s,a) \tag{2.53}$$

and from Theorem 2.2

$$d_1^*(s) \in \arg\max_{a \in A_s} q(s,a). \tag{2.54}$$

Finding $d_1^*(s)$ using (2.54) is equivalent to greedy action choice with the q-function.

Figure 2.6 provides another way to distinguish state-action value functions from optimal value functions. The optimal value function corresponds to the largest total reward that can be obtained when the system starts in state s *before* choosing an action at the first decision epoch, while the (optimal) state-action value function refers to the optimal total reward *after* choosing action a in state s. Since there are no further decisions after action choice at decision epoch 1, the state-action value function does not involve a maximization.

As covered in Part III of this book, one might choose to estimate $q(s,a)$ by simulation. In such a situation, (2.52) can be used to find the value of a policy and (2.54) and (2.53) to find an optimal policy and its value, respectively.

2.4.4 Summary of results for a one-period problem

Putting this all together, the following has been demonstrated for a one-period model:

1. There exists a Markovian deterministic policy $\pi^* = (d_1^*)$ that is optimal in the class of all policies.

2. To "solve" the one-period problem under the expected total reward criterion, compute $v^*(s)$ for all $s \in S$ according to

$$v^*(s) = \max_{a \in A_s} \left\{ \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\} \tag{2.55}$$

   and choose $d_1^*(s)$ so that

$$d_1^*(s) \in \arg\max_{a \in A_s} \left\{ \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\}. \tag{2.56}$$

3. The expected total reward of the optimal policy $\pi^* = (d_1^*)$, satisfies

$$\begin{aligned} v^{\pi^*}(s) &= \sum_{j \in S} p_1(j|s, d_1^*(s))(r_1(s, d_1^*(s), j) + r_2(j)) \\ &= \max_{a \in A_s} \left\{ \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\} = v^*(s). \end{aligned}$$

The goal in later parts of the book will be to generalize these ideas to multi-period models under different optimality criteria.

2.5 A two-state model

This section introduces a simple example, frequently referred to as *the two-state model*, that illustrates model components and notation, and previews calculations that will be seen later in the book. It formulates a finite horizon version; an infinite horizon version is analyzed in depth in later chapters. To simplify notation assume the transition probabilities and rewards are stationary, that is, they do not vary from decision epoch to decision epoch. Moreover, the reward is a function of the current state, the current chosen action, and the subsequent state.

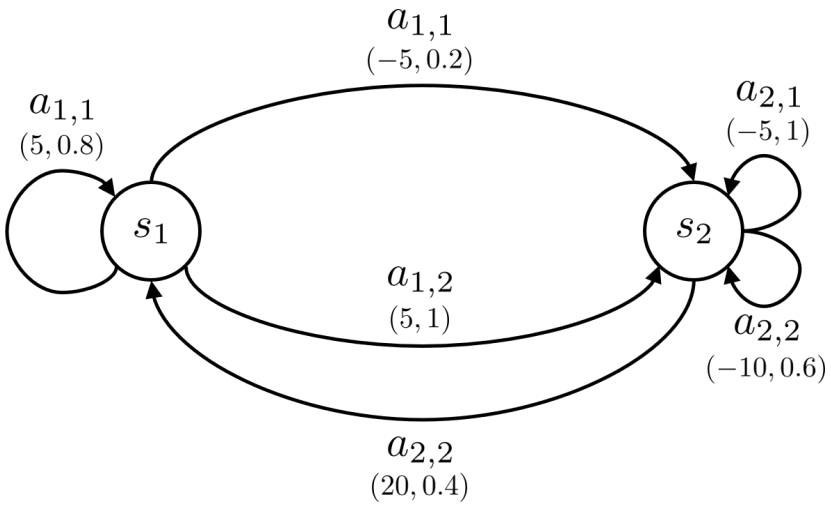

Figure 2.7: Graphical representation of two-state model. Circles denote states, arcs denote actions and transitions between states, and the expressions in parentheses denote rewards and transition probabilities, respectively. Zero probability transitions have been omitted.

Example 2.2. This example provides a concrete realization of the two-state model graphically represented in Figure 2.7. In this model, there are two states, and two actions to choose from in each state. Actions $a_{1,2}$ and $a_{2,1}$ result in deterministic outcomes – when the decision maker chooses these actions, transitions to a specified state occur with certainty. Consequently, rewards $r_n(s_1, a_{1,2}, s_1)$ and $r_n(s_2, a_{2,1}, s_1)$ are superfluous since they correspond to outcomes that cannot occur. Assume terminal rewards of 0. A precise formulation follows.

Decision epochs:

$$\{1, 2, \ldots, N\}, \quad N < \infty$$

States:

$$S = \{s_1, s_2\}$$

Actions:

$$A_{s_1} = \{a_{1,1}, a_{1,2}\}, \quad A_{s_2} = \{a_{2,1}, a_{2,2}\}$$

Rewards: For $n = 1, 2, \ldots, N$;

$$r_n(s_1, a_{1,1}, s_1) = 5, \quad r_n(s_1, a_{1,1}, s_2) = -5$$

$$r_n(s_1, a_{1,2}, s_1) = 0, \quad r_n(s_1, a_{1,2}, s_2) = 5$$
$$r_n(s_2, a_{2,1}, s_1) = 0, \quad r_n(s_2, a_{2,1}, s_2) = -5$$
$$r_n(s_2, a_{2,2}, s_1) = 20, \quad r_n(s_2, a_{2,2}, s_2) = -10$$
$$r_N(s_1) = 0, \quad r_N(s_2) = 0$$

Transition probabilities: For $n = 1, 2, \ldots, N$;

$$p_n(s_1|s_1, a_{1,1}) = 0.8, \quad p_n(s_2|s_1, a_{1,1}) = 0.2$$

$$p_n(s_1|s_1, a_{1,2}) = 0, \quad p_n(s_2|s_1, a_{1,2}) = 1$$
$$p_n(s_1|s_2, a_{2,1}) = 0, \quad p_n(s_2|s_2, a_{2,1}) = 1$$
$$p_n(s_1|s_2, a_{2,2}) = 0.4, \quad p_n(s_2|s_2, a_{2,2}) = 0.6$$

2.5.1 A one-period version

This section shows how to compute optimal value functions and policies directly, and with state-action value functions as intermediaries, in Example 2.2. Recall that in a one-period model, $N = 2$.

Optimal value functions

From (2.45), $v^*(s)$ satisfies

$$\begin{aligned} v^*(s_1) &= \max_{a \in \{a_{1,1}, a_{1,2}\}} \{p_1(s_1|s_1, a)(r_1(s_1, a, s_1) + r_2(s_1)) + p_1(s_2|s_1, a)(r_1(s_1, a, s_2) + r_2(s_2))\} \\ &= \max\{0.8 \times (5 + 0) + 0.2 \times (-5 + 0), 5 + 0\} = \max\{3, 5\} = 5. \end{aligned}$$

and

$$v^*(s_2) = \max_{a \in \{a_{1,1}, a_{1,2}\}} \{p_1(s_1|s_1, a)(r_1(s_1, a, s_1) + r_2(s_1)) + p_1(s_2|s_1, a)(r_1(s_1, a, s_2) + r_2(s_2))\}$$

$$= \max\{-5+0, 0.4\times(20+0)+0.6\times(-10+0)\} = \max\{-5,2\} = 2.$$

The first term in the "max" corresponds to action $a_{i,1}$, $i = 1, 2$ and the second term in the "max" corresponds to $a_{i,2}$, $i = 1, 2$. Thus, from (2.48), $\pi^* = (d_1^*)$ where $d_1^*(s_1) = a_{1,2}$ and $d_1^*(s_2) = a_{2,2}$ and from (2.38), $v^{\pi^*}(s_1) = 5$ and $v^{\pi^*}(s_2) = -2$.

Observe that this calculation was particularly simple because the terminal reward in each state equals 0. Exercise 5 asks you to explore the sensitivity of the optimal decision to the terminal reward.

State-action value functions

These calculations are now repeated using state-action value functions. From (2.51),

$$\begin{aligned}
q(s_1, a_{1,1}) &= p(s_1|s_1, a_{1,1})r(s_1, a_{1,1}, s_1) + p(s_2|s_1, a_{1,1})r(s_1, a_{1,1}, s_2) = 3\\
q(s_1, a_{1,2}) &= p(s_1|s_1, a_{1,2})r(s_1, a_{1,2}, s_1) + p(s_2|s_1, a_{1,2})r(s_1, a_{1,2}, s_2) = 5\\
q(s_2, a_{2,1}) &= p(s_1|s_2, a_{2,1})r(s_1, a_{2,1}, s_1) + p(s_2|s, a_{2,1})r(s_1, a_{2,1}, s_2) = -5\\
q(s_2, a_{2,2}) &= p(s_1|s_2, a_{2,2})r(s_2, a_{2,1}, s_1) + p(s_2|s_2, a_{2,2})r(s_2, a_{2,2}, s_2) = 2.
\end{aligned}$$

Hence from (2.54)

$$\begin{aligned}
d_1^*(s_1) &= \underset{a\in\{a_{1,1},a_{1,2}\}}{\arg\max}\ q(s_1, a) = a_{1,2}\\
d_1^*(s_2) &= \underset{a\in\{a_{1,1},a_{1,2}\}}{\arg\max}\ q(s_2, a) = a_{2,2}
\end{aligned}$$

and from (2.53)

$$\begin{aligned}
v^*(s_1) &= \max_{a\in\{a_{1,1},a_{1,2}\}} q(s_1, a) = 5\\
v^*(s_2) &= \max_{a\in\{a_{1,1},a_{1,2}\}} q(s_2, a) = 2
\end{aligned}$$

in agreement with direct calculation of optimal value functions.

As an aside, when coding algorithms in more complex problems, it was easier to organize calculations by first specifying $q(s, a)$.

2.5.2 Policies in a two-period version of the two-state model

In a one-period model, history-dependent and Markovian decision rules (and policies) are equivalent since the history at decision epoch 1 is the same as the state at decision epoch 1. Thus, to provide examples of the different types of policies, this section considers a two-period version. In a two-period example $N = 3$ representing two-decision epochs and a subsequent point in time when the terminal state is realized.

A Markovian deterministic policy in $\Pi^{\mathbf{MD}}$

Let $\pi_1 = (d_1, d_2)$, where

$$d_1(s) = \begin{cases} a_{1,1}, & s = s_1 \\ a_{2,1}, & s = s_2 \end{cases} \quad \text{and} \quad d_2(s) = \begin{cases} a_{1,2}, & s = s_1 \\ a_{2,1}, & s = s_2. \end{cases} \tag{2.57}$$

In state s_1, π_1 chooses action $a_{1,1}$ at the first decision epoch and $a_{1,2}$ at the second decision epoch. In state s_2, π chooses action $a_{2,1}$ at both decision epochs.

The expected total reward generated by policy π_1 for each starting state was computed by enumerating sample paths and assigning probabilities to each. These simple calculations are summarized in Table 2.1.

In state s_1 π_1 chooses $a_{1,1}$ and remains in state s_1 with probability 0.8 and jumps to state s_2 with probability 0.2. The corresponding rewards for these transitions are 5 and -5, respectively. At decision epoch 2, if the system is in state s_1, policy π_1 chooses $a_{1,2}$ resulting in a transition to state s_2 with certainty and a reward of 5. This sample path, which occurs with probability 0.8, has a total reward of $5 + 5 = 10$.

If the state at decision epoch 2 is s_2, this policy chooses action $a_{2,1}$, resulting in a self-transition and a reward of -5. This sample path occurs with probability 0.2 and has a total reward of $-5 - 5 = -10$. Since the terminal rewards are 0, the expected total reward of this Markovian deterministic policy is $0.8(10) + 0.2(-10) = 6$. Note that in these calculations the state at the end of period 2 was required to evaluate the reward in period 2.

Starting state	Sample path	Total reward	Probability	Expected total reward
s_1	$(s_1, a_{1,1}, s_1, a_{1,2}, s_2)$	10	0.8	6
	$(s_1, a_{1,1}, s_2, a_{2,1}, s_2)$	-10	0.2	
s_2	$(s_1, a_{2,1}, s_2, a_{2,1}, s_2)$	-10	1	-10

Table 2.1: Evaluation of expected total reward for Markovian deterministic policy π_1 by enumerating sample paths.

Clearly these calculations are tedious when N and the states and action sets were larger. Inductive methods in Chapter 4 provide a more efficient way to evaluate the expected rewards in a finite horizon model.

If instead, the initial state is chosen *randomly* with $\rho(s_1) = p$ and $\rho(s_2) = 1 - p$ for some $0 \leq p \leq 1$, the expected total reward corresponding to π_1 equals $16p - 10$.

A Markovian randomized policy in $\Pi^{\mathbf{MR}}$

Next is an example of a Markovian randomized policy $\pi_2 = (d_1, d_2)$. The randomized decision rule d_n, for $n = 1, 2$, choose action $a_{1,1}$ in state s_1, with probability q_1^n and action $a_{1,2}$ with probability $1 - q_1^n$ and in state s_2, choose action $a_{2,1}$ with probability

q_2^n and action $a_{2,2}$ with probability $1 - q_2^n$. Then, in the notation of Section 2.2.1, for $n = 1, 2$;

$$w_{d_n}^n(a|s) = \begin{cases} q_1^n, & a = a_{1,1},\ s = s_1 \\ 1 - q_1^n, & a = a_{1,2},\ s = s_1 \\ q_2^n, & a = a_{2,1},\ s = s_2 \\ 1 - q^2, & a = a_{2,2},\ s = s_2. \end{cases} \tag{2.58}$$

Computing the expected total reward of a Markovian randomized policy is slightly more complicated than in the Markovian deterministic case because it requires taking into account action choice probabilities specified by d_1 and d_2. Table 2.2 exhibits these calculations when the process starts in state s_1 and $q_1^1 = 0.1$, $q_1^2 = 0.5$ and $q_2^2 = 0.7$. Note that since the system starts in s_1 with certainty, q_2^1 is not used in the calculations below. In this table, a sample path represents a realization of the stochastic process $(X_1, Y_1, X_2, Y_2, X_3)$. As above, the realization of X_3 is required to compute the reward in period 2 which depends on the state at end of the horizon s^3.

Sample path	Total reward	Probability
$(s_1, a_{1,1}, s_1, a_{1,1}, s_1)$	$5 + 5 = 10$	$0.1 \times 0.8 \times 0.5 \times 0.8 = 0.0320$
$(s_1, a_{1,1}, s_1, a_{1,1}, s_2)$	$5 - 5 = 0$	$0.1 \times 0.8 \times 0.5 \times 0.2 = 0.0080$
$(s_1, a_{1,1}, s_1, a_{1,2}, s_2)$	$5 + 5 = 10$	$0.1 \times 0.8 \times 0.5 \times 1.0 = 0.0400$
$(s_1, a_{1,1}, s_2, a_{2,1}, s_2)$	$-5 - 5 = -10$	$0.1 \times 0.2 \times 0.7 \times 1.0 = 0.0140$
$(s_1, a_{1,1}, s_2, a_{2,2}, s_1)$	$-5 + 20 = 15$	$0.1 \times 0.2 \times 0.3 \times 0.4 = 0.0024$
$(s_1, a_{1,1}, s_2, a_{2,2}, s_2)$	$-5 - 10 = -15$	$0.1 \times 0.2 \times 0.3 \times 0.6 = 0.0036$
$(s_1, a_{1,2}, s_2, a_{2,1}, s_2)$	$5 - 5 = 0$	$0.9 \times 1.0 \times 0.7 \times 1.0 = 0.6300$
$(s_1, a_{1,2}, s_2, a_{2,2}, s_1)$	$5 + 20 = 25$	$0.9 \times 1.0 \times 0.3 \times 0.4 = 0.1080$
$(s_1, a_{1,2}, s_2, a_{2,2}, s_2)$	$5 - 10 = -5$	$0.9 \times 1.0 \times 0.3 \times 0.6 = 0.1620$

Table 2.2: Sample paths, rewards and probabilities corresponding to Markovian randomized policy π_2 starting in state s_1. Zero-probability sample paths are not shown.

The following expression mathematically describes the quantities used to obtain the probability in the first row of Table 2.2.

$$\begin{aligned} P^{\pi_2}\left[(X_1, Y_1, X_2, Y_2, X_3) = (s_1, a_{1,1}, s_1, a_{1,1}, s_1)\right] \\ = w_{d_1}^1(a_{1,1}|s_1)p(s_1|a_{1,1}, s_1)w_{d_2}^2(a_{1,1}|s_1)p(s_1|a_{1,1}, s_1) \end{aligned}$$

Combining sample paths which generate the same reward provides the following probability distribution of total rewards generated by π_2 and enables computation of its mean and standard deviation.

From Table 2.3 it follows that the expected total reward corresponding to starting π_2 in s_1 is 2.45 and its standard deviation is 9.35.

Reward	Probability
25	0.1080
15	0.0024
10	0.0720
0	0.6380
−5	0.1620
−10	0.0140
−15	0.0036

Table 2.3: Derived probability distribution of total reward over decision epochs 1 and 2 for Markovian randomized policy π_2 starting in state s_1.

A history-dependent deterministic policy in $\Pi^{\mathbf{HD}}$

Let $\pi_3 = (d_1, d_2)$, where

$$d_1(h) = \begin{cases} a_{1,1}, & h = s_1 \\ a_{2,1}, & h = s_2 \end{cases} \quad \text{and} \quad d_2(h) = \begin{cases} a_{1,2}, & h = (s_1, a_{1,1}, s_1) \\ a_{2,1}, & h = (s_1, a_{1,1}, s_2) \\ a_{2,2}, & h = (s_2, a_{2,1}, s_2) \end{cases} \tag{2.59}$$

For this policy, the chosen action (at decision epoch 2) depends on the entire history. At decision epoch 1, the history is simply the initial state. But at decision epoch epoch 2, the history includes the initial state, the first action chosen, and the subsequent state. If the state is s_1 at decision epoch 2, there is only one way for this to happen, given d_1 (initial state s_1 with action $a_{1,1}$). However, there are two histories that lead to s_2 at epoch 2. Thus, the action chosen in state s_2 at epoch 2 depends on whether the process started in s_1 or s_2: starting in s_1 leads to choosing action $a_{2,1}$ in the second decision epoch, otherwise $a_{2,2}$. Notice that if the policy selects action $a_{2,1}$ for the history $(s_2, a_{2,1}, s_2)$, then this policy coincides with the Markovian deterministic policy described previously.

It is left as an exercise to derive the probabilities of each realization and expected total reward.

A history-dependent randomized policy in $\Pi^{\mathbf{HR}}$

Define the policy $\pi_4 = (d_1, d_2)$ as follows. At the first decision epoch, the randomized decision rule is described by the following probability distribution:

$$w^1_{d_1}(a|h) = \begin{cases} q^1_1, & a = a_{1,1},\ h = s_1 \\ 1 - q^1_1, & a = a_{1,2},\ h = s_1 \\ q^1_2, & a = a_{2,1},\ h = s_2 \\ 1 - q^1_2, & a = a_{2,2},\ h = s_2. \end{cases} \tag{2.60}$$

In state s_1, action $a_{1,1}$ is chosen with probability q_1^1 and action $a_{1,2}$ is chosen with probability $1 - q_1^1$. In state s_2, action $a_{2,1}$ is chosen with probability q_2^1 and action a_2^1 is chosen with probability $1 - q_{1,2}$.

At the second decision epoch, there are eight histories to consider, since there are two states and two actions in each state. However, $a_{1,2}$ and $a_{2,1}$ result in deterministic transitions to state s_2, so two of the eight histories are impossible. The remaining six are listed below:

$$\begin{aligned} h_1 &:= (s_1, a_{1,1}, s_1) \\ h_2 &:= (s_1, a_{1,1}, s_2) \\ h_3 &:= (s_1, a_{1,2}, s_2) \\ h_4 &:= (s_2, a_{2,1}, s_2) \\ h_5 &:= (s_2, a_{2,2}, s_1) \\ h_6 &:= (s_2, a_{2,2}, s_2). \end{aligned}$$

Given these possible histories at the second decision epoch, a possible specification for a randomized decision rule is

$$w_{d_2}^2(a|h) = \begin{cases} q_1^2, & a = a_{1,1},\ h = h_i \\ 1 - q_i^2, & a = a_{1,2},\ h = h_i \end{cases} \tag{2.61}$$

for $i \in \{1, 5\}$ and

$$w_{d_2}^2(a|h) = \begin{cases} q_i^2, & a = a_{2,1},\ h = h_i \\ 1 - q_i^2, & a = a_{2,2},\ h = h_i \end{cases} \tag{2.62}$$

for $i \in \{2, 3, 4, 6\}$.

Again it is left as an exercise to derive the probabilities of each realization and expected total reward.

A stationary deterministic policy $\pi \in \Pi^{\mathbf{SD}}$

Suppose $\pi_5 = (d, d)$, where

$$d(s) = \begin{cases} a_{1,1}, & s = s_1 \\ a_{2,1}, & s = s_2 \end{cases} \tag{2.63}$$

This policy uses the same decision rule in both decision epochs. It is similar to the Markovian deterministic policy above, except that if the system is in state s_1 at epoch 2, the SD policy chooses action $a_{1,1}$, whereas the MD policy chooses $a_{1,2}$.

A stationary randomized policy $\pi \in \Pi^{\mathbf{SR}}$

The Markovian randomized policy above can be transformed into a stationary randomized policy $\pi = (d, d)$ by simply omitting its dependence on n, for example:

$$w_d(a|s) = \begin{cases} q_1, & a = a_{1,1},\ s = s_1 \\ 1 - q_1, & a = a_{1,2},\ s = s_1 \\ q_2, & a = a_{2,1},\ s = s_2 \\ 1 - q_2, & a = a_{2,2},\ s = s_2. \end{cases} \tag{2.64}$$

This policy has a stationary probability distribution of choosing action $a_{1,1}$ versus $a_{1,2}$ when the system is in state s_1, and similarly when the state is s_2.

Bibliographic remarks

See the historical summary in Section 1.1.1.

Exercises

1. Consider the following three-state model with $S = \{s_1, s_2, s_3\}$, actions $A_{s_i} = \{a_{i,1}, a_{i,2}\}$ for $i = 1, 2, 3$, rewards $r_n(s_i, a_{i,1}, s_j) = i - j, r_n(s_i, a_{i,2}, s_j) = j - i$ for $n = 1, 2, \ldots, N-1$ and $r_N(s_i) = -i^3$. Transition probabilities are given by $p_n(s_i|s_i, a_{i,1}) = 1 - 1/i$, $p_n(s_j|s_i, a_{i,1}) = 1/(2i)$ for $j \neq i$ and $p_n(s_i|s_i, a_{i,2}) = 1 - 1/(i+1)$, $p_n(s_j|s_i, a_{i,2}) = 1/(2(i+1))$ for $j \neq i$ for $n = 1, 2, \ldots, N$.

   (a) Provide a graphical representation of the model as in Figure 2.7.

   (b) Represent a one- and two-period version of the model as a decision tree when $\rho(s_i) = P[X_1 = s_i] = 1/3$ for $i = 1, 2, 3$. Compute the expected total reward by rolling back calculations on each path through the tree.

   (c) In a one-period version, find the distribution of X_2 for each possible initial state s_1, s_2, s_3 when the deterministic decision rule $d(s_i) = a_{i,1}$ is used at decision epoch 1.

   (d) Use (2.38) to find the expected total reward $v^{\pi}(s)$ for each $s \in S$ in a one-period version for the policy π that uses the above decision rule d at decision epoch 1.

   (e) In a one-period version, find the distribution of X_2 for each possible initial state s_1, s_2, s_3 when the randomized decision rule d' with distribution $P[d'(s_i) = a_{i,1}] = 1 - P[d'(s_i) = a_{i,2}] = e^{-0.5i}$ for $i = 1, 2, 3$ is used at decision epoch 1.

   (f) Use (2.39) to find the expected total reward $v^{\pi'}(s)$ for each $s \in S$ in a one-period version for the policy π' that uses the above decision rule d' at decision epoch 1.

   (g) Find the optimal policy in a one-period version by computing optimal value functions and state-action value functions.

2. Using 5,000 replications of Algorithm 2.1, simulate the total reward for the policies π and π' from Exercise 1 when $N = 2$.

   (a) Estimate the expected total reward of each policy and compare your results to those in Exercise 1.

   (b) Provide histograms of your estimates and comment on their shape and any differences you observe between the histograms of π and π'.

   (c) Compute the standard deviation and 95th percentile of the total returns for each policy. Interpret these quantities verbally and note why one might be interested in such quantities.

   (d) Suppose one measures the value of a reward stream (R_1, R_2) by multiplicative utility $e^{\gamma R_1} e^{\gamma R_2}$. Compare policies on the basis of their expected utility. Repeat parts (a) to (c) above using this utility function.

3. Show that when $\pi \in \Pi^{\text{MR}}$, the sequence of state and action pairs is a discrete time Markov Chain. Devise an example that shows that when $\pi \in \Pi^{\text{HR}}$, the sequence may not be Markov Chain.

4. Construct an example where an epoch-dependent state space S_n is appropriate (i.e., where using $S = \cup_n S_n$ would unnecessarily enlarge the state space at each decision epoch).

5. Write out (2.55) and (2.56) for the one-period version of the two-state problem in which $r_2(s_1) = c_1$ and $r_2(s_2) = c_2$. Plot the optimal policy as a function of the terminal rewards c_1 and c_2.

6. Evaluate the probability distributions of stochastic process generated by the deterministic history-dependent policy π_3 and the randomized history-dependent policy π_4 for the two-period model in Section 2.5.2. To organize calculations create analogs of Tables 2.2 and 2.3.

7. *Consider the following deterministic model. Let $S = \{s_1, s_2\}, A_{s_1} = \{a_{1,1}, a_{1,2}\}, A_{s_2} = \{a_{2,1}, a_{2,2}\}, r(s_1, a_{1,1}) = r(s_1, a_{1,2}) = 2, r(s_2, a_{2,1}) = r(s_2, a_{2,2}) = -2,$ and $p(s_1|s_1, a_{1,1}) = p(s_2|s_1, a_{1,2}) = p(s_1|s_2, a_{2,1}) = p(s_2|s_2, a_{2,2}) = 1$.

   (a) Provide a graphical representation of the model as in Figure 2.7.

   (b) Show that for each stationary policy π and each state $s \in S$ that the following limit exists

$$\lim_{N\to\infty} \frac{1}{N} E^{\pi} \left[\sum_{n=1}^{N} r(X_n, Y_n) \;\middle|\; X_1 = s \right]. \tag{2.65}$$

   (c) Consider a history-dependent policy π that when the initial state is s_1 chooses action $a_{1,1}$ for one period, then chooses $a_{1,2}$ so that the system

proceeds to s_2 and then chooses action $a_{2,2}$ so the system remains in s_2 for three periods, at which point it chooses action $a_{2,1}$ so that the system returns to s_1 and then chooses action $a_{1,1}$ so that it stays in state s_1 for $3^2 = 9$ periods and then chooses actions so that it proceeds to s_2 and remains there for $3^3 = 27$ periods and so forth.

Show that for π the limit in (2.65) does not exist by showing that

$$\liminf_{N\to\infty} \frac{1}{N} E^{\pi}\left[\sum_{n=1}^{N} r(X_n, Y_n) \;\middle|\; X_1 = s\right] \neq \limsup_{N\to\infty} \frac{1}{N} E^{\pi}\left[\sum_{n=1}^{N} r(X_n, Y_n) \;\middle|\; X_1 = s\right].$$