Introduction to Markov Decision Processes

Martin L. Puterman and Timothy C. Y. Chan

December 26, 2022

Chapter 2

Model Foundations

We welcome all feedback and suggestions at:
martin.puterman@sauder.ubc.ca and tcychan@mie.utoronto.ca

One must first have a strong foundation.
Sri Auribindo, Indian Philosopher 1872-1950

To appreciate the richness of Markov decision processes, one must learn the fundamentals. This chapter strives to facilitate this understanding by describing the basic components of a Markov decision process: decision epochs, states, actions, transition probabilities, and rewards. To apply the Markov decision process model to concrete examples, one needs to identify each of these components.

After describing the basic components, we introduce several other fundamental concepts, namely decision rules, policies, derived stochastic processes, and reward processes. These are not part of the model definition, but arise naturally when using this model and are derived from its basic components. We also present optimality criteria, which provide a basis for comparing the quality of decisions. Detailed analysis of Markov decision processes under these different criteria will be the focus of Chapters ?? – ??.

Our development focuses on discrete time models with discrete states and discrete actions. We briefly discuss generalizations of this setting where appropriate. We also

consider models with continuous state spaces which arise naturally in partially observed Markov decision processes, which are discussed in Chapter **??**.

2.1 Basic Model Components

A Markov decision process model describes the fundamental elements of a recurrent problem faced by a decision maker. We represent it graphically in Figure 2.1.

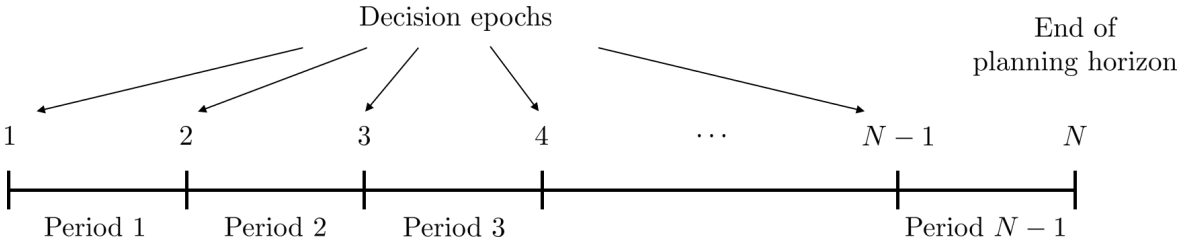

Figure 2.1: The planning horizon and decision epochs in a finite horizon setting.

Time is divided into *periods* or *stages* with a period beginning at one decision epoch and ending at the next decision epoch. At a decision epoch, a decision maker 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 making a decision, the decision maker receives a reward during that period that depends on the current state, current action and possibly the subsequent state. These steps repeat at each subsequent decision epoch. Figure 2.2 summarizes these steps within one period. We formalize these notions in the following subsections.

In this book, we use the expression *decision maker* to refer to a possibly animate entity who is making 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 the reality in which an inanimate entity chooses and executes decisions.

2.1.1 Planning Horizons and Decision Epochs

The *planning horizon* is the time interval over which decisions are made. We assume that a Markov decision process model is either:

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

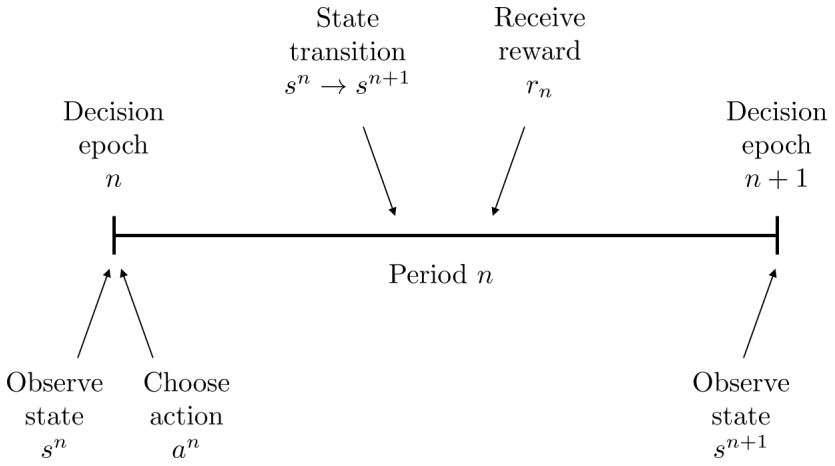

Figure 2.2: Timeline showing key events in one period. The decision maker observes state s^n in epoch n and chooses action a^n. The state transitions to a state s^{n+1} according to the transition probabilities, the decision maker receives a reward r_n, and then the decision maker gets ready to choose a new action in epoch $n+1$ after observing state s^{n+1}.

Both of these model types are considered *discrete time*, since decisions are made at discrete time points.

Each period begins at a *decision epoch*, a point in time at which the decision maker notes 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.

When the planning horizon is of finite length, our convention is to divide it into $N-1$ periods with $N-1$ decision epochs. Since no decision is made at the end of the planning horizon, epoch N, which we refer to as the *terminal epoch*, is included to evaluate the consequences of the decision at epoch $N-1$. Using $N-1$ also leads to cleaner formulas. Accordingly, our convention is to refer to finite horizon problems as $N-1$-period problems. 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 **??** focuses on finite horizon models, while Chapters **??** – **??** focus on infinite horizon models.

Throughout the book, when indexing states and action, epochs will be represented by superscripts. For transition probabilities and rewards, we will refer to the epoch using subscripts[1].

[1]For example a^n will denote the action action chosen at decision epoch n and the expression a_m will denote a specific action.

2.1.2 States

The state of the system contains all *relevant* information available to the decision maker at a decision epoch. By relevant, we mean 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 and s denote a specific state in S. We refer to S as the *state space* and an $s \in S$ as a *state*. States are necessarily *exhaustive* and *mutually exclusive*. At each decision epoch the system must be in one state in S [2].

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 being 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). We will assume 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 $s \in S$. We refer to A_s as an *action set* and each $a \in A_s$ as an an *action*. Actions are mutually exclusive; the decision maker cannot choose two actions at the same time.

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. Actions may be physically interpretable (e.g., instructing a robot to turn left) or merely a list of symbols. The action set may be independent of s, as in infinite capacity queuing control models where the action is to decide whether to admit or not admit a job. Our development will focus on A_s being discrete and finite. Assuming the states and actions are ordered, we will denote by $a_{s,j}$ 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". We refer to such states as *non-actionable*. These situations occur most often when the system reaches a state s_0 from which it cannot exit. In that case, $A_{s_0} = \{\text{Do nothing}\}$. This happens in episodic models that terminate at random times such as robotic control or optimal stopping.

[2] 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 for $n = 1, 2, \ldots, N-1$. 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. As a probability, $p_n(j|s,a)$ has the following properties:

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

. Note that in a non-actionable state, s_0, with a_0 representing the "Do nothing" action, $p_n(s_0|s_0,a_0) = 1$.

When S represents an uncountable subset of finite-dimensional Euclidean space, the transition probability may be represented by a probability density function. We will require this level of generality in Chapter **??** on partially observed models.

When the transition probabilities do not vary over decision epochs, we refer to them as *stationary*. For infinite horizon models presented in this book, we assume transition probabilities are stationary, and thus drop the subscript n.

2.1.5 Rewards

We 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 at decision epoch $n \leq N-1$ 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 **??** will illustrate both cases.

We view rewards as intrinsic parts of the model that arise naturally from the application. However, in reinforcement learning models, the modeller may need to construct an *artificial* reward function that conforms with the objectives of the task.

We 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 will assign a numerical value to each item and be indifferent between collections of items with the same total value.

We refer to these quantities as *rewards* because we model a decision maker seeking to *maximize* rewards. We represent *costs* as negative rewards to allow for a decision maker who seeks to minimize costs. Consequently, minimizing costs corresponds to maximizing rewards.

When using optimality criteria based on expected rewards[3], the quantity $r_n(s, a)$ can 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, we specify a *terminal reward* or *scrap value* $r_N(s)$ to represent the consequence of ending the planning horizon in state s. In infinite horizon models, we omit the terminal reward. Furthermore, we delete the subscript n from the reward function in the infinite horizon setting, since our focus in that case is on *stationary* rewards.

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 possible trajectories through the tree as we increase the length of the planning horizon.

2.2 Derived Objects

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

- the planning horizon N,
- 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

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

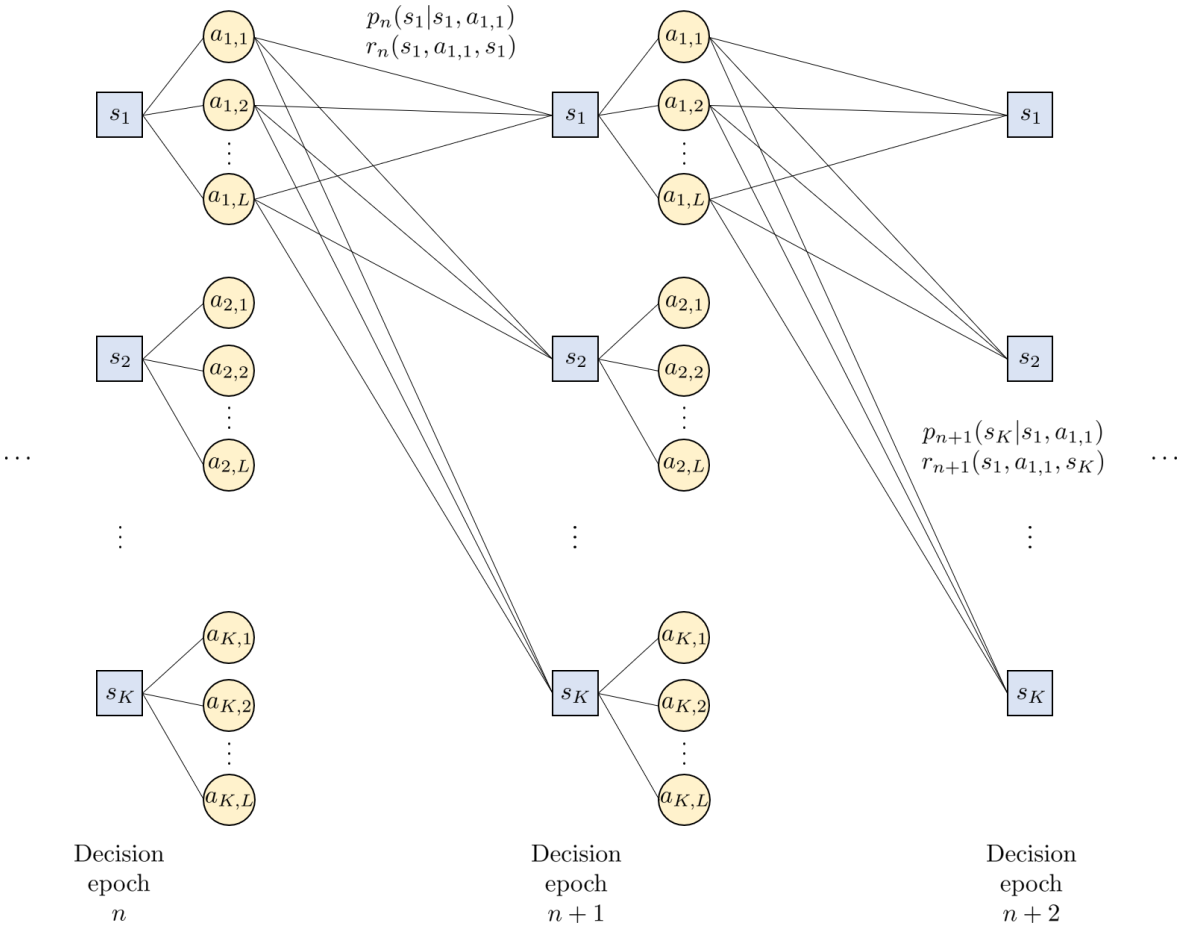

Figure 2.3: Representing a Markov decision process by a decision tree.

- the rewards $r_n(s, a, j)$ or $r_n(s, a)$.

In this section, we describe decision rules, policies, derived stochastic processes and reward processes, which 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 these two independent dimensions:

- Information: Markovian vs. history-dependent
- Mechanism: Deterministic vs. randomized

Histories

The Markovian versus 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 uses 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[4] of s^n and a history-dependent decision rule is a function of some or all of $(s^1, a^1, s^2, \ldots, a^{n-1}, s^n)$. 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. We write

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

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, we view its inclusion as redundant since the history of states and actions alone allows us to reconstruct the past rewards through the reward functions. Note that $h^1 = s^1$ and for $n = 2, 3, \ldots, N$,

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

We will use this recursion explicitly in Section **??**.

Randomization

The deterministic versus 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[5].

Types of decision rules

The above taxonomy leads to four classes of decision rules.

- A *Markovian deterministic* decision rule (MD), 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* decision rule (HD), 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

[4]Recall our convention that superscripts refer to the state and/or action chosen at decision epoch n.

[5]A degenerate probability distributions places all of the probability on one action.

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 $h^{n\prime}$ that both arrive at state s^n, $d_n(h^n)$ and $d_n(h^{n\prime})$ may choose different actions, but each action will be chosen from the same action set A_{s^n}.

- A *Markovian randomized* decision rule (MR), d_n, is a mapping from the state space to the set of probability distributions over the action set. More formally, it specifies a probability distribution

$$w^n_{d_n}(a^n|s^n) := P[Y_n = a^n],$$

where the random variable Y_n denotes the action chosen at decision epoch n, $a^n \in A_{s^n}$ and s^n is the system state at decision epoch n.

- A *history-dependent randomized* decision rule (HR), d_n, is a mapping from the set of histories to the set of probability distributions over the action set. It specifies a probability distribution

$$w^n_{d_n}(a^n|h^n) := P[Y_n = a^n], \tag{2.4}$$

where the random variable Y_n denotes the action chosen at decision epoch n, $a^n \in A_{s^n}$ when the system history is $h^n \in H_n$ and s^n is the system state at decision epoch n.

We denote these classes of decision rules 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 on this point.

We will argue in subsequent chapters that it is not necessary to consider randomized decision rules when seeking optimal policies. However, the linear programming formulation of the infinite-horizon Markov decision process modeland some policy-based reinforcement learning algorithms are based on the continuity properties of randomized decision rules (see Chapter **??**).

2.2.2 Policies

A *policy* π, also referred to as a *contingency plan* or *strategy*, is a sequence of decision rules, one for each decision epoch. In finite horizon models, we write $\pi = (d_1, d_2, \ldots, d_{N-1})$. In infinite horizon models, $\pi = (d_1, d_2, \ldots)$. Once an optimality

criterion has been specified, the decision maker's goal is to find an optimal policy – one that maximizes or minimizes the particular criterion.

The four classes of decision rules defined in Section 2.2.1 form four classes of policies, Π^{MD}, Π^{HD}, Π^{MR} and Π^{HR}. In addition, we define stationary deterministic (SD) and stationary randomized (SR) policies, denoted by the classes Π^{SD} and Π^{SR}, respectively. A *stationary policy* chooses the same decision rule at every decision epoch and will be represented as $\pi = (d, d, \dots)$[6].

Some comments about policies follow:

1. The class Π^{HR} denotes the most general class of policies; all policies we consider in the book are in this class. We require this level of generality to define optimality, but we will often not require this level of generality 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).

2. 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.

3. Stationary policies are most relevant to infinite horizon models. Proofs and methods in Chapters **??** – **??** 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.

4. 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. Chapter **??** on approximate dynamic programming provides some approaches for addressing this challenge.

5. Figure 2.4 summarizes the relationship between policy classes. Although not represented in the figure, there may exist policies that are stationary and history-dependent (i.e., non-Markovian). **(I think we should omit this possibility)** See Exercise 3. However, they represent "edge cases" and are uncommon. Thus, in this book, we only consider stationary policies within the Markovian class.

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

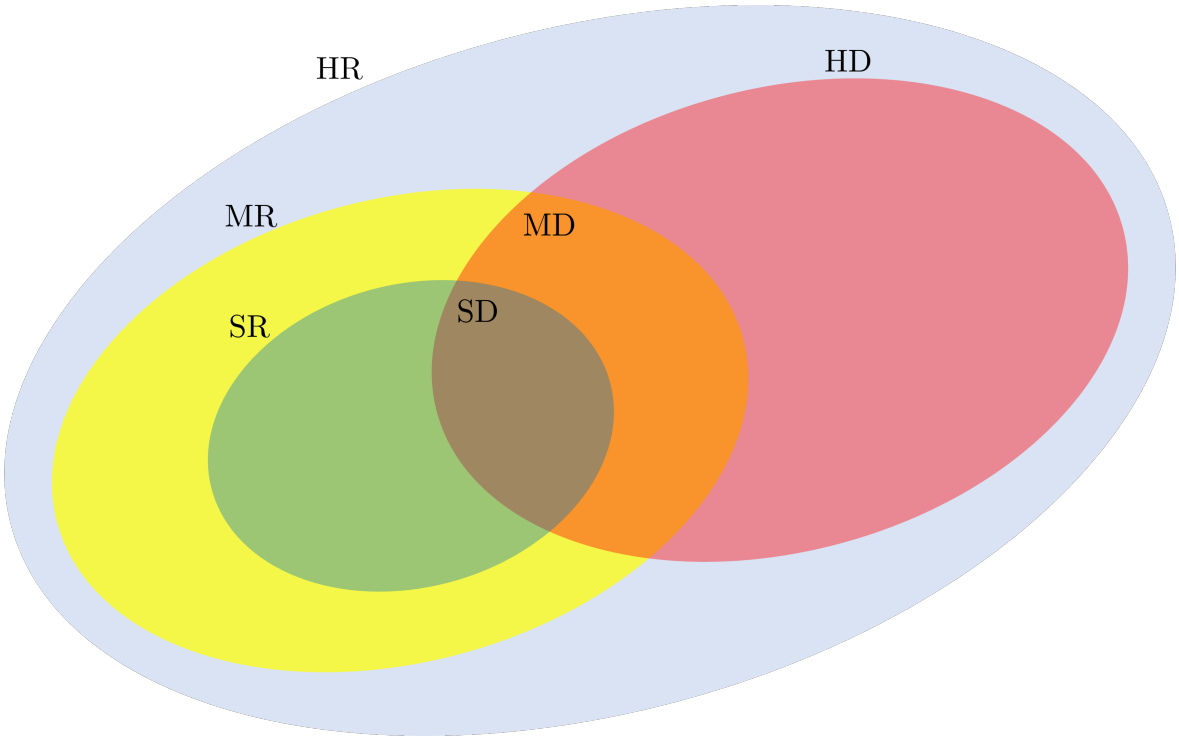

Figure 2.4: Relationships between policy classes. Labels immediately outside of an oval denote the entire oval, whereas labels inside the intersection of two ovals denote the entire intersection between them. For example, SD is the intersection of SR and HD. MD is the intersection of MR and HD, which includes SD (i.e., any region with orange tint).

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 $\gamma(s) := P[X_1 = s]$ denote the initial state probability distribution, which we assume is independent of policy choice. When, as in most applications, the system starts in a specific state at decision epoch 1, say s^1, we write $\gamma(s) = 1$ for $s = s^1$ and $\gamma(s) = 0$ for $s \neq s^1$. Let π denote a policy for either a finite or infinite horizon problem. 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:

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, i.e., when the system starts in a specific state s^1 and d_1 is deterministic, respectively. Consequently, the second action Y_2 will be random as well, and so on. Thus, a policy and initial state distribution generates 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.

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 4.

2. When π is history-dependent, the stochastic process need not be a Markov chain. That is, at each decision epoch, state and action, the transition probabilities may depend on all or some of the past. See Exercise 4.

3. The policy $\pi = (d_1, d_2, \ldots)$[7] induces a probability distribution p^π on $(X_1, Y_1, X_2, Y_2, \ldots)$. To determine it requires enumerating realizations and multiplying the transition probabilities and action randomization distributions corresponding to each realization as follows:

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

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

   where $h^1 = s^1$, $h^2 = (s^1, a^1, s^2)$, and so on. Recall that $w_{d_n}(a|h)$, defined in Section 2.2.1, denotes the probability that decision rule d chooses action a in state s at epoch n given history h.

   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 $\pi \in \Pi^{\mathrm{MD}}$

   $$p^\pi(s^1, a^1, s^2, a^2, s^3, \ldots) = \gamma(s^1) p_1(s^2|s^1, a^1) p_2(s^3|s^2, a^2) \ldots \tag{2.8}$$

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

[7] We emphasize that d_n denotes the decision rule chosen by π at decision epoch n.

2.2.4 Reward Processes

As noted in Section 2.2.3, a policy π generates a stochastic process of states and actions represented by (2.5) for 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

$$(r_1(X_1, Y_1, X_2), r_2(X_2, Y_2, X_3) \ldots, r_{N-1}(X_{N-1}, Y_{N-1}, X_N), r_N(X_N)) \tag{2.9}$$

for finite horizon models, and

$$(r_1(X_1, Y_1, X_2), r_2(X_2, Y_2, X_3), \ldots) \tag{2.10}$$

for infinite horizon models. We can write these stochastic processes more compactly as

$$(R_1, R_2, \ldots, R_N) \tag{2.11}$$

and

$$(R_1, R_2, \ldots), \tag{2.12}$$

respectively, where R_n is the random variable denoting the reward in epoch n. The distribution of rewards can be derived from $p^\pi(\cdot)$ by noting which states and actions map into the same reward value. We illustrate this approach in Table **??** when analyzing a two-state model.

In a simulated environment, the value of each realized sequence may differ. The probability distribution of this sequence of random variables describes how these values vary from realization to realization. It may be estimated empirically by the relative frequency of each realization in the simulation. We will see that the computational algorithms that appear below avoid calculating or estimating this distribution. We refer to it so as to enhance understanding of the basic principles.

Following convention, we refer to a stochastic process together with a reward function as a *reward process*. When the stochastic process is a Markov chain, we refer to it as a *Markov reward process*. Markov reward processes mostly 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

Next, we show how reward processes lead to decision making criteria. As noted in Section 2.1.5, we will assume that the reward functions r_n are real-valued scalars so that each R_n, as defined implicitly through (2.11) or (2.12), is real-valued. Thus, the random reward sequence $(R_1, R_2, \ldots, R_N)$ takes values in $\Re^N$ and $(R_1, R_2, \ldots)$ takes values in $\Re^\infty$.

To assess the value of a sequence of rewards, a decision maker may use *expected utility*, defined as follows. Let $u(\cdot)$ denote a real-valued function on $\Re^N$ or $\Re^\infty$, and let $E[\cdot]$ denote expectation with respect to the probability distribution of $(R_1, R_2, \ldots, R_N)$ or

$(R_1, R_2, \dots)$. The expected utility of these reward sequences equals $E[u(R_1, R_2, \dots, R_N)]$ or $E[u(R_1, R_2, \dots)]$, respectively.

Often utility functions are additive, so

$$u(R_1, R_2, \dots, R_N) = \sum_{n=1}^{N} u_n(R_n) \tag{2.13}$$

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

$$E[u(R_1, R_2, \dots, R_N)] = E\left[\sum_{n=1}^{N} u_n(R_n)\right] \tag{2.14}$$

and when N is infinite, it becomes

$$E[u(R_1, R_2, \dots)] = E\left[\sum_{n=1}^{\infty} u_n(R_n)\right]. \tag{2.15}$$

Note that when N is infinite, we require conditions on $u_n(R_n)$ to ensure the limit implicit in the infinite sum exists. We elaborate on this issue in Sections 2.3.1 and 2.3.2.

A decision maker chooses u_n to reflect their attitude towards risk and towards receiving rewards at different points of time. Common choices for u_n include $u_n(r) = r$ and $u_n(r) = \lambda^{n-1} r$ where $0 \leq \lambda < 1$. The quantity λ is referred to as a *discount factor*. The discount factor λ represents the present value of one unit of reward received one time period in the future. For example, if $\lambda = 0.95$, the decision maker would be indifferent between receiving one unit of reward next period and 0.95 units of reward in this period.

By using $E[u_n(R_n)] = E[R_n]$ 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
- 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 to Gamble B. Convex increasing utility functions such as $u(r) = r^2$, correspond to risk seeking decision making while concave increasing utility functions such as $u(r) = \sqrt{r}$ correspond to risk averse decision making. Which would better reflects your attitude towards gambles? Why?

The Markov decision process models we consider in this book will focus on decision making by risk neutral decision makers, i.e., where the utility function is replaced by the rewards directly. The majority of the literature on Markov decision processes

considers this case. 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 with respect to a different utility function. For example, consider coaching decisions in a football game. If the team is behind near the end of the game, the coach may be risk seeking in an attempt to catch up. Alternatively, if the team is ahead, the coach may choose to make more conservative decisions, based on a risk averse utility function, in order to hold on to the lead. See Exercise 2d for an example.

2.3 Optimality Criteria: Turning a Markov Decision Process into a Markov Decision Problem

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

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

We first state the criteria for a random reward sequence. Since policies generate random reward sequences, we then generalize these definitions to policies using the construction in Section 2.2.4.

The expected total reward criterion applies to both finite and infinite horizon models, while the latter two are most appropriate for infinite horizon models. Thus a policy is *optimal* when it maximizes the appropriate criterion over the set of all history-dependent randomized policies. To be precise, we use the expression *Markov decision problem* to correspond to a Markov decision process *together* with a optimality criterion[8]. However, as with most of the published literature, we will continue to 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

We define the expected total reward of a reward sequence and policy. Because of technical considerations we distinguish finite horizon and infinite models.

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

Finite horizon models

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

$$E\left[\sum_{n=1}^{N} R_n\right]. \tag{2.16}$$

where the expectation is over the distribution of reward sequences.

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 $n < N$ and $R_N = r_N(X_N)$.

Thus we can define the expected total reward of a policy $\pi \in \Pi^{HR}$ by

$$v^{\pi}(s) := E^{\pi}\left[\sum_{n=1}^{N} R_n \,\middle|\, X_1 = s\right] \tag{2.17}$$

where 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).

Note that we will most often see $v^{\pi}(s)$ written as either

$$v^{\pi}(s) = E^{\pi}\left[\sum_{n=1}^{N-1} r_n(X_n, Y_n, X_{n+1}) + r_N(X_n) \,\middle|\, X_1 = s\right] \quad \text{or} \tag{2.18}$$

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

depending on the form of r_n. Note that to simplify notation, we will often write $E_s^{\pi}[\cdot]$ instead of $E^{\pi}[\cdot|X_1 = s]$.

We 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[9].

Simulating the expected total reward

The following algorithm describes how you might use simulation to estimate $v^{\pi}(s)$. It illustrate a single replicate. An estimate (often referred to as the*Monte Carlo* estimate)

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

is obtained by averaging v over many replicates.

Algorithm 2.1: Simulation of a single replicate of the total reward of a Markovian policy in a finite horizon Markov decision process starting from a given state s^1.

Set $n = 1$ and $v = 0$. Fix $\pi = (d_1, d_2, \ldots, d_{N-1})$ and s^1.
while $n < N$ **do**
  If d_n is deterministic, set $a^n = d(s^n)$, otherwise sample a^n from $w_{d_n}(\cdot|s^n)$
  Sample s^{n+1} from $p_n(\cdot|s^n, a^n)$
  $v \leftarrow v + r_n(s^n, a^n, s^{n+1})$
  $n \leftarrow n + 1$
$v \leftarrow v + r_N(s^N)$

Some comments about this simulation follow:

1. 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.

2. Although you might not wish to simulate this process, this algorithm describes how the process would evolve in an implementation.

3. Chapter **??** focuses on simulation-based methods for estimating values and solving MDPs.

As an alternative to simulation, Chapter **??** 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

Similarly to the finite horizon setting, we write the expected total reward of a policy π in the infinite horizon setting as

$$v^\pi(s) := \lim_{N\to\infty} E_s^\pi \left[\sum_{n=1}^{N} R_n\right] := E_s^\pi \left[\sum_{n=1}^{\infty} R_n\right] \tag{2.20}$$

We can express this in terms of r_n as

$$v^\pi(s) = E_s^\pi \left[\sum_{n=1}^{\infty} r_n(X_n, Y_n, X_{n+1})\right] \quad \text{or} \quad v^\pi(s) = E_s^\pi \left[\sum_{n=1}^{\infty} r_n(X_n, Y_n) + r_N(X_n)\right]$$

In contrast to the finite horizon setting, we now have to be concerned with whether the above limit exists. Some possible limiting behaviors for this (or any) sequence include:

- **Convergence:** occurs when $E[R_n]$ decreases sufficiently quickly or becomes zero eventually,
- **Divergence:** occurs when $E[R_n]$ remains sufficiently positive or negative,
- **Oscillation:** occurs 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 (2.20) exists. Most examples of this kind are *optimal stopping problems*, which contain *reward-free* absorbing states[10]. More specifically, there exist policies when ensure that the expected time to reach one of the absorbing stares is finite so that effectively they behave like finite horizon problems. 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 are also referred to as *stochastic shortest path* models. See Sections **??**, **??**, and **??** for concrete examples of such problems.

The reinforcement learning literature refers to these optimal stopping models as *episodic*. This is because the decision maker learns optimal behavior by "playing" or simulating the decision process from initial state to absorption over many episodes and developing methods for identifying which policies produce the best outcomes. We will illustrate such models and approaches in Chapters **??** and **??**.

Optimal policies

We say that a policy π^* is *optimal* if

$$v^{\pi^*}(s) \geq v^{\pi}(s) \tag{2.21}$$

for all $s \in S$ and $\pi \in \Pi^{\text{HR}}$. Chapter **??** analyzes finite horizon Markov decision process models and Chapter **??** considers infinite horizon models under this optimality criterion .

2.3.2 Expected Total Discounted Reward

In contrast to the expected total reward criterion, the expected total discounted rewards 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.

[10] Appendix **??** defines basic Markov chain concepts.

Finite Horizon

We define the *expected total discounted reward* of the reward sequence $(R_1, R_2, \ldots)$ by

$$E\left[\sum_{n=1}^{N} \lambda^{n-1} R_n\right], \tag{2.22}$$

where the expectation is respect to the distribution of the sequence of rewards and the quantity $0 \leq \lambda < 1$ is referred to as a *discount factor*[11]. 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 closer 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.

Analogously to the expected total reward case, the expected total discounted reward of policy π, for each $s \in S$, is written

$$v_\lambda^\pi(s) := E_s^\pi\left[\sum_{n=1}^{N} \lambda^{n-1} R_n\right], \tag{2.23}$$

where 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. Similar expressions as (**??**) and (**??**) can be written for the expected total discounted total rewards cases, which we omit here for brevity.

Infinite Horizon

Discounting plays a fundamental role in the application and analysis of infinite horizon models. In the infinite horizon setting, the expected total discounted reward of policy π for each $s \in S$ is given by

$$v_\lambda^\pi(s) := \lim_{N\to\infty} E_s^\pi\left[\sum_{n=1}^{N} \lambda^{n-1} R_n\right] = E_s^\pi\left[\sum_{n=1}^{\infty} \lambda^{n-1} R_n\right] \tag{2.24}$$

Unlike in the expected total reward case, we do not require that $E[R_n]$ o decay sufficiently quickly or that there exist reward-free absorbing states for the infinite sum (2.24) to converge, only that $E[\lambda^{n-1} R_n]$ decays sufficiently quickly.

Throughout this book, we will assume that the absolute value of the rewards to be bounded. This means that for some finite M, $|r(s, a, j)| \leq M$ for all $a \in A_s$, $s \in S$ and

[11]The discount factor λ explicitly captures the present value of one unit of reward received one time period in the future. For example, if $\lambda = 0.95$, a decision maker using the expected discounted reward criterion would be indifferent between receiving one unit of reward next period and 0.95 units of reward in this period.

$j \in S$. Since the sum of a geometric series satisfies $\sum_{n=1}^{\infty} \lambda^{n-1} = (1-\lambda)^{-1}$ we have that

$$v_\lambda^\pi(s) \leq \sum_{n=1}^{\infty} \lambda^{n-1} M = \frac{M}{1-\lambda} \tag{2.25}$$

for all $\pi \in \Pi^{\text{HR}}$. Equation (2.25) 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.

An alternative 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 we now discuss.

Suppose the planning horizon length is not fixed but represented by a random variable, T, distributed according to a geometric distribution[12] with parameter λ, independent of $(R_1, R_2, \ldots)$, or $(X_1, Y_1, X_2, Y_2, \ldots)$ when the rewards are generated by a policy in a Markov decision process.

We can write the expected total reward over a random horizon of length T as

$$E_T \left[E \left[\sum_{n=1}^{T} R_n \right] \right], \tag{2.27}$$

where E_T denotes the expectation with respect to T. It turns out that[13]

$$E_T \left[E \left[\sum_{n=1}^{T} R_n \right] \right] = E \left[\sum_{n=1}^{\infty} \lambda^{n-1} R_n \right] \tag{2.28}$$

or in the case when the rewards are generated by policy π that

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

[12] A random variable T follows a *geometric distribution* with parameter λ if

$$P(T = k) = (1-\lambda)\lambda^{k-1} \tag{2.26}$$

for $k = 1, 2, \ldots$.

[13] 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 $\sum_{k=n}^{\infty} \lambda^{k-1} = \frac{\lambda^{n-1}}{(1-\lambda)}$. The assumptions on R_n and λ allow interchange of the order of summation.

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 total discounted reward . This is particularly appropriate in applications where the process can terminate suddenly for exogenous reasons. Examples include:

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

Thus discounting makes sense in models which do not use economic values for rewards.

Optimal policies

We say that a policy π^* is *discount optimal* if

$$v_\lambda^{\pi^*}(s) \geq v_\lambda^{\pi}(s) \tag{2.29}$$

for all $s \in S$ and $\pi \in \Pi^{\mathrm{HR}}$. Chapter **??** analyzes infinite horizon Markov decision process models under this optimality criterion.

2.3.3 Long-Run Average Reward

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, average reward is equivalent to 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 **??** provides a rigorous formulation of such a problem. For such a problem:

- **Expected total rewards** would not be able to distinguish between policies because costs or rewards would be unbounded.
- **Expected discounted rewards** 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.

We define the *average expected total reward*[14]. of the reward sequence $(R_1, R_2, \ldots)$ 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 we define the *average expected total reward*, $g^\pi(s)$ of policy π for $s \in S$ as

$$g^\pi(s) := \lim_{N\to\infty} E_s^\pi\left[\frac{1}{N}\sum_{n=1}^{N} R_n\right]. \tag{2.30}$$

Note that the limit in (2.30) exists for all stationary policies when S is finite, as will be shown in Chapter 3. It need not exist when S is countable or policies are history-dependent. In such cases we replace the limit above by the "lim inf" or the "lim sup" (See Chapter 3). The quantity g^π is sometimes referred to as the *gain* because the expected total reward in state s grows at rate $g^\pi(s)$ per epoch in the limit.

Optimal policies

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

$$g^{\pi^*}(s) \geq g^\pi(s) \tag{2.31}$$

for all $\pi \in \Pi^{\text{HR}}$ and $s \in S$. Chapter 3 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 basic building blocks of Markov decision process models. Most of the algorithms we present 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 consists of a single Markovian decision rule[15], and the derived stochastic process is (X_1, Y_1, X_2). Figure 2.5 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 $s \in S$. We assume for simplicity that S and each A_s are discrete and finite. Let $r_1(s, a, j)$

[14]We interchangeably refer to this quantity as the *long-run average reward* or *gain*.

[15]In a one period model, there is no need for history-dependent policies because the only information available to the decision maker is the state at decision epoch 1.

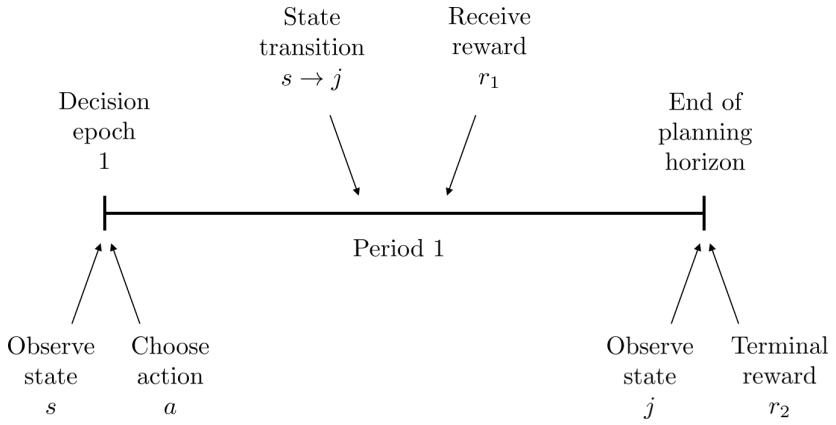

Figure 2.5: Illustration of timing of events and notation for the one-period problem. Note that action a may be chosen detrministically or probabilistically.**(This figure may be better with s^1, a^1 etc.)**

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. We use this simple model to help build intuition for more complex models.

For a policy π, let $v^\pi(s)$ be the expected total reward when the process starts with certainty in state s at decision epoch 1. It is given by

$$v^\pi(s) := E_s^\pi[r_1(X_1, Y_1, X_2) + r_2(X_2)], \tag{2.32}$$

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 $s \in S$. The assumption that the system starts in state s with certainty implies that $X_1 = s$.

Suppose $\pi = (d_1)$ is deterministic and $d_1(s) = a'$. Since s and $d_1(s) = a'$ are specified, the only random quantity above is X_2. Under this assumption, (2.32) is equivalent to

$$v^\pi(s) = \sum_{j \in S} p_1(j|s,a')(r_1(s,a',j) + r_2(j)). \tag{2.33}$$

Because a' has been specified, to compute $v^\pi(s)$, requires only $p_1(j|s,a') = P[X_2 = j]$; we do not need to derive the distribution of $r_1(X_1, Y_1, X_2) + r_2(X_2)$. Consequently, this avoids an enumeration of all of the latter's possible values as described in Section 2.2.4. Expressions of the form (2.33) appear frequently in the book, so it is important to understand why (2.33) is equivalent to (2.32).

When d_1 is randomized, Y_1 would also be random and $P[Y_1 = a|X_1 = s] = w_{d_1}(a|s)$. In this case

$$v^\pi(s) = \sum_{a \in A_s} w_{d_1}(a|s) \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)). \tag{2.34}$$

In either case, $v^\pi(s)$ is bounded above by the largest value possible for the right hand side of (2.33) over all actions $a \in A_s$, that is,

$$v^\pi(s) \leq \max_{a \in A_s} \left\{ \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\}. \tag{2.35}$$

This bound holds for any choice of d_1 (or equivalently π) in the one-period model. In the next section we expand on this concept for randomized policies.

Finding an optimal policy

Define $d_1^*(\cdot)$ to be a deterministic decision rule with the property that for any $s \in S$,

$$d_1^*(s) \in \arg\max_{a \in A_s} \left\{ \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\}, \tag{2.36}$$

where the *argmax* function returns the set of actions in A_s that maximize the expression inside the braces[16]. In other words for any $d_1^*(s)$ satisfying (2.36),

$$\sum_{j \in S} p_1(j|s, d_1^*(s))(r_1(s, d_1^*(s), j) + r_2(j)) = \max_{a \in A_s} \left\{ \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\} \tag{2.37}$$

Consequently $\pi^* = (d_1^*)$ maximizes the one-period reward among all polices. That is,

$$v^{\pi^*}(s) = \max_{\pi \in \Pi^{\mathrm{MD}}} v^\pi(s) \tag{2.38}$$

and

$$v^{\pi^*}(s) = \max_{a \in A_s} \left\{ \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\} \tag{2.39}$$

Actually $v^\pi(s)$ is the largest possible value over all randomized decision rules since no randomized decision rule can take on a greater value than the right hand side of (2.33). We will expand on this point in the next section.

[16]Formally, let B denote an arbitrary set and $f(b)$ denote a real valued function on B. Then

$$b^* \in \arg\max_{b \in B} f(b)$$

if $f(b^*) = \max_{b \in B} f(b)$. When b^* is the unique maximizer, we replace $\in$ by $=$ in the above expression.

Some key observations

We note the following:

1. **Greedy Actions:** We refer to any action that achieves the maximum in the right hand side of (2.36) as a *greedy* action.

2. **Independent Problems:** To find an optimal policy in this model, one solves an independent problem (2.36) for each state $s \in S$.

3. **Trade-offs:** The expression on the right-hand side of (2.35) appears frequently in solution algorithms for Markov decision processes. This maximization highlights the trade-off between the immediate reward $r_1(s, a, j)$ and a 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". We elaborate on this key concept in Chapters **??** – **??**.

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 "off line". Then to determine the next action in state s "on line", the decision maker solves the one period problem and uses the greedy action. Such an approach has been used to determine good strategies in chess, go and backgammon Bertsekas [2022].

2.4.1 Randomized policies are not necessary*

Our approach in the previous section found an optimal policy in the class of Markovian deterministic policies by applying (2.36) for each $s \in S$. What remains is to show that a policy chosen by (2.36) is optimal within the larger class of Markovian *randomized* policies. We now do this formally although the result should be obvious from some of the discussion above.

Let $\mathscr{P}(A_s)$ denote the set of probability distributions on A_s. Each $w \in \mathscr{P}(A_s)$ corresponds to a different randomized decision rule. Since $\mathscr{P}(A_s)$ is not a finite set, we write *sup* instead of *max* in (2.40) below[17]

[17]When maximizing over a non-finite set, we write "sup" even when the "sup" is attained to emphasize that the set is not finite. Hopefully this won't cause confusion.

This section shows that

$$\sup_{u\in\mathscr{P}(A_s)}\left\{\sum_{a\in A_s}u(a)\sum_{j\in S}p_1(j|s,a)(r_1(s,a,j)+r_2(j))\right\} \tag{2.40}$$

$$=\max_{a\in A_s}\left\{\sum_{j\in S}p_1(j|s,a)(r_1(s,a,j)+r_2(j))\right\}. \tag{2.41}$$

To see this, first note that (2.40) is greater than or equal to (2.41). This is because any maximizing action in (2.41) corresponds to a degenerate probability distribution[18] in (2.40) with all the probability mass placed on the maximizing action. To see that these two expressions are in fact equal, consider an optimal probability distribution from (2.40) that puts positive probability mass on two or more actions. All of these actions must have the same expected total reward. Thus, a degenerate probability distribution that puts all mass on any of those actions would also have the same expected total reward. Put more simply, the weighted average of a set of values cannot be larger than every one of the individual values.

Lemma 2.1 which will be used often in the rest of the book, formalizes the above argument in greater generality.

Lemma 2.1. Let U be an arbitrary finite set, let $f(\cdot)$ be a real-valued function on U, and $w(\cdot)$ be a probability distribution on U. Then,

$$\max_{u\in U}f(u)\geq\sum_{u\in U}w(u)f(u) \tag{2.42}$$

and

$$\max_{u\in U}f(u)=\sup_{w\in\mathscr{P}(U)}\sum_{u\in U}w(u)f(u). \tag{2.43}$$

Proof. To prove (2.42), let $\bar{f}:=\max_{u\in U}f(u)$. Then

$$\bar{f}=\sum_{u\in U}w(u)\bar{f}\ \geq\max_{u\in U}f(u).$$

To prove (2.43), 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 (2.42), for any $w\in\mathscr{P}(U)$

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

from which the result follows. □

[18] A *degenerate* probability distribution places all of its mass on a single element.

What the equivalence between (2.40) and (2.41) means is that a decision maker cannot receive a larger expected reward by randomizing over actions than by just using a deterministic decision rule determined by (2.36).

This argument establishes for the one-period model that if d_1^* satisfies (2.36) and $\pi^* = (d_1^*)$, then the *optimal expected total reward*, denoted $v^*(s)$, satisfies

$$v^*(s) = \sup_{\pi \in \Pi^{\mathrm{MR}}} v^\pi(s) = \max_{\pi \in \Pi^{\mathrm{MD}}} v^\pi(s) = v^{\pi^*}(s) \tag{2.44}$$

for all $s \in S$. (Be sure you can identify each expression in (2.44) and distinguish what it represents.) *Recall that for a one-period problem there is no distinction between history-dependent and Markovian policies.*

2.4.2 Summary of results for a one-period problem

Putting this all together, we have demonstrated the following 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, we compute $v^*(s)$ for all $s \in S$ as follows

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

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

$$d^*(s) \in \arg\max_{a \in A_s} \left\{ \sum_{j \in S} p_1(j|s,a)(r_1(s,a,j) + r_2(j)) \right\}. \tag{2.46}$$

3. The expected total reward of the optimal policy satisfies

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

Our goal in later parts of the book will be to generalize these ideas to more complex models.

2.5 A Two-State Model

In this section, we provide a simple concrete example that illustrates basic model components and notation, and previews calculations that will be seen later in the book. We formulate this model as a finite horizon Markov decision process; we analyze an infinite horizon version in later chapters. We assume the transition probabilities and rewards are stationary, that is, they do not vary from decision epoch to decision epoch. Moreover r is a function of the state the action and the subsequent state.

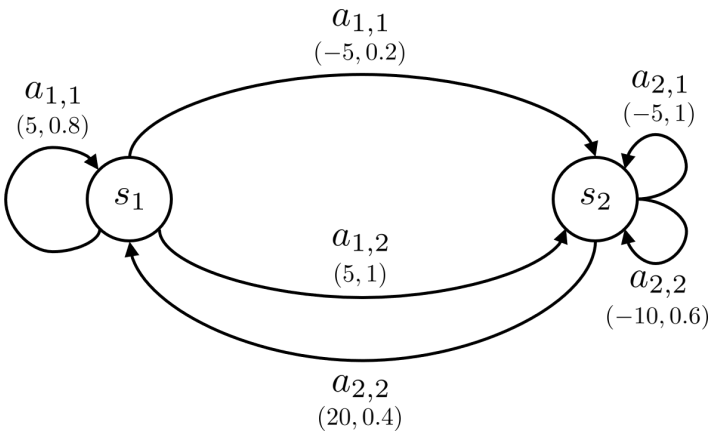

Figure 2.6: Graphical representation of two-state model. Circles denote states, arcs denote actions and the expressions in parentheses denote rewards and transition probabilities, respectively. Zero probability transitions have been omitted.

Example 2.1. Consider the two-state model illustrated in Figure 2.6. 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 occur to a specified state 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. We assume terminal rewards of 0. The formal 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 $n < N$

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

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

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

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

$$r_N(s_1) = 0, \quad r_N(s_2) = 0$$

Transition Probabilities: For $n < N$

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

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

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

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

2.5.1 A One-Period Model

Here, we illustrate calculations from Section 2.4 applied to Example 2.1. From (2.45), we compute $v^*(s)$ as

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

and

$$\begin{aligned} v^*(s_2) &= \max_{a=a_{2,1},a_{2,2}} \{p_1(s_1|s_1, a)(r_1(s_1, a, s_1) + r_2(s_1)) + p_1(s_2|s_1, a)(r_1(s_1, a, s_2) + r_2(s_2))\} \\ &= \max\{-5 + 0, 0.4 \times (20 + 0) + 0.6 \times (-10 + 0)\} = \max\{-5, 2\} = 2. \end{aligned}$$

The first term in the maxima correspond to action $a_{i,1}$, $i = 1, 2$ and the second term in the maxima correspond to $a_{i,2}$, $i = 1, 2$. Thus, from (2.46), $\pi^* = (d^*)$ where $d^*(s_1) = a_{1,2}$ and $d^*(s_2) = a_{2,2}$ and from (2.45), $v^{\pi^*}(s_1) = 5$ and $v^{\pi^*}(s_2) = -2$.

Observe that this calculation was particularly simple because the terminal reward in each state equals 0. Exercise 6 asks you to explore the sensitivity of the optimal decision to the terminal reward.

2.5.2 A Two-Period Model

We require a two-period model to provide a concrete example of each type of policy. In a one-period model, history-dependent and Markovian decision rules (and policies) are the equivalent since the history at decision epoch 1 is the same as the state at decision epoch 1. Thus, to illustrate all of the different types of policies that are possible in our two-state model, we now consider a two-period version of it.

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

Suppose $\pi = (d_1, d_2)$, where

$$d_1(s) = \begin{cases} a_{1,1}, & s = s_1 \\ a_{2,1}, & s = s_2 \end{cases} \quad \text{and} \quad d_2(s) = \begin{cases} a_{1,2}, & s = s_1 \\ a_{2,1}, & s = s_2 \end{cases} \tag{2.47}$$

In state s_1, action $a_{1,1}$ is chosen at the first decision epoch and $a_{1,2}$ is chosen at the second decision epoch. In contrast, if the state is s_2, then action $a_{2,1}$ is chosen at both decision epochs.

Next, we compute the expected total reward generated by this policy π. Suppose the process starts in state s_1. Then $a_{1,1}$ is chosen, which results in a self-transition to state s_1 with probability 0.8 and a transition to state s_2 with probability 0.2. The corresponding rewards for these transitions are 5 and -5, respectively. At epoch 2, if the state is s_1, then $a_{1,2}$ is chosen and the system transitions to state s_2 with certainty and a reward of 5. This first sample path ($s_1 \to s_1 \to s_2$), which occurs with probability 0.8, has a total reward of $5 + 5 = 10$. If the state is s_2 at epoch 2, then action $a_{2,1}$ is chosen, which results in a self-transition and a reward of -5. This second sample path ($s_1 \to s_2 \to s_2$), which occurs with probability 0.2, has a total reward of $-5-5 = -10$. Since the terminal rewards are 0, the expected total reward of this particular Markov deterministic policy π is $0.8(10) + 0.2(-10) = 6$.

A similar set of calculations will show that if the process starts in state s_2, the expected total reward of policy π is -10. Finally, an initial probability distribution that specifies the starting state as s_1 with probability p and s_2 with probability $1 - p$ would have an expected total reward of $16p - 10$. Calculations similar to this will be fundamental to analyzing partially observed MDP models in Chapter **??**.

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

Suppose that at decision epoch $n = 1, 2$, in state s_1, d_n chooses action $a_{1,1}$ with probability $q_{n,1}$ and action $a_{1,2}$ with probability $1 - q_{n,1}$ and in state s_2, d_n chooses action $a_{2,1}$ with probability $q_{n,2}$ and action $a_{2,2}$ with probability $1 - q_{n,2}$. Then, in the

notation of Section 2.2.1, we have that for $n = 1, 2$,

$$w^n_{d_n}(a|s) = \begin{cases} q_{n,1}, & a = a_{1,1},\ s = s_1 \\ 1 - q_{n,1}, & a = a_{1,2},\ s = s_1 \\ q_{n,2}, & a = a_{2,1},\ s = s_2 \\ 1 - q_{n,2}, & a = a_{2,2},\ s = s_2. \end{cases} \tag{2.48}$$

This set of probabilities can be expressed in matrix form as

$$W^n_{d_n} = \begin{bmatrix} q_{n,1} & 1 - q_{n,1} & 0 & 0 \\ 0 & 0 & q_{n,2} & 1 - q_{n,2} \end{bmatrix}, \tag{2.49}$$

where rows correspond to states and columns correspond to actions. In this notation, the Markovian deterministic policy from the previous subsection can be represented as

$$W^1_{d_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad W^2_{d_2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Computing the expected total reward of a Markovian randomized policy is similar to the Markovian deterministic case, except that an additional expectation has to be taken with respect to the probability distribution over action selection specified by $w^n_{d_n}(a|s)$.

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

Suppose $\pi = (d_1, d_2)$, where

$$d_1(h) = \begin{cases} a_{1,1}, & h = s_1 \\ a_{2,1}, & h = s_2 \end{cases} \quad \text{and} \quad d_2(h) = \begin{cases} a_{1,2}, & h = (s_1, a_{1,1}, s_1) \\ a_{2,1}, & h = (s_1, a_{1,1}, s_2) \\ a_{2,2}, & h = (s_2, a_{2,1}, s_2) \end{cases} \tag{2.50}$$

For this policy, the choice action depends on the entire history. In epoch 1, the history is simply the initial state. But in epoch 2, the history includes the initial state, the first action chosen, and then the subsequent state. If the state is s_1 at 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 $a_{2,1}$, 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.

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

Suppose at the first decision epoch, the randomized decision rule is described by the following probability distribution:

$$w_{d_1}^1(a|h) = \begin{cases} q_{1,1}, & a = a_{1,1},\ h = s_1 \\ 1 - q_{1,1}, & a = a_{1,2},\ h = s_1 \\ q_{1,2}, & a = a_{2,1},\ h = s_2 \\ 1 - q_{1,2}, & a = a_{2,2},\ h = s_2. \end{cases} \tag{2.51}$$

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 $q_{1,2}$ and action $a_{2,2}$ is chosen with probability $1 - q_{1,2}$.

At the second decision epoch, there are up to 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 randomized decision rule can be written as

$$w_{d_2}^2(a|h) = \begin{cases} q_{2,i}, & a = a_{1,1},\ h = h_i \\ 1 - q_{2,i}, & a = a_{1,2},\ h = h_i \end{cases} \tag{2.52}$$

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

$$w_{d_2}^2(a|h) = \begin{cases} q_{2,i}, & a = a_{2,1},\ h = h_i \\ 1 - q_{2,i}, & a = a_{2,2},\ h = h_i \end{cases} \tag{2.53}$$

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

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

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

$$d(s) = \begin{cases} a_{1,1}, & s = s_1 \\ a_{2,1}, & s = s_2 \end{cases} \tag{2.54}$$

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:

$$w_d(a|s) = \begin{cases} q_1, & a = a_{1,1},\ s = s_1 \\ 1 - q_1, & a = a_{1,2},\ s = s_1 \\ q_2, & a = a_{2,1},\ s = s_2 \\ 1 - q_2, & a = a_{2,2},\ s = s_2. \end{cases} \tag{2.55}$$

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 for when the state is s_2.

2.6 Bibliographic Remarks

The expression "Markov decision process" originates with Bellman [1957]. His comprehensive book formulates a wide range of problems as Markov decision processes and introduces many of the basic concepts including states, actions, transition probabilities and optimality equations. Antecedents include Wald [1947] on statistical decision models, Massé [1946] and Gessford and Karlin [1958] on reservoir management and Shapley [1953] on stochastic games.

Subsequently the book Howard [1960], which was based on his MIT doctoral dissertation, discussed the average and discounted reward model, developed the policy improvement algorithm (see Chapter **??**) for both and provided a range of colorful applications.

Inspired by Howard's book, Blackwell [1962] provides a rigorous analysis of infinite horizon, finite state and action, discounted and undiscounted Markov decision process. He formulates the model using matrix notation and establishes the optimality of stationary policies in a discounted model by showing that policy improvement converges. However, the most significant results in this paper concern the existence and computation of optimal policies as the discount factor approaches 1 in which case the limit in (2.24) diverges. To do so, he uses Markov chain results from Kemeny and Snell [1960] concerning the fundamental matrix to relate the average and discounted cases through a partial Laurent expansion. This chain of arguments was subsequently refined in A.F. Veinott [1969] where he introduces the concepts of n-discount optimality.

Several noteworthy books include Derman [1970],Sennott [1999],Bertseksas [2012], Powell [2007], and Sutton and Barto [2018]. Puterman [1994] further elaborates on historical issues.

2.7 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 2.6.

   (b) Represent a one- and two-period model as a decision tree when $P(X_1 = s_i) = \gamma(s_i) = \frac{1}{3}$ for $i = 1, 2, 3$. Compute the total reward of each path through the tree.

   (c) In a one-period model, 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 (2.33) to find the expected total reward $v^\pi(s)$ for each $s \in S$ in a one-period model for the policy π that uses the above decision rule d at decision epoch 1.

   (e) In a one-period model, 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 (2.34) to find the expected total reward $v^{\pi'}(s)$ for each $s \in S$ in a one-period model for the policy π' that uses the above decision rule d' at decision epoch 1.

   (g) Find the optimal policy in a one-period model using (2.45) and (2.46).

2. Using 5000 replications of Algorithm 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 we might be interested in such quantities.

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

3. Construct a deterministic and randomized history-dependent policy for a two-period version of the model in Example 1.

4. 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.

5. 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). Hint: consider a shortest path problem.

6. Write out (2.45) and (2.46) 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.

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 2.6.

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

$$\lim_{N\to\infty} \frac{1}{N} E^{\pi}\left[\sum_{n=1}^{N} r(X_n, Y_n) \,|\, X_1 = s\right]. \tag{2.56}$$

   (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 (2.56) does not exist by showing that

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