Foundations of Reinforcement Learning

Amir-massoud Farahmand

August 25, 2025

Contents

Preface **i**

1 Introduction **1**
1.1 Setup . . . 3
1.2 Markov Decision Process (MDP) . . . 5
1.2.1 Following a Sequence of Policies (†) . . . 11
1.3 From Immediate to Long-Term Reward . . . 12
1.3.1 Finite Horizon Tasks . . . 14
1.3.2 Episodic Tasks . . . 17
1.3.3 Continuing Tasks . . . 18
1.4 Optimal Policy and Optimal Value Function . . . 20
1.5 An Instance of an RL Algorithm: Q-Learning . . . 22
1.6 A Few Remarks on the MDP Assumption . . . 24
1.6.1 On State . . . 24
1.6.2 On Reward . . . 26
1.6.3 On Time . . . 29
1.7 Applications of Reinforcement Learning . . . 29

A Mathematical Background **33**
A.1 Probability Space 33
A.2 Norms and Function Spaces 33
A.3 Functional Analysis: Spaces, Operators, and Contraction Mapping . . 35
A.3.1 Operators 38
A.3.2 Contraction Mapping 39
A.4 Matrix Norm and Some of its Properties 42
A.5 Incremental Matrix Inversion 45
A.6 Concentration Inequalities 45
A.7 Information Theory 48
A.8 Algebraic Inequalities 49

Bibliography **51**

Preface

This book, tentatively titled *Foundations of Reinforcement Learning*, started as lecture notes for a graduate-level Introduction to Reinforcement Learning (RL) course, taught at the Department of Computer Science, University of Toronto, in Spring 2021. Since many students found the lecture notes very useful for learning a solid and modern foundation of RL, I decided to further develop and expand it into a book. What you are reading is its draft, which will gradually evolve into a full book.

This is an introductory book in the sense that I do not assume prior exposure to RL. It does, however, go beyond providing the high-level intuition or collection of algorithms, and instead tries to mathematically develop the foundation behind many important ideas and concepts in RL. Throughout the book, you and I go through the proof of many many basic, or sometimes not so basic, results in RL. The goal is to have formal statements and proofs for many major concepts and algorithms in RL.

This book should be accessible to someone who is mathematically mature and has the knowledge of probability theory, linear algebra, basics of analysis, and statistics and supervised machine learning. As a bonus material, you can find the accompanied video lectures and the course webpage, based on the Spring 2021 course.

The book is a work in progress. I intend to expand the content of existing chapters, for example by adding more exercises. I would also like to add several new chapters, including chapters on the important topics of model-based RL and exploration-exploitation tradeoff. I add a footnote at the beginning of each chapter showing what stage of maturity the chapter is. The version 0.05 is for the first full draft, version 0.1 is after its first proofread and possible minor revisions, and the versions below 0.05 are for incomplete chapters (which means that I have the content ready, but I haven't typed it yet). Versions with higher number, such as 0.2, show significant revisions compared to the first draft. Hopefully the version of all chapters will eventually converge to 1.

If you find any typos or unclear parts, please send an email to me at amirmasoud.farahmand@gmail.com. I would appreciate your feedback.

XXX Before proceeding, I would like to mention that there are very good textbooks on RL, which I encourage you to consult. A very well-known textbook is by Sutton and Barto [2018]. It provides a good intuition on many of the concepts and algorithms that we discuss in book. XXX

Amir-massoud Farahmand
March 2025

Chapter 1

Introduction

Chapter Introduction

This chapter introduces the main ideas of the book, including the key concepts that will be explored in subsequent chapters.

How should an intelligent system act such that some notion of long-term performance is maximized? This is the Reinforcement Learning (RL) problem, and is the main topic of this book.[1] To make this more clear and concrete, let us consider some examples. **RL Problem**

Consider a robot manipulator in an automobile factory. The robot perceives its workspace through cameras. It also has sensors that measure its joints angles as well as force sensors at the tip of its hand. It can send commands to its motors in order to move the joints to a certain position or velocity, or perhaps exert a certain amount of force on objects. Its goal is to successfully build a car as fast as possible with the minimum cost.

As another example, consider a smart HVAC (Heating, Ventilation, and Air Conditioning) system in a large office building. It can observe the temperature of the room using several thermometers, humidity sensors, and CO2 meters across the office. It may even have infrared cameras that can measure the temperature on the surfaces, hence providing a high-resolution temperature profile of the room. It can act in its world by varying the temperature, humidity, and the airflow rates of the vents distributed across the building. Its goal is to maximize the long-term comfort and health of people working at the office, which are measured through occasional voice feedback (`Too hot!` or `I feel a bit cold!`), or perhaps through a smart

[1] Chapter's Version: 0.15 (2025 March 12).

watch that measures their heart rate and blood oxygen level.[a]

These two were examples of artificial systems. We may also consider an animal, a cat or dog perhaps, that observes its world through its various sensors (eyes, ears, nose, whiskers, etc.) and has a musculoskeletal system that allows it to move and change the world around it. The goal of the animal can be defined at various time scales: in the short term, it is to find food and water for its next meal, which of course can be seen as just a part of the longer plan of maximizing its chance of survival and reproduction.

All these examples can be interpreted as an entity (a robot, an HVAC system, or a animal) being a nexus of causal pathways. Let us relax the definition of an animal from a biological one to include certain artificial systems too. So a robot or even a smart HVAC system are animals too, albeit an artificial ones.

As the animal perceives its environment through its various sensors (cameras; thermometers; eyes and ears), it collects information about its surrounding. It becomes the convergent point, in a non-mathematical sense, of the information around it. The perceived information has a causal power on the animal, as it affects how the animal acts. This action is based on the animal processing its perceived information and making a decision. When the animal acts (moving a joint; blowing hot air; pouncing on a bird), it becomes the point where the causal pathways diverge from the animal and affects the world around it (a screw is picked; a corner and gradually the rest of a room warms up; a bird flies away). Now as time passes, the affected world offers new information to the animal, and this in turn leads to new decisions and actions by the animal.[b]

A central part of this causal convergent and divergent process is how the animal decides on how to act based on what it perceives. The animal should act such that its goals are achieved. Successfully achieving an animal's goals often requires the animal to consider the long-term consequences of its actions. For example, it is XXX

To have such an animal, one needs to design (or evolve) many components and processes. It has various sensors and actuators, whose suitability greatly affects the animal's chance of successfully achieving its long-term goals. These, we ignore. In this book, we solely focus on the decision making aspect of the animal. Specifically, we ask the question of how this animal should act so that some notion of long-term performance is maximized. This is the RL problem. This is admittedly a very general objective. One may argue that the computational aspect of solving the AI problem is equivalent to the RL problem.

RL Methods

It is notable that in addition to the RL problem, we may use RL to refer to a set of computational methods for solving the RL problem. What kind of computation an agent needs to perform in order to ensure that its actions lead to good (or even

optimal) long-term performance? The methods that achieve these are known as RL methods.

Historically, only a subset of all computational methods that attempt to solve the RL problem have been known as the RL methods. For example, a method such as Q-Learning, which we shall soon encounter as Algorithm 1.1 in Section 1.5, is a well-regarded RL method, but an evolutionary computation method, such as a genetic algorithm, is not. One can argue that evolutionary computation methods do not have much of a "learning" component, or that they do not act at the timescale of an agent's lifetime, but act at the timescale of generations. While these are true distinctions, this way of demarcation is somewhat arbitrary. In this book, we focus on methods that are commonly studied within the "RL Community", though we have a short discussion of some of the evolutionary computation methods later in the book in Section **??**.

So far, our explanation has been high-level, not very precise, and perhaps even a bit philosophical. Next, we discuss the setup of the RL problem in a more concrete way, before formalizing it precisely in the rest of this chapter.

1.1 Setup

In reinforcement learning, we often talk about an agent and its environment, and their interaction. Figure 1.1 depicts the schematic of how they are related. The agent is the decision maker and/or learner, and the environment is anything outside it with which the agent interacts. For example, an agent can be the decision-maker part of a robot. Or it can be the decision-maker of a medical diagnosis and treatment system. For the robot agent, the environment is whatever is outside the robot, i.e., the physical system. For the medical agent, the environment is the patient.

The interaction of the agent and its environment follows a specific protocol. The current discussion is somewhat informal, but may help you understand the concept before we formalize it. At time t, which we consider to be discrete, the agent observes its state X_t in the environment. For example, this is the position of the robot in the environment. Or it can be the vital information of a patient such as their temperature, blood pressure, EKG signal, etc.

The agent then picks an action A_t according to an action-selection mechanism. This mechanism is called a policy π. It usually depends on the agent's current state X_t. The policy can be *deterministic*, which means that π is a function from the state space to the action space and $A_t = \pi(X_t)$, or it can be *stochastic* (or *randomized*), which means that π defines a probability distribution over the action space that depends on the state variable, i.e., $A_t \sim \pi(\cdot|X_t)$. Here $\sim$ refers to the

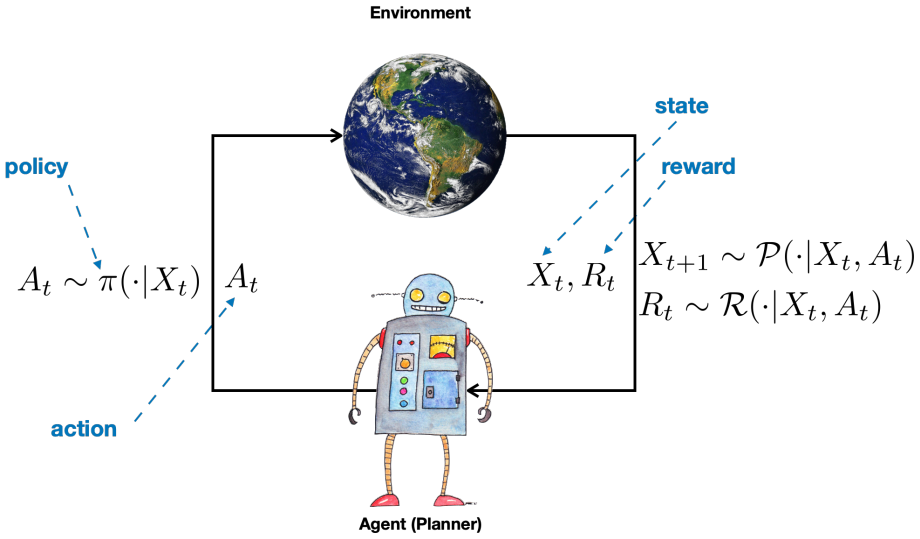

Figure 1.1: Reinforcement Learning Agent

random variable (r.v.) A_t being drawn from the distribution $\pi(\cdot|X_t)$. For example, the action can be a "move forward with velocity of 1m/s" command for the robot problem, or "inject 10mg of Amoxicillin".

Based on the selected action, the state of the agent in the environment changes and becomes X_{t+1}. The state evolves according to the dynamics of the agent in the environment, which is shown by $\mathcal{P}$ in the figure. This means that $X_{t+1} \sim \mathcal{P}(\cdot|X_t, A_t)$. The conditional distribution $\mathcal{P}$ is called *transition probability kernel* (or distribution). For the robot example, the dynamics can be described by a set of electromechanical equations that describe how the position of the robot (including its joints) change when a certain command is sent to its motor. For the medical agent, the dynamics is described by how the patient's physiology changes after the administration of the treatment. This is a very complex dynamics, which we may not have a set of equation to describe.

The agent also receives a reward signal R_t. The reward signal is a real number, and it specifies how "desirable" the choice of action A_t at state X_t (possibly leading to state X_{t+1}) has been. Therefore, $R_t \sim \mathcal{R}(\cdot|X_t, A_t)$ or $R_t \sim \mathcal{R}(\cdot|X_t, A_t, X_{t+1})$. We use the former in the rest, as it simplify our notations. All the developed theory and algorithms also work with the latter form of the reward with minor modifications. The reward is a measure of the performance of the agent at time t. For example, if

our goal is for the robot to go to a specific location and pick up an object, the reward might be defined as a positive value whenever the robot achieves that goal. And it can be zero whenever the robot is not doing anything relevant to the goal. It may even be negative when it does something that ruins achieving the goal, for example breaks the object. In this case, the negative reward is actually a punishment. For the medical agent case, the reward might be defined based on the vital signs of the patient. For example, if the patient at time t had an infection, and the action was an appropriate choice of antibiotics, and at the next time $t+1$ (maybe a day later), the infection has subsided, the agent receives a positive reward, say, $+10$.[c]

This process repeats and as a result the agent receives a sequence of states, actions, and rewards:

$$X_1, A_1, R_1, X_2, A_2, R_2, \cdots .$$

This sequence might terminate after a fixed number of time steps (say, T), or until the agent gets to a certain region of the state space, or it might continue forever.

The reward is a measure of immediate (or short-term) performance of the agent. This can be different from the long-term performance. It is possible for an agent to receive some low (or negative) rewards initially before receiving much larger rewards later on. What we often care in the RL is the long-term performance. Next we formalize this description.

1.2 Markov Decision Process (MDP)

In this section, we formally define some of the important concepts that we require throughout the course. The first important concept is the Markov Decision Process (MDP). An MDP essentially defines the environment with which the agent interacts and the problem that it should solve. In other words, an MDP encodes the decision problem.

In the rest of this book, we denote $\mathcal{M}(\Omega)$ as the space of all probability distributions defined over the space Ω, and $\mathcal{B}(\Omega)$ as the space of all bounded functions defined over Ω. So, for example, $\mathcal{M}(\mathbb{R})$ is the space of all probability distribution over real numbers, and similar for $\mathcal{B}(\mathbb{R})$. Refer to Appendix A.1 for more formal definition. We are ready to formally define elements of MDP.

Definition 1.1. *A* discounted MDP *is a 5-tuple* $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$*, where* $\mathcal{X}$ *is a measurable state space,* $\mathcal{A}$ *is the action space,* $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$ *is the transition*

probability kernel with domain $\mathcal{X} \times \mathcal{A}$, $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathbb{R})$ *is the immediate reward distribution, and* $0 \leq \gamma < 1$ *is the discount factor.*[2]

MDPs encode the temporal evolution of a discrete-time stochastic process controlled by an *agent*. The dynamical system starts at time $t = 1$ with random initial state $X_1 \sim \rho$ where " $\sim$ " denotes that X_1 is drawn from the initial state distribution $\rho \in \mathcal{M}(\mathcal{X})$.[3] At time t, action $A_t \in \mathcal{A}$ is selected by the agent controlling the process. As a result, the agent goes to the next state X_{t+1}, which is drawn from $\mathcal{P}(\cdot|X_t, A_t)$ The agent also receives an immediate reward drawn from $R_t \sim \mathcal{R}(\cdot|X_t, A_t)$.[4] Note that in general X_{t+1} and R_t are random, unless the dynamics is deterministic. This procedure continues and leads to a random *trajectory* $\xi = (X_1, A_1, R_1, X_2, A_2, R_2, \cdots)$. We denote the space of all possible trajectories as Ξ.

This definition of MDP is quite general. If $\mathcal{X}$ is a finite state space, the result is called a *finite MDP*. The state space $\mathcal{X}$ can be more general. If we consider a measurable subset of $\mathbb{R}^d$ ($\mathcal{X} \subseteq \mathbb{R}^d$), such as $(0, 1)^d$, we get the so-called continuous state-space MDPs. We can talk about other state spaces too, e.g., the binary lattices $\{0, 1\}^d$, the space of graphs, the space of strings, the space of distributions, etc. In this course, we switch back and forth between finite MDPs and continuous MDPs.

Example 1.1. *When* $\mathcal{X}$ *is finite (i.e.,* $\mathcal{X} = \{x_1, x_2, \ldots, x_m\}$*), the transition probability kernel* $\mathcal{P}(\cdot|\cdot, a)$ *is a matrix for any* $a \in \mathcal{A}$*.*

As another example, consider a dynamical system described by the following equation:

$$x_{t+1} = f(x_t, a_t), \tag{1.1}$$

where $x \in \mathbb{R}^m$, $a \in \mathbb{R}^n$, and $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$. For example, if

$$f(x, a) = Mx + Na,$$

[2] We do not use measure theoretical arguments in this book, but we use quantifiers such as *measurable* in order to make our statements precise and to avoid pathological cases. We can simply think of a measurable space as stating that the space is "nice enough". The finite and countable spaces as well as the usually used subsets of $\mathbb{R}^d$ are nice.

[3] The initial distribution ρ is not a part of the definition of MDPs. When we talk about MDPs as the descriptor of temporal evolution of dynamical systems, we usually implicitly or explicitly define the initial state distribution.

[4] We could slightly modify the interaction protocol, so that the reward R_t depends on X_t and A_t as well as X_{t+1}, i.e., $R_t \sim \mathcal{R}(\cdot|X_t, A_t, X_{t+1})$. This does not change the formalism.

with $M \in \mathbb{R}^{m\times m}$ and $N \in \mathbb{R}^{m\times n}$, we have a linear (time-invariant) dynamical system. Such a general formulation can represent a wide range of deterministic physical systems. Such dynamical systems are familiar for those with background in control theory. They can be represented in the MDP framework.

Example 1.2 (Deterministic Dynamics)**.** *We can represent deterministic dynamics such as* (1.1) *within the MDP framework. If* $x_{t+1} = f(x_t, a_t)$ *for* $f : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$*, then the transition probability kernel conditioned on a pair of* (x, a) *puts a probability mass of* 1 *at* $f(x, a)$*. Using Dirac's delta function's notation,*

$$\mathcal{P}(x'|x, a) = \delta(x' - f(x, a)).$$

Remark 1.1. *We use 'x' to denote the state and 'a' to denote the action. This is similar to how Szepesvári [2010] uses it too. These are not the only notation used in the literature, and definitely not the most commonly used one. Sutton and Barto [2018] use 's' for the state and 'a' for the action. The authors from the control theory background tend to use 'u' for the action, and 'i' [Bertsekas and Tsitsiklis, 1996] or 'x' for the state [Bertsekas, 2018].*

The reason I use 'x' for state is partly historical and partly because of the following justification: I find it more aligned with how the rest of ML, and even applied math, use x *as the input to a function. The fact that the input is an agent's state does not mean that we have to use a different notation. I find it slightly more appealing to see* $f(x)$ *instead of* $f(s)$*, though nothing is inherently wrong with the latter usage. The reason I stick to 'a' for the action, instead of 'u' commonly used in control theory, does not have much of a justification other than a nod to the CS/AI roots of RL.*

Let us tend to the policy π. Recall from Section 1.1 that the policy is the action-selection mechanism of the agent. The goal of the RL agent is to find a "good" policy, to be defined what it exactly means. Let us formally define it.

Definition 1.2 (Definition 8.2 and 9.2 of Bertsekas and Shreve [1978])**.** *A* ***policy*** *is a sequence* $\bar{\pi} = (\pi_1, \pi_2, \ldots)$ *such that for each* t*,*

$$\pi_t(a_t|X_1, A_1, X_2, A_2, \ldots, X_{t-1}, A_{t-1}, X_t)$$

is a universally measurable stochastic kernel on $\mathcal{A}$ *given* $\underbrace{\mathcal{X} \times \mathcal{A} \times \cdots \times \mathcal{X} \times \mathcal{A} \times \mathcal{X}}_{2t-1 \text{ elements}}$ *satisfying*

$$\pi_t(\mathcal{A}|X_1, A_1, X_2, A_2, \ldots, X_{t-1}, A_{t-1}, X_t) = 1$$

for every $(X_1, A_1, X_2, A_2, \ldots, X_{t-1}, A_{t-1}, X_t)$.

If π_t is parametrized only by X_t, that is

$$\pi_t(\cdot|X_1, A_1, X_2, A_2, \ldots, X_{t-1}, A_{t-1}, X_t) = \pi_t(\cdot|X_t),$$

$\bar{\pi}$ is a Markov *policy.*

If for each t and $(X_1, A_1, X_2, A_2, \ldots, X_{t-1}, A_{t-1}, X_t)$, the policy π_t assigns mass one to a single point in $\mathcal{A}$, $\bar{\pi}$ is called a deterministic (nonrandomized) *policy; if it assigns a distribution over $\mathcal{A}$, it is called* stochastic *or* randomized *policy.*

If $\bar{\pi}$ is a Markov policy in the form of $\bar{\pi} = (\pi, \pi, \ldots)$, it is called a stationary *policy.*

This definition categorizes whether the policy is time-dependent or not, whether it uses only the current state X_t or looks at the previous state and action pairs too, and whether it is deterministic or stochastic.

To understand this definition better, let us start from the simplest form of the policy and gradually get to more general cases. The simplest form of a policy is a stationary Markov deterministic policy. This is a function from the state space $\mathcal{X}$ to the action space $\mathcal{A}$, that is $\pi : \mathcal{X} \to \mathcal{A}$, and we use $\pi(x)$ to refer to it. This policy is a function (deterministic property) that does not depend on time (stationary property). It ignores the past states and actions $X_{t-1}, A_{t-1}, X_{t-2}, \cdots$ and only looks at the current state X_t (Markov property). A slightly more complex form is when we allow this policy to be stochastic, instead of deterministic. A stationary Markov stochastic policy is a conditional distribution over the action space depending on the state, that is, $\pi(\cdot|x) \in \mathcal{M}(\mathcal{A})$.

In most of this book, we only focus on stationary Markov policies, and simply use "policy" to refer to a stationary Markov policy $\pi(\cdot|x)$, without any adjectives. It turns out that the class of stationary Markov policies is rich enough to allow the agent make optimal decisions, under the condition that we have access to the actual state of the MDP X_t. Neither non-stationarity nor non-Markovity does not bring any extra performance to the table. As we shall discuss later in Section 1.6, if the agent does not have access to the state of the MDP and only observe some aspects of the state, this is not necessarily true, and the use of a non-Markov policy might be needed for optimal decision making. This means that the policy should look not only at the most recent observation, but at the past observations too.

We define the following terminology and notations in order to simplify our exposition.

Definition 1.3. *We say that an agent is "following" a Markov stationary policy π whenever A_t is selected according to the policy $\pi(\cdot|X_t)$, i.e., $A_t = \pi(X_t)$ (deterministic) or $A_t \sim \pi(\cdot|X_t)$ (stochastic). The policy π induces two transition probability*

kernels $\mathcal{P}^\pi : \mathcal{X} \to \mathcal{M}(\mathcal{X})$ *and* $\mathcal{P}^\pi : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X} \times \mathcal{A})$. *For a measurable subset* A *of* $\mathcal{X}$ *and a measurable subset* B *of* $\mathcal{X} \times \mathcal{A}$ *and a deterministic policy* π*, denote*

$$(\mathcal{P}^\pi)(A|x) \triangleq \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}y|x, \pi(x)) \mathbb{I}_{\{y \in A\}},$$
$$(\mathcal{P}^\pi)(B|x, a) \triangleq \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}y|x, a) \mathbb{I}_{\{(y, \pi(y)) \in B\}}.$$

If π *is stochastic, we have*

$$(\mathcal{P}^\pi)(A|x) \triangleq \int_{\mathcal{X} \times \mathcal{A}} P(\mathrm{d}y|x, a) \pi(\mathrm{d}a|x) \mathbb{I}_{\{y \in A\}},$$
$$(\mathcal{P}^\pi)(B|x, a) \triangleq \int_{\mathcal{X} \times \mathcal{A}} P(\mathrm{d}y|x, a) \pi(\mathrm{d}a'|y) \mathbb{I}_{\{(y, a') \in B\}}.$$

The m*-step transition probability kernels* $(\mathcal{P}^\pi)^m : \mathcal{X} \to \mathcal{M}(\mathcal{X})$ *and* $(\mathcal{P}^\pi)^m : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X} \times \mathcal{A})$ *for* $m = 2, 3, \cdots$ *for a deterministic policy* π *are inductively defined as*[5]

$$(\mathcal{P}^\pi)^m(A|x) \triangleq \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}y|x, \pi(x)) (\mathcal{P}^\pi)^{m-1}(A|y),$$
$$(\mathcal{P}^\pi)^m(B|x, a) \triangleq \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}y|x, a) (\mathcal{P}^\pi)^{m-1}(B|y, \pi(y)).$$

The difference between the transition probability kernels $\mathcal{P}^\pi : \mathcal{X} \to \mathcal{M}(\mathcal{X})$ and $\mathcal{P}^\pi : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X} \times \mathcal{A})$ is in the way the policy affects the action selection: in the former, the action of the first step is chosen according to the policy, while in the latter the first action is pre-chosen and the policy chooses the action in the second step.

The m-step transition probability kernels $(\mathcal{P}^\pi)^m(A|x)$ is the probability that the agent starts at state x, chooses actions according to the policy π for m steps, and falls within the set A. Similarly, $(\mathcal{P}^\pi)^m(B|x, a)$ is the probability of the agent starting from state x, choosing action a at the first step, and for the next $m-1$ steps, chooses actions according to the policy π.

We may sometimes use $\mathcal{P}^\pi(A|x; m)$ and $\mathcal{P}^\pi(B|x, a; m)$ to refer to $(\mathcal{P}^\pi)^m(A|x)$ and $(\mathcal{P}^\pi)^m(B|x, a)$, if having a superscript reduces the clutter.

In case thinking about countable space is more intuitive, the definition $(\mathcal{P}^\pi)^m(A|x)$ for A being a state z ($A = \{z\}$) is

$$(\mathcal{P}^\pi)^m(z|x) \triangleq \sum_{y \in \mathcal{X}} \mathcal{P}(y|x, \pi(x)) (\mathcal{P}^\pi)^{m-1}(z|y).$$

[5]The definition for the stochastic policy would be similar.

If we arrange the probabilities in a matrix, the definition of $(\mathcal{P}^\pi)^m$ takes a perhaps more familiar form. Consider a state space $\mathcal{X} = \{x_1, \ldots, x_{|\mathcal{X}|}\}$. Let us identify $(\mathcal{P}^\pi)(x_j|x_i)$ (the probability of starting from x_i and going to x_j) with an $|\mathcal{X}| \times |\mathcal{X}|$ matrix $P^\pi \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ with its i-th row and j-th

$$[P^\pi]_{i,j} = (\mathcal{P}^\pi)(x_j \mid x_i). \tag{1.2}$$

Consider $(\mathcal{P}^\pi)^2$, the 2-step transition probability kernel (or matrix), and let us calculate the probability of starting from state x_i and moving to state x_j after 2 steps:

$$(\mathcal{P}^\pi)^2(x_j|x_i) = \sum_{y\in\mathcal{X}} \mathcal{P}^\pi(y|x_i)\mathcal{P}^\pi)(x_j|y = \sum_{k\in\{1,\ldots,|\mathcal{X}|\}} P^\pi_{i,k}P^\pi_{k,j} = [P^\pi P^\pi]_{i,j} = [(P^\pi)^2]_{i,j},$$

where the penultimate equality is due to the definition of matrix multiplication. This shows that the 2-step transition probability kernel is the same as taking the matrix P^π and raising it to the power of two. This argument can be performed for any $m \geq 1$ to conclude that for countable state spaces, $(\mathcal{P}^\pi)^m$ can be identified with the matrix P^π raised to the power of m, i.e., $(P^\pi)^m$. In the rest of this notes, we do not use a different font for $\mathcal{P}$ and P, and use $\mathcal{P}$ for both cases.

A useful, and intuitive, property of following a policy π is that if the agent follows it for m_1 steps and then it continues following it for another m_2 steps, from wherever it landed after the first m_1 steps, it is the same as following the agent following π for $m_1 + m_2$ steps. This can be written as

$$(\mathcal{P}^\pi)^{m_1}(\mathcal{P}^\pi)^{m_2} = (\mathcal{P}^\pi)^{m_1+m_2}.$$

We define another notation, which shall be helpful in our discussions.

Definition 1.4. *Given the transition probability kernel $\mathcal{P}$ and a bounded measurable function $f \in \mathcal{B}(\mathcal{X})$, we define $\mathcal{P}f : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ as the function*

$$(\mathcal{P}f)(x,a) \triangleq \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}y|x,a)f(y), \qquad \forall (x,a) \in \mathcal{X} \times \mathcal{A}.$$

Likewise, given the transition probability kernel induced by a policy π, we define $\mathcal{P}^\pi f : \mathcal{X} \to \mathbb{R}$ as

$$(\mathcal{P}^\pi f)(x) \triangleq \int_{\mathcal{X}} \mathcal{P}^\pi(\mathrm{d}y|x)f(y), \qquad \forall x \in \mathcal{X}.$$

In other words, $\mathcal{P}^\pi f$ is the function whose value at a state x is the expected value of function f according to the distribution $\mathcal{P}^\pi(\cdot|x)$, that is, $(\mathcal{P}^\pi f)(x) = \mathbb{E}_{X' \sim \mathcal{P}^\pi(\cdot|x)}[f(X')]$. The interpretation of $\mathcal{P}f$ is similar.

For a countable state space $\mathcal{X}$, we have

$$(\mathcal{P}^\pi f)(x) \triangleq \sum_{y \in \mathcal{X}} \mathcal{P}^\pi(y|x) f(y), \qquad \forall x \in \mathcal{X}. \tag{1.3}$$

We may sometimes use $\mathcal{P}(\cdot|x,a)f$ or $\mathcal{P}^\pi(\cdot|x)f$ to refer to the same functions.

Exercise 1.1. *Consider a 2-state MDP with a policy that induces*

$$\mathcal{P}^\pi = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}.$$

Assume that the reward at state x_1 is zero and the reward at state x_2 is equal to 1, that is, $r^\pi = [0; 1]$. Answer the following questions assuming that the agent starts at state x_1:

- *What is the immediate reward it receives?*
- *What is the expected reward it receives after moving 1 step in the environment? What about 2? And 10? (You probably need to write one or two lines of code to compute this.)*
- *What is the expected reward it receives after moving infinite steps in the environment?*

Now answer the same questions for when the agent start at state x_2.

Exercise 1.2. *Suppose that $f(x) = c$ for a constant real-valued number $c \in \mathbb{R}$. What is $\mathcal{P}^\pi f$?*

1.2.1 Following a Sequence of Policies (†)

For a sequence of policies $\pi_{1:m} = (\pi_1, \ldots, \pi_m)$, the transition probability kernel of following them in the order of π_1, then π_2, etc., is denoted by $\mathcal{P}^{\pi_1:\pi_m}$ or $\mathcal{P}^{\pi_{1:m}}$ and is

$$\mathcal{P}^{\pi_{1:m}}(A|x) \triangleq \int_{\mathcal{X}} \mathcal{P}^{\pi_1}(\mathrm{d}y|x) \mathcal{P}^{\pi_{2:m}}(A|y),$$
$$\mathcal{P}^{\pi_1:\pi_m}(B|x,a) \triangleq \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}y|x,a) \mathcal{P}^{\pi_2:\pi_m}(A|y),$$

for deterministic policies, and similar for stochastic policies.

The interpretation of $\mathcal{P}^{\pi_{1:m}}(A|x)$ is that this is the probability of starting from a state x, following the sequence of policies $\pi_{1:m}$, and ending up in a set A after exactly m steps (and similar interpretation for $\mathcal{P}^{\pi_1:\pi_m}(B|x)$). When the state space is countable, we can also write it in the matrix form:

$$\mathcal{P}^{\pi_{1:2}}(x_j|x_i) = \sum_{y\in\mathcal{X}} \mathcal{P}^{\pi_1}(y|x_i)\mathcal{P}^{\pi_2}(x_j|y) = \sum_{k\in\{1,\dots,|\mathcal{X}|\}} P^{\pi_1}_{i,k} P^{\pi_2}_{k,j} = [P^{\pi_1} P^{\pi_2}]_{i,j}.$$

As the matrices are not commutative in general $P^{\pi_1}P^{\pi_2} \neq P^{\pi_2}P^{\pi_1}$, which is intuitive, as following a policy π_1 and then π_2 (which induces $\mathcal{P}^{\pi_{1:2}}$) is not the same as following π_2 and then π_1 (which induces $\mathcal{P}^{\pi_{2:1}}$).

The value of function $\mathcal{P}^{\pi_{1:m}} f : \mathcal{X} \to \mathbb{R}$ at state x is the expected value of f at the distribution of an agent that starts at x and follows the policy sequence $\pi_1, \dots, \pi_m$.

1.3 From Immediate to Long-Term Reward

Recall that the RL problem is the problem of how to act so that some notion of long-term performance is maximized. In this section, we elaborate on the meaning of "long-term". Along the way, we learn about important concepts such as *return* and *value functions*. It turns out that we can define long-term in different ways. We discuss some of them here. Before that, however, let us start with a simpler problem of maximizing the immediate (or short-term) performance first.

Suppose that an agent starts at state $X_1 \sim \rho \in \mathcal{M}(\mathcal{X})$, chooses action $A_1 = \pi(X_1)$ (deterministic policy), and receives a reward of $R_1 \sim \mathcal{R}(\cdot|X_1, A_1)$. This ends one round of interaction of the agent and its environment. The agent then restarts, samples another (independent) $X_1 \sim \rho \in \mathcal{M}(\mathcal{X})$, and repeats as before again and again. We call each of these rounds an *episode*. Here the episode only lasts one time-step.

How should this agent chooses its policy in order to maximize its performance? To answer this question, we need to specify what performance actually refers too. There are several sensible ways to define the performance of the agent, one of which is to talk about the *average (expected) reward* that the agent receives within one episode. The meaning of average here is that if the agent repeats this interaction with the environment for many episodes (infinitely), how much reward it receives in average. So the averaging is over the episodes.

expected reward

Answering the question of how the agent should act to maximize this notion of performance is easy. Let us define *expected reward* as

$$r(x,a) \triangleq \mathbb{E}\left[R|X=x, A=a\right]. \tag{1.4}$$

Here the r.v. R is distributed according to $\mathcal{R}(\cdot|x,a)$. Paying attention that this is the expected reward is important: even if the agent starts at a state x and chooses action a, the actual reward it receives can be higher or lower than $r(x,a)$. In expectation, however, it receives $r(x,a)$.

In order to maximize the expected reward, the best action depends on the state the agent initially starts with. At state x, it should choose an action that maximizes the average reward $r(x,a)$ at that state. That is,

$$a^* \leftarrow \operatorname*{argmax}_{a\in\mathcal{A}} r(x,a).$$

This is the best, or *optimal*, action at state x.[6] By the definition of argmax, no choice of action can gather more rewards in expectation. With this choice, we can define the optimal policy $\pi^* : \mathcal{X} \to \mathcal{A}$ as the function that at each state x returns

$$\pi^*(x) \leftarrow \operatorname*{argmax}_{a\in\mathcal{A}} r(x,a). \tag{1.5}$$

Note that the optimal policy is only a function of the agent's state x. It does not depend on the initial distribution ρ.

Exercise 1.3. *Describe a similar setup where the optimal policy depends on ρ. The performance measure should still be the expected reward that the agent receives. But feel free to change some crucial aspect of the agent.*

Exercise 1.4. *Explain how a standard supervised learning problem can be formulated as finding the policy that maximizes the immediate expected reward. To be concrete, focus on the binary classification problem. What is the state x? What is the action a? And what is the reward $r(x,a)$?*

Exercise 1.5 ($\star\star$)**.** *Come up with a real-world application where the goal is to find an optimal policy maximizing the immediate reward.*

Exercise 1.6 ($\star\star$)**.** *We equate the performance as maximizing the expected reward. What other sensible performance measures can you think of? It should still be related to the rewards that the agent receives in its episode.*

Let us consider some setups where the agent interacts with the environment for multiple steps.

[6] If there are more than one action that attains $\max_{a\in\mathcal{A}} r(x,a)$, the agent can choose any of them.

1.3.1 Finite Horizon Tasks

The discussion of this section so far has been for when the episode length is $T = 1$. When $T = 1$, as soon as the agent chooses its action A_1, it receives a reward R_1, and then the episode terminates. We can extend the setup to when $T > 1$. In that case, within each episode, the interaction of the agent following a policy π goes like this:

- The agent starts at $X_1 \sim \rho \in \mathcal{M}(\mathcal{X})$.
- It chooses action $A_1 = \pi(X_1)$ (or $A_1 \sim \pi(\cdot|X_1)$ for a stochastic policy).
- The agent goes to the next-state $X_2 \sim \mathcal{P}(\cdot|X_1, A_1)$ and receives reward $R_1 \sim \mathcal{R}(\cdot|X_1, A_1)$.
- The agent chooses $A_2 = \pi(X_2)$ (or $A_2 \sim \pi(\cdot|X_2)$ for a stochastic policy).
- The agent goes to the next-state $X_3 \sim \mathcal{P}(\cdot|X_2, A_2)$ and receives reward $R_2 \sim \mathcal{R}(\cdot|X_2, A_2)$.
- This process repeats for several steps until the agent gets to the last state $X_T \sim \mathcal{P}(\cdot|X_{T-1}, A_{T-1})$, chooses action $A_T = \pi(X_T)$ (or $A_T \sim \pi(\cdot|X_T)$ for a stochastic policy), and receives $R_T \sim \mathcal{R}(\cdot|X_T, A_T)$.

Afterward, the agent starts a new episode.[7,8]

How should we evaluate the performance of the agent as a function of the reward sequence $(R_1, R_2, \ldots, R_T)$? A common choice is to compute the sum of rewards:

$$G^\pi \triangleq R_1 + \ldots + R_T. \tag{1.6}$$

return

The r.v. G^π is called the *return* of following policy π. As it is random, its value in each new episodes would be different (unless the dynamics and policy are deterministic, and ρ always selects the same initial state; or other similar cases).

Another choice is to consider the *discounted* sum of rewards. Given a discount factor $0 \leq \gamma \leq 1$, we define the return as

$$G^\pi \triangleq R_1 + \gamma R_2 + \ldots + \gamma^{T-1} R_T. \tag{1.7}$$

[7] We could generalize the interaction by allowing $\mathcal{P}$ and $\mathcal{R}$ to be time-dependent. In that case, $X_{t+1} \sim \mathcal{P}_t(\cdot|X_t, A_t)$ and $R_t \sim \mathcal{R}_t(\cdot|X_t, A_t)$. Also the policy might be non-stationary $\bar{\pi} = (\pi_1, \ldots, \pi_T)$, so at each time step t, the action is selected according to π_t. We comment on this further in Remark 1.2 at the end of this section.

[8] If the reward R_t depended on (X_t, A_t, X_{t+1}), as opposed to (X_t, A_t) that we consider here, the agent would get to X_{T+1} and terminates; it would not require to choose an action at the last step.

Whenever $\gamma < 1$, the reward that is received earlier in an episode contributes more to the return. Or similarly, the contribution of later rewards are discounted and contribute less (when $\gamma = 1$, there is no discounting, and all rewards contribute equality). Intuitively, this means that such a definition of return pays more emphasis on earlier rewards. An everyday example is that we prefer to get a cookie today instead of tomorrow, and we prefer a cookie tomorrow to a cookie a week later – assuming that we like cookies after all. How much exactly your preference changes depends on the value of γ. This is an example of a delayed gratification, which has been observed in humans. The Marshmallow test is famous example of it.[d]

The discount factor has a financial interpretation too and is related to the inflation rate.The inflation is the rise over time in the average price (usually over a large part of the market, for example, the consumer goods and services). If the price of a certain set of goods has changed from \$1 to $\$(1+\text{rate}_{\text{inflation}})$ next year, the inflation is $\text{rate}_{\text{inflation}}$ per year. This means that whenever $\text{rate}_{\text{inflation}} > 0$, the value of a dollar this year is more than a value of dollar next year. So if you have a choice in receiving a dollar this year or some amount of dollar next year, you need to consider the inflation rate, and discount the value of dollar next year by $\gamma = \frac{1}{1+\text{rate}_{\text{inflation}}}$. Of course, this is all based on the assumption that you do not have an immediate need for that dollar, so you can potentially postpone the time you receive it.

The return (1.7) (and (1.6) as a special case) is a random variable. To define a performance measure that is not random, we compute its expectation. We define

$$V^{\pi}(x) \triangleq \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} R_t | X_1 = x\right]. \tag{1.8}$$

This is the expected value of return if the agent starts at state x and follows policy π. The function $V^{\pi} : \mathcal{X} \to \mathbb{R}$ is called the *value* function of π.

More generally, we can define the return from time $\tau \in \{1, \ldots, T\}$ until the end of episode, which is time T, as

$$G_{\tau}^{\pi} \triangleq \sum_{t=\tau}^{T} \gamma^{t-\tau} R_t. \tag{1.9}$$

And likewise, we define the value function at time τ to be

$$V_{\tau}^{\pi}(x) \triangleq \mathbb{E}\left[G_{\tau}^{\pi} | X_{\tau} = x\right]. \tag{1.10}$$

Clearly, V_1^{π} is the same as V^{π} in (1.8).

Comparing the expected reward (1.4) and the value function (1.10) is instructive. We first focus on $T = 1$. We get that

$$V^\pi(x) = \mathbb{E}\left[R_1 | X_1 = x\right].$$

This is similar to $r(x, a) = \mathbb{E}\left[R | X = x, A = a\right]$ with the difference that $r(x, a)$ is conditioned on both x and a, whereas V^π is conditioned on x. The choice of action a in V^π is governed by the policy π, and is $a = \pi(x)$ (deterministic) or $A \sim \pi(\cdot|x)$ (stochastic). If we define

$$r^\pi(x) \triangleq \mathbb{E}\left[R | X = x\right] \tag{1.11}$$

with $A \sim \pi(\cdot|x)$, we get that

$$r^\pi = V^\pi.$$

Of course, this equality is only true for $T = 1$. For $T > 1$, V^π captures the long-term (discounted) average of the rewards, instead of the expected immediate reward captures by r^π.

For $T = 1$, finding the optimal policy given $r(x, a)$ is easy because we can simply find the maximizing action, as in (1.5).[9] Finding the optimal policy given V^π may seem less straightforward. We need to search over the space of all deterministic or stochastic policies. For example, if we denote the space of all stochastic policies by

$$\Pi = \left\{ \pi \,:\, \pi(\cdot|x) \in \mathcal{M}(\mathcal{A}), \forall x \in \mathcal{X} \right\}, \tag{1.12}$$

we need to find

$$\pi^* \leftarrow \operatorname*{argmax}_{\pi \in \Pi} V^\pi.$$

If we find such a π^*, it is an optimal policy. But how can we solve this optimizatoin problem when the search is over the large policy space (1.12)?

It turns out that this problem is not too difficult when $T = 1$. As the values of V^π at two different states $x_1, x_2 \in \mathcal{X}$ do not have any interaction with each other, we find the optimal policy at each state separately. Note that for each $x \in \mathcal{X}$,

$$V^\pi(x) = \int \mathcal{R}(\mathrm{d}r|x, a)\pi(\mathrm{d}a|x) = \int \pi(\mathrm{d}a|x) \int \mathcal{R}(\mathrm{d}r|x, a) = \int \pi(\mathrm{d}a|x) r(x, a).$$

Find a $\pi(\cdot|x)$ that maximizes $V^\pi(x)$ means that

$$\sup_{\pi(\cdot|x) \in \mathcal{M}(\mathcal{A})} \int \pi(\mathrm{d}a|x) r(x, a). \tag{1.13}$$

[9] Assuming that finding the maximizer is easy. For a finite (and small) action space, it is. But for a general action spaces, it is not.

The maximizing distribution concentrates all its mass at the action a^* that maximizes $r(x, a)$, assuming it exists.[e] Therefore,

$$\pi^*(a|x) = \delta(a - \operatorname*{argmax}_{a' \in \mathcal{A}} r(x, a')),$$

or equivalently,

$$\pi^*(x) = \operatorname*{argmax}_{a \in \mathcal{A}} r(x, a),$$

is an optimal policy at state x.

When $T > 1$, this argument does not hold anymore and finding the optimal policy is more difficult. The reason is that the choice of action at each time step affects the future states, so we have to be careful in choosing the policy. We spend a great deal of time on algorithms to solving this problem (though not for the finite horizon problems, but for another type that we shall introduce next).

Remark 1.2. *In this section, we described the finite horizon task when the action is selected by a stationary policy: $A_t = \pi(X_t)$ or $A_t \sim \pi(\cdot|X_t)$. This is only for simplicity of exposition. More generally, the policy can be non-stationary, so $\bar{\pi} = (\pi_1, \dots, \pi_T)$ (Definition 1.2). The definition of the value function (1.8) would be the same, with the understanding that the selected action at each time step t comes from policy π_t. We may occasionally use $V^{\bar{\pi}}$ to emphasize that we are talking about a non-stationary policy. We also occasionally use $\pi_1 : \pi_T$ or $\pi_{1:T}$ to refer to the policy sequence $(\pi_1, \cdots, \pi_T)$ and $V^{\pi_1:\pi_T}$ or $V^{\pi_{1:T}}$ to its corresponding policy.*

1.3.2 Episodic Tasks

In some scenarios, there is a time T that the episode ends (or terminates), but it is not fixed a priori. For example, think of playing of a board game such as chess (it ends whenever one side checkmates the other or they reach a draw), moving through a maze (it ends whenever the agent reaches a goal state), or a robot successfully picks up an object and places it in another location. For these problems, the episode terminates whenever the agent reaches a certain state x_{terminal} within the state space, that is, it terminates whenever $X_T = x_{\text{terminal}}$.[10] These are called *episodic* tasks. In episodic problems, the length of the episode T is a random variable. We define the

episodic tasks

[10] It might be more intuitive to think about the terminal states $\mathcal{X}_{\text{terminal}}$, instead of a singular one. Mathematically, it does not matter.

return and the value function as before: For $0 \leq \gamma \leq 1$, we have

$$G^\pi \triangleq \sum_{t=1}^{T} \gamma^{t-1} R_t, \tag{1.14}$$

$$V^\pi(x) \triangleq \mathbb{E}\left[G^\pi | X_1 = x\right]. \tag{1.15}$$

If $\gamma < 1$, these definitions are always well-defined. If $\gamma = 1$, we need to ensure that the termination time T is finite. Otherwise, the summation might be divergent (think of the case that that all R_t are equal to 1). We do not get into analysis of episodic problem with $\gamma = 1$, so we do not get into more detail here anymore. Refer to Section 2.2 (Stochastic Shortest Path Problems) by Bertsekas and Tsitsiklis [1996].

Exercise 1.7. *Describe several real-world applications that are best modelled as an episodic task.*

Exercise 1.8. *Suppose that we want to solve a goal reaching task, as an episodic task with the choice of $\gamma = 1$. We formulate it in two different ways. In the first way, we set the immediate reward -1 whenever we are not at the goal, and 0 whenever we reach the goal, which is the terminal state too. This encourages the agent to get to the goal sooner. In the second way, we add a constant reward $+1$ to the reward function. So, at all states, the reward is 0, except at the goal/terminal state where it is $+1$. This does not encourage getting to the goal state faster, as a later goal reaching has the same value as an earlier one. This sounds like a contradiction. A constant change in the reward function should not change the optimal policy, yet the intuitive argument here indicates that it does. What is happening?*

1.3.3 Continuing Tasks

Sometimes the interaction between the agent and its environment does not break into episodes that terminates. It goes on continually forever. For example, this might be the case for a life-long robot or a chemical plant that is supposed to work for a long time. Of course, nothing in real world lasts forever, even the livable universe itself, so the mathematical framework on continuing tasks is an abstract idealization of tasks that may take a long time.

Consider the sequence of rewards $(R_1, R_2, \dots)$ generated after the agent starts at state $X_1 = x$ and follows policy π. Given the discount factor $0 \leq \gamma < 1$, we define

the return from time τ forward as

$$G_\tau^\pi \triangleq \sum_{t \geq \tau} \gamma^{t-\tau} R_t. \tag{1.16}$$

We can also define two value functions. One of them is V^π and similar to what we have seen so far. We call it state-value function or simply value function. Another one is called the action-value function. Since these are the value functions that we will use for the rest of the book, we define them formally.

action-value function

Definition 1.5 (Value Functions)**.** *The (state-)value function V^π and the action-value function Q^π for a policy π are defined as follows: Let $(R_t; t \geq 1)$ be the sequence of rewards when the process is started from a state X_1 (or (X_1, A_1) for the action-value function) drawn from a positive probability distribution over $\mathcal{X}$ (or $\mathcal{X} \times \mathcal{A}$) and follows the policy π for $t \geq 1$ (or $t \geq 2$ for the action-value function). Then,*

$$\begin{aligned} V^\pi(x) &\triangleq \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t | X_1 = x\right], \\ Q^\pi(x,a) &\triangleq \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t | X_1 = x, A_1 = a\right]. \end{aligned}$$

In words, the value function V^π evaluated at state x is the expected discounted return of following the policy π from state x. The other value function is called the action-value function, and is very useful in our further developments. The action-value Q^π function evaluated at (x, a) is the expected discounted return when the agent starts at state x, takes action a, and then follows policy π. Note than

$$\begin{aligned} V^\pi(x) &= \mathbb{E}\left[G_1^\pi | X = x\right], \\ Q^\pi &= \mathbb{E}\left[G_1^\pi | X = x, A = a\right], \end{aligned} \tag{1.17}$$

by definition.

The action-value function Q^π and value function V^π are closely related. The difference is that the first action A_1 in V^π is selected according to $\pi(\cdot|X_1)$, but the first action in $Q^\pi(x, a)$ is the pre-specified action a. So

$$V^\pi(x) = \mathbb{E}\left[Q^\pi(x, A)\right] = \int \pi(\mathrm{d}a|x) Q^\pi(x, a). \tag{1.18}$$

If $\gamma = 0$, $Q^\pi = \mathbb{E}\left[R_1 | X_1 = x, A_1 = a\right]$. This is the same as the expected immediate reward $r(x, a)$. The same way that we could easily compute the optimal action using

$r(x, a)$ in the finite-horizon problem with $T = 1$, we shall see that we can use the action-value function of the optimal policy, which shall be defined soon in Section 1.4, in order to easily compute the optimal policy.

Note that an episodic task can be seen as a continuing task with a special state x_{terminal} from which the agent cannot escape and it always gets a reward of zero, i.e.,

$$\mathcal{P}(x_{\text{terminal}}|x_{\text{terminal}}, a) = 1, \qquad \forall a \in \mathcal{A}$$
$$\mathcal{R}(r|x_{\text{terminal}}, a) = \delta(r), \qquad \forall a \in \mathcal{A}.$$

Exercise 1.9. *Describe several real-world applications that are best modelled as a continuing task.*

Exercise 1.10. *Consider the MDP in Exercise 1.1. Compute the value function V^π for the discount factors $\gamma = \{0, 0.5, 0.9\}$.*

Exercise 1.11. *Consider a set of states $x_1, x_2, \ldots, x_N$. The agent has two actions a_{Left} and a_{Right}. Whenever the agent chooses action a_{Right} at state x_i, it goes to state x_{i+1}, unless $i = N$, in which case it stays there. Similarly for a_{Left}, except that it moves to x_{i-1}, unless $i = 1$, in which case it stays there. The reward function is 0 everywhere except at state x_N, in which it is $r(x_N) = +1$, and x_1, in which it is $r(x_1) = -1$.*

- *What are $\mathcal{P}(\cdot|\cdot; a_{\mathsf{Left}})$ and $\mathcal{P}(\cdot|\cdot; a_{\mathsf{Right}})$? The answers should be an $N \times N$ matrix.*
- *Consider π_{Left}, which always chooses action a_{Left}. What are V^π_{Left} and Q^π_{Left}?*
- *Answer the previous question for π_{Right}.*
- *Consider policy π_{Uniform}, which at each state, chooses each action with the same probability of $\frac{1}{2}$. What are V^π_{Uniform} and Q^π_{Uniform}?*

1.4 Optimal Policy and Optimal Value Function

What does it mean for an agent to act optimally? To start thinking about this question, let us first think about how we can compare two policies π and π'. For the moment, we can assume that they are Markov stationary policies, so the action selection is based on $A_t \sim \pi(\cdot|X_t)$, and not, for example, $A_t \sim \pi(\cdot|X_t, X_{t-1}, X_{t-2}, \ldots)$ or $A_t \sim \pi_t(\cdot|X_t)$. We say that π is better than or equal to π' (i.e., $\pi \geq \pi'$) iff $V^\pi(x) \geq V^{\pi'}(x)$ for all states $x \in \mathcal{X}$.[11] This is shown in Figure 1.2. We also use a

[11] This is a partial order relationship. It is possible that for two policies, none of them is better than the other one.

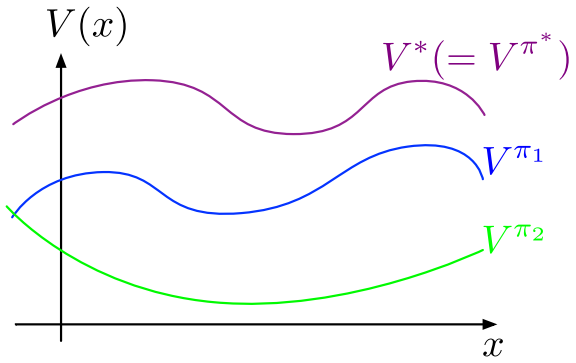

Figure 1.2: For any policy π, we have that $V^{\pi^*} \geq V^{\pi}$. Here the values V^{π_1} and V^{π_2} of two sub-optimal policies π_1 and π_2 are shown.

strict inequality $\pi > \pi'$ if $V^{\pi}(x) \geq V^{\pi'}(x)$ for all states $x \in \mathcal{X}$ and there exists at least a single state $x' \in \mathcal{X}$ such that the inequality is strict, that is $V^{\pi}(x') > V^{\pi'}(x')$.

If we can find a policy π^* that satisfies $\pi^* \geq \pi$ for any π, we call it an *optimal policy*. There may be more than one optimal policy. Despite that, their values should be the same, i.e., if we have two different π_1^* and π_2^*, we have $V^{\pi_1^*}(x) \geq V^{\pi_2^*}(x)$ and $V^{\pi_1^*}(x) \leq V^{\pi_2^*}(x)$ for all $x \in \mathcal{X}$, which entails that $V^{\pi_1^*} = V^{\pi_2^*}$.

If we denote Π as the space of all stationary Markov polices, the goal of finding an optimal policy can be written down as the following optimization problem:

$$\pi^* \leftarrow \operatorname*{argmax}_{\pi \in \Pi} V^{\pi}, \tag{1.19}$$

where one of the maximizers is selected in an arbitrary way. The value function of this policy is the called the *optimal value function*, and is denoted by V^{π^*} or simply V^*. We can also define the optimal policy based on Q^{π}, i.e.,

$$\pi^* \leftarrow \operatorname*{argmax}_{\pi \in \Pi} Q^{\pi}. \tag{1.20}$$

The optimal action-value function is denoted by Q^{π^*} or Q^*.

For the immediate reward maximization problem (or equivalently, when $T = 1$ for a finite horizon problem), the solution was easy to find, see (1.5) and (1.13). It is not obvious, however, that such a policy exists for the continuing discounted tasks. It might be the case that no single policy can dominate (that is, being better than) all others for all states. For example, it is imaginable that at best we can only hope to find a π^* that is better than any other policy π only on a proper subset of $\mathcal{X}$, which perhaps depends on π, but not at all states in $\mathcal{X}$.

It is also not obvious why we should focus on stationary policies. Isn't it possible to have a policy $\bar{\pi} = \{\pi_1, \pi_2, \ldots\}$ that depends on the time step and acts better than any stationary policy $\bar{\pi} = \{\pi, \pi, \ldots\}$?

Even if we find satisfactory answers to these questions, a more pragmatic question remains: Suppose that we know the MDP, which means that we know $\mathcal{P}$ and $\mathcal{R}$? How can we can compute π^*?

And even more interesting question is how we can *learn* π^*, or a close approximation thereof, without actually knowing the MDP, but only by using samples coming from the interaction of the agent with the MDP. This is the RL problem.

And even more interesting is the question of how we can *learn* π^*, or a close approximation thereof, without actually knowing the MDP, but only have samples coming from interacting with the MDP.

We study the question about the existence and properties of the optimal policy in Chapter **??**. The short answer is that for continuing discounted problems, the optimal policy is indeed a stationary Markov policy. Moreover, we can always find a deterministic optimal policy too.

Chapter **??** introduces several methods for computing the optimal policy given a known model $\mathcal{P}$ and $\mathcal{R}$. We study some of their properties, and prove their convergence to the optimal policy. We call the setting when the model is known as the *planning* setting, and the corresponding methods are called *Planning algorithms*.

planning

When we do not know $\mathcal{P}$ or $\mathcal{R}$, we are in the *reinforcement learning* setting. In that setting, we do not have a direct access to the model, but instead we can only interact with the MDP by selecting action A_t at state X_t, and getting a reward $R_t \sim \mathcal{R}(\cdot|X_t, A_t)$ and going to the next state X_{t+1} according to the transition probability kernel. It turns out that many of the planning algorithms can be modified to become a learning algorithm. Therefore, it is good to get a good grasp of planning algorithms first instead of delving into RL from the beginning. We introduce and analyze some methods for solving RL problems in Chapter **??**. The focus of that chapter is on the RL problems with finite state and action spaces. We turn to problems with large state and action spaces (e.g., when $\mathcal{X}$ is a subset of $\mathbb{R}^d$) in Chapter **??**. [12]

1.5 An Instance of an RL Algorithm: Q-Learning

It takes a while before we get into the detail of any RL algorithm, so it is good to see an example of such an algorithm before starting our excursion into the properties of

[12] The detail of chapter information will be determined later.

Algorithm 1.1 Q-Learning (Simplified)

Require: Step size update rule $\alpha \in (0, 1]$

1: Initialize $Q : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ arbitrary, except that for x_{terminal}, set $Q(x_{\text{terminal}}, \cdot) = 0$.
2: **for** each episode **do**
3: Initialize $X_1 \sim \rho$
4: **for** each step t of episode **do**
5: $A_t \sim \pi(\cdot|X_t)$, ▷ Action selection
6: Take action A_t, observe X_{t+1} and R_t ▷ The environment chooses $X_{t+1} \sim \mathcal{P}(\cdot|X_t, A_t)$ and $R_t \sim \mathcal{R}(\cdot|X_t, A_t)$.
7: $Q(X_t, A_t) \leftarrow Q(X_t, A_t) + \alpha \left[R_t + \gamma \max_{a' \in \mathcal{A}} Q(X_{t+1}, a') - Q(X_t, A_t)\right]$. ▷ Q-Learning Update Rule
8: **end for**
9: **end for**

MDPs (Chapter **??**) and the planning methods (Chapter **??**) until we finally get to RL algorithms in Chapter **??**.

Q-Learning (Algorithm 1.1) is the quintessential RL algorithm, introduced by Christopher Watkins [Watkins, 1989, Chapter 7 – Primitive Learning]. Q-Learning itself is an example of the Temporal Difference (TD) learning [Sutton, 1988].

The choice of policy π in Line 1.1.5 is not specified. The Q-Learning algorithm can work with variety of choices for π. A common choice is to use the ε-greedy policy. The ε-greedy policy $\pi_\varepsilon(Q)$ for an $0 \leq \varepsilon \leq 1$ chooses the action as follows: Given the current estimate of the action-value function Q, it chooses the action that maximizes the action-value function at the current state X_t with probability $1 - \varepsilon$, and chooses a (possibly uniformly) random action with probability ε. Mathematically,

$$A_t = \begin{cases} \text{argmax}_{a \in \mathcal{A}} Q(X_t, a) & \text{w.p. } 1 - \varepsilon \\ \text{uniform}(\mathcal{A}) & \text{w.p. } \varepsilon \end{cases} \tag{1.21}$$

Usually the value of ε is small and may go to zero as the agent learns more about its environment. This occasional random choice of actions ensures that the agent explores its environment. Studying exploration is the subject of Chapter **??**.

The update rule for Q-Learning is Line 1.1.7. We notice that it does not directly use the model $\mathcal{P}$ or $\mathcal{R}$, but uses the tuple (X_t, A_t, R_t, X_{t+1}) in order to update the action-value function Q.

Under certain conditions, the Q-Learning algorithm is guaranteed to converge to

the optimal action-value function Q^*.[13] As we shall see later, we can use Q^* to find the optimal policy π^*. We shall try to understand why this is the case in the next few chapters.

Exercise 1.12 (Programming)**.** *Implement the Q-Learning algorithm (Algorithm 1.1), and try it for the deterministic MDP described in Exercise 1.11. As that is a continuing task, the episode does not end. You can let the algorithm run for different number of steps, for example,* 1000*,* 10000*, and* 100000*. Answer the following questions for each different number of steps.*

- *Plot* $Q(\cdot, a_{\mathsf{Left}})$ *and* $Q(\cdot, a_{\mathsf{Right}})$*.*
- *For which states* $Q(\cdot, a_{\mathsf{Left}})$ *is larger than* $Q(\cdot, a_{\mathsf{Right}})$*? How do you interpret it?*

1.6 A Few Remarks on the MDP Assumption

Before finishing this chapter, we have several remarks about the MDP assumption. Specifically, we ask the following questions:

- What is the state variable?
- Where does the reward signal come from?

Although most of our focus in this book will be on methods to design an RL agent, assuming that the MDP is given to us, thinking about these questions is nonetheless important for any RL practitioner and researcher.[14]

1.6.1 On State

What is a state variable?

Perhaps the most crucial remark on the MDP assumption is the definition of state. What is a state? Is any variable that the agent observes a state? The way we use the state here is that the state of the agent at time t is a variable that summarizes whatever has happened to the agent up to that time step, that is, its *history*. Knowing the state is enough to know (probabilistically) what will happen to the agent in the future. In other words, the state is a *sufficient statistic* of the history.

[13] For convergence of the Q-Learning algorithm, the step size should gradually converge to zero. We skip these detail here.

[14] And perhaps we expand on these in a future edition of this book.

To make this more clear, let us introduce another concept called *observation*. An observation O_t is the variable that the agent actually observes using its various sensors. For example, it might be the camera input for the robot agent, or the temperature and blood pressure for the medical agent. The observation alone may not be sufficient to know "everything" that we could know about the agent given the information so far. For example, by only having an access to the current camera image, we do not know whether the robot is moving forward or backward or something is getting close or far from it (as the velocity information cannot be inferred from the position information alone). Or as another example, if the agent can only observe the blood pressure and heart rate at the moment, we cannot know everything that could be known about the patient, for example, whether the heart rate and blood pressure is suddenly spiking up or they have been up for a long time. The information might be there, if we looked at the previous observations.

Whatever has happened to the agent up to time t is in its history H_t variable **history**

$$H_t = (O_1, A_1, R_1, \ldots, O_{t-1}, A_{t-1}, R_{t-1}, O_t).$$

The history H_t summarizes whatever has happened to the agent up to time t. Given H_t, we can inquire about the probability distribution

$$\mathbb{P}\{O_{t+1}|H_t, A_t\}.$$

This is all we can hope to know about the future, given the information that we have. Now, if we do not look at H_t, but only look at the current observation O_t, we can still form $\mathbb{P}\{O_{t+1}|O_t, A_t\}$, but it has more "uncertainty" about the probability of O_{t+1}. We are losing information by not looking at H_t.

The variable H_t is a state of the agent at time t. But it is not a compact one, as its size gradually increases.[15] If it happens that we can find another variable X_t, which is a function of H_t but perhaps of a compact form, that satisfies

$$\mathbb{P}\{O_{t+1}|H_t, A_t\} = \mathbb{P}\{O_{t+1}|X_t, A_t\},$$

we can replace H_t with X_t. This X_t is the state of the system in the sense described above. In the rest of the book, we assume that the agent has access to such a state variable.

Finding such a summary is not always complicated. Consider a dynamical system described by equation

$$z_{t+1} = f(z_t, a_t),$$

[15] We do not use "compact" in the formal sense used in topology, but in an informal sense meaning being concise.

where $z \in \mathbb{R}^m$, $a \in \mathbb{R}^n$, and $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$. This is the same dynamics we encountered before (1.1).

Suppose that the observation is $o_t = z_t$. In this case, we do not need to keep $h_t = (z_1, a_1, \ldots, z_{t-1}, a_{t-1}, z_t)$ as a state of the system; the observation o_t alone is enough to know whatever has happened to the system up to time t. We can disregard $z_{t-1}, a_{t-1}, z_{t-2}$, etc. Now suppose that the observation is $o_t = g(z_t)$ with $g : \mathbb{R}^m \to \mathbb{R}^d$. In this case, depending on the function g, the observation o_t may or may not be a state. For example, if g is not a bijection (one-to-one correspondence), it is likely that we lose o having a property of being a state. However it may be possible that we can still process h_t and find a compact representation x_t that is a state of the agent.

Most (all?) physical systems can be written by an equation similar to (1.1).[16] If we have such a description of the dynamics, the state is often clear as long as we observe the right variable.

Exercise 1.13. *A ball is free falling under the Earth's gravity. The state is the vector described by its location $x(t)$ and velocity $v(t) = \dot{x}(t)$. If we only observe $x(t)$, that is not enough to know the (physical) state of the ball. How can you estimate the state using only the location information?*

Exercise 1.14. *In Atari games, a single frame is not a state of the agent. Explain why.*

Exercise 1.15. *We just claimed that $\mathbb{P}\{O_{t+1}|O_t, A_t\}$ has more uncertainty than $\mathbb{P}\{O_{t+1}|H_t, A_t\}$.*

- *Formalize this claim. (Hint: Take a look at Appendix A.7.)*
- *Prove it.*

1.6.2 On Reward

Where does reward come from?

How do we determine the reward signal R_t? The reward signal encodes the desire of the problem designer, so it is a part of the problem specification. Therefore, how to come up with a reward signal is a separate question from how to solve the planning or learning problem. In practice, however, when an agent designer wants to design the whole system, they have to both choose the learning algorithm as well as design a

[16] To be more accurate, almost all physical systems are written in the form of a differential equation, so we have $\frac{\mathrm{d}z}{\mathrm{d}t}(t) = f(z_t, a_t)$ instead. But this is not a crucial difference here.

good reward signal to specify the task. Depending on the task, designing the reward signal can be easy or difficult. Let us briefly comment on it.

In some tasks, the reward function is relatively easy to define. For example, in control engineering, one is often interested in ensuring that the state of a system, described by a dynamical system such as (1.1), reaches a predefined state as quick as possible. For example, if the desired state is state $x = 0$, the reward might be defined as

$$R_t = - \|X_t\|_2^2 .$$

This, however, ignores the cost associated with the choice of actions. To incorporate that cost, we can define the reward as

$$R_t = - \left[\|X_t\|_2^2 + c_a \|A_t\|_2^2 \right] ,$$

for some $c_a > 0$. This is known as the quadratic cost model in control engineering.

For some other problems, the reward might be defined at a higher and more abstract level. For instance, in the robot manipulator in an automobile factor example mentioned earlier in this chapter, the reward might be defined as successfully building a car according to the desired specifications. Whenever such a car is successfully built, the agent receives a reward of $+1$, and at other times, it receives a reward of 0. This is a valid definition for the reward, but since the reward is extremely *sparse*, in the sense that the agent does not receive a non-zero reward very often, the RL agent might have difficulty learning how to the problem. To see this, when the agent has just begun learning, the chance of successfully solving the task, and receiving a non-zero reward, is very slim. When the agent only receives zero reward, it cannot learn much. More concretely, consider what happens to an agent using the Q-Learning algorithm (Algorithm 1.1) when the reward R_t is equal to zero. The update rule for the action-value function would be

$$\begin{aligned} Q(X_t, A_t) \leftarrow & Q(X_t, A_t) + \alpha \left[0 + \gamma \max_{a' \in \mathcal{A}} Q(X_{t+1}, a') - Q(X_t, A_t) \right] \\ & = (1 - \alpha) Q(X_t, A_t) + \alpha \gamma \max_{a' \in \mathcal{A}} Q(X_{t+1}, a'). \end{aligned}$$

If the Q function is initialized as any constant function (zero, for example), as long as the agent has not received any non-zero R_t, the right-hand side (RHS) is equal to the same constant function – the action-value function does not change. No change in the value function shows that the agent does not learn anything.

This is an extreme example to show that the sparse reward might cause difficulty in learning, even though from the goal specification standpoint, the reward is correctly specified.

As another example, consider the smart HVAC system that optimizes the comfort of the occupants. For that problem, the reward might be directly provided by the occupant of the space, given in the form of occasional voice feedback about their level of comfort.

Sometimes we can avoid directly specifying the reward function, but instead try to infer it from other information available to the agent.

One approach is to assume that we have access to an *expert* who knows how to solve the problem, and we can observe their behaviour. The goal is then to find a reward function whose optimal policy leads to a behaviour similar to the expert's. This is called the *inverse reinforcement learning* problem.

The expert data can be in the form of observing the set of states and actions the expert has selected, or only the set of states the behaviour of the expert generated. As a concrete example, consider that the expert has a policy π_E, which is unavailable to the agent. The expert follows policy π_E in an environment with the transition dynamics $\mathcal{P}$. As a result, a sequence of data in the form of $X_1^E, A_1^E, X_2^E, A_2^E, \ldots$ is generated. Assume that the agent can only observe the states of the expert, and not its actions: $X_1^E, X_2^E, \ldots$ (observing the actions is also a possible setup of IRL). The goal of the IRL problem is to find the reward distribution $\mathcal{R}$ such that the optimal policy $\pi^*(\mathcal{P}, \mathcal{R})$ leads to a similar distribution of states as $X_1^E, X_2^E, \ldots$. Different IRL methods use different notions of similarity and take different approaches to compute this unobserved reward distribution. [f]

Instead of providing a scalar reward, we may provide the agent with our preference over its choice of actions or induced trajectory. This is called *preference-based reinforcement learning*. In one approach to this problem, we first learn a scalar reward model that conforming to the preferences, and then use a regular RL algorithm to optimize using this learned reward model instead. The preference-based RL approach in the context of training and aligning the Large Language Models (LLM) with human preferences has attracted much attention under the name of RL from Human Feedback (RLHF) in recent years.[g]

Finally, we would like to remark that in biological animals, the reward signal has not been designed, but has been evolved. Their reward mechanism has been evolved so that the chance of survival and successful reproduction increases. Other than a few exceptions, the RL community does not tend to combine evolution of rewards and a learning-based RL algorithm.[h]

Throughout this course, we assume that the reward signal is given, but we note that much research has been done in how to specify reward.

In biological systems, however, the reward signal has not been designed, but has been evolved. The reward mechanism of animals has been evolved so that the chance

of survival and successful reproduction increases. Throughout this course, we assume that the reward signal is given, but we note that much research has been done in how to specify reward. [i]

1.6.3 On Time

A modelling assumption of the MDP model is that the agent makes decision on its actions at regular time steps $1, 2, \ldots$. What does time actually mean here?

For certain problems, such as board games, the meaning of these decision times is clear: each decision time corresponds to a move of a piece on a chess board. In those problems, the notion of time is abstract and corresponds to the number of steps the game is played.

For an agent living in a physical world such as a robot, or a simulation thereof, the correspondence is usually through a discretization of the continuous flow of time. Each decision time $t = 1, 2, \ldots$ correspond to the physical time $\mathtt{t} = t\Delta t$, where Δt is the discretization resolution. For some tasks, such a control of a robot, the discretization resolution might be in the order of milliseconds, while for a slower process such as the temperature of a room in the smart HVAC problem, the discretization resolution might be in the order of minutes.

1.7 Applications of Reinforcement Learning

Chapter Summary

Summarize the main points that the reader needs to remember from this chapter.

Notes and Remarks

a Relevant citations: Farahmand et al. [2016, 2017]; Pan et al. [2018]. More by others?

b The viewpoint of an animal as the nexus of causal pathways is from Chapter 5 (The Origin of Subjects) of Godfrey-Smith [2020]. I include it because I believe it is a good metaphor on what an intelligent being is, no matter biological or artificial. Of course, it is not a mathematically rigorous definition of what an animal is, and should not be interpreted as such.

c Designing a reward function that reflects our intended goals clearly is not always easy. This is the problem of agent alignment and is a subject of active research in the AI safety community. . If the animal is evolved, however, the problem of designing a proper reward function is taken care of by the evolutionary process: if the reward function does not lead to successful reproduction, the genes that encode that animal, and its reward function, will not survive for long.

d The original Stanford Marshmallow test is conducted by Mischel et al. [1972]. It appears that the discounting model of humans is better modelled by a hyperbolic model, instead of the geometric one. In that model, the influence of reward received after time t is $\frac{R_t}{1+\lambda t}$, instead of $\gamma^t R_t$, which we consider here. It also seems that humans have a relatively stable discounting parameters [Kirby, 2009]. The neural correlates of this discounting has also been studied, for example, by Casey et al. [2011].

e The assumption that the maximizer exists is technical, and may appear when the maximizing action does not belong to the action set. As an example, suppose that the state space is $\mathcal{X} = \mathbb{R}$ and the action set is $\mathcal{A} = (-1, +1)$, an open set. If the reward function is $r(x, a) = -(x - a)^2$, the maximizing action depends on whether $x \in (-1, +1)$ or outside it. If it is inside, the optimal action is to choose a being equal to x. But when x is outside that set, the optimal action is to be as close as $+1$ (for $x > 1$) or as close as -1 (for $x < -1$). But since the maximizing action cannot be realized within the action set, we can choose an action that is arbitrary close to $+1$ or -1.

f Some relevant papers for inverse RL are Russell [1998]; Ng et al. [2000]; Ramachandran and Amir [2007]; Ziebart et al. [2013]; Huang et al. [2015].

g Wirth et al. [2017] survey preference-based RL just before the recent surge of interest in RL from Human Feedback (RLHF). Some related papers directly related to RLHF are Wirth et al. [2017]; Christiano et al. [2017]; Ouyang et al. [2022]; Gheshlaghi Azar et al. [2024].

h What are good papers for this? Singh et al. has "Where Do Rewards Come From?", which seems relevant. And also "Intrinsically Motivated Reinforcement Learning: An Evolutionary Perspective" from Singh et al. I did some research about it during my MS (2003 or 2004), but unfortunately I didn't publish.

i There are several approaches to define or design a reward signal. In one approach, we assume that we have access to an expert who knows how to solve the problem, and we can observe their behaviour. The goal is then to find a reward function whose optimal policy leads to a behaviour similar to the expert's. This is called *inverse reinforcement learning* problem [Russell, 1998; Ng et al., 2000; Ziebart et al., 2013; Huang et al., 2015]. An active area of research is to convert the preference of For example, one may try to convert the preference of humans into reward function. , for example, in order to capture human We should mention that there has been work in evolving the reward signal itself.

Appendix A

Mathematical Background

A.1 Probability Space

For a space Ω, with σ-algebra σ_Ω, we define $\mathcal{M}(\Omega)$ as the set of all probability measures over σ_Ω. Further, we let $\mathcal{B}(\Omega)$ denote the space of bounded measurable functions w.r.t. (with respect to) σ_Ω and we denote $\mathcal{B}(\Omega, L)$ as the space of bounded measurable functions with bound $0 < L < \infty$.

We write $\nu_1 \ll \nu_2$ if $\nu_2(A) = 0$ implies that $\nu_1(A) = 0$ as well. For two σ-finite measures ν_1 and ν_2 on some measurable space (Ω, σ_Ω), ν_1 is *absolutely continuous* w.r.t. ν_2 if there is a non-negative measurable function $f : \Omega \to \mathbb{R}$ such that $\mu_1(A) = \int f d\nu_2$ for all $A \in \sigma_\Omega$. It is known that ν_1 is absolutely continuous w.r.t. ν_2 if and only if $\nu_1 \ll \nu_2$. We write $\frac{d\nu_1}{d\nu_2} = f$ and call it the *Radon-Nikodym* derivative of ν_1 w.r.t. ν_2 [Rosenthal, 2006, Chapter 12].

A.2 Norms and Function Spaces

We use $\mathcal{F} : \mathcal{X} \to \mathbb{R}$ to denote a subset of measurable functions.[1] The exact specification of this space should be clear from the context. We usually denote $\mathcal{F}$ as the space of value functions.

For a probability distribution $\nu \in \mathcal{M}(\mathcal{X})$, and a measurable function $V \in \mathcal{F}$, we define the $L_p(\nu)$-norm of V with $1 \leq p < \infty$ as

$$\|V\|_{p,\nu}^p \triangleq \int_{\mathcal{X}} |V(x)|^p d\nu(x). \tag{A.1}$$

[1] This section is quoted almost verbatim from Section 2.1 of Farahmand [2011].

When $p = 2$, this is the norm induced by the inner product $\langle \cdot\,, \cdot \rangle : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$: For any V_1, V_2, we have

$$\langle V_1\,,\, V_2 \rangle_\nu = \int_{\mathcal{X}} V_1(x) V_2(x) \mathrm{d}\nu(x). \tag{A.2}$$

It is clear that $\|V\|_{2,\nu}^2 = \langle V\,,\, V \rangle_\nu$.

The $L_\infty(\mathcal{X})$-norm is defined as

$$\|V\|_\infty \triangleq \sup_{x \in \mathcal{X}} |V(x)|. \tag{A.3}$$

If we want to emphasize that the probability distribution is defined on the state space $\mathcal{X}$, we use $\nu_{\mathcal{X}}$ and $\|V\|_{p,\nu_{\mathcal{X}}}$.

We define $\mathcal{F}^{|\mathcal{A}|} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{|\mathcal{A}|}$ as a subset of vector-valued measurable functions with the following identification:

$$\mathcal{F}^{|\mathcal{A}|} = \left\{ (Q_1, \ldots, Q_{|\mathcal{A}|}) \,:\, Q_i \in \mathcal{F},\ i = 1, \ldots, |\mathcal{A}| \right\}.$$

We use $Q_j(x) = Q(x, j)\,(j = 1, \ldots, |\mathcal{A}|)$ to refer to the j^{th} component of $Q \in \mathcal{F}^{|\mathcal{A}|}$. We often denote $\mathcal{F}^{|\mathcal{A}|}$ as a space of action-value functions. If there is no chance of ambiguity, we may use $\mathcal{F} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{|\mathcal{A}|}$ for a space of action-value functions though.

Let $z_{1:n}$ denote the $\mathcal{Z}$-valued sequence $(z_1, \ldots, z_n)$. We define the empirical measure as the measure that assigns the following probability to any (measurable) set $B \subset \mathcal{Z}$:

$$\nu_n(B) \triangleq \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{Z_i \in B\}.$$

The empirical norm of function $f : \mathcal{Z} \to \mathbb{R}$ is then

$$\|f\|_{p,z_{1:n}}^p = \|f\|_{p,\mathcal{D}_n}^p \triangleq \|f\|_{p,\nu_n}^p = \frac{1}{n} \sum_{i=1}^{n} |f(z_i)|^p. \tag{A.4}$$

When there is no chance of confusion about $\mathcal{D}_n$, we may simply use $\|f\|_{p,n}^p$. Based on this definition, one may define $\|V\|_n$ (with $\mathcal{Z} = \mathcal{X}$) and $\|Q\|_n$ (with $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$).

If $\mathcal{D}_n = Z_{1:n}$ is random with $Z_i \sim \nu$, the empirical norm is random as well. For any fixed function f, we have $\mathbb{E}\left[\|f\|_{p,n}\right] = \|f\|_{p,\nu}$.

We sometimes use the shorthand notation of $\nu|Q|^p = \|Q\|_{p,\nu}^p$ (similar for $\nu_{\mathcal{X}}$ and other probability distributions). In this book, most results are stated for $p = 1$ or $p = 2$. The symbols $\|\cdot\|_\nu$ and $\|\cdot\|_n$ refers to an L_2-norm.

Finally, define the projection operator $\Pi_{\mathcal{F}^{|\mathcal{A}|},\nu} : B(\mathcal{X} \times \mathcal{A}) \to B(\mathcal{X} \times \mathcal{A})$ as

$$\Pi_{\mathcal{F},\nu} Q \triangleq \operatorname*{argmin}_{Q' \in \mathcal{F}^{|\mathcal{A}|}} \|Q' - Q\|_\nu^2$$

for $Q \in B(\mathcal{X} \times \mathcal{A})$. The definition of $\Pi_{\mathcal{F},\nu_{\mathcal{X}}} : B(\mathcal{X}) \to B(\mathcal{X})$ is similar. If the distribution $\nu_{\mathcal{X}}$ or ν are clear from the context, we may simply write $\Pi_{\mathcal{F}}$ and $\Pi_{\mathcal{F}^{|\mathcal{A}|}}$ instead.

A.3 Functional Analysis: Spaces, Operators, and Contraction Mapping

The contraction mapping (or operator) is a mapping that maps points (i.e., vectors, functions) closer to each other. As the Bellman operators for discounted tasks are contraction mapping, it is useful to have a good understanding on what such a mapping is, and what their properties are. We briefly review some basic concepts from functional analysis. Our discussion here freely borrows from Hunter and Nachtergaele [2001].

First, let us recall the definition of a *metric space*. Let $\mathcal{Z}$ be an arbitrary non-empty set.

Definition A.1 (Metric – Definition 1.1 of Hunter and Nachtergaele 2001)**.** *A metric or a distance function on $\mathcal{Z}$ is a function $d : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ with the following properties:*

- *$d(x, y) \geq 0$ for all $x, y \in \mathcal{Z}$; and $d(x, y) = 0$ if and only if $x = y$.*
- *$d(x, y) = d(y, x)$ for all $x, y \in \mathcal{Z}$ (symmetry).*
- *$d(x, y) \leq d(x, z) + d(z, y)$ for all $x, y, z \in \mathcal{Z}$ (triangle inequality).*

Given a metric, we can define a metric space.

Definition A.2 (Metric Space)**.** *A metric space $(\mathcal{Z}, d)$ is a set $\mathcal{Z}$ equipped with a metric d.*

Example A.1. *Let $\mathcal{Z} = \mathbb{R}$ and $d(x, y) = |x - y|$. These together define a metric space $(\mathbb{R}, d)$.*

Example A.2. *Let $\mathcal{Z}$ be a discrete set and define*

$$d(x,y) = \begin{cases} 0 & x = y, \\ 1 & x \neq y. \end{cases}$$

We often work with *linear vector spaces* and *norms* in this book. Let us define them.

Definition A.3 (Linear Space – Definition 1.7 of Hunter and Nachtergaele 2001)**.** *A linear space $\mathcal{Z}$ over the scalar field $\mathbb{R}$ (or $\mathbb{C}$) is a set of points (or vectors), on which operations of vector additions and scalar multiplications with the following properties are defined:*

(a) The set $\mathcal{Z}$ is a commutative group with the operation of $+$ of vector addition, that is,

- $x + y = y + x$.
- $x + (y + z) = (x + y) + z$
- *There exists an element $0 \in \mathcal{Z}$ such that for any $x \in \mathcal{Z}$, we have $x + 0 = x$.*
- *For each $x \in \mathcal{Z}$, there exists a unique vector $-x \in \mathcal{Z}$ such that $x + (-x) = 0$.*

(b) For all $x, y \in \mathcal{Z}$ and $a, b \in \mathbb{R}$ (or $\mathbb{C}$), we have

- $1.x = x$.
- $(a + b)x = ax + bx$.
- $a(bx) = (ab)x$.
- $a(x + y) = ax + by$.

Next we define a notion of the length or size of a vector.

Definition A.4 (Norm – Definition 1.8 of Hunter and Nachtergaele 2001)**.** *A norm on a linear space $\mathcal{Z}$ is a function $\|\cdot\| : \mathcal{Z} \to \mathbb{R}$ with the following properties:*

(a) (non-negative) For all $x \in \mathcal{Z}$, $\|x\| \geq 0$.

(a) (homogenous) For all $x \in \mathcal{Z}$ and $\lambda \in \mathbb{R}$ (or $\mathbb{C}$), $\|\lambda x\| = |\lambda| \, \|x\|$.

(a) (triangle inequality) For all $x, y \in \mathcal{Z}$, $\|x + y\| \leq \|x\| + \|y\|$.

(a) (strictly positive) If for a $x \in \mathcal{Z}$, we have that $\|x\| = 0$, it implies that $x = 0$.

The same way that we used a metric space given a metric, we can define a normed linear space given a norm.

Definition A.5 (Normed Linear Space)**.** *A normed linear space $(\mathcal{Z}, \|\cdot\|)$ is a linear space $\mathcal{Z}$ equipped with a norm $\|\cdot\|$.*

We can use a norm to define a distance between two points in a linear space $\mathcal{Z}$, simply by defining $d(x, y) = \|x - y\|$. This gives us a metric space $(\mathcal{Z}, d)$.

Example A.3. *Let $\mathcal{Z} = \mathbb{R}^d$ $(d \geq 1)$. The following norms are often used:*

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^{d} |x_i|^p}, \qquad 1 \leq p < \infty,$$
$$\|x\|_\infty = \max_{i=1,\dots,d} |x_i|.$$

This can be generalized to infinite sequences too.

Example A.4. *For $p \geq 1$, the sequence space ℓ_p is the set of all sequences $(x_i)_{i\geq 1}$ such that $\sum_{i\geq 1} |x_i|^p < \infty$. The norm is defined as*

$$\|x\|_p = \begin{cases} \sqrt[p]{\sum_{i=1}^{\infty} |x_i|^p}, & 1 \leq p < \infty, \\ \max_{i\geq 1} |x_i|. & p = \infty \end{cases}$$

Example A.5. *Consider the space of continuous functions with domain $[0, 1]$. It is denoted by $\mathcal{C}([0, 1])$. This plays the rule of $\mathcal{Z}$. We define the following norm for a function $f \in \mathcal{C}([0, 1])$:*

$$\|f\|_\infty = \sup_{x\in[0,1]} |f(x)|.$$

This is called the supremum *or* uniform *norm. Given this norm, $(\mathcal{C}([0, 1]), \|\cdot\|_\infty)$ would be a normed linear space. This norm is similar to $\|x\|_\infty$ with $x \in \mathbb{R}^d$ (previous example), but it is for the space of continuous functions, which is an infinite dimensional object, as opposed to for a finite dimensional vector.*

We often use the supremum norm of value functions. For $V \in \mathcal{B}(\mathcal{X})$ and $Q \in \mathcal{B}(\mathcal{X} \times \mathcal{A})$, their supremum norms are

$$\|V\|_\infty = \sup_{x\in\mathcal{X}} |V(x)|,$$
$$\|Q\|_\infty = \sup_{(x,a)\in\mathcal{X}\times\mathcal{A}} |Q(x, a)|,$$

Using this norm, we can define a supremum-norm-based distance between two value functions V_1 and V_2 as $d_\infty(V_1, V_2) = \|V_1 - V_2\|_\infty$ (and similarly for the action-value functions).

We can also define

A.3.1 Operators

Operators, or mappings, are transformations from one linear space $\mathcal{Z}$ to another linear space $\mathcal{W}$. An operator $L : \mathcal{Z} \to \mathcal{W}$ is a *linear* operator when

$$L(c_1 z_1 + c_2 z_2) = c_1 L z_1 + c_2 L z_2,$$

for all $c_1, c_2 \in \mathbb{R}$ and $z_1, z_2 \in \mathcal{Z}$.

A simple example is the operator $L : \mathbb{R} \to \mathbb{R}$ defined as $Lz = az$ with $a \in \mathbb{R}$ and $z \in \mathbb{R}$. It is a mapping from the space of real numbers to the space of real numbers. Here both $\mathcal{Z}$ and $\mathcal{W}$ are $\mathbb{R}$.

A generalization of this is $L : \mathbb{R}^m \to \mathbb{R}^n$ (with m, n being integer numbers) defined as the mapping that takes $z \in \mathcal{Z} = \mathbb{R}^m$ and maps it to $w \in \mathcal{W} = \mathbb{R}^n$ with the i-th component of w being

$$w_i = \sum_{j=1}^{m} l_{i,j} z_j,$$

for $l_{i,j} \in \mathbb{R}$ for $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$. Of course, this is the same as matrix-vector multiplication that we can succinctly write as $w = Lz$ with L being a matrix with components $l_{i,j}$. These are examples of linear operators.

Not all operators are linear though. For example, the operator $L : \mathbb{R} \to \mathbb{R}$ defined as $Lz = az + b$ (with $a, b \in \mathbb{R}$) or $Lz = z^2$ or $Lz = \sin(z)$ or $Lz = \max\{0, z\}$ are not linear operators. We call them nonlinear operators. The first one is called *affine*.

The operators are not necessarily limited to finite dimensional linear spaces, but they can also be defined over the space of functions. An example is the linear integral operators. Consider a continuous function $k : [0, 1] \times [0, 1] \to \mathbb{R}$, and define $K : C([0, 1]) \to C([0, 1])$ (cf. Example A.5) as the operator that takes a continuous function $f \in C([0, 1])$ and returns another continuous function $Kf : [0, 1] \to \mathbb{R}$, whose value at $x \in [0, 1]$, that is $(Kf)(x)$, is

$$(Kf)(x) = \int_0^1 L(x, y) f(y) \mathrm{d}y.$$

Note the similarity of Kf with $\mathcal{P}f$ (Definition 1.4) and the Bellman operators applied to a value function, such as $T^\pi V$ or $T^* V$ (Definition **??**). The Bellman operators, however, are not linear operators, but they are affine.

If the spaces $\mathcal{Z}$ and $\mathcal{W}$ are also equipped with norms, that is, we have $(\mathcal{Z}, \|\cdot\|_{\mathcal{Z}})$ and $(\mathcal{W}, \|\cdot\|_{\mathcal{W}})$, we can define a notion of operator norm based on the norm of these two spaces. First, we say that the operator L is bounded if there exists a finite $B < \infty$ such that for any $z \in \mathcal{Z}$, the norm of the output of the operator L, that is $w = Lz \in \mathcal{W}$, satisfies

$$\|Lz\|_{\mathcal{W}} \leq B \|z\|_{\mathcal{Z}} .$$

Note that as $w = Lz$ belongs to the space $\mathcal{W}$, its size should be measured with $\|\cdot\|_{\mathcal{Z}}$.

We define the *operator norm* of L as

$$\|L\| = \inf \{ B \, : \, \|Lz\|_{\mathcal{W}} \leq B \|z\|_{\mathcal{Z}} \} . \tag{A.5}$$

This is equivalent to

$$\|L\| = \sup_{z \neq 0} \frac{\|Lz\|_{\mathcal{W}}}{\|z\|_{\mathcal{Z}}} = \sup_{\|z\|_{\mathcal{Z}}=1} \|Lz\|_{\mathcal{W}} . \tag{A.6}$$

The operator norm measures the maximum any input z can be expanded after going through the operator L and creating $z = Lw$.

An immediate consequence of this definition is that for any $z \in \mathcal{Z}$, we have

$$\|Lz\|_{\mathcal{W}} \leq \|L\| \|z\|_{\mathcal{Z}} . \tag{A.7}$$

Another useful property is that for two operators $L_1 : \mathcal{Z} \to \mathcal{W}$ and $L_2 : \mathcal{W} \to \mathcal{Y}$, whose joint operation $L_2 L_1 : \mathcal{W} \to \mathcal{Y}$, their norm is sub-multiplicative:

$$\|L_2 L_1\|_{\mathcal{W} \to \mathcal{Y}} \leq \|L_1\|_{\mathcal{Z} \to \mathcal{W}} \|L_2\|_{\mathcal{W} \to \mathcal{Y}} . \tag{A.8}$$

Here we used subscripts $\mathcal{Z} \to \mathcal{W}$ and $\mathcal{W} \to \mathcal{Y}$ to emphasize that these two operator norms are defined over different spaces. A further discussion of this XXX

A.3.2 Contraction Mapping

We are ready to define the contraction mapping/operator formally.

Definition A.6 (Contraction Mapping – Definition 3.1 of Hunter and Nachtergaele 2001)**.** *Let $(\mathcal{Z}, d)$ be a metric space. A mapping $L : \mathcal{Z} \to \mathcal{Z}$ is a contraction mapping (or contraction) if there exists a constant $0 \leq a < 1$ such that for all $z_1, z_2 \in \mathcal{Z}$, we have*[2]

$$d(L(z_1), L(z_2)) \leq a d(z_1, z_2).$$

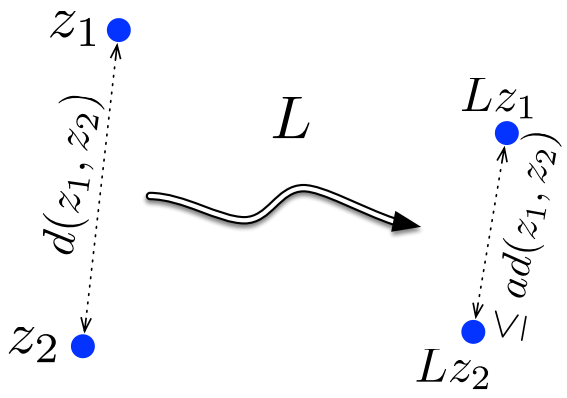

Figure A.1: Visualization of an a-contraction mapping L.

This is visualized in Figure A.1.

Example A.6. *Let $\mathcal{Z} = \mathbb{R}$ and $d(z_1, z_2) = |z_1 - z_2|$. Consider the mapping $L : z \mapsto az$ for $a \in \mathbb{R}$. We have Let us see if/when this mapping is a contraction or not.*

For any $z_1, z_2 \in \mathbb{R}$, we have

$$d(L(z_1), L(z_2)) = |L(z_1) - L(z_2)| = |az_1 - az_2| = |a||z_1 - z_2| = |a|d(z_1, z_2).$$

So if $|a| < 1$, this is a contraction mapping.

Exercise A.1 ($\star$)**.** *Consider the same $(\mathbb{R}, |\cdot|)$ as before, but let the mapping be $L : z \mapsto az + b$ for $a, b \in \mathbb{R}$. What is condition on a and b for this mapping to be a contraction.*

Exercise A.2 ($\star\star$)**.** *Consider the same $(\mathbb{R}, |\cdot|)$ as before, and let $L : z \mapsto az^2 + b$ for $a, b \in \mathbb{R}$. Is this a contraction mapping for some choice of a and b? If yes, specify a and b. If not, can you consider another space $\mathcal{Z}$ (a subset of $\mathbb{R}$) that makes this a contraction (possibly with an appropriate choice of a and b)?*

Exercise A.3 ($\star\star$)**.** *Consider $\mathcal{Z} = \mathbb{R}^d$. Given a matrix $A \in \mathbb{R}^{d\times d}$ and a vector $b \in \mathbb{R}^d$, define the mapping $L : z \mapsto Az + b$. Using the vector norm $\|\cdot\|_p$ ($1 \le p \le \infty$, define the metric $d_p(z_1, z_2) = \|z_1 - z_2\|_p$. Then, $d_p(L(z_1), L(z_2)) = \|Az_1 - Az_2\|_p$.*

What is the condition that L is a contraction? Note that this depends on the choice of p.

[2]Sometimes the condition of having $a < 1$ is called strict contraction [Berinde, 2007], and the condition that $d(L(z_1), L(z_2)) < d(z_1, z_2)$ is called contractive.

Exercise A.4 ($\star\star$)**.** *Consider* $(\mathbb{R}, |\cdot|)$*. Let* $r \geq 0$ *and define the mapping*

$$L : z \mapsto rz(1-z).$$

When is this a contraction mapping?

Why do we care about a contraction mapping? We have two reasons in mind.

The first is that we can describe the behaviour of a dynamical system depending on whether the mapping describing it is a contraction or not. To be concrete, let $z_0 \in \mathcal{Z}$ and consider a mapping $L : z \mapsto az$ for some $a \in \mathbb{R}$. Define the dynamical system

$$z_{k+1} = Lz_k, \qquad k = 0, 1, \ldots .$$

The dynamical system described by this mapping generates

$$\begin{aligned} &z_0 \\ &z_1 = az_0 \\ &z_2 = az_1 = a^2 z_0 \\ &\quad\vdots \\ &z_k = az_{k-1} = a^k z_0. \end{aligned}$$

If $|a| < 1$, z_k converges to zero, no matter what z_0 is. If $a = 1$, we have $z_k = z_0$. So depending on z_0, it converges to different points. For $a = -1$, the sequence would oscillate between $+z_0$ and $-z_0$. And if $|a| > 1$, the sequence diverges (unless $z_0 = 0$).

An interesting observation is that the case of converge is the same as the case of L being a contraction map (see Example A.6). This is not an isolated example, as we shall see. A dynamical system defined based on a contraction mapping converges. We call such a system *stable*.[3]

The second reason we care about contraction is that we can sometimes use it to solve equations. We can convert an equation that we want to solve (think of solving $f(z) = 0$) as the fixed point equation, as we shall see. If it happens that the underlying mapping is contraction, we can define an algorithm based on a dynamical system in order to solve the equation. Let us make this idea more concrete.

Definition A.7 (Fixed Point)**.** *If* $L : \mathcal{Z} \to \mathcal{Z}$*, then a point* $z \in \mathcal{Z}$ *such that*

$$Lz = z$$

is called a fixed point of L*.*

[3]There are various notions of stability in control theory. What we consider as stable is the same as globally exponentially stable.

In general, a mapping may have more one, many, or no fixed point.

Given an equation $f(z) = 0$, we can convert it to a fixed point equation $Lz = z$ by defining $L : z \mapsto f(z) + z$. Then, if $Lz^* = z^*$ for a z^*, we get that $f(z^*) = 0$, i.e., the fixed point of L is the same as the solution of $f(z) = 0$.

Example A.7. *Suppose that we want to solve $cz + b = 0$ for $z \in \mathbb{R}$ and constants $c, b \in \mathbb{R}$. We can choose $L : z \mapsto (c + 1)z + b$. The mapping L is a contraction if $|c+1| < 1$ (or $-2 < c < 0$). As a numerical example, if we want to solve $-0.5z+1 = 0$ (which has $z^* = 2$), we can write it as $L : z \mapsto 0.5z + 1$. If we start from $z_0 = 0$, we get the sequence of $(z_0, z_1, \dots) = (1, 1.5, 1.75, 1.875, 1.9375, 1.96875, \dots)$.*

Of course, this is a very simple example, and we may not use such an iterative method to solve that equation.

The next theorem formalizes what we discussed about the convergence property of a contraction mapping. This is a simple, yet very important, result. It is known as the *contraction mapping* or *Banach fixed point* theorem.

Theorem A.1 (Banach Fixed Point Theorem – Theorem 3.2 of Hunter and Nachtergaele 2001)**.** *If $L : \mathcal{Z} \to \mathcal{Z}$ is a contraction mapping on a complete metric space (Z, d), then there exists a unique $z^* \in \mathcal{Z}$ such that $Lz^* = z^*$.*

Furthermore, the point z^* can be found by choosing an arbitrary $z_0 \in \mathcal{Z}$ and defining $z_{k+1} = Lz_k$. We have $z_k \to z^*$.

Note that the convergence is in norm, and it means that $\lim_{k\to\infty} d(z_k, z^*) = 0$.

There are extensions of this result, for example, when L is not a contraction per se, but is non-expansion, i.e., $d(L(z_1), L(z_2)) \leq d(z_1, z_2)$. With a relaxed assumption on the contraction property, we may lose some of the properties (e.g., uniqueness of the fixed point) or we may need extra conditions on the space, e.g., its compactness.[4]

A.4 Matrix Norm and Some of its Properties

Let us recall some results from linear algebra regarding the matrix norm, and the inverse of $\mathbf{I} - A$ and its matrix norm. The material here is mostly from Section 2.3 of Golub and Van Loan [2013].

[4] As an example, we quote Theorem 3.1 of Berinde [2007]: Let $\mathcal{Z}$ be a closed bounded convex subset of the Hilbert space $\mathcal{H}$ and $L : \mathcal{Z} \to \mathcal{Z}$ be a non-expansion mapping. Then L has at least one fixed point. This does not, however, mean that we can find it by an iterative application of L.

The vector induced p-norm of a matrix $A \in \mathbb{R}^{d\times d}$ is defined as

$$\|A\|_p = \sup_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \sup_{\|x\|_p=1} \|Ax\|_p . \tag{A.9}$$

The intuition is that we find a unit vector $x \in \mathbb{R}^d$ (according to the ℓ_p-norm) that maximizes the sizes of the mapped vector $Ax \in \mathbb{R}^d$, measured according to the same ℓ_p-norm. We could generalize this definition to have different dimensions of domain and range ($\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$) and use different vector norms to measure the length of the vectors before and after mapping. As we do not use them such results, we do not present them. These are all examples of operator norm (A.6) for linear operator A between the normed spaces $(\mathcal{Z} = \mathbb{R}^d, \|\cdot\|_p)$ and $(\mathcal{W} = \mathbb{R}^d, \|\cdot\|_p)$, as we discussed in Appendix A.3.1.

We have the following identities for the matrix norms:

- $\|A\|_1 = \max_{1\leq j\leq d} \sum_{i=1}^{d} |a_{i,j}|$ (maximum of the sum over rows)
- $\|A\|_\infty = \max_{1\leq i\leq d} \sum_{j=1}^{d} |a_{i,j}|$ (maximum of the sum over columns)
- $\|A\|_2 = \sqrt{\lambda_{\max}(A^\top A)}$ (the maximum eigenvalue of $A^\top A$).

If A is a stochastic matrix, the sum over columns (next state) is equal to one. So

$$\|\mathcal{P}^\pi\|_\infty = 1. \tag{A.10}$$

A useful property of any vector-induced p-norm is that for any $x \in \mathbb{R}^d$,

$$\|Ax\|_p \leq \|A\|_p \|x\|_p . \tag{A.11}$$

It is worth paying attention that these norms are semantically different: the norms $\|Ax\|_p$ and $\|x\|_p$ are the p-norms on the vector space $\mathbb{R}^d$, while $\|A\|_p$ is a matrix norm on the space of $\mathbb{R}^{d\times d}$ matrices. This result is essentially the same as (A.7).

Another useful property of the vector induced p-norms is that they are sub-multiplicative: For two matrices A and B, we have

$$\|AB\|_p \leq \|A\|_p \|B\|_p . \tag{A.12}$$

This is the analogous result to (A.8), specialized to matrices.

As an example of how these are relevant to the topic of this book, consider two policies π_1 and π_2, their policy induced transition kernels (matrices) $\mathcal{P}^{\pi_1}, \mathcal{P}^{\pi_2} \in$

$\mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$, and a value function $V \in \mathbb{R}^{|\mathcal{X}|}$, all for a finite state space $\mathcal{X} = \{x_1, \ldots, x_d\}$. The expression

$$\mathcal{P}^{\pi_1}\mathcal{P}^{\pi_2}V$$

is an $|\mathcal{X}|$-dimensional vector whose i-th component is the expected value of V for the agent starting from x_i and follows π_1 for the first step and π_2 for the second step. From (A.11) and (A.12), we have

$$\left\|\mathcal{P}^{\pi_1}\mathcal{P}^{\pi_2}V\right\|_p \leq \left\|\mathcal{P}^{\pi_1}\mathcal{P}^{\pi_2}\right\|_p \|V\|_p \leq \left\|\mathcal{P}^{\pi_1}\right\|_p \left\|\mathcal{P}^{\pi_2}\right\|_p \|V\|_p .$$

If $p = \infty$, by (A.10) we have $\left\|\mathcal{P}^{\pi_1}\right\|_\infty = \left\|\mathcal{P}^{\pi_2}\right\|_\infty = 1$, so overall, we get

$$\left\|\mathcal{P}^{\pi_1}\mathcal{P}^{\pi_2}V\right\|_\infty \leq \|V\|_\infty .$$

This is intuitive: the maximum value of the expectation of the value function V that the agent gets following any policies is not going to be larger than the maximum of V itself. If the norm was $p < 1$, this may not hold.

We also note that this argument is not limited to finite state problems, as we could use the properties of operator norms to get the same results for more general state spaces.

The following result shows that if a matrix A has a norm that is smaller than 1, the inverse of $\mathbf{I} - A$ exists, it has a Neumann expansion, and we can provide a bound on its norm.

Lemma A.2 (Lemma 2.3.3 of Golub and Van Loan 2013)**.** *If $A \in \mathbb{R}^{d\times d}$ and $\|A\|_p < 1$, then $\mathbf{I} - A$ is non-singular, and*

$$(\mathbf{I} - A)^{-1} = \sum_{k=0}^{\infty} A^k.$$

We also have

$$\left\|(\mathbf{I} - A)^{-1}\right\|_p \leq \frac{1}{1 - \|A\|_p}.$$

The consequence of this result for us is that we can write

$$(\mathbf{I} - \gamma\mathcal{P}^{\pi})^{-1} = \sum_{k\geq 0}(\gamma\mathcal{P}^{\pi})^k,$$

and conclude that

$$\left\|(\mathbf{I} - \gamma\mathcal{P}^{\pi})^{-1}\right\|_\infty \leq \frac{1}{1 - \gamma}.$$

A.5 Incremental Matrix Inversion

There are some formulae that allow us to incrementally update a matrix inversion (Section 2.1.4 of Golub and Van Loan 2013).

The Sherman-Morrison-Woodbury formula states that for a matrix $A_{d\times d}$ and two $d \times k$ matrices U and V, we have

$$(A + UV^\top)^{-1} = A^{-1} - A^{-1}U(\mathbf{I} + V^\top A^{-1}U)^{-1}V^\top A^{-1},$$

assuming that A and $(\mathbf{I} + V^\top A^{-1}U)$ are invertible. As $UV^\top$ is a $k \times k$ matrix (so of rank at most k), $A + UV^\top$ can be thought of as a rank-k update of the matrix A. The update of its inverse requires the computation of the inverse of $k \times k$ matrix $(\mathbf{I} + V^\top A^{-1}U)$, which can be much cheaper that directly inverting the new $d \times d$ matrix $A + UV^\top$ when k is smaller than d.

A special case of this formula is known as the Sherman-Morrison formula. It states that for an invertible matrix $A_{d\times d}$ and vectors $u, v \in \mathbb{R}^d$, the matrix $A + uv^\top$ is invertible if and only if $1 + v^\top A^{-1}u \neq 0$. And if it is invertible, we can compute it as

$$\left(A + uv^\top\right)^{-1} = A^{-1} - \frac{A^{-1}uv^\top A^{-1}}{1 + v^\top A^{-1}u}.$$

Note that the denominator is a scalar.

A.6 Concentration Inequalities

Consider $X_1, \ldots, X_n$ be independent real-valued random variables. Their average

$$S_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

is a random variable itself, and it tends to be *concentrated* arounds its expectation $\mathbb{E}[S_n]$. To see what this means, we provide a series of results that quantifies a notion of concentration.

First, for the simplicity of exposition, assume that all of X_i have the same mean μ and variance σ^2. By the linearity of the expectation, we have

$$\mathbb{E}[S_n] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[X_i] = \frac{1}{n}\mathbb{E}[\mu] = \mu.$$

By benefitting from the independence of X_i and X_j, we get that the variance of S_n is

$$\begin{aligned}
\mathrm{Var}\left[S_n\right] &= \mathbb{E}\left[(S_n - \mathbb{E}\left[S_n\right])^2\right] = \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}(X_i-\mu)\right)^2\right] \\
&= \frac{1}{n^2}\mathbb{E}\left[\sum_{i,j=1}^{n}(X_i-\mu)(X_j-\mu)\right] \\
&= \frac{1}{n^2}\mathbb{E}\left[\sum_{i=1}^{n}(X_i-\mu)(X_i-\mu) + \sum_{i,j=1;i\neq j}^{n}(X_i-\mu)(X_j-\mu)\right] \\
&= \frac{1}{n^2}\left[\sum_{i=1}^{n}\sigma^2 + \sum_{i,j=1;i\neq j}^{n}\mathbb{E}\left[(X_i-\mu)(X_j-\mu)\right]\right] \\
&= \frac{\sigma^2}{n} + \frac{1}{n^2}\sum_{i,j=1;i\neq j}^{n}\mathbb{E}\left[(X_i-\mu)\right]\mathbb{E}\left[(X_j-\mu)\right] = \frac{\sigma^2}{n}.
\end{aligned}$$

This shows that as n increases, the variance of S_n decreases with a rate of $\frac{1}{n}$. Variance is a notion of dispersion of a random variable arounds its mean, so this result shows that S_n is increasingly more concentrated around μ.

We can use these results on the mean and variance of S_n to derive a high probability notion of concentration.

Recall the Markov's inequality, which states that for a non-negative random variables Z, for any $\varepsilon > 0$, we have

$$\mathbb{P}\left\{Z > \varepsilon\right\} \leq \frac{\mathbb{E}\left[Z\right]}{\varepsilon}. \tag{A.13}$$

This means that the probability that a non-negative r.v. Z is much larger than its expectation is decreasing. For instance, $\mathbb{P}\left\{Z > k\mathbb{E}\left[Z\right]\right\} \leq \frac{1}{k}$.

A direct consequence of the Markov's inequality, applied to the non-negative r.v. $Z = |S_n - \mu|^2$ is that

$$\mathbb{P}\left\{|S_n-\mu| > \varepsilon\right\} = \mathbb{P}\left\{|S_n-\mu|^2 > \varepsilon^2\right\} \leq \frac{\mathbb{E}\left[|S_n-\mu|^2\right]}{\varepsilon^2} = \frac{\mathrm{Var}\left[S_n\right]}{\varepsilon^2} = \frac{\sigma^2}{n\varepsilon^2}. \tag{A.14}$$

This shows that for any $\varepsilon > 0$, as $n \to \infty$,

$$\lim_{n\to\infty}\mathbb{P}\left\{|S_n-\mu| > \varepsilon\right\} \to 0.$$

This means that asymptotically, the probability that S_n is more than ε different from μ is zero, no matter how small ε is. This is the *convergence in probability* of S_n to μ. This result is known as the *weak Law of Large Number (LLN)*.

We also have the *strong* LLN, which states that

$$S_n \to \mu \quad \text{almost surely}$$

under mild assumptions, such as $\mathbb{E}[|X_i|] < \infty$ for all i.

Both versions of LLN are about the convergence of S_n to μ, but they do not specify how different S_n from μ is. The statement (A.14) provides a rather loose upper bound on the deviation of S_n from μ. There are way to provide a tighter statements.

The first is the *Central Limit Theorem*, which states that as $n \to \infty$, the distribution of S_n converges to the Gaussian (normal) distribution with mean μ and variance $\frac{\sigma^2}{n}$, that is

$$S_n \xrightarrow{d} N\left(\mu, \frac{\sigma^2}{n}\right). \tag{A.15}$$

Here $\xrightarrow{d}$ denotes the convergence in distribution, which means that for any $t \in \mathbb{R}$

$$\lim_{n\to\infty} \mathbb{P}\left\{\frac{S_n - \mu}{\sigma\sqrt{n}} \le t\right\} \to \Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{x^2}{2}} \mathrm{d}x.$$

Two remarks are worth mentioning. The first is that the CLT is an asymptotic result and holds exactly only when $n \to \infty$. When n is finite, say 100 or 10,000, the distribution of S_n is only approximately Gaussian. The second is that (A.15) shows that the tail behaviour of S_n is (approximately) like a Gaussian distribution, which means that

$$\mathbb{P}\{|S_n - \mu| > \varepsilon\} \approx 2\exp(-$$

The following result is a simplified form of Lemma 6.3 of Györfi et al. 2002, which itself is Theorem 2 of Hoeffding [1963].

Lemma A.3. *(Hoeffding's Inequality) Let $X_1, \ldots, X_n$ be independent real-valued random variables bounded by B almost surely, i.e., $|X_i| < B$. For any $\varepsilon > 0$, we have*

$$\mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbb{E}[X_i])\right| > \varepsilon\right\} \le 2\exp\left(-\frac{n\varepsilon^2}{2B^2}\right).$$

A.7 Information Theory

We provide a very brief overview of some definition from information theory, which are occasionally used in the book. For a more detailed treatment, including intuition about these concepts, refer to MacKay [2003]; Cover and Thomas [2006].

Given a discrete random variable X with probability distribution p, its entropy is defined as

$$\mathbb{H}[X] \triangleq \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)}. \tag{A.16}$$

The joint entropy of (X, Y) is defined similarly:

$$\mathbb{H}[X, Y] \triangleq \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p(x, y) \log \frac{1}{p(x, y)}. \tag{A.17}$$

The conditional entropy of X given Y is defined as

$$\mathbb{H}[X|Y] \triangleq \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p(x, y) \log \frac{1}{p(x|y)}. \tag{A.18}$$

The KL divergence between distribution p and distribution q is defined as

$$\mathsf{KL}(p||q) = \int p(\mathrm{d}x) \log \frac{p(x)}{q(x)}. \tag{A.19}$$

The KL divergence is always non-negative. Whenever it is zero, it means that two distributions are the same, almost surely.

The mutual information between X and Y is

$$\mathbb{I}[X; Y] = \mathbb{H}[X|Y] - \mathbb{H}[X]. \tag{A.20}$$

The mutual information can also be written as

$$\mathbb{I}[X; Y] = \mathsf{KL}(p(x, y)||p(x)p(y)).$$

It can be shown than the mutual information is symmetric ($\mathbb{I}[X; Y] = \mathbb{I}[Y; X]$). It is also non-negative $\mathbb{I}[X; Y] \geq 0$, as a direct consequence of non-negativity of the KL divergence. When $\mathbb{I}[X; Y] = 0$, the X and Y are independent.

A.8 Algebraic Inequalities

In the theoretical analysis of ML and Statistics algorithms, in general, and in RL algorithms, in particular, we often go through some steps of upper bounding (or occasionally lower bounding) certain quantities. To prove those upper bounds, we use techniques that can be categorized, at the high level, as probabilistic arguments or algebraic ones. In many cases, to complete a proof, we need to use both types of arguments.

An example of the probabilistic argument is to benefit from the observation that a random variable, such as $G^\pi(x)$, is concentrated around its mean, which is $V^\pi(x) = \mathbb{E}\left[G^\pi(x)|x\right]$, and the probability of G^π being very different from the mean V^π is small, and becomes even smaller when we average multiple independent samples from that random variable. We discuss this in Appendix A.6.

An example of the algebraic one is to show that if r is $R_{\max}$-bounded and Q is $Q_{\max} = \frac{R_{\max}}{1-\gamma}$-bounded, we also have $T^\pi Q = r + \gamma \mathcal{P}^\pi Q$ is $(1 + \frac{\gamma}{1-\gamma}) = 1$-bounded, hence the Bellman error $\|Q - T^\pi\|_2^2$ is also $(2Q_{\max})^2$-bounded. Many of the algebraic proofs use inequalities, which we briefly summarize here.

We review Cauchy-Schwarz, Hölder, and Jensen inequalities. They all appear in different forms, such as an inequality for a finite dimensional vector, an infinite sequence, functions, or random variables. They can all be unified with the appropriate selection of the function space (and measure), but we present them separately. Considering that all of them are

In the following, let $u, v \in \mathbb{R}^d$ with $u = (u_1, \dots, u_d)$ and $v = (v_1, \dots, v_d)$; let $a = (a_1, a_2, \dots)$ and $b = (b_1, b_2, \dots)$ be infinite sequences, and $f, g : \mathcal{X} \to \mathbb{R}$ be functions. XXX

Lemma A.4.

- *For finite dimensional vectors $u, v \in \mathbb{R}^d$, we have*

$$\sum_{i=1}^{d} u_i v_i \le \sqrt{\sum_{i=1}^{d} |u_i|^2} \sqrt{\sum_{i=1}^{d} |v_i|^2},$$

  or more compactly, $\langle u, v \rangle \le \|u\|_2 \|v\|_2$, with the norms defined as in Example A.3.

- *The same holds for sequences $u = (u_1, u_2, \dots)$ and $v = (v_1, v_2, \dots)$ that satisfy $\|u\|_2, \|v\|_2 < \infty$: $\langle u, v \rangle \le \|u\|_2 \|v\|_2$ (cf. Example A.4).*

- *For functions $f, g : \mathcal{X} \to \mathbb{R}$ with $\int f^2(x)\mathrm{d}x$ and $\int g^2(x)\mathrm{d}x$ both being finite, we have*

$$\int f(x)g(x)\mathrm{d}x \leq \sqrt{\int |f(x)|^2\mathrm{d}x}\sqrt{\int |g(x)|^2\mathrm{d}x}.$$

- *If the random variables X and Y satisfy $\mathbb{E}\left[|X|^2\right], \mathbb{E}\left[|Y|^2\right] < \infty$, we have*

$$\mathbb{E}\left[XY\right] \leq \mathbb{E}\left[|XY|\right] \leq \mathbb{E}\left[|X|^2\right]^{\frac{1}{2}} \mathbb{E}\left[|Y|^2\right]^{\frac{1}{2}}.$$

  This can be written as $\langle X, Y \rangle \leq \|XY\|_1 \leq \|X\|_2 \|Y\|_2$ (cf. (A.1)).

- *Given a random variables $X \sim \nu$, and two functions $V, U : \mathcal{X} \to \mathbb{R}$ that have finite second order moments $\mathbb{E}\left[|V(X)|^2\right], \mathbb{E}\left[|U(X)|^2\right] < \infty$, we have*

$$\langle U, V \rangle_\nu \leq \|V\|_{2,\nu} \|U\|_{2,\nu},$$

  with inner product and norm defined as in (A.1) and (A.2).

- *We have*

$$\sum_{i=1}^{d} u_i v_i \leq \sqrt{\sum_{i=1}^{d} |u_i|^2}\sqrt{\sum_{i=1}^{d} |v_i|^2} \quad \textit{or more compactly} \langle u, v \rangle \leq \|u\|_2 \|v\|_2,$$

$$\int f(x)g(x)\mathrm{d}x \leq \sqrt{\int |f(x)|^2\mathrm{d}x}\sqrt{\int |g(x)|^2\mathrm{d}x}$$

$$\mathbb{E}\left[XY\right] \leq \mathbb{E}\left[|X|^2\right]^{\frac{1}{2}} \mathbb{E}\left[|Y|^2\right]^{\frac{1}{2}}$$

Bibliography

Vasile Berinde. *Iterative approximation of fixed points*, volume 1912. Springer, 2007. 40, 42

Dimitri P. Bertsekas. *Abstract dynamic programming*. Athena Scientific Belmont, 2nd edition, 2018. 7

Dimitri P. Bertsekas and Steven E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Academic Press, 1978. 7

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996. 7, 18

B J Casey, Leah H Somerville, Ian H Gotlib, Ozlem Ayduk, Nicholas T Franklin, Mary K Askren, John Jonides, Marc G Berman, Nicole L Wilson, Theresa Teslovich, Gary Glover, Vivian Zayas, Walter Mischel, and Yuichi Shoda. Behavioral and neural correlates of delay of gratification 40 years later. *Proceedings of the National Academy of Sciences of the United States of America*, 108(36): 14998–15003, 2011. 30

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 30

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006. 48

Amir-massoud Farahmand. *Regularization in Reinforcement Learning*. PhD thesis, University of Alberta, 2011. 33

Amir-massoud Farahmand, Saleh Nabi, Piyush Grover, and Daniel N. Nikovski. Learning to control partial differential equations: Regularized fitted Q-iteration

approach. In *IEEE Conference on Decision and Control (CDC)*, pages 4578–4585, December 2016. 29

Amir-massoud Farahmand, Saleh Nabi, and Daniel N. Nikovski. Deep reinforcement learning for partial differential equation control. In *American Control Conference (ACC)*, 2017. 29

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Remi Munos. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024. 30

Peter Godfrey-Smith. *Metazoa: Animal life and the Birth of the Mind*. Farrar, Straus and Giroux, 2020. 29

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The John Hopkins University Press, 4th edition, 2013. 42, 44, 45

László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Verlag, New York, 2002. 47

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. 47

De-An Huang, Amir-massoud Farahmand, Kris M Kitani, and J. Andrew Bagnell. Approximate MaxEnt inverse optimal control and its application for mental simulation of human interactions. In *AAAI Conference on Artificial Intelligence*, January 2015. 30, 31

John K. Hunter and Bruno Nachtergaele. *Applied analysis*. World Scientific Publishing Company, 2001. 35, 36, 39, 42

Kris N. Kirby. One-year temporal stability of delay-discount rates. *Psychonomic Bulletin & Review*, 16(3):457–462, 2009. 30

David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. 48

Walter Mischel, Ebbe B. Ebbesen, and Antonette Raskoff Zeiss. Cognitive and attentional mechanisms in delay of gratification. *Journal of personality and social psychology*, 21(2):204, 1972. 30

Andrew Y. Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2000. 30, 31

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 30

Yangchen Pan, Amir-massoud Farahmand, Martha White, Saleh Nabi, Piyush Grover, and Daniel Nikovski. Reinforcement learning with function-valued action spaces for partial differential equation control. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 3986–3995, Jul 2018. 29

Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2586–2591, 2007. 30

Jeffrey S. Rosenthal. *A Fist Look at Rigorous Probability Theory*. World Scientific Publishing, 2nd edition, 2006. 33

Stuart Russell. Learning agents for uncertain environments. In *Annual conference on Computational Learning Theory (COLT)*, pages 101–103, 1998. 30, 31

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988. 23

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. ii, 7

Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Morgan Claypool Publishers, 2010. 7

Christopher J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, University of Cambride, 1989. 23

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research (JMLR)*, 18(136):1–46, 2017. 30

Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. The principle of maximum causal entropy for estimating interacting processes. *Information Theory, IEEE Transactions on*, 59(4):1966–1980, April 2013. ISSN 0018-9448. 30, 31