Reinforcement Learning and Empirical Processes

Mathukumalli Vidyasagar

September 28, 2021

Preface

This is a *draft*. To quote Oliver Goldsmith, "There are a hundred faults in this Thing and a hundred things might be said to prove them beauties." I hope that, as time passes, the faults will decrease while the "beauties" will increase. In the meantime, "caveat emptor" is the watchword for the reader.

Feedback of all kinds would be gratefully received at m.vidyasagar@iith.ac.in

Contents

Chapter 1

Introduction

1.1 Introduction to Reinforcement Learning

As with many phrases in common usage, there is no precise definition of what constitutes "reinforcement learning," often abbreviated to just RL. In the present set of notes, this phrase is used to refer to decision-making with uncertain models, *and in addition, current decisions alter the model of the system.* One consequence of this alteration is that, if the same decision is taken at a future time, the consequences might not be the same. This additional feature, namely that current decisions alter the dynamics of the system under study, usually though not always by altering the surrounding environment, is what distinguishes RL from "mere" decision-making under uncertainty.

Figure 1.1 rather arbitrarily divides decision-making problems into four quadrants. Examples from each quadrant can be given.

- Many if not most decision-making problems fall into the lower-left quadrant of "good model, no alteration." For example, a well-studied control system such as a fighter aircraft has an excellent model thanks to aerodynamical modelling and/or wind tunnel tests. To be specific, the dynamical model of a fighter aircraft depends on the so-called "flight condition," consisting of the altitude and velocity (measured as its Mach number). While the dependence of the dynamical model on the flight condition is nonlinear and somewhat complex, usually sufficient modelling studies are carried out, both before the aircraft is flown and afterwards, that the dynamical model can be assumed to be "known." In turn this permits the control system designers to formulate an optimal (or some other form of) control problem, which can be solved.

- Controlling a chemical reactor would be an example from the lower-right quadrant. As a traditional control system, it can be assumed that the dynamical model of such a reactor does not change as a consequence of the control strategy adopted. However, due to the complexity of a reactor, it is difficult to obtain a very accurate model, in contrast with a fighter aircraft for example. In such a case, one can adopt one of two approaches. The first, which is a traditional approach in control system theory, is to use a nominal model of the system and to treat the deviations from the nominal model as uncertainties in the model. The second, which would move the problem from the upper left to the upper right quadrant, is to attempt to "learn" the unknown dynamical model by probing its response to various inputs. This approach is suggested in [33, Example 3.1]. A similar statement can be made about robots, where the geometry determines the *form* of the dynamical equations describing it, but not the parameters in the equations; see for example SHV20. In this case too, it is possible to "learn" the dynamics through experimentation. In practice, such an approach is far slower than the traditional control systems approach of using a nominal model and designing a "robust" controller. However, "learning control" is a popular area in the world of machine learning.

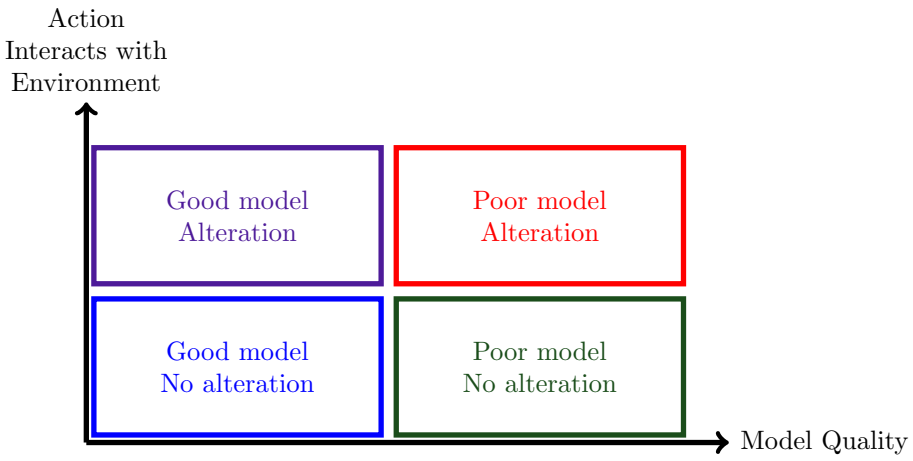


Figure 1.1: The four quadrants of decision-making under uncertainty

- A classic example of a problem belonging to the upper-left corner is a Markov Decision Process (MDP). In this class of problems, at each time instant the actor (or agent) decides on the action to be taken at that time. In turn the action affects the probabilities of the future evolution of the system. As this class of problems forms the starting point for RL (upper-right quadrant), the first part of these notes are addressed to a study of MDPs. Board games without an element of randomness would also belong to the upper-left quadrant, at least in principle. Games such as tic-tac-toe belong here, because the rules of the game are clear, and the number of possible games is manageable. *In principle*, games such as chess which are "deterministic" (i.e., there is no throwing of dice as in Backgammon for example) would also belong here. Chess is a two-person game in which, for each board position, it is possible to assign the likelihood of the three possible outcomes: White wins, Black wins, or it is a draw. However, due to the enormous number of possibilities, it is often not possible to *determine* these likelihoods precisely. It is pointed out explicitly in [30] that, merely because we cannot explicitly compute this likelihood function, that does not mean that it does not exist! However, as a practical matter, it is not a bad idea to treat this likelihood function as being unknown, and to infer it on the basis of experiment / experience. Thus, as with chemical reactors, it is not uncommon to move chess-playing from the lower-right corner to the upper-right corner.
- The upper-right quadrant is the focus of these notes. Any problems where the actions taken by the learner alter the environment, in ways that are not known to the learner, are referred to as "reinforcement learning" (RL). Despite the lack of knowledge about the consequences, the learner has no option but to keep trying out various actions in order to "explore" the environment in which the unknown system is operating. As time goes on, some amount of knowledge is gained, and it is therefore possible, at least in principle, to "exploit" the knowledge to improve decision making. The trade-off between exploration and exploitation is a standard topic in RL. A canonical example is MDPs where the underlying parameters are not known, and these occupy a major part of these notes. As mentioned above, often complex problems from the lower-right quadrant (such as chemical reactors), or the upper-left quadrant (such as Chess), are also treated as RL problems.

Now we will give a general description of the problem. In a RL problem, there are "states" and then there are "actions." At each time t, the agent, also sometimes referred to as the learner, measures the state X_t at time t, which belongs to a state space $\mathcal{X}$. Based on this measurement, the agent chooses a control U_t from a menu of "actions," which we denote by $\mathcal{U}$. While it is possible for the state space $\mathcal{X}$ and the range of possible actions $\mathcal{U}$ to be infinite, in these notes we simplify our lives by restricting $\mathcal{U}$ to be a finite set.

In the same way, it is possible to treat "time" as a continuum, but again we simplify life by treating t as a discrete variable assuming values in the set of natural numbers $\mathbb{N} = \{0, 1, \cdots\}$. Thus RL requires the agent to take a set of sequential decisions from a finite menu, at discrete instants of time. When the agent chooses an action $U_t \in \mathcal{U}$, two things happen.

1. The agent receives a "reward" R_t. The reward could either be deterministic, or random, and both possibilities are permitted in these notes. The reward could be a negative number, suggesting a penalty instead of a reward, but the phrase "reward" is standard phraseology. In case the reward is random, it is assumed that the reward lies in a bounded interval in $\mathbb{R}$ which is known *a priori*, in which case the reward can be translated to belong to an interval $[0, M]$. The same transformation can of course be applied if the reward is deterministic. Note that some authors speak of a "cost" which is to be minimized, rather than a reward which is to be maximized. The modifications required to tackle this situation are obvious and we will not comment upon this further. The reward depends not just on the action chosen U_t, but also the state X_t of the environment at time t. There can be two sources of uncertainty in the reward. In a Markov Decision Problem (MDP), the reward could be a random function of X_t and U_t, but with a known probability distribution. In an RL problem, even the probability distribution of the reward is not necessarily known. However, for technical reasons, it is assumed that the upper bound M on the reward is known.

2. The action U_t affects the dynamics of the system. A consequence is that the same action taken at a different time need not lead to the same reward, because in the meantime the "state" of the environment may have changed.

Over the years, the RL research community has given some "structure" to the above rather vague and general description. Specifically:

1. The environment is taken as a Markov process (see Section 8.2) in which the state transition matrix depends on the action taken. So there are $|\mathcal{U}|$ state transition matrices, one for each possible action.

2. If X_t denotes the state of the Markov process at time t and U_t is the action taken at time t, then the reward R is taken to be a function $R(X_t, U_t)$. This formalism explains why the same action $U_t \in \mathcal{U}$ taken at a different time may lead to a different reward, because the state X_t may have changed. It is also possible for R to be a "random" function of X_t and U_t, so that X_t, U_t only specify the probability distribution of $R(X_t, U_t)$. In such a case, even if the same state-action pair (X_t, U_t) were to occur at a different time, the resulting reward need not the same.

3. Yet another variation is that the reward $R(X_t, U_t)$ (whether random or deterministic) is paid at the *next* time instant $t + 1$. This is the case in some books, notably [35, 33]. In other words, if the Markov process is in state X_t and the action U_t is applied, the reward is $R_{t+1} = R(X_t, U_t)$. This allows those authors to consider the situation where the "next state" X_{t+1} and "next reward" R_{t+1} can share a joint probability distribution, which depends on X_t and U_t. This is the convention adopted in these notes. Note that some other authors assume that the reward is *immediate*, so that $R_t = R(X_t, U_t)$.

4. There are two distinct types of Markov Decision Processes that are widely studied, namely: Discounted reward processes and average reward processes. Each of them has rather a distinct behavior from the other. In discounted reward processes, there is a "discount factor" $\gamma \in (0, 1)$ that is applied to future rewards. The objective is to maximize the sum of the future rewards, where the reward at time t is discounted by the factor γ^t. Because this future discounted reward is itself random, we maximize the *expected value* of this random variable. In the average reward process, the objective is to minimize the expected value of the average of future rewards over time. Because there is no discounting of future rewards, a reward paid at any time contributes just as much to the average as a reward paid at any other time.

5. In the simplest version of the problem, the $|\mathcal{U}|$ state transition matrices, one for each possible action, are assumed to be known, as is the reward function. In the case where the reward is a random function of X_t and U_t, it is assumed that the probability distribution of $R(X_t, U_t)$ is known. It is also assumed that the state X_t of the Markov process can be observed by the agent, and can be used to decide the action U_t. A key concept in RL is that of a "policy" π which is a map from the state space $\mathcal{X}$ of the Markov process to the set of actions $\mathcal{U}$. The objective here is to choose the optimal policy, which maximizes the expected value of the discounted future reward over all possible policies. This version of the problem is usually known as a **Markov Decision Process (MDP)**.[1] It is usually viewed as a precursor to RL. In "proper" RL, neither the Markovian dynamics nor the reward are assumed to be known, and must be learned on the fly so to speak. However, knowing the solution approaches to the MDP is very useful in solving RL problems. It should be pointed out that some authors also use the phrase RL to the problem of finding the optimal policy in an MDP where the parameters of the problem are completely known.

A dominant theme in RL is the trade-off between "exploration" and "exploitation." By definition, the agent in an RL problem is operating in an unknown environment. However, after sometime a reasonably good model of the environment is available, and a set of actions that is reasonably "rewarding" is also identified. Should the agent then persist with this set of actions, or occasionally attempt something new, just on the off-chance that there is a better set of actions available? Let us take a concrete example. A successful chess player would have evolved, over the years, a set of strategies that work well for him/her. Should the player persist with the time-proven strategies (exploitation) until someone starts beating him/her, or occasionally try something completely different just to see what happens (exploration)? The answer is not clear, and is likely to vary from one domain to another. To illustrate the domain dependence of the solution, suppose a person moves to a new town and wishes to find the best coffee shop. Then it is probably sufficient to try each nearby coffee shop just once (or just a few times), because most coffee shops have standardized protocols for preparing coffee, so that the quality is not likely to vary very much from one visit to the next. Therefore a person can stick to the coffee shop that is most appealing after a few visits, and there is very little incentive for further "exploration," only "exploitation." In contrast, it can be assumed that the course of a chess match between two players at the highest level almost invariably leads to a previously unexplored set of positions. Thus persisting with a stock strategy would invariably lead to suboptimal results, and there must be greater emphasis on exploration than in the coffee shop example.

There are a couple of methods for quantifying the trade-off between exploration and exploitation. We begin with the observation that almost any "sensible" learning algorithm would converge to a nearly optimal policy within a finite number of time steps. Here are two ways to measure how good the algorithm is:

1. Given an accuracy ϵ, one can measure how many time steps are required for the policy to be within ϵ of the optimal policy.[2] The faster a policy becomes ϵ-suboptimal, the better it is. Implicit in this characterization is the assumption that a policy is *not* penalized for how badly it performs before it achieves ϵ-suboptimality – just the time it takes to achieve ϵ-suboptimality status.

2. The other measure is to see what the reward would have been, had the learner somehow magically implemented the optimal policy right at the outset, and compare it against the actually achieved performance. This quantity is called the "regret" and is defined precisely later on. The difference between minimizing the regret and minimizing the time for achieving ϵ-optimality is that in the latter, the performance of the algorithm before achieving ϵ-optimality is not penalized, whereas it is counted as a part of the regret.

Clearly, the two criteria are not the same. A learning strategy that converges relatively quicky, but performs poorly along the way would be rated highly under the first criterion, and poorly under the second criterion.

[1] There is a variant where the state X_t cannot be observed directly; instead one observes an output Y_t which is either a deterministic or a random function of X_t. This problem is known as a Partial Observed Markov Decision Process (POMDP). This problem is not discussed at all in these notes.

[2] This idea is made precise in subsequent chapters.

Such issues are examined in Chapter 7.

Within the broad area of Machine Learning (ML) or Artificial Intelligence (AI), RL stands quite distinctly apart from other popular areas such as supervised learning (which is what many people mean when they talk about ML), and unsupervised learning. In supervised learning, the main goal is **generalization**. Thus the learner is shown an amount of "training data" which consists of labelled data. After the training phase, the learner is then shown "testing data" for which the correct labels are known to the evaluator, and the learner is asked to predict these correct labels. The extent to which the learner is able to match the correct labels serves as a measure of the quality of the learning algorithm. A well-known recent example is the ImageNet database [19], created as a part of the LSVRC (Large Scale Visual Recognition Challenge). It consists of roughly 14 million images that are hand-curated. The full set, or some subset thereof, is presented to some supervised learning algorithm, whose parameters are then adjusted to achieve good performance on the training inputs. At the other end of the spectrum from supervised learning lies "unsupervised learning," a popular part of which is "clustering." In this application, unlabelled inputs are collected into several groups or clusters, whereby the elements of each cluster are closer to the centroid of their own cluster than they are to that of any other cluster. Then a new input is assigned to the cluster whose centroid is closest to the test input. Reinforcement learning lies in-between supervised and unsupervised learning. Expand. There are several excellent texts on these two topics and we will not discuss these two branches of ML hereafter. Book-length treatments of RL can be found, among others, in the following: [35, 33]. A book length treatment of MDPs can be found in [28]. Add some references of Bertsekas-Tsitsiklis here.

1.2 Some Examples of Reinforcement Learning

In this section we briefly discuss a few motivating problems that can serve as illustrations of reinforcement learning. We will return to a couple of these problems again in future chapters.

There are several examples of reinforcement learning available in the literature. The books [28, 33] contain several examples, while the book [10] is primary devoted to examples of RL in a variety of areas, including healthcare, transportation, finance etc. Perhaps the most "famous" application of RL is a general-purpose algorithm that can be taught to play a variety of games, including Chess, Shogi and Go [12, 31]. Robot control, including path-planning in the presence of (possibly unknown) obstacles is another popular application. Some RL texts and papers study the problem of balancing a stick on a moving cart, which is known in control theory as the "inverse pendulum" problem. This might not be a good application of RL, because the system can be modelled very precisely, which in turn leads to very efficient control laws. However, by viewing this well-studied problem in control theory as a problem in RL, the research community has developed several new and interesting learning paradigms. Another application, which is of moderate size, is that of deciding an optimal strategy for the game of Blackjack, sometimes also called Twenty One. We will study this example, either in its full form or in a simplified form, in detail at appropriate places in these notes.

1.2.1 Multi-Arm Bandit Problems

This problem is a generalization of the "slot machine" in gambling casinos around the world, whereby the player pulls a lever and receives a random payoff. In order to pull the lever, the player has to insert some money, and the expected value of the payoff is less than the amount to be inserted; that is how the casino makes money. However, in our model, we ignore the fact that a player has to pay to play, and focus strictly on the payout part of it.

Suppose a player is facing m slot machines, or "bandits," each of which has random payout. Specifically, let X_i denote the random payout of the i-th bandit. Then X_i has an unknown expected (mean value) payout, as well as an unknown probability distribution around this mean value. To avoid unnecessary technicalities, it is assumed that all returns are nonnegative, and that there is a fixed known upper bound M on the payout of each machine, which can be taken as 1 without any loss of generality. Therefore the return of each arm

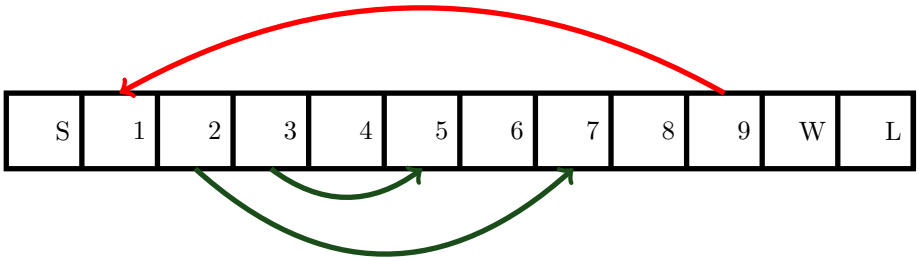


Figure 1.2: Toy Snakes and Ladders Game

has a probability distribution ϕ_i is supported on the set $[0, 1]$. Define

$$\mu_i = \int_0^1 x\phi_i(x)dx$$

to be the mean or expected value of X_i. Of course, the player does not know either μ_i or $\phi_i(\cdot)$. But the player is able to "pull the arm" of each bandit and see what happens. This generates (we assume) statistically independent samples $x_{i1}, \cdots, x_{im}$ of the random variable X_i. Based on the outcome of these experiments, the player is able to make *some estimate* of μ_i for each bandit i. These estimates can be used to determine future strategies.

Note that if the quantities $\mu_1, \cdots, \mu_m$ are known, then the problem is simple: The player should always play the machine that has the highest expected payout. But the challenge is to determine which machine this is, on the basis of experimentation. As stated above, there are many reasonable algorithms that will asymptotically (as the number of trials increases towards infinity) determine the arm(s) with the best return(s). Therefore one way to assess the performance of an algorithm is its "regret," that is, the return achieved over the course of learning, subtracted from the optimal return of always choosing the arm with the highest return. Interestingly, there are theorems that give quite tight upper and lower bounds on the achievable regret, and these are discussed in Section 7.1.

1.2.2 Snakes and Ladders

We all know the ancient snakes and ladders game, where the objective is for a player to pass from the start to the end while avoiding the snakes and taking advantage of the ladders. We will modify the game slightly by adding the possibility of losing if the player overshoots the last square. A toy version of the game is shown below (it is also studied in Section 8.2.

The rules of the game are as follows:

- Initial state is S.
- A four-sided, fair die is thrown at each stage.
- Player advances as many squares as the outcome of the throw, followed by the impact of the snake or ladder, if any.
- Player must land exactly on W to win.
- If implementing a move causes the crossing of L, then the player loses. Landing exactly on L also loses.
- Hitting the square W leads to a reward of 5 and hitting the square L leads to a reward of -5. The reward in every other square is 0.

(P, H)	R
$P < H$	-2
$P = H$	1
$P > H, P \neq W$	2
$(P, H) = (W, *)$	5
$(P, H) = (L, *)$	-5

Table 1.1: Reward Table for Simplified Blackjack Game

At each stage of the game, the player has two choices: to roll the die and take a chance on the outcome, or not to roll it. We can ask: What is the best strategy for a player as a function of the square currently being occupied? Clearly, it depends on whether the expected return from playing exceeds the expected return from not playing.

1.2.3 Blackjack

Blackjack is a popular game in gambling casinos around the world. The player plays against the "house."[3] The player and the house draw cards in alternation. The objective is to draw cards such that the total of the cards is as close to 21 as possible without exceeding it. That is why sometimes Blackjack is also called "Twenty-One." The formulation of Blackjack as a problem in RL is discussed in [33, Example 5.1]. At each time instant, the player has ony two possible actions: To ask for one more card, or not. These are known as "hit" and "stick" respectively. So the set of possible actions $\mathcal{U}$ has cardinality two. If the player draws a card, the outcome is obviously random. Either way, the house also draws a card whose outcome is random. It is shown in [33, Example 5.1] that the process can be modelled by a Markov process with 200 states, so that $|\mathcal{X}| = 200$. However, tracing out all possible future evolutions of the game, starting from the current state, is nearly impossible, and simulations are the only way to analyze the problem.

We now present a simplified version of Blackjack. Obviously, drawing a card leads to the player's total increasing by anywhere from 1 to 11.[4] So if the player's current total is 10 or less, the player cannot possibly lose by drawing, and may get closer to winning. So the optimal strategy from such a position is not in doubt. With that in mind, we replace the drawing of a card by the rolling of a fair four-sided die, with all four outcomes being equally probable. It does not matter what the "target" total is, because if the target total is T, then so long as the player's total is $T - 4$ or less, the player should roll the die. With this in mind, we can think of the player's states as $\{0, 1, 2, 3, W, L\}$, with W and L denoting Win and Lose respectivey. If the player's current total plus the outcome of the die *exactly equals* 4, the player wins, and if the total exceeds 4, the player loses. But there is an added complication, which is the total of the "House." Let us assume that the House policy is to "stick" whenever it gets within 3 of the designated total. Hence it can be assumed that the House total is in $\{1, 2, 3\}$. Now the object of the game is not merely to get as close to W without going over, but also to beat the House total. Hence the reward for this game can be specified as shown in Table 1.1. With this reward structure, at each position, the player has the option of rolling the die, or not. It turns out that this game is more complex than just the player playing snakes and ladders. We will analyze this game also in later chapters.

1.2.4 Backgammon

Backgammon is a board game played by two players on a board with (essentially) 24 positions, with each player throwing two six-sided dice at each turn. Figure 1.3 shows a typical board position. The game

[3] Actually, it is possible to have more than one player plus the "house." However, to simplify the problem, we study only the case of one player against the "house."

[4] An Ace can be counted as either 1 or 11 as per the player's choice.

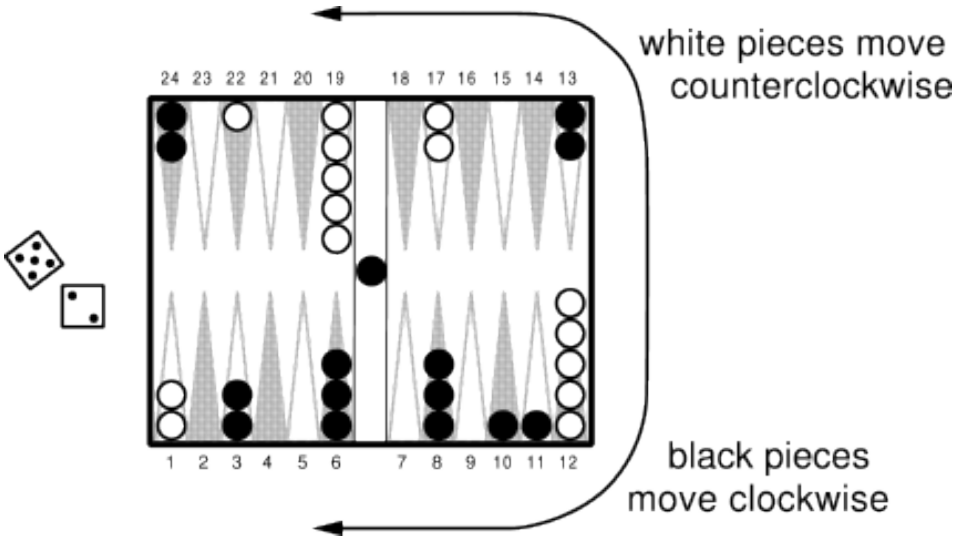


Figure 1.3: A typical board position in backgammon

combines chance (random outcome of throwing the dice) and strategy (what a player does based on the outcome of the dice).

Unlike in Blackjack, the range of possible actions available to a player at each turn is quite large. This game is well-suited to a technique called "temporal difference" or TD-learning, which is studied in Section 4.2. Tesauro has published several articles on how to program a computer to play backgammon, including [36, 37, 38]. See [33, Section 16.1] for a detailed description of the rules of backgammon and the TD implementation of Tesauro.

1.2.5 AlphaGo and AlphaZero

It would not be a exaggeration to say that a great deal of the public attention to artificial intelligence arises from the success of two programs, namely AlphaGo and AlphaZero. In 2016, a UK-based company called Deep Mind (since acquired by Google) created a program called AlphaGo to play Go, a board game played on a grid of 19×19 places. In a five-game match held in Seoul, Korea between the 9th and 15th of March, AlphaGo played against Lee Sedol, who was an eighteen-time world champion, though he was not world champion at that time. AlphaGo won four out of the five games. It was the first instance of a computer defeating a ranking Go player. A year later, in 2017, AlphaGo defeated the top-ranked player Ke Jie. In a series of three matches played between 23rd and 27th May, AlphaGo won all three matches.

Twenty years earlier IBM had developed the Deep Blue platform to play chess. Obviously, over such a long period of time, there would be massive improvements in computing hardware. Indeed, AlphaGo ran on a collection of Tensor Processing Units (TPUs), which are specially designed to carry out the type of computations required by AlphaGo (as opposed to general-purpose CPUs, or Central Processing Units).

Even at that time, Deep Mind had in its possession a more advanced program called AlphaZero, but did not deploy it against Ke Jie. AlphaZero could be programmed to play chess, Go and shogi (Japanese chess). AlphaZero defeated AlphaGo while playing Go, defeated Stockfish (a popular chess-playing program), and Elmo (a popular program to play shogi). However, in the eyes of many, the real interest in AlphaZero arose from the manner in which is trained itself. Recall that the Deep Blue platform developed by IBM relied on human inputs, and a search technique, in order to analyze board positions and determine its next move. In contrast, AlphaZero used an entirely different approach, whereby it improved itself through "self-play", through a mathematical method known as Monte Carlo tree search (MCTS) algorithm. Thus *the same*

Figure 1.4: Deep Mind's AlphaGo program playing Lee Sedol. Source [50].

program is able to "teach itself" to play different games. A popular description how AlphaZero goes about its self-appointed task can be found in [12]. Those interested in the mathematical details can find them in [31].

One of the intriguing philosophical aspects of AlphaZero is the fact that, as its name implies, AlphaZero starts from zero, that is, *without any prior knowledge*. Its superior performance compared to other programs that make use of prior knowledge has been interpreted by some AI researchers to claim that "prior knowledge" is not necessary to achieve top performance. To understand why this is interesting, let us consider the same question, but changing "chess" to "cooking." Suppose you wish to become a master chef. Should you first learn under someone who is already a master chef, and experiment on your own only *after* you have achieved some level of proficiency? Or is it better for you to undertake trial and error right from Day One? Most of us would instinctively answer that learning from a master (i.e., tapping domain knowledge) would be better. One of the intriguing aspects of the success of AlphaZero is that, when it comes to a computer learning to play chess, domain knowledge *apparently* does not confer any advantage. However, at the moment the role of prior domain knowledge in AI is still a topic for further research. It is not clear whether the success of AlphaZero is a one-off phenomenon, or a manifestation of a more universally applicable principle.

1.3 About These Notes

This section is to be rewritten in its entirety. In particular, the role of stochastic approximation in RL is to be highlighted. In the first part of these notes, the emphasis is on solution techniques for the conventional Markov Decision Processes (MDPs) with known dynamics, and then MDPs with unknown dynamics. The latter problems are the domain of Reinforcement Learning (RL). The techniques presented in this part of the notes can either lead to "exact" solutions to the MDP under study, or to "approximate" solutions. These techniques are useful when the number of possible policies (the number of possible maps from the state space to the action space) is not overly large. When the size of the set of policies is too large to be handled by a computer, the usual approach is to reduce the dimensionality of the problem. Specifically, instead of considering *all possible* policies, attention is restricted to *a subset*. Suppose that the state space $\mathcal{X}$ and the action space $\mathcal{U}$ are both finite. Then the number of possible policies is the set of all possible

maps from $\mathcal{X}$ into $\mathcal{U}$, which has cardinality $|\mathcal{U}|^{|\mathcal{X}|}$. The value function $V : \mathcal{X} \to \mathbb{R}$ (see Chapter 2) can be viewed as a vector of dimension $|\mathcal{X}|$. However, the action-value function $Q : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ can be viewed as a vector of dimension $d := |\mathcal{X}| \times |\mathcal{U}|$, which can be a large number even if $|\mathcal{X}|$ and $|\mathcal{U}|$ are individually rather small. In such a case, instead of examining *all possible* vectors of dimension d, we can choose a set of linearly independent functions $\psi_i(\cdot, \cdot) : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ for $i = 1, \cdots, m$ (basically, just a set of $m \ll d$ linearly independent vectors, and then express Q as a linear combination of these m vectors. By resticting Q to be a linear combination of these vectors, we lose generality but gain tractability of the problem. At some point we may decide that limiting Q to be a *linear combination* of some basis functions is too restrictive, and opt for a *nonlinear function*, that is, a multi-layer neural network. To over-simplify grossly, that is one way to think of deep reinforcement learning. The final theme studied in these notes is that of PAC (Probably Approximately Correct) solutions to Markov Decision Problems. The idea here is that it is not necessary to strive for absolutely the best possible solution – it is sufficient if the solution is nearly optimal. Moreover, if these nearly optimal solutions are to be found using probabilistic methods, then it is permissible if these probabilistic methods work *with high probability*. To put it in colloquial terms, a PAC algorithm is one that gives more or less the right answer most of the time. PAC learning is a well-established subject, and [45] offers a fairly complete (though advanced and rather abstract) treatment of the topic. For reinforcement learning, it is not clear whether so much generality and abstraction are required. Perhaps there are ways to present the theory in a simplified setting, and at the same time, explore questions that are relevant to reinforcement learning. These questions are still being studied, and an attempt will be made to summarize the current research in PAC-MDP in the last part of the notes.

Chapter 2

Markov Decision Processes

A widely used mathematical formalism for reinforcement learning problems is Markov Decision Processes (MDPs) where the dynamics of the Markov process are not known, and must somehow be "inferred" on the fly. Before tackling that problem, we must first understand MDPs when the dynamics *are* known. That is the aim of the present chapter. In the interests of simplicity, the discussion is limited to the situation where the state and action spaces underlying the MDP are finite sets. MDPs where the underlying state space and/or action space is countable, or an arbitrary measurable space, are also of interest in some applications. However, we do not study the more general situations in these notes. The area of MDP is quite well-studied, and there are several excellent books on the subject. The reader is directed to [28] for a comprehensive treatment of the subject, which also studies the case of infinite state and action spaces. The book [10] contains several practical examples of MDPs. The theory of MDPs is also studied in [33] and [35].

2.1 Markov Reward Processes

Recall the introduction to Markov processes in Section 8.2. Further facts about Markov processes can be found in [46].

Suppose $\mathcal{X}$ is a finite set of cardinality n, written as $\{x_1, \ldots, x_n\}$. If $\{X_t\}_{t \geq 0}$ is a stationary Markov process assuming values in $\mathcal{X}$, then the corresponding state transition matrix A is defined by

$$a_{ij} = \Pr\{X_{t+1} = x_j | X_t = x_i\}. \tag{2.1}$$

Thus the i-th row of A is the conditional probability vector of X_{t+1} when $X_t = x_i$. Clearly the row sums of the matrix A are all equal to one. Therefore the induced norm $\|A\|_{\infty \to \infty}$ also equals one.

Up to now there is nothing new beyond the contents of Section 8.2. Now suppose that there is a "reward" function $R : \mathcal{X} \to \mathbb{R}$ associated with each state. There is no consensus within the community about whether the reward corresponding to the state X_t is paid at time t, or time $t+1$. We choose to follow [28, 33] and assume that the reward is paid at time $t+1$.[1] This allows us to talk about the *joint* probability distribution $\Pr\{(X_{t+1}, R_{t+1}) | X_t\}$, where both the next state X_{t+1} and the reward R_{t+1} are random functions of the current state X_t. In particular, if R_{t+1} is a deterministic function of X_t, then we have that

$$\Pr\{(X_{t+1}, R_{t+1}) = (x_j, r_l) | X_t = x_i\} = \begin{cases} a_{ij} & \text{if } r_l = R(x_i), \\ 0 & \text{otherwise,} \end{cases}$$

where $R : \mathcal{X} \to \mathbb{R}$ is the reward function.

Two kinds of Markov reward processes are widely studied, namely: Discounted reward processes, and average reward processes. Each of these is studied in a separate subsection.

[1] Note that in [35], the reward is assumed to be "immediate," that is, paid at time t.

2.1.1 Discounted Reward Processes

To study discounted Markov Reward Processes, we choose a "discount factor" $\gamma \in (0,1)$. Suppose $x_i \in \mathcal{X}$ is the "state of interest." Then the **expected discounted future reward** $V(x_i)$ is defined as

$$V(x_i) = E\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | X_0 = x_i\right]. \tag{2.2}$$

We often just use "discounted reward" instead of the longer phrase. Note that, because the set $\mathcal{X}$ is finite, the reward function R_{t+1} is bounded if it is a deterministic function of X_t. If R_{t+1} is a random variable dependent on X_t, then it customary to assume that it is bounded. With these assumptions, because $\gamma < 1$, the above summation converges and is well-defined. The quantity $V(x_i)$ is referred to as the **value function** associated with x_i, and the vector

$$\mathbf{v} = [\ V(x_1) \quad \cdots \quad V(x_n)\ ]^\top, \tag{2.3}$$

is referred to as the **value vector**. Note that, throughout these notes, we view the value as both a *function* $V : \mathcal{X} \to \mathbb{R}$ as well as a *vector* $\mathbf{v} \in \mathbb{R}^n$. The relationship between the two is given by (2.3). We shall use whichever interpretation is more convenient in a given context.

This raises the question as to how the value function and/or value vector is to be determined.

Define the vector $\mathbf{r} \in \mathbb{R}^n$,

$$\mathbf{r} := [\ r_1 \quad \cdots \quad r_n\ ]^\top, \tag{2.4}$$

where, if R_{t+1} is a random function of X_t, then

$$r_i := E[R_{t+1} | X_t = x_i]. \tag{2.5}$$

If R_{t+1} is a deterministic function $R_d(X_{t+1})$, then

$$r_i = \sum_{j=1}^{n} a_{ij} R_d(x_j).$$

Of course, if R_{t+1} is a deterministic function $R(X_t)$, then r_i is just $R(x_i)$.

Theorem 2.1. *The vector* $\mathbf{v}$ *satisfies the recursive relationship*

$$\mathbf{v} = \mathbf{r} + \gamma A \mathbf{v}, \tag{2.6}$$

or, in expanded form,

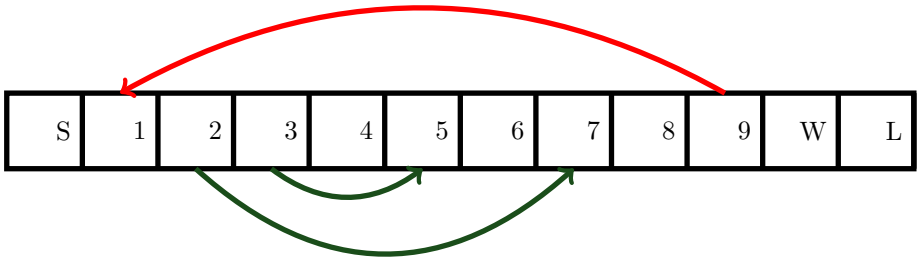
$$V(x_i) = r_i + \gamma \sum_{j=1}^{n} a_{ij} V(x_j). \tag{2.7}$$

Proof. Let $x_i \in \mathcal{X}$ be arbitrary. Then by definition we have

$$V(x_i) = E\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | X_0 = x_i\right] = r_i + E\left[\sum_{t=1}^{\infty} \gamma^t R_{t+1} | X_0 = x_i\right]. \tag{2.8}$$

However, if $X_0 = x_i$, then $X_1 = x_j$ with probability a_{ij}. Therefore we can write

$$\begin{aligned} E\left[\sum_{t=1}^{\infty} \gamma^t R_{t+1} | X_0 = x_i\right] &= \sum_{j=1}^{n} a_{ij} E\left[\sum_{t=1}^{\infty} \gamma^t R_{t+1} | X_1 = x_j\right] \\ &= \gamma \sum_{j=1}^{n} a_{ij} E\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | X_0 = x_j\right] \\ &= \gamma \sum_{j=1}^{n} a_{ij} V(x_j). \end{aligned} \tag{2.9}$$



Example 2.1.

In the second step we use fact that the Markov process is stationary. Substituting from (2.9) into (2.8) gives the recursive relationship (2.14). □

Until now it has been assumed that the discount factor γ is strictly less than one. However, if the Markov process has one or more absorbing states, then (8.40) gives an explicit formula for the average time before a sample path terminates in an absorbing state. Moreover, this time is always finite. Therefore, in Markov processes with absorbing states, it is possible to use an *undiscounted* sum, with the understanding that the summation ends as soon as X_t is an absorbing state. Equivalently, it is possible to assign a reward of zero to every absorbing state, so that the infinite sum of rewards contains only a finite number of nonzero terms. It is left to the reader to state and prove the analog of Theorem 2.1 for this case.

We analyze the toy snakes and ladders game of Example 8.3. As shown therein, the state transition matrix of this game is given by

	S	1	4	5	6	7	8	W	L
S	0	0.25	0.25	0.25	0	0.25	0	0	0
1	0	0	0.25	0.50	0	0.25	0	0	0
4	0	0	0	0.25	0.25	0.25	0.25	0	0
5	0	0.25	0	0	0.25	0.25	0.25	0	0
6	0	0.25	0	0	0	0.25	0.25	0.25	0
7	0	0.25	0	0	0	0	0.25	0.25	0.25
8	0	0.25	0	0	0	0	0.25	0.25	0.25
W	0	0	0	0	0	0	0	1	0
L	0	0	0	0	0	0	0	0	1

To define a reward function for this problem, we will set $R_{t+1} = f(X_{t+1})$, where f is defined as follows: $f(W) = 5$, $f(L) = -2$, $f(x) = 0$ for all other states. Thus there is no *immediate* reward. However, there is an expected reward *depending on the state at the next time instant*. For example, if $X_0 = 6$, then the expected value of R_1 is 5/4, whereas if $X_0 = 7$ or $X_0 = 8$, then the expected value of R_1 is 3/4.

Now let us see how the implicit equation (2.6) can be solved to determine the value vector $\mathbf{v}$. Since the induced matrix norm $\|A\|_{\infty\to\infty} = 1$ and $\gamma < 1$, it follows that the matrix $I - \gamma A$ is nonsingular. Therefore, for every fixed assignment of rewards to states, there is a unique $\mathbf{v}$ that satisfies (2.6). In principle it is possible to deduce from (2.6) that

$$\mathbf{v} = (I - \gamma A)^{-1}\mathbf{r}. \tag{2.10}$$

The difficulty wth this formula however is that in most actual applications of Markov Decision Problems, the integer n denoting the size of the state space $\mathcal{X}$ is quite large. Moreover, inverting a matrix has cubic complexity in the size of the matrix. Therefore it may not be practicable to invert the matrix $I - \gamma A$. So we are forced to look for alternate approaches. A feasible approach is provided by the Contraction Mapping Theorem (CMT), namely Theorem 8.13. With the contraction mapping theorem in hand, we can apply it to the problem of computing the value of a discounted Markov reward process.

Theorem 2.2. *The map* $\mathbf{y} \mapsto T\mathbf{y} := \mathbf{r} + \gamma A\mathbf{y}$ *is monotone and is a contraction with respect to the* ℓ_∞*-norm, with contraction constant* γ*.*

Proof. The first statement is that if $\mathbf{y}_1 \leq \mathbf{y}_2$ componentwise (and note that the vectors $\mathbf{y}_1, \mathbf{y}_2$ need not consist of only positive components), then $T\mathbf{y}_1 \leq T\mathbf{y}_2$. This is obvious from the fact that the matrix A has only nonnegative components, so that $A\mathbf{y}_1 \leq A\mathbf{y}_2$. For the second statement, note that, because the matrix A is row-stochastic, the induced norm of A with respect to $\|\cdot\|_\infty$ is equal to one. Therefore

$$\|T\mathbf{y}_1 - T\mathbf{y}_2\|_\infty = \|\gamma A(\mathbf{y}_1 - \mathbf{y}_2)\|_\infty \leq \gamma \|\mathbf{y}_1 - \mathbf{y}_2\|_\infty.$$

This completes the proof. □

Therefore one can solve (2.6) by repeated application of the contraction map T. In other words, we can choose some vector $\mathbf{y}^0$ arbitrarily, and then define

$$\mathbf{y}^{i+1} = \mathbf{r} + \gamma A\mathbf{y}^i.$$

Then the contraction mapping theorem tells us that $\mathbf{y}^i$ converges to the value vector $\mathbf{v}$. Moreover, from (8.49) one can estimate how far the current iteration is from the solution $\mathbf{v}$. Note that the contraction constant ρ in the statement of the theorem can be taken as the discount factor γ. Define the constant

$$c := \|\mathbf{r} + \gamma A\mathbf{y}^0 - \mathbf{y}^0\|_\infty,$$

which measures how far away the initial guess $\mathbf{y}^0$ is from satisfying (2.6). Then we have the estimate

$$\|\mathbf{y}^i - \mathbf{v}\|_\infty \leq \frac{\gamma^i}{1-\gamma} c. \tag{2.11}$$

In this approach to finding the value function, each iteration has quadratic complexity in n, the size of the state space. Moreover, (2.11) can be used to decide how many iterations should be run to get an acceptable estimate for $\mathbf{v}$. This approach to determining $\mathbf{v}$ (albeit approximately) is known as "value iteration." Thus if we use I iterations, then the complexity of value iteration is $O(In^2)$ as opposed to $O(n^3)$ for using (2.10). Hence the value iteration approach is preferable if $I \ll n$. Note that the faster future rewards are discounted (i.e., the smaller γ is), the faster the iterations will converge. Moreover, if the reward vector $\mathbf{r}$ is nonnegative, and we choose $\mathbf{y}^0 = \mathbf{r}$, then the value iterations result in a sequence of estimates $\{\mathbf{y}_i\}$ that is componentwise monotonically nondecreasing. (See Problem 2.1.)

2.1.2 Average Reward Markov Processes

Now we discuss average reward Markov processes. As before, there is a Markov process $\{X_t\}_{t\geq 0}$ on a finite space $\mathcal{X}$ of cardinality n, with the state transition matrix $A \in [0,1]^{n\times n}$, and a reward function $R : \mathcal{X} \to \mathbb{R}$. If the reward is random, it is assumed that the reward is bounded almost surely (to avoid technicalities), and the symbol r_i is used to denote the *expected value* of the reward to be paid at time $t+1$, when $X_t = x_i$.

The objective is to compute the **average reward**

$$c^* := \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} E[R(X_t)|X_0 \sim \boldsymbol{\phi}], \tag{2.12}$$

where $\boldsymbol{\phi} \in \mathbb{S}(\mathcal{X})$ is a probability distribution on $\mathcal{X}$. Compared with the definition (2.2) of the discounted reward, two points of contrast would strike us at once.

1. In (2.2), the existence of the sum is not in question, because $\gamma < 1$. However, in the present instance, there is no *a priori* reason to assume that the limit in (2.12) exists.

2. The value function V in (2.2) is associated with an initial state x_i. It is implicit in the definition that $V(x_i)$ need not equal $V(x_j)$ if $x_i \neq x_j$. In (2.12), the initial state is replaced by an initial distribution $\boldsymbol{\phi}$, which is more general. However, we write c^*, instead of $c^*(\boldsymbol{\phi})$, suggesting that the limit, if it exists, is independent of $\boldsymbol{\phi}$.

Theorem 2.3 presents a simple sufficient condition to address both of the above observations.

Theorem 2.3. *Suppose A is irreducible, and let $\boldsymbol{\mu}$ denote its unique stationary distribution. Then*

$$c^* = \boldsymbol{\mu}\mathbf{r} = E[R, \boldsymbol{\mu}], \ \forall \boldsymbol{\phi} \in \mathbb{S}(\mathcal{X}), \tag{2.13}$$

where $\mathbf{r}$ is the reward vector defined in (2.4)*.*

Proof. If $X_0 \sim \boldsymbol{\phi}$, then $X_t \sim \boldsymbol{\phi}A^t$. Therefore

$$E[R(X_t)|X_0 \sim \boldsymbol{\phi}] = \boldsymbol{\phi}A^t\mathbf{r}.$$

Also, as stated in Theorem 8.8, we have

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} A^t = \mathbf{1}_n\boldsymbol{\mu}.$$

Therefore

$$c^* = \boldsymbol{\phi}\left[\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} A^t\right]\mathbf{r} = \boldsymbol{\phi}\mathbf{1}_n\boldsymbol{\mu}\mathbf{r} = \boldsymbol{\mu}\mathbf{r} = E[R, \boldsymbol{\mu}], \tag{2.14}$$

because $\boldsymbol{\phi}\mathbf{1}_n = 1$. This is the desired result. □

Next we introduce an important concept known variously as the **bias** or the **transient reward**. For a discussion (albeit with "reward" replaced by "cost"), see [28, Section 8.2.3] or [1, Section 4.1].

Definition 2.1. Suppose A is primitive,[2] and define c^* as in (2.14) For each index i, the **transient reward** $J_i^* \in \mathbb{R}$ is defined as

$$J_i^* = \sum_{t=0}^{\infty}\{E[R(X_t|X_0 = x_i] - c^*\}. \tag{2.15}$$

A priori it is not clear why the sum in (2.15) is well-defined, because there is no averaging over time. It is now shown that the transient reward is indeed well-defined, and several explicit expressions are given for it.

Theorem 2.4. *Suppose A is primitive, and let $\boldsymbol{\mu}$ denote its stationary distribution. Define $M := \mathbf{1}_n\boldsymbol{\mu} \in [0,1]^{n\times n}$, and $\mathbf{J}^* \in \mathbb{R}^n$ as $[J_i^*]$. Then the following statements are true:*

1. *The vector $\mathbf{J}^*$ is well-defined.*

2. *An explicit expression for $\mathbf{J}^*$ is given by*

$$\mathbf{J}^* = (I - A + M)^{-1}(I - M)\mathbf{r} = (I - A + M)^{-1}(\mathbf{r} - c^*\mathbf{1}_n). \tag{2.16}$$

3. *The vector $\mathbf{J}^*$ satisfies the "Poisson equation"*

$$\mathbf{J} = \mathbf{r} - c^*\mathbf{1}_n + A\mathbf{J}. \tag{2.17}$$

Moreover, $\mathbf{J}^*$ is the unique solution of (2.17) *that satisfies*

$$\boldsymbol{\mu}\mathbf{J} = 0. \tag{2.18}$$

[2]This is equivalent to assuming that A is irreducible and aperiodic; see Theorem 8.7.

Proof. Note that $\boldsymbol{\mu}, \mathbf{1}_n$ are row and column eigenvectors of A corresponding to the eivenvalue $\lambda = 1$, and that all other eigenvalues of A have magnitude less than one. So if we define

$$A_2 = A - \mathbf{1}_n \boldsymbol{\mu} = A - M,$$

then the spectrum of A_2 is the same as that of A, except that the eigenvalue at 1 is replaced by 0. In particular, $\rho(A_2) < 1$, and as a consequence

$$\sum_{t=0}^{\infty} A_2^t = (I - A_2)^{-1} = (I - A + M)^{-1}. \tag{2.19}$$

Next, suppose $\mathbf{v} \in \mathbb{R}^n$ satisfies $\boldsymbol{\mu}\mathbf{v} = 0$. Then it is easy to verify that $A\mathbf{v} = A_2\mathbf{v}$, and moreover, $\boldsymbol{\mu} A_2 \mathbf{v} = 0$. Repeated application of this relationship shows that $A^t\mathbf{v} = A_2^t\mathbf{v}$, for all $t \geq 1$. Therefore, for every such $\mathbf{v}$, we have that

$$\sum_{t=0}^{\infty} A^t\mathbf{v} = \sum_{t=0}^{\infty} A_2^t\mathbf{v} = (I - A + M)^{-1}\mathbf{v}. \tag{2.20}$$

Now in particular, choose

$$\mathbf{v} = \mathbf{r} - c^*\mathbf{1}_n = (I - M)\mathbf{r}.$$

Then it follows from (2.14) that $\boldsymbol{\mu}\mathbf{v} = 0$. Hence (2.20) implies that

$$\sum_{t=0}^{\infty} A^t(\mathbf{r} - c^*\mathbf{1}_n) = (I - A + M)^{-1}(I - M)\mathbf{r}.$$

To prove Statements 1 and 2, let $\mathbf{e}_i$ denote the i-th elementary basis vector. Then $X_0 = x_i$ is equivalent to $X_0 \sim \mathbf{e}_i^\top$. Then $X_t \sim \mathbf{e}_t^\top A^t$, and

$$J_i^* = \sum_{t=0}^{\infty} [\mathbf{e}_i^\top A^t \mathbf{r} - c^*],$$

$$\begin{aligned} \mathbf{J}^* &= \sum_{t=0}^{\infty} (A^t\mathbf{r} - c^*\mathbf{1}_n) = \sum_{t=0}^{\infty} A^t(\mathbf{r} - c^*\mathbf{1}_n) \\ &= (I - A + M)^{-1}(I - M)\mathbf{r}. \end{aligned} \tag{2.21}$$

Here we use the fact that $c^*\mathbf{1}_n = c^*A^t\mathbf{1}_n$ for all t. This establishes Statements 1 and 2.

Now we come to Statement 3. From (2.15), we get

$$\begin{aligned} J_i^* &= \sum_{t=0}^{\infty} \{E[R(X_t|X_0 = x_i] - c^*\} \\ &= r_i - c^* + \sum_{t=1}^{\infty} \{E[R(X_t|X_0 = x_i] - c^*\} \\ &= r_i - c^* + \sum_{j=1}^{n} a_{ij} \sum_{t=1}^{\infty} \{E[R(X_t|X_1 = x_j] - c^*\} \\ &= r_i - c^* + \sum_{j=1}^{n} a_{ij} J_j^*, \end{aligned}$$

which is just (2.17) written out in component form. Hence $\mathbf{J}^*$ is a particular solution of (2.17).

Finally, observe that if J is another solution of (2.17), then $(\mathbf{J}^* - \mathbf{J}) = A(\mathbf{J}^* - \mathbf{J})$, which implies that $\mathbf{J} = \mathbf{J}^* + \alpha \mathbf{1}_n$ for some constant α. Thus $\{\mathbf{J}^* + \alpha \mathbf{1}_n : \alpha \in \mathbb{R}\}$ is the set of all solutions to (2.17). Now, since $\boldsymbol{\mu}(\mathbf{r} - c^* \mathbf{1}_n) = 0$, it follows that

$$\boldsymbol{\mu}\mathbf{J}^* = \boldsymbol{\mu} \sum_{t=0}^{\infty} A^t(\mathbf{r} - c^* \mathbf{1}_n) = \sum_{t=0}^{\infty} \boldsymbol{\mu}(\mathbf{r} - c^* \mathbf{1}_n) = 0.$$

Moreover, if $\boldsymbol{\mu}(\mathbf{J}^* + \alpha \mathbf{1}_n) = 0$, then $\alpha = 0$. Hence $\mathbf{J}^*$ is the unique solution of (2.17) that satisfies $\boldsymbol{\mu}\mathbf{J} = 0$. □

It is possible to give an alternate proof of Statement 3, and we do so now. Suppose $\mathbf{J}^*$ is given by (2.21). Observe that

$$\boldsymbol{\mu}(I - A + M) = \boldsymbol{\mu} M = \boldsymbol{\mu}, \text{ or } \boldsymbol{\mu}(I - A + M)^{-1} = \boldsymbol{\mu}.$$

Also, $\boldsymbol{\mu}(I - M) = \mathbf{0}$. Therefore

$$\boldsymbol{\mu}\mathbf{J}^* = \boldsymbol{\mu}(I - A + M)^{-1}(I - M)\mathbf{r} = \boldsymbol{\mu}(I - M)\mathbf{r} = 0.$$

Next, (2.21) implies that

$$(I - A + M)\mathbf{J}^* = (I - M)\mathbf{r}.$$

However, $M\mathbf{J}^* = \mathbf{1}_n \boldsymbol{\mu}\mathbf{J}^* = \mathbf{0}$, and $(I - M)\mathbf{r} = \mathbf{r} - c^* \mathbf{1}_n$. Therefore

$$\mathbf{J}^* - A\mathbf{J}^* = \mathbf{r} - c^* \mathbf{1}_n.$$

This is just (2.17). The above derivation avoids infinite sums.

Problem 2.1. Suppose the reward vector $\mathbf{r} \geq \mathbf{0}$. Show that if we carry out value iteration with $\mathbf{y}^0 = \mathbf{r}$, then the sequence of iterations $\{\mathbf{y}^i\}$ is componentwise nondecreasing, that is, $\mathbf{y}^{i+1} \geq \mathbf{y}^i$.

2.2 Markov Decision Processes

2.2.1 Markov Decision Processes: Problem Formulation

In a Markov process, the state X_t evolves on its own, according to a predetermined state transition matrix. In contrast, in a MDP, there is also another variable called the "action" which affects the dynamics. Specifically, in addition to the state space $\mathcal{X}$, there is also a finite set of actions $\mathcal{U}$. Each action $u_k \in \mathcal{U}$ leads to a distinct state transition matrix $A^{u_k} = [a_{ij}^{u_k}]$. So at time t, if the state is X_t, and an action $U_t \in \mathcal{U}$ is applied, then

$$\Pr\{X_{t+1} = x_j | X_t = x_i, U_t = u_k\} = a_{ij}^{u_k}. \tag{2.22}$$

Obviously, for each fixed $u_k \in \mathcal{U}$, the corresponding state transition matrix A^{u_k} is row-stochastic. In addition, there is also a "reward" function $R : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$. Note that in a Markov reward process, the reward depends only on the current state, whereas in a Markov decision process, the reward depends on both the current state as well as the action taken. As in Markov reward processes studied in Section 2.1, it is possible to permit R to be a random function of X_t and U_t as opposed to a deterministic function. Moreover, to be consistent with the earlier convention, it is assumed that the reward $R(X_t, U_t)$ is paid at the *next* time instant $t + 1$ and not at time t.

The most important aspect of an MDP is the concept of a "policy," which is just a systematic way of choosing U_t given X_t. One can make a distinction between deterministic and probabilistic policies. A deterministic policy is just a map from $\mathcal{X}$ to $\mathcal{U}$. A probabilistic policy is a map from $\mathcal{X}$ from the set of probability distributions on $\mathcal{U}$, denoted by $\mathbb{S}(\mathcal{U})$. Let Π_d, Π_p denote respectively the set of deterministic, and the set of probabilistic, policies. Clearly the number of deterministic policies is $|\mathcal{U}|^{|\mathcal{X}|}$, while Π_p is uncountable. Observe that a policy $\pi \in \Pi_d$ can be represented by a $|\mathcal{X}| \times |\mathcal{U}|$ matrix P, where each row of P contains a single one and the rest are zeros. Thus in row i, the one is in column $\pi(x_i)$ and the rest are

zeros. If $\pi \in \Pi_p$, then P need not be binary, but P must have only nonnegative elements, and the sum of each row must equal one.

Now we make an important observation. Whether a policy π is deterministic or probabilistic, the resulting stochastic process $\{X_t\}$ is Markov with the state transition matrix determined as follows: If $\pi \in \Pi_d$, then

$$\Pr\{X_{t+1} = x_j | X_t = x_i, \pi\} = a_{ij}^{\pi(x_i)}. \tag{2.23}$$

If $\pi \in \Pi_p$ and

$$\pi(x_i) = [\ \phi_{i1} \quad \cdots \quad \phi_{im}\ ], \tag{2.24}$$

where $m = |\mathcal{U}|$, then

$$\Pr\{X_{t+1} = x_j | X_t = x_i, \pi\} = \sum_{k=1}^{m} \phi_{ik} a_{ij}^{u_k}. \tag{2.25}$$

Equation (2.25) contains (2.24) as a special case, by setting $\phi_{ij} = 1$ if $\pi(x_i) = u_j$, and zero otherwise. In a similar manner, for every policy π, the reward function $R : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ can be converted into a reward map $R_\pi : \mathcal{X} \to \mathbb{R}$, as follows: If $\pi \in \Pi_d$, then

$$R_\pi(x_i) = R(x_i, \pi(x_i)), \tag{2.26}$$

whereas if $\pi \in \Pi_p$, then

$$R_\pi(x_i) = \sum_{k=1}^{m} \phi_{ik} R(x_i, u_k). \tag{2.27}$$

Despite its simplicity, the above observation is very useful, because it states that a Markov process with an action variable remains a Markov process under any choice of policy, deterministic or probabilistic. The reason why this is so is that the policy has no memory. To illustrate, suppose $\pi : \mathcal{X} \to \mathcal{U}$ is a deterministic policy, with (for example) $\pi(x_i) = u_k$. Then the action variable $U(t)$ equals u_k, whenever the state $X(t)$ equals x_i; the time t at which this happens is immaterial. Similar remarks apply to the reward function. For this reason, we define A^π to be the state transition matrix that results from applying the policy π, and R_π to be the reward function that results from applying the policy π.

For a MDP, one can pose three questions:

1. **Policy evaluation:** For a given policy π, define $V_\pi(x_i)$ to be the "value" associated with the policy π and initial state x_i, that is, the expected discounted future reward with $X_0 = x_i$. How can $V_\pi(x_i)$ be computed for each $x_i \in \mathcal{X}$?

2. **Optimal Value Determination:** For a specified initial state x_i, define

$$V^*(x_i) := \max_{\pi \in \Pi_p} V_\pi(x_i), \tag{2.28}$$

   to be the **optimal value** over all policies for that initial state. How can $V^*(x_i)$ be computed? Note that in (2.28), the optimum is taken over all *probabilistic* policies. It is shown in Theorem 2.9 in the sequel that the optimum can actually be achieved by a *deterministic* policy.

3. **Optimal Policy Determination:** Define the **optimal policy** map $\mathcal{X} \to \Pi_d$ via

$$\pi^*(x_i) := \arg\max_{\pi \in \Pi_d} V_\pi(x_i). \tag{2.29}$$

   How can the optimal policy map π^* be determined? Note that we can restrict to $\pi \in \Pi_d$ because, as stated above, the maximum over $\pi \in \Pi_p$ is not any larger. Moreover, it is again shown in Theorem 2.9 that there exists *one common optimal policy for all initial states.*

2.2.2 Markov Decision Processes: Solution

In this subsection we present answers to the three questions above.

Policy Evaluation:

Suppose a policy $\pi \in \Pi_d$ is specified. Then the corresponding state transition matrix and reward are given by (2.23) and (2.26) respectively. Now suppose we define the vector $\mathbf{v}_\pi$ by

$$\mathbf{v}_\pi = [\ V_\pi(x_1) \quad \dots \quad V_\pi(x_n)\ ], \tag{2.30}$$

and the reward vector $\mathbf{r}_\pi$ by

$$\mathbf{r}_\pi = [\ R_\pi(x_1) \quad \dots \quad R_\pi(x_n)\ ], \tag{2.31}$$

where $R(x_i)$ is defined by (2.26) or (2.27) as appropriate. Then it readily follows from Theorem 2.1 that $\mathbf{v}_\pi$ satisfies an equation analogous to (2.6), namely

$$\mathbf{v}_\pi = \mathbf{r}_\pi + \gamma A^\pi \mathbf{v}_\pi. \tag{2.32}$$

As before, it is inadvisable to compute $\mathbf{v}_\pi$ via $\mathbf{v}_\pi = (I - \gamma A^\pi)^{-1}\mathbf{r}_\pi$. Instead, one should use value iteration to solve (2.32).

For future use we introduce another function $Q : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, known as the **action-value function**, which is defined as follows:

$$Q_\pi(x_i, u_k) := R(x_i, u_k) + E_\pi \left[\sum_{t=1}^{\infty} \gamma^t R_\pi(X_t) | X_0 = x_i, U_0 = u_k \right]. \tag{2.33}$$

Apparently this function was first defined in [47]. Note that Q_π is defined only for deterministic policies. In principle it is possible to define it for probabilistic policies, but this is not commonly done. In the above definition, the expectation E_π is with respect to the evolution of the state X_t under the policy π. When the reward is a random function of X_t and U_t, then inside the summation we would need to take the expected value of $R(X_t, \pi(X_t))$ for a deterministic policy.

The way in which a MDP is set up is that at time t, the Markov process reaches a state X_t, based on the previous state X_{t-1} and the state transition matrix A^π corresponding to the policy π. Once X_t is known, the policy π determines the action $U_t = \pi(X_t)$, and then the reward $R_\pi(X_t) = R(X_t, \pi(X_t))$ is generated at time $t+1$. In particular, when defining the value function $V_\pi(x_i)$ corresponding to a policy π, we start off the MDP in the initial state $X_0 = x_i$, *and choose the action* $U_0 = \pi(x_i)$. However, in defining the action-value function Q, we do not feel compelled to set $U_0 = \pi(X_0) = \pi(x_i)$, and can choose an arbitrary action $u_k \in \mathcal{U}$. From $t = 1$ onwards however, the action U_t is chosen as $U_t = \pi(X_t)$. This seemingly small change leads to some simpifications. Specifically, it will be seen in later chapters that it is often easier to approximate (or to "learn") the action-value function than it is to approximate the value function.

Just as we can interpret $V : \mathcal{X} \to \mathbb{R}$ as a $|\mathcal{X}|$-dimensional vector, we can interpret $Q : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ as an $|\mathcal{X}| \cdot |\mathcal{U}|$-dimensional vector. Consequently the Q-vector has higher dimension than the value vector.

Theorem 2.5. *The function Q satisfies the recursive relationship*

$$Q_\pi(x_i, u_k) = R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} Q_\pi(x_j, \pi(x_j)). \tag{2.34}$$

Proof. Observe that at time $t = 0$, the state transition matrix is A^{u_k}. So, given that $X_0 = x_i$ and $U_0 = u_k$, the next state X_1 has the distribution

$$X_1 \sim [a_{ij}^{u_k}, j = 1, \cdots, n].$$

Moreover, $U_1 = \pi(X_1)$ because the policy π is implemented from time $t = 1$ onwards. Therefore

$$\begin{aligned} Q_\pi(x_i, u_k) &= R(x_i, u_k) + E_\pi \left[\sum_{j=1}^n a_{ij}^{u_k} \left(\gamma R(x_j, \pi(x_j)) + \sum_{t=2}^\infty \gamma^t R_\pi(X_t) | X_1 = x_j, U_1 = \pi(x_j) \right) \right] \\ &= R(x_i, u_k) + E_\pi \left[\gamma \sum_{j=1}^n a_{ij}^{u_k} \left(R(x_j, \pi(x_j)) + \sum_{t=1}^\infty \gamma^t R_\pi(X_t) | X_1 = x_j, U_1 = \pi(x_j) \right) \right] \\ &= R(x_i, u_k) + \gamma \sum_{j=1}^n a_{ij}^{u_k} Q(x_j, \pi(x_j)). \end{aligned}$$

This is the desired conclusion. Note that in the above summation, we have written $R(X_t)$ for reward to be paid at time $t + 1$. □

Theorem 2.6. *The functions V_π and Q_π are related via*

$$V_\pi(x_i) = Q_\pi(x_i, \pi(x_i)). \tag{2.35}$$

Proof. If we choose $u_k = \pi(x_i)$ then (2.34) becomes

$$Q_\pi(x_i, \pi(x_i)) = R_\pi(x_i) + \gamma \sum_{j=1}^n a_{ij}^{\pi(x_j)} Q(x_j, \pi(x_j)).$$

This is the same as (2.22) written out componentwise. We know that (2.22) has a unique solution. This shows that (2.35) holds. □

In view of (2.35), the recursive equation for Q_π can be rewritten as

$$Q_\pi(x_i, u_k) = R(x_i, u_k) + \gamma \sum_{j=1}^n a_{ij}^{u_k} V_\pi(x_j). \tag{2.36}$$

Optimal Value Determination:

For a policy $\pi \in \Pi_d$ or $\pi \in \Pi_p$, define the associated map $T_\pi : \mathbb{R}^n \to \mathbb{R}^n$ via

$$T_\pi \mathbf{v} = \mathbf{r}_\pi + \gamma A^\pi \mathbf{v}. \tag{2.37}$$

Then it follows from Theorem 2.2 that T_π is monotone and is a contraction with respect to the ℓ_∞-norm, with contraction constant γ.

Now we introduce one of the key ideas in Markov Decision Processes. Define the **Bellman iteration map** $B : \mathbb{R}^n \to \mathbb{R}^n$ via

$$(B\mathbf{v})_i := \max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j=1}^n a_{ij}^{u_k} v_j \right]. \tag{2.38}$$

Theorem 2.7. *The map B is monotone and a contraction with respect to the ℓ_∞-norm.*

Proof. The theorem has two claims: The first claim is that the map B is monotone, meaning that if $\mathbf{v}_1 \leq \mathbf{v}_2$ componentwise, then $B(\mathbf{v}_1) \leq B(\mathbf{v}_2)$ componentwise. The second claim is that B is a contraction with respect to the ℓ_∞-norm. Note that, unlike the value iteration map T_π defined in (2.37), the map B is not affine.

Let us begin with the first claim. Suppose $\mathbf{v}_1 \leq \mathbf{v}_2$. Then

$$\begin{aligned}(B(\mathbf{v}_1))_i &= \max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} v_{1j} \right] \\ &\leq \max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} v_{2j} \right] \\ &= (B(\mathbf{v}_2))_i.\end{aligned}$$

Here we use the fact that $a_{ij}^{u_k} \geq 0$ for all i, j. This establishes that B is monotone, which is the first claim.

The proof of the second claim is a bit more elaborate. We begin by establishing that

$$\left| \max_{u_k \in \mathcal{U}} g(x_i, u_k) - \max_{u_k \in \mathcal{U}} h(x_i, u_k) \right| \leq \max_{u_k \in \mathcal{U}} |g(x_i, u_k) - h(x_i, u_k)|, \ \forall x_i \in \mathcal{X}. \tag{2.39}$$

To prove (2.39), we begin with the obvious observation that, if α, β are real numbers, then

$$\alpha - \beta \leq |\alpha - \beta| \implies \alpha \leq |\alpha - \beta| + \beta.$$

Note that this inequality holds irrespective of the signs of α and β. Fix $x_i \in \mathcal{X}, u_k \in \mathcal{U}$ and apply the above inequality with $\alpha = g(x_i, u_k), \beta = h(x_i, u_k)$. This gives

$$g(x_i, u_k) \leq |g(x_i, u_k) - h(x_i, u_k)| + h(x_i, u_k).$$

Now take the maximum of both sides over $u_k \in \mathcal{U}$. This gives

$$\begin{aligned}\max_{u_k \in \mathcal{U}} g(x_i, u_k) &\leq \max_{u_k \in \mathcal{U}} [|g(x_i, u_k) - h(x_i, u_k)| + h(x_i, u_k)] \\ &\leq \max_{u_k \in \mathcal{U}} |g(x_i, u_k) - h(x_i, u_k)| + \max_{u_k \in \mathcal{U}} h(x_i, u_k).\end{aligned}$$

Rearranging gives

$$\max_{u_k \in \mathcal{U}} g(x_i, u_k) - \max_{u_k \in \mathcal{U}} h(x_i, u_k) \leq \max_{u_k \in \mathcal{U}} |g(x_i, u_k) - h(x_i, u_k)|.$$

By symmetry, we can interchange g and h, which gives

$$\max_{u_k \in \mathcal{U}} h(x_i, u_k) - \max_{u_k \in \mathcal{U}} g(x_i, u_k) \leq \max_{u_k \in \mathcal{U}} |g(x_i, u_k) - h(x_i, u_k)|.$$

Combining these two inequalities gives (2.39).

Now we make use of (2.39) to show that B is a contraction with respect to the ℓ_∞-norm. Let $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^n$ be arbitrary, and fix $x_i \in \mathcal{X}$. Then, by using the definition of B and (2.39), we get

$$\begin{aligned}|(B(\mathbf{v}_1))_i - (B(\mathbf{v}_2))_i| &= \left| \max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} v_{1j} \right] - \max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} v_{2j} \right] \right| \\ &\leq \max_{u_k \in \mathcal{U}} \left| R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} v_{1j} - R(x_i, u_k) - \gamma \sum_{j=1}^{n} a_{ij}^{u_k} v_{2j} \right| \\ &= \max_{u_k \in \mathcal{U}} \gamma \left| \sum_{j=1}^{n} a_{ij}^{u_k} (v_{1j} - v_{2j}) \right| \leq \max_{u_k \in \mathcal{U}} \gamma \left| \sum_{j=1}^{n} a_{ij}^{u_k} |v_{1j} - v_{2j}| \right| \\ &\leq \gamma \|\mathbf{v}_1 - \mathbf{v}_2\|_\infty.\end{aligned} \tag{2.40}$$

Here we use the facts

$$|v_{1j} - v_{2j}| \leq \|\mathbf{v}_1 - \mathbf{v}_2\|_\infty \ \forall j, \sum_{j=1}^{n} a_{ij}^{u_k} = 1, \ \forall i, \ \forall u_k \in \mathcal{U}$$

Because the inequality (2.40) holds for *every* index i, it follows that

$$\|B(\mathbf{v}_1) - B(\mathbf{v}_2)\|_\infty \leq \gamma \|\mathbf{v}_1 - \mathbf{v}_2\|_\infty.$$

This shows that the map B is a contraction with respect to the ℓ_∞-norm, which is the second claim. □

Theorem 2.8. *Define $\bar{\mathbf{v}} \in \mathbb{R}^n$ to be the unique fixed point of B, and define $\mathbf{v}^* \in \mathbb{R}^n$ to equal $[V^*(x_i), x_i \in \mathcal{X}]$, where $V^*(x_i)$ is defined in* (2.28)*. Then $\bar{\mathbf{v}} = \mathbf{v}^*$.*

Proof. By definition, for every $\pi \in \Pi_d$, we have that

$$\begin{aligned} [T_\pi(\bar{\mathbf{v}})]_i &= R(x_i, \pi(x_i)) + \sum_{j=1}^{n} a_{ij}^{\pi(x_i)} \bar{V}_j \\ &\leq \max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} \bar{V}_j \right] = (B(\bar{\mathbf{v}}))_i = \bar{V}_i, \end{aligned} \tag{2.41}$$

because $\bar{\mathbf{v}}$ is a fixed point of the map B. If $\pi \in \Pi_p$, say

$$\pi(x_i) = [\ \phi_{i1} \quad \cdots \quad \phi_{im} \ ] \in \mathbb{S}_m,$$

then

$$\begin{aligned} [T_\pi(\mathbf{v})]_i &= \sum_{l=1}^{l} \phi_{il} \left[R(x_i, u_l) + \sum_{j=1}^{n} a_{ij}^{u_l} \bar{V}_j \right] \\ &\leq \max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \sum_{j=1}^{n} a_{ij}^{u_k} \bar{V}_j \right] \\ &= (B(\bar{\mathbf{v}}))_i = \bar{V}_i. \end{aligned} \tag{2.42}$$

Because (2.41) and (2.42) hold for every index i, it follows that

$$T_\pi(\bar{\mathbf{v}}) \leq \bar{\mathbf{v}}.$$

Next, because T_π is monotone as per Theorem 2.2, it follows that

$$T_\pi^2(\bar{\mathbf{v}}) = T_\pi(T_\pi(\bar{\mathbf{v}})) \leq T_\pi(\bar{\mathbf{v}}) \leq \bar{\mathbf{v}}.$$

The reasoning can be repeated to show that

$$T_\pi^l(\bar{\mathbf{v}}) \leq \bar{\mathbf{v}}, \ \forall l.$$

Now let $l \to \infty$. Then the left side approaches the fixed point of the map T_π, which is $\mathbf{v}_\pi$. Thus we conclude that, for all policies in Π_d or Π_p, we have that

$$\mathbf{v}_\pi \leq \bar{\mathbf{v}}. \tag{2.43}$$

Therefore, for each $x_i \in \mathcal{X}$, we infer that

$$V^*(x_i) = \max_\pi V(x_i) \leq \bar{V}_i, \ \forall i, \text{ or } \mathbf{v}^* \leq \bar{\mathbf{v}}. \tag{2.44}$$

To show that $\bar{\mathbf{v}} \leq \mathbf{v}^*$, define a deterministic policy $\bar{\pi} \in \Pi_d$ by

$$\bar{\pi}(x_i) = \arg\max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \sum_{j=1}^{n} a_{ij}^{u_k} \bar{V}_j \right]. \tag{2.45}$$

In case of ties, choose any deterministic tie-breaking rule, e.g., choose the u_k with the lowest index. Then, since the right side of (2.45) equals $(B(\bar{\mathbf{v}}))_i = \bar{V}_i$, we conclude that

$$\bar{V}_i = R(x_i, \bar{\pi}(x_i)) + \sum_{j=1}^{n} a_{ij}^{\bar{\pi}(x_i)} \bar{V}_j, \ \forall i. \tag{2.46}$$

Hence $T_{\bar{\pi}}(\bar{\mathbf{v}}) = \bar{\mathbf{v}}$. But since $T_{\bar{\pi}}$ is a contraction, it has a unique fixed point, which shows that $\bar{V}_i = V_{\bar{\pi}}(x_i)$ for all i. Therefore, for each index i, we have that

$$\bar{V}_i = V_{\bar{\pi}}(x_i) \leq V^*(x_i), \ \forall i, \text{ or } \bar{\mathbf{v}} \leq \mathbf{v}^*.$$

Taken together with (2.43), this shows that $\bar{\mathbf{v}} = \mathbf{v}^*$. □

By replacing $\bar{\mathbf{v}}$ in Theorem 2.8 by $\mathbf{v}^*$ (which equals $\bar{\mathbf{v}}$), we derive the following fundamental result for Markov Decision Processes.

Theorem 2.9. *Define the optimal value function $V^*(x_i)$ as in (2.28). Then*

1. *The optimal value function $V^* : \mathcal{X} \to \mathbb{R}$ is the unique solution of the following recursive relationship, known as the **Bellman optimality equation**:*

$$V^*(x_i) = \max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} V^*(x_j) \right]. \tag{2.47}$$

2. *There is at least one deterministic policy $\pi \in \Pi_d$ such that*

$$V_\pi(x_i) = V^*(x_i), \ \forall i \in \mathcal{X}. \tag{2.48}$$

Specifically, the policy $\bar{\pi}$ defined by restating (2.45) with $\bar{V}_j$ replaced by V_j^*, namely

$$\pi^*(x_i) = \arg\max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \sum_{j=1}^{n} a_{ij}^{u_k} V_j^* \right]. \tag{2.49}$$

satisfies (2.48) and is thus an optimal policy.

Note that Item 2 of the theorem states that enlarging the policy space to include probabilistic policies *does not* increase the maximum value. Also, there is *one common policy* that achieves the optimal value for *every* state x_i. Perhaps neither of these statements is obvious on the surface.

In analogy with the optimal value function, we can also define an optimal action-value function.

Theorem 2.10. *Define $Q^* : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ by*

$$Q^*(x_i, u_k) = R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} V^*(x_j). \tag{2.50}$$

Then $Q^*(\cdot,\cdot)$ satisfies the following relationships:

$$Q^*(x_i,u_k) = R(x_i,u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} \max_{w_l \in \mathcal{U}} Q^*(x_j,w_l). \tag{2.51}$$

$$V^*(x_i) = \max_{u_k \in \mathcal{U}} Q^*(x_i,u_k), \tag{2.52}$$

Moreover, every policy $\pi \in \Pi_d$ such that

$$\pi^*(x_i) = \arg\max_{u_k \in \mathcal{U}} Q^*(x_i,u_k) \tag{2.53}$$

is optimal.

Proof. Since $Q^*(\cdot,\cdot)$ is defined by (2.50), it follows that

$$\max_{u_k \in \mathcal{U}} Q^*(x_i,u_k) = \max_{u_k \in \mathcal{U}} \left[R(x_i,u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} V^*(x_j) \right] = V^*(x_i),$$

by (2.47). This establishes (2.52) and (2.53). Substituting from (2.52) into (2.50) gives (2.51). □

Now we define an iteration on action-functions that is analogous to (2.38) for value functions. As with the value function, the action-value function can either be viewed as a map $Q : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, or as a vector in $\mathbb{R}^{|\mathcal{X}|\cdot|\mathcal{U}|}$. Define $F : \mathbb{R}^{|\mathcal{X}|\times|\mathcal{U}|} \to \mathbb{R}^{|\mathcal{X}|\times|\mathcal{U}|}$ by

$$[F(Q)](x_i,u_k) := R(x_i,u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} \max_{w_l \in \mathcal{U}} Q(x_j,w_l). \tag{2.54}$$

Theorem 2.11. *The map F is monotone and is a contraction. Therefore for all $Q_0 : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, the sequence of iterations $\{F^t(Q_0)\}$ converges to Q^* as $t \to \infty$.*

Proof. The proof is very similar to that of Theorem 2.9. Given a map $Q : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, define the associated map $\mathcal{M}(Q) : \mathcal{X} \to \mathbb{R}$ by

$$[\mathcal{M}(Q)](x_i) = \max_{u_k \in \mathcal{U}} Q(x_i,u_k),$$

and rewrite (2.54) as

$$[F(Q)](x_i,u_k) := R(x_i,u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} [\mathcal{M}(Q)](x_j). \tag{2.55}$$

Also, if $Q, Q' : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, let $Q \le Q'$ denote that $Q(x_i,u_k) \le Q'_i(x_i,u_k)$ for all x_i, u_k. Then it is clear that if $Q \le Q'$, then $\mathcal{M}(Q) \le \mathcal{M}(Q')$. Because $a_{ij}^{u_k}$ is always nonnegative, it follows that the map F is monotone.

Next, as in the proof of Theorem 2.7, for arbitrary maps $Q_1, Q_2 : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, we have

$$\begin{aligned} |[\mathcal{M}(Q_1)](x_i) - [\mathcal{M}(Q_2)](x_i)| &= \left| \max_{u_k \in \mathcal{U}} Q_1(x_i,u_k) - \max_{u_k \in \mathcal{U}} Q_2(x_i,u_k) \right| \\ &\le \max_{u_k \in \mathcal{U}} |Q_1(x_i,u_k) - Q_2(x_i,u_k)|, \ \forall x_i \in \mathcal{X}. \end{aligned}$$

As a result

$$\|\mathcal{M}(Q_1) - \mathcal{M}(Q_2)\|_\infty \le \|Q_1 - Q_2\|_\infty.$$

Substituting this into (2.55) gives

$$\|F(Q_1) - F(Q_2)\|_\infty \le \gamma \|Q_1 - Q_2\|_\infty. \tag{2.56}$$

The desired conclusion now follows. □

If we were to rewrite (2.47) and (2.51) in terms of expected values, the advantages of the Q-functioon would become apparent. We can rewrite (2.47) as

$$V^*(X_t) = \max_{U_t \in \mathcal{U}} \{R(X_t, U_t) + \gamma E[V^*(X_{t+1})|X_t]\}, \tag{2.57}$$

and (2.51) as

$$Q^*(X_t, U_t) = R(X_t, U_t) + \gamma E\left[\max_{U_{t+1} \in \mathcal{U}} Q^*(X_{t+1}, U_{t+1})\right]. \tag{2.58}$$

Thus in the Bellman formulatioon and iteration, the maximization occurs *outside* the expectation, whereas with the Q-formulation and F-iteration, the maximization occurs *inside* the expectation. As shown in later chapter, learning Q^* is easier than learning V^*.

The idea of learning Q^* instead of learning V^* is introduced in [47].

Optimal Policy Determination:

Theorems 2.8 and 2.9 together show the following: Start with any initial guess $\mathbf{v}_0 \in \mathbb{R}^n$, and apply the Bellman iteration B defined in (2.38). Then the sequence $\{\mathbf{v}_k\}$ with $\mathbf{v}_{k+1} = B\mathbf{v}_k$ converges monotonically to the optimal value $\mathbf{v}^*$. Once $\mathbf{v}^*$ is determined, then an optimal policy can be determined using (2.49). This approach to determining $\mathbf{v}^*$ is known as **value iteration**. While this is a useful result, a shortcoming is that the intermediate vectors $\mathbf{v}_k$ do not necessarily correspond to any policy. An easy remedy is to choose the starting point of the iterations $\mathbf{v}_0$ to be the value of some policy π_0. Then each successive iteration $\mathbf{v}_k$ also corresponds to a policy π_k. In this way, we generate a sequence of suboptimal policies π_k with the property that the associated value vector $\mathbf{v}_k = \mathbf{v}_{\pi_k}$ converges to the optimal value. This approach is known as **policy iteration**. This is made precise as follows:

Theorem 2.12. *Choose an arbitrary policy $\pi_0 \in \Pi_d$, and compute the corresponding value $\mathbf{v}_{\pi_0}$. At the k-th iteration, choose an updated policy $\pi_{k+1} \in \Pi_d$ according to*

$$\pi_{k+1}(x_i) = \arg\max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} (\mathbf{v}_{\pi_k})_j\right]. \tag{2.59}$$

Then

1. *$\mathbf{v}_{\pi_{k+1}} \geq \mathbf{v}_{\pi_k}$, where the dominance is componentwise.*
2. *$\{\mathbf{v}_{\pi_k}\} \uparrow \mathbf{v}^*$ as $k \to \infty$.*

The proof is quite straightforward. The key step is to verify that if we define the updated policy π_{k+1} according to (2.59), then the corresponding value $\mathbf{v}_{\pi_{k+1}}$ is just $B\mathbf{v}_{\pi_k}$; but this is obvious.

Example 2.2. Now we return to the game of Blackjack. A detailed discussion of the game is given in [33, Example 5.1]. To describe the original game briefly, it is played between a player and the "House." (It is possible to have more than one player playing against the House, but we don't study that problem in the interests of simplicity.) At each turn, the player and the House have the option of drawing a card ("hit") or not drawing ("stick"). Each card is counted as its face value, with picture cards counted as 10. An ace can count as either 1 of 11 at the player's preference. The objective of the player is to exceed the total of the House without going over 21.

From the description, it is obvious that if the player's current total is eleven or less, then the best strategy is to hit, because there is no chance of losing on the next draw. Hence the issue of what to do arises only when the player's total reaches 12 or higher. Indeed, if the target were to be changed to some number N, then it is clear that if the player's total is $N - 10$ or less, then the correct solution is to hit. It can also be assumed that the probability of any particular card being the next card drawn is the same, no matter what

cards have been drawn until then (infinitely many card decks being used). In the original Blackjack game, only one card of the House is visible. In what follows, for the purposes of illustration, we eliminate all of these complications, and introduce a simplified game.

Suppose that, instead of drawing a card, the player rolls a fair four-sided die. Since there are only four possible outcomes, irrespective of what the target total might be, it is reasonable to suppose that the state P_t of the player lies in the set $\{0, 1, 2, 3, W, L\}$, with 0 being the start state. It can be assumed that the *current* state is in $\{0, 1, 2, 3\}$, while W and L are terminal states. To simplify the problem further, suppose that the House adopts the strategy that it does not roll the die further once its state is in $\{1, 2, 3\}$ (i.e., it does not try for a win from any of these states). Therefore the state H_t of the house lies in the set $\{1, 2, 3\}$. The overall state (P_t, H_t) lies in the Cartesian product $\{0, 1, 2, 3, W, L\} \times \{1, 2, 3\}$. Out of these, there are twelve possible *current* states, namely $\{0, 1, 2, 3\} \times \{1, 2, 3\}$ where the first number is the state of the player and the second is the state of the House. If the player rolls the die, the possible *next* states are $\{1, 2, 3, W, L\} \times \{1, 2, 3\}$, or a total of fifteen states. In this game, as in the snakes and ladders game, the reward is random and is a function of the next state.

As a part of the problem statement, we need to specify the dynamics of the Markov process. For the House, it does not play, so its state transition matrix is the 3×3 identity matrix, which ensures that $H_{t+1} = H_t$. As for the player's state P_t, if the action is to "stick," then the state transition matrix A^S is the 5×5 identity matrix. If the action is to "hit," then the state transition matrix A^H is given by

$A^H =$

	0	1	2	3	W	L
0	0	0.25	0.25	0.25	0.25	0
1	0	0	0.25	0.25	0.25	0.25
2	0	0	0	0.25	0.25	0.50
3	0	0	0	0	0.25	0.75
W	0	0	0	0	1	0
L	0	0	0	0	0	1

.

To complete the problem formulation, we need to specify the reward. Unlike the state transition matrix above, which is based on nothing more than the assumption that all four outcomes of the die are equally likely, the reward is to some extent arbitrary. Let us assign the following rewards:

$P_t > H_t$	2
$P_t = H_t$	1
$P_t < H_t$	0
$P_t = W$	5
$P_t = L$	-5

With this problem specification, we should strive to find an optimal policy. Note that the action space $\mathcal{U} = \{H, S\}$ (for "hit" or "stick") has cardinality two. Hence the number of policies is $2^{12} = 4,096$, which is already large enough that simply enumerating all possibilities is not practicable.[3] Hence some kind of policy iteration is the only way.

For evaluating *a specific* policy, it can be noted that the duration of the game cannot exceed four time steps. This is because the player's position has to advance by at least one at each time step. So discount factors very close to 1 do not make sense. The discount γ should be chosen much smaller, say 0.5.

Problem 2.2. Suppose that a Markov decision problem has four states and two actions. Suppose further that the two row-stochastic matrices corresponding to the two actions are as follows:

$$A^{u_1} = \begin{bmatrix} 0.1 & 0.3 & 0.3 & 0.3 \\ 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.4 & 0.4 & 0.2 \\ 0.4 & 0.2 & 0.2 & 0.2 \end{bmatrix}, A^{u_2} = \begin{bmatrix} 0.3 & 0.2 & 0 & 0.5 \\ 0.1 & 0.1 & 0.2 & 0.6 \\ 0.2 & 0.5 & 0.1 & 0.2 \\ 0 & 0.1 & 0.5 & 0.4 \end{bmatrix}.$$

[3]For the full Blackjack game, the number of policies is 2^{200} as shown in [33, Example 5.1].

Suppose further that the reward map $R : \mathcal{X} \times \mathcal{U}$ is as follows (note that we write e.g., $(3,1)$ instead of (x_3, u_1) to save space):

$$R = \begin{array}{|c|c|c|c|c|c|c|c|} \hline (1,1) & (1,2) & (2,1) & (2,2) & (3,1) & (3,2) & (4,1) & (4,2) \\ \hline 2 & 5 & -1 & 4 & 3 & 3 & 6 & -1 \\ \hline \end{array}.$$

- Suppose we define a deterministic policy π by

$$\pi = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

  In other words, $\pi(x_1) = u_2, \pi(x_2) = u_1, \pi(x_3) = u_1, \pi(x_4) = u_2$. Compute the corresponding state transition matrix A^π and reward map R_π.

- Suppose we define a probabilistic policy π by

$$\pi = \begin{bmatrix} 0.3 & 0.4 & 0.2 & 0.6 \\ 0.7 & 0.6 & 0.8 & 0.4 \end{bmatrix}.$$

  Compute the corresponding state transition matrix A^π and reward map R_π.

- How many deterministic policies can there be for this problem?

- With a discount factor of $\gamma = 0.9$, compute the optimal value and optimal policy using Theorem 2.12.

Problem 2.3. Prove Theorem 2.11.

Problem 2.4. Using the policy iteration method of Theorem 2.12, compute the optimal value function and optimal policy for the Markov decision process of Problem 2.2.

Problem 2.5. Formulate the simplified blackjack game as a Markov Decision Problem with discount factor $\gamma = 0.5$ and find the optimal policy using policy iteration.

Chapter 3

Stochastic Approximation

In this chapter, we discuss stochastic approximation (SA), which provides the mathematical foundation for many of the reinforcement learning (RL) algorithms presented in subsequent chapters. In RL, the learner attempts to identify an optimal (or nearly optimal) policy for an MDP with possibly unknown dynamics. Even if the MDP dynamics were to be known, computing an optimal policy using the policy iteration approach described in Theorem 2.12 would require, at each iteration, the *exact* computation of the value function associated with the policy. One of the applications of SA is that the policy can be updated even as the value function is being estimated. In the case where the MDP dynamics are not known, the learner has two possible approaches:

1. Estimate the MDP parameters and use this estimate to formulate the corresponding optimal policy. This is often referred to as "indirect" RL, drawing inspiration from the phrase "indirect adaptive control" from control theory.

2. Directly estimate the optimal policy without attempting to estimate the MDP parameters. This is often referred to as "direct" RL, again drawing upon the phrase "direct adaptive control."

We will see that SA is very useful in analyzing direct RL.

3.1 An Overview of Stochastic Approximation

Stochastic approximation (SA) can be viewed as an iterative technique for finding a zero of a function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ using noisy measurements. It is not necessary for the function to be "known" (e.g., in closed form). All that is required is that, given an argument $\boldsymbol{\theta} \in \mathbb{R}^d$, an "oracle" gives us a noise-corrupted measurement in the form

$$\mathbf{y}_{t+1} = \mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}, \tag{3.1}$$

where $\{\boldsymbol{\xi}_t\}_{t\geq 1}$ is a noise sequence. We defer a discussion of the nature of the noise sequence to a later point in this chapter. For the moment, let us focus on how the noisy measurements could be used to construct a sequence of approximations $\{\boldsymbol{\theta}_t\}$ that we hope would converge to a $\boldsymbol{\theta}^* \in \mathbb{R}^d$ such that $\mathbf{f}(\boldsymbol{\theta}^*) = \mathbf{0}$.

Observe that a solution to $\mathbf{f}(\boldsymbol{\theta}) = \mathbf{0}$ is also a minimizer of $h(\boldsymbol{\theta}) = (1/2)\|\mathbf{f}(\boldsymbol{\theta})\|_2^2$. Moreover, the gradient of $h(\cdot)$ is given by

$$\nabla h(\boldsymbol{\theta}) = \mathbf{f}(\boldsymbol{\theta}).$$

If we had access to noise-free measurements of $\mathbf{f}(\cdot)$, we could apply the steepest descent algorithm to minimize $h(\cdot)$, as follows: Pick some $\boldsymbol{\theta}_0 \in \mathbb{R}^d$, and at step t, define

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \nabla h(\boldsymbol{\theta}) = \boldsymbol{\theta}_t - \alpha_t \mathbf{f}(\boldsymbol{\theta}_t), \tag{3.2}$$

where $\{\alpha_t\}$ is a predetermined sequence of step sizes.[1] Note that α_t is permitted to be a random number; in this case, its probability distribution depends only on information up to and including time t. This is made precise later on. If only noisy measurements of the form (3.1) are available, then (3.2) is replaced by

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \mathbf{y}_{t+1} = \boldsymbol{\theta}_t - \alpha_t[\mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}]. \tag{3.3}$$

However, there is some ambiguity about the updating rule (3.3). Note that $h(\boldsymbol{\theta})$ also equals $\| - \mathbf{f}(\boldsymbol{\theta})\|_2^2$. So we could just as easily use the updating rule

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \mathbf{y}_{t+1} = \boldsymbol{\theta}_t + \alpha_t[\mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}]. \tag{3.4}$$

Which one should we choose?

The answer is that it does not matter so long as we are consistent. More to the point, the answer depends on what we think the shape of the function $\mathbf{f}(\cdot)$ is. As above, let $\boldsymbol{\theta}^*$ satisfy $\mathbf{f}(\boldsymbol{\theta}^*) = \mathbf{0}$. If $\mathbf{f}(\cdot)$ satisfies

$$\langle \boldsymbol{\theta} - \boldsymbol{\theta}^*, \mathbf{f}(\boldsymbol{\theta}) - \mathbf{f}(\boldsymbol{\theta}^*) \rangle < 0 \tag{3.5}$$

if $\boldsymbol{\theta} \neq \boldsymbol{\theta}^*$, then we should use the updating (3.4). However, if the sign is reversed in (3.5), then we should use the updating rule (3.3). Going forward, we will use (3.4). The main reason is that, if the function $\mathbf{f}(\cdot)$ satisfies (3.5), then under mild additional conditions $\boldsymbol{\theta}^*$ is a globally attractive equilibrium of the associated differential equation

$$\dot{\boldsymbol{\theta}} = \mathbf{f}(\boldsymbol{\theta}). \tag{3.6}$$

We will refer to a function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ for all $\boldsymbol{\theta}$ as a **passive function**, borrowing a term from circuit theory.

Stochastic approximation theory is devoted to the study of conditions under which a sequence $\{\boldsymbol{\theta}_t\}$ defined as in (3.3) converges to a zero of the function $\mathbf{f}(\cdot)$. Due to the presence of the noise, the convergence can only be probabilistic. The SA algorithm was introduced in [29], and some generalizations and/or simplifications followed very quickly; see [49, 20, 15, 13]. An excellent survey can be found in [26]. Book-length treatments of SA can be found in [24, 3, 25, 8].

The above formulation of SA can be used to address some related problems. For instance, suppose $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^d$ is some function, and it is desired to find a *fixed point* of the map $\mathbf{g}$, that is, a vector $\boldsymbol{\theta}^*$ such that $\mathbf{g}(\boldsymbol{\theta}^*) = \boldsymbol{\theta}^*$. As shown in Chapter 2, computing the value of a Markov reward problem, or the value of a policy in an MDP, both fall into this category. This problem can be formulated as that finding a zero of the function $\mathbf{f}(\boldsymbol{\theta}) = \mathbf{g}(\boldsymbol{\theta}) - \boldsymbol{\theta}$. If we were to substitute this expression into (3.3), we get what might be called the "fixed point version" of SA, namely

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t[\mathbf{g}(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t + \boldsymbol{\xi}_{t+1}] = (1 - \alpha_t)\boldsymbol{\theta}_t + \alpha_t[\mathbf{g}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}]. \tag{3.7}$$

Another application is that of finding a stationary point of a function $J : \mathbb{R}^d \to \mathbb{R}$, that is, finding a $\boldsymbol{\theta}^* \in \mathbb{R}^d$ such that $\nabla J(\boldsymbol{\theta}^*) = \mathbf{0}$. Again, the above problem can be formulated in the present framework by defining $\mathbf{f}(\boldsymbol{\theta}) := -\nabla J(\boldsymbol{\theta})$. Here again, one can ask by $\mathbf{f}(\boldsymbol{\theta}) := -\nabla J(\boldsymbol{\theta})$ and not $\mathbf{f}(\boldsymbol{\theta}) := \nabla J(\boldsymbol{\theta})$. The answer is that if we write $\mathbf{f}(\boldsymbol{\theta}) := -\nabla J(\boldsymbol{\theta})$, then under suitable conditions we can expect SA to find a local (or global) minimum of J. If we wish to maximize J, then of course we should choose $\mathbf{f}(\boldsymbol{\theta}) := \nabla J(\boldsymbol{\theta})$. As in the previous case, the learner has available only noisy measurements of the gradient, in the form $\mathbf{y}_{t+1} = -\nabla J(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}$. For this reason, the above formulation is sometimes referred to as "stochastic gradient descent." The reader is cautioned that the same phrase is also used with an entirely different meaning in the deep learning literature.

Equation 3.3 describes what might be called the "standard" SA. Two variants of SA are germane to RL, namely "asynchronous" SA and "two time-scale" SA. Each of these is briefly described next.

[1] In earlier years, methods such as steepest descent used to consist of two parts: (i) a choice of the search direction, and (ii) solution of a minimum along the search direction to determine the step size. However, in recent times, one-dimensional minimization has been dispensed with. Instead, the step size is chosen according to a predetermined schedule.

We begin with a description of "asynchronous" SA. Let $[d]$ denote the set $\{1, \cdots, d\}$, and let there be a rule I that maps $\mathbb{Z}_+$, the set of nonnegative integers, into $[d]$. At time $t \in \mathbb{Z}_+$, let $I(t)$ be the corresponding element of $[d]$. Then the update rule (3.3) is applied only to the $I(t)$-th component of $\boldsymbol{\theta}$. Thus

$$\theta_{t+1,j} = \begin{cases} \theta_{t,j} - \alpha_t y_{t+1,j}, & \text{if } j = I(t), \\ \theta_{t,j}, & \text{if } j \neq I(t). \end{cases} \tag{3.8}$$

From an implementation standpoint, the asynchronous update rule (3.8) presumably requires less storage. However, convergence with the asynchronous update rule might be slower than with (3.3).

Note that there is a great deal of flexibility in the update rule, which could either be deterministic or probabilistic. To illustrate, updating each component of $\boldsymbol{\theta}$ sequentially would be an example of a deterministic update rule, while choosing an index i from $[d]$ according to some probability distribution would be an example of a probabilistic rule. We shall see in subsequent chapters that, in some RL applications, the process $\{I(t)\}$ is itself a Markov process assuming values in $[d]$. For a given integer T, let $T(i)$ denote the number of times that component $i \in [d]$ is chosen to be updated, until time T. Then we insist that there exists a $\nu > 0$ such that

$$\liminf_{T \to \infty} \frac{T(i)}{T} \geq \nu > 0, \ \forall i \in [d]. \tag{3.9}$$

If i is chosen in a random fashion, for example in accordance with some probability distribution on $[d]$, or as a Markov process on $[d]$, then we insist only that (3.9) must hold almost surely. The purpose of (3.9) is to ensure that, though only one component of $\boldsymbol{\theta}_t$ is updated at time t, as time goes on, the fraction of times that each component of $\boldsymbol{\theta}_t$ is updated is bounded below by some positive constant.

This variant of asynchronous SA is introduced in [39], and builds on "asynchronous optimization" introduced in [40]. Another variant of asynchronous SA is introduced in [7], and studied further in [9]. In this variant, the update rule (3.8) is changed to

$$\theta_{t+1,j} = \begin{cases} \theta_{t,j} - \alpha_{\nu(t+1;i)} y_{t+1,j}, & \text{if } j = I(t), \\ \theta_{t,j}, & \text{if } j \neq I(t), \end{cases} \tag{3.10}$$

where

$$\nu(t+1;i) = \sum_{\tau=1}^{t+1} I_{\{I(\tau)=i\}}.$$

Thus $\nu(t+1;i)$ counts the number of time instants up to time $t+1$ when component i is selected for updating. The difference between (3.8) and (3.10) is this: In (3.8), the step size is α_t, which depends on the "global counter" t. In contrast, in (3.10), the step size is $\alpha_{\nu(t+1;i)}$, which depends on the "local counter" $\nu(t+1;i)$. Establishing the convergence of this particular variant makes use of far more stringent requirements on the noise $\{\boldsymbol{\xi}_t\}$, compared to the version in (3.8). This can be seen by comparing the contents of [39] with those of [7]. Moreover, standard RL algorithms such as Q-learning, introduced in subsequent chapters, correspond to the update rule (3.8) and not (3.10). For this reason, the update rule in (3.10) is not studied further.

The third variant of SA studied here is "two time-scale" SA, in which we attempt to solve "coupled" equations of the form

$$\mathbf{f}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbf{0}, \mathbf{g}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbf{0}, \tag{3.11}$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$, $\boldsymbol{\phi} \in \mathbb{R}^l$, $\mathbf{f} : \mathbb{R}^d \times \mathbb{R}^l \to \mathbb{R}^d$, and $\mathbf{g} : \mathbb{R}^d \times \mathbb{R}^l \to \mathbb{R}^l$. As before, for given "current" guesses $\boldsymbol{\theta}_t, \boldsymbol{\phi}_t$, we have access only to noise-corrupted measurements of the form

$$\mathbf{y}_{t+1} = \mathbf{f}(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t) + \boldsymbol{\xi}_{t+1}, \mathbf{z}_{t+1} = \mathbf{g}(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t) + \boldsymbol{\zeta}_{t+1}. \tag{3.12}$$

We update the current guesses in a manner analogous with (3.3), namely

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \mathbf{y}_{t+1}, \boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t - \beta_{t+1} \mathbf{z}_{t+1}. \tag{3.13}$$

Note the provision to have different step sizes for updating $\boldsymbol{\theta}_t$ and $\boldsymbol{\phi}_t$. Now, if we were to choose the step sizes in such a way that $\alpha_t = \beta_t$, or that the ratio α_t/β_t is bounded above and also below away from zero, then nothing much would be gained by permitting two different step sizes. However, suppose that $\alpha_t/\beta_t \to 0$ as $t \to \infty$. Then $\boldsymbol{\phi}_t$ is updated more rapidly than $\boldsymbol{\theta}_t$ (or $\boldsymbol{\theta}_t$ is updated more slowly than $\boldsymbol{\phi}_t$). In this setting, (3.13) is said to represent "two time-scale" SA.

The final section of this chapter deals with "finite-time" SA. Traditional results in SA are asymptotic, and assert that, under suitable conditions, the iterates converge to a solution of the problem under study as $t \to \infty$. In contrast, in "finite-time" SA, the emphasis is on providing (probabilistic) estimates of how far the current guess is from a solution. The finite-time approach is applicable to each of the three variants of SA mentioned above.

Throughout this chapter, the emphasis is on stating, and wherever possible proving, theorems about the behavior of various SA algorithms. The application of these results to problems in RL is deferred to later chapters.

3.2 Introduction to Martingales

Because martingale difference sequences play a central role in SA, in this subsection we quickly summarize some of the key aspects. In particular, we state without proof a couple of very useful results on the convergence of martingales. Further details about this topic can be found in [48, 11, 6, 14]. In particular, [48, Part B] is a very good source of theorems and examples, while the corresponding exercises in [48, Part E] provide additional useful material. Similarly, [14, Chapter 4] has a wealth of material, including several examples and problems, that is relevant to the material below.

Suppose that $(\Omega, \mathcal{F}, P)$ is a probability space, as described in Section 8.1. A sequence of σ-algebras $\{\mathcal{F}_t\}_{t\geq 0}$ on Ω is called a **filtration** if

$$\mathcal{F}_t \subseteq \mathcal{F}_{t+1} \subseteq \mathcal{F}, \ \forall t \geq 0. \tag{3.14}$$

Clearly (3.14) implies that

$$\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F}_t \subseteq \mathcal{F}_{t+1} \subseteq \mathcal{F}, \ \forall t \geq 0. \tag{3.15}$$

Now suppose that $\{Z_t\}_{t\geq 0}$ is an $\mathbb{R}^d$-valued stochastic process on $(\Omega, \mathcal{F}, P)$. We say that $\{Z_t\}$ is **adapted** to the filtration $\{\mathcal{F}_t\}$, or that the pair $(\{Z_t\}, \{\mathcal{F}_t\})$ is adapted, if Z_t is measurable with respect to $(\Omega, \mathcal{F}_t)$, (i.e., with $\mathcal{F}$ replaced by $\mathcal{F}_t$). Since the underlying set Ω and probability measure P are fixed, and the only thing varying is $\mathcal{F}_t$, we denote this by $Z_t \in \mathcal{M}(\mathcal{F}_t)$. In view of (3.14), we can make the following observations:

1. $Z_t \in \mathcal{M}(\mathcal{F}_\tau)$ whenever $\tau \geq t$.

2. Let $Z_0^t \in \mathbb{R}^{d(t+1)}$ denote $(Z_0, Z_1, \cdots, Z_t)$. Then $Z_0^t \in \mathcal{M}(\mathcal{F}_t)$.

If $\{Z_t\}_{t\geq 0}$ is an $\mathbb{R}^d$-valued stochastic process on $(\Omega, \mathcal{F}, P)$, then we can define the "natural filtration" by

$$\mathcal{F}_t = \sigma(Z_0^t),$$

where $\sigma(Z_0^t) \subseteq \mathcal{F}$ is the σ-algebra generated by Z_0^t. However, we do not always use the natural filtration.

Suppose $\{\mathcal{F}_t\}$ is a filtration on $(\Omega, \mathcal{F})$, and that $\{Z_t\}_{t\geq 0}$ is an $\mathbb{R}^d$-valued stochastic process on $(\Omega, \mathcal{F}, P)$. Then the pair $(\{Z_t\}, \{\mathcal{F}_t\})$ is said to be a **martingale** if the following hold:

(M1.) $E(|Z_t|, P) < \infty$, for all $t \geq 0$.

(M2.) The pair $(\{Z_t\}, \{\mathcal{F}_t\})$ is adapted.

(M3.) We have that

$$E(Z_{t+1}|\mathcal{F}_t) = Z_t, \text{ a.s., } \forall t \geq 0. \tag{3.16}$$

If we use the natural filtration $\mathcal{F}_t = \sigma(Z_0^t)$, then (3.16) can be replaced by Condition (M3'), namely

$$E(Z_{t+1}|Z_0^t) = Z_t, \text{ a.s., } \forall t \geq 0. \tag{3.17}$$

If (3.16) is replaced by

$$E(Z_{t+1}|\mathcal{F}_t) \leq Z_t, \text{ a.s., } \forall t \geq 0, \tag{3.18}$$

then $\{Z_t\}_{t\geq 0}$ is called a **supermartingale**, whereas if (3.17) is replaced by

$$E(Z_{t+1}|\mathcal{F}_t) \geq Z_t, \text{ a.s., } \forall t \geq 0, \tag{3.19}$$

then $\{Z_t\}_{t\geq 0}$ is called a **submartingale**.

Several useful consequences of the definition are obtained by applying Theorem 8.1. If $\{Z_t\}$ is a martingale, then by the iterated conditioning property (Item 5 of Theorem 8.1), it follows that

$$E(Z_\tau|\mathcal{F}_t) = Z_t, \text{ a.s., } \forall \tau \geq t+1, \ \forall t \geq 0. \tag{3.20}$$

The equality is replaced by $\leq$ for a supermartingale, and by $\geq$ for a submartingale. Next, by the expected value preservation property (Item 3 of Theorem 8.1), it follows that[2]

$$E[Z_t, P] = E[Z_0, P], \ \forall t \geq 0. \tag{3.21}$$

It similarly follows that if $\{Z_t\}$ is a supermartingale, then

$$E[Z_t, P] \leq E[Z_0, P], \ \forall t \geq 0, \tag{3.22}$$

where as if $\{Z_t\}$ is a submartingale, then

$$E[Z_t, P] \geq E[Z_0, P], \ \forall t \geq 0. \tag{3.23}$$

Thus, in a supermartingale, $\{E[Z_t, P]\}$ is a nonincreasing sequence of real numbers, while in a submartingale, $\{E[Z_t, P]\}$ is a nondecreasing sequence of real numbers.

Next, let $\{\xi_t\}_{t\geq 0}$ be a stochastic process adaptated to a filtration $\{\mathcal{F}_t\}$, such that $E[|\xi_t|, P] < \infty$ for all t, and define

$$Z_t = \sum_{\tau=0}^{t} \xi_\tau. \tag{3.24}$$

Then it is obvious that $\{Z_t\}$ is also adapted to $\{\mathcal{F}_t\}$. The sequence $(\{\xi_t\}, \{\mathcal{F}_t\})$ is said to be a **martingale difference sequence** if $(\{Z_t\}, \{\mathcal{F}_t\})$ is a martingale. It is easy to show using (3.16) that, if $\{\xi_t\}$ is a martingale difference sequence, then

$$E(\xi_{t+1}|\mathcal{F}_t) = 0, \text{ a.s., } \forall t \geq 0. \tag{3.25}$$

If $\xi_0 = 0$ almost surely (that is, if ξ_0 is a constant), then it follows that $E[\xi_t, P] = 0$ for all $t \geq 1$. The picture is clearer if each ξ_t belongs to $L_2(\Omega, \mathcal{F}_t)$. Then, by the projection property (Item 9) of Theorem 8.1, (3.26) is equivalent to the statement that ξ_{t+1} is orthogonal to every element of $L_2(\Omega, \mathcal{F}_t)$.

Example 3.1. Suppose $\{\xi_t\}$ is a sequence of independent (but not necessariy identically distributed) zero-mean random variables. Then $\{\xi_t\}$ is a martingale difference sequence, and the sequence $\{Z_t\}$ is a martingale. If $E[\xi_t, P] \geq 0$ for all t, then $\{Z_t\}$ is submartingale, whereas if $E[\xi_t, P] \leq 0$ for all t, then $\{Z_t\}$ is supermartingale.

[2]The reader is reminded that, wherever possible, we use parentheses for the conditional expectation, which is a random variable, and square brackets for the expected value, which is a real number.

Now we present some results on the convergence of martingales, which are useful in proving the convergence of various SA algorithms. The material is taken from [48, Chapter 12] and/or [14, Chapter 4] and is stated without proof. Citations from these sources are given for individual results stated below.

We begin with a preliminary concept. Given a filtration $\{\mathcal{F}_t\}$ and a stochastic process $\{A_t\}$ that is adapted to $\mathcal{F}_t$, we say that $\{(A_t, \mathcal{F}_t)\}$ is **predictable** if $A_t \in \mathcal{M}(\mathcal{F}_{t-1})$ for all $t \geq 1$. Note that there is no A_0. Also, note that in [48], such processes are said to be "previsible." However, the phrase "predictable" is used in [14] and appears to be more commonly used. We say that a martingale $\{Z_t\}$ (adapted to $\mathcal{F}_t$) is **null at zero** if $Z_0 = 0$ a.s., and that a predictable process $\{A_t\}$ is **null at zero** if $A_1 = 0$ a.s..[3] With these preliminaries, we can now state the following:

Theorem 3.1. *(Doob decomposition theorem. See [48, Theorem 12.11] or [14, Theorem 4.3.2].) Suppose $\{\mathcal{F}_t\}$ is a filtration and $\{Y_t\}$ is a stochastic process adapted to $\{\mathcal{F}_t\}$. Then Y_t can be expressed as*

$$Y_t = Y_0 + Z_t + A_t, \tag{3.26}$$

where $\{Z_t\}_{t\geq 0}$ is a martingale null at zero, and $\{A_t\}$ is a predictable process null at zero. If $\{Z'_t\}$ and $\{A'_t\}$ also satisfy the above conditions, then

$$P\{\omega : Z_t(\omega) = Z'_t(\omega) \& A_t(\omega) = A'_t(\omega),\ \forall t\} = 1. \tag{3.27}$$

Moreover, $\{Y_t\}$ is a submartingale if and only if $\{A_t\}$ is an increasing process, that is

$$P\{\omega : A_{t+1}(\omega) \geq A_t(\omega),\ \forall t\} = 1. \tag{3.28}$$

Note that an "explicit" expression for A_t is given by

$$A_t = \sum_{\tau=0}^{t-1} E((Y_{\tau+1} - Y_\tau)|\mathcal{F}_\tau) = \sum_{\tau=0}^{t-1} [E(Y_{\tau+1}|\mathcal{F}_\tau) - Y_\tau]. \tag{3.29}$$

Next, suppose $Y_t = M_t^2$, where $\{M_t\}$ is a martingale in $L_2(\Omega, P)$ null at zero. Then it is easy to show using the conditional Jensen's inequality (not covered here) that $\{Y_t\}$ is a submartingale null at zero. Therefore the Doob decomposition of $Y_t = M_t^2$ is

$$M_t^2 = Z_t + A_t, \tag{3.30}$$

where $\{Z_t\}$ is a martingale and $\{A_t\}$ is an increasing predictable process, both null at zero. It is customary to refer to $\{A_t\}$ as the **quadratic variation process** and to denote it by $\langle M_t \rangle$. Note that

$$A_{t+1} - A_t = E((M_{t+1}^2 - M_t^2)|\mathcal{F}_t) = E((M_{t+1} - M_t)^2|\mathcal{F}_t). \tag{3.31}$$

Define $A_\infty(\omega) = \lim_{t\to\infty} A_t(\omega)$ for (almost all) $\omega \in \Omega$. Then we have the following:

Theorem 3.2. *(See [48, Theorem 12.13].) If $A_\infty(\cdot)$ is bounded almost everywhere as a function of ω, then $\{M_t(\omega)\}$ converges almost everywhere at $t \to \infty$.*

Actually [48, Theorem 12.13] is more powerful and gives "almost necessary and sufficient" conditions for convergence. We have simply extracted what is needed for present purposes.

Theorem 3.3. *(See [14, Theorem 4.2.12].) If $\{\mathbf{Z}_t\}$ is nonnegative (i.e., $Z_t \geq 0$ a.s.) supermartingale, then there exists a $Z \in L_1(\Omega, P)$ such that $Z_t \to$ almost surely, and $E[Z, P] \leq E[Z_0, P]$.*

Theorem 3.4. *Suppose $\{\mathbf{Z}_t\}$ is a martingale wherein $Z_t \in L_p(\Omega, P)$ for some $p > 1$, and suppose further that the martingale is bounded in $\|\cdot\|_p$, that is*

$$\sup_t E[Z_t^p, P] < \infty. \tag{3.32}$$

Then there exists a $Z \in L_p(\Omega, P)$ such that $Z_t \to Z$ as $t \to \infty$, almost surely and in the p-th mean.

The above theorem is false if $p = 1$. The convergence is almost sure but need not be in the mean. See [14, Example 4.2.13].

[3] This is a slight inconsistency because actually A_t is "null at time one," but the usage is common.

3.3 Standard Stochastic Approximation

Recall the problem under study: There is a function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ for which only noisy measurements are available, and it is desired to find a zero of this function, using these noisy measurements. Specifically, if at time t if the current guess is $\boldsymbol{\theta}_t$, then we get a measurement

$$\mathbf{y}_{t+1} = \mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}, \tag{3.33}$$

where $\boldsymbol{\xi}_{t+1}$ is the measurement noise. The guess is updated according to

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \mathbf{y}_{t+1} = \boldsymbol{\theta}_t + \alpha_t(\mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}), \tag{3.34}$$

where $\{\alpha_t\}_{t\geq 1}$ is a predetermined sequence of step sizes. It is desired to study the limit behavior of the sequence $\{\boldsymbol{\theta}_t\}$. More specifically, suppose $\boldsymbol{\theta}^*$ is the only solution to the equation $f(\boldsymbol{\theta}) = \mathbf{0}$. We would like to find suitable conditions to ensure that the iterates $\{\boldsymbol{\theta}_t\}$ converge almost surely to $\boldsymbol{\theta}^*$.

In this section, we provide two different approaches to analyzing the SA algorithm. First, we analyze the situation where the function $\mathbf{f}$ is "passive," and second, where there exists a "Lyapunov" function V.

3.3.1 Stochastic Approximation for Passive Functions

In this subsection we show that the stochastic approximation algorithm of (3.34) converges when the function $\mathbf{f}(\cdot)$ is "passive," which is made precise in (3.35) below. The proof given here follows that in [16]. However, the reader is cautioned that in [16], the SA algorithm uses a minus sign in front of α_t, that is, it uses the formulation (3.3).

Definition 3.1. Suppose $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$, and suppose further that $\boldsymbol{\theta}^*$ is the only solution to the equation $f(\boldsymbol{\theta}) = \mathbf{0}$. Then $\mathbf{f}$ is said to be **passive with a zero at $\boldsymbol{\theta}^*$** if

$$\inf_{\epsilon < \|\boldsymbol{\theta}\|_2 < \epsilon^{-1}} \langle \boldsymbol{\theta} - \boldsymbol{\theta}^*, \mathbf{f}(\boldsymbol{\theta}) \rangle < 0, \ \forall \epsilon > 0. \tag{3.35}$$

Note that if $\mathbf{f}(\cdot)$ is continuous, then (3.35) can be replaced by

$$\langle \boldsymbol{\theta} - \boldsymbol{\theta}^*, \mathbf{f}(\boldsymbol{\theta}) \rangle < 0 \ \forall \boldsymbol{\theta} \neq \boldsymbol{\theta}^*. \tag{3.36}$$

In circuit theory, a nonlinear characteristic that satisfies (3.36) with $\boldsymbol{\theta}^* = \mathbf{0}$ would be called "passive," so we borrow that terminology.

To determine $\boldsymbol{\theta}^*$, we use the iterations defined by (3.34). The step sizes α_t are positive, and satisfy the conditions (referred to hereafter as the Robbins-Monro or RM conditions)

$$\sum_{t=1}^{\infty} \alpha_t = \infty, \sum_{t=1}^{\infty} \alpha_t^2 < \infty. \tag{3.37}$$

Theorem 3.5. *(See [16, Theorem 1].) Suppose the following assumptions hold:*

1. *$\mathbf{f}(\cdot)$ is passive with a zero at $\boldsymbol{\theta}^*$, that is,* (3.35) *holds.*
2. *There exists a constant d_1 such that*

$$\|\mathbf{f}(\boldsymbol{\theta})\|_2^2 \leq d_1(1 + \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2). \tag{3.38}$$

3. *Let $\mathcal{F}_t := \sigma(\boldsymbol{\theta}_0^t, \boldsymbol{\xi}_1^t)$.[4] Then the noise sequence $\{\xi_{t+1}\}$ satisfies two conditions: First,*

$$E(\boldsymbol{\xi}_{t+1} | \mathcal{F}_t) = 0 \ a.s.. \tag{3.39}$$

Second, there exists a constant $d_2 > 0$ such that

$$E(\|\boldsymbol{\xi}_{t+1}\|_2^2 | \mathcal{F}_t) \leq d_2(1 + \|\boldsymbol{\theta}_t\|_2^2) \ a.s., \ \forall t. \tag{3.40}$$

[4]The reader is reminded that $\boldsymbol{\theta}_0^t$ is a shorthand for $(\boldsymbol{\theta}_0, \cdots, \boldsymbol{\theta}_t$ etc., and $\sigma(\boldsymbol{\theta}_0^t, \boldsymbol{\xi}_1^t)$ denotes the σ-algebra generated by $\boldsymbol{\theta}_0^t$ and $\boldsymbol{\xi}_1^t$.

4. *The step size sequence* $\{\alpha_t\}$ *satisfies the RM conditions* (3.37).

Under these assumptions, we have that

1. *The sequence* $\{\boldsymbol{\theta}_t\}$ *is bounded almost surely.*

2. *Further,*

$$\boldsymbol{\theta}_t \to \boldsymbol{\theta}^* \text{ w.p. 1 as } t \to \infty. \tag{3.41}$$

For notational simplicity, suppose $\boldsymbol{\theta}$ is changed to $\boldsymbol{\theta} - \boldsymbol{\theta}^*$, and $\mathbf{f}(\boldsymbol{\theta})$ is changed to $\mathbf{f}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$, so that the unique solution to $\mathbf{f}(\boldsymbol{\theta}) = \mathbf{0}$ is $\boldsymbol{\theta} = \mathbf{0}$. Then (3.40) continues to hold with possibly a different (but still finite) constant d_2. Next,

$$\begin{aligned} \|\boldsymbol{\theta}_{t+1}\|_2^2 &= \|\boldsymbol{\theta}_t\|_2^2 + \alpha_t^2 \|\mathbf{f}(\boldsymbol{\theta}_t)\|_2^2 + \alpha_t^2 \|\boldsymbol{\xi}_{t+1}\|_2^2 \\ &+ 2\alpha_t \boldsymbol{\theta}_t^\top \boldsymbol{\xi}_{t+1} + 2\alpha_t^2 (\mathbf{f}(\boldsymbol{\theta}_t))^\top \boldsymbol{\xi}_{t+1} + 2\alpha_t \boldsymbol{\theta}_t^\top \mathbf{f}(\boldsymbol{\theta}_t) \end{aligned}$$

Now we use (3.38), (3.39), and (3.40), and define $d = d_1 + d_2$. This gives

$$\begin{aligned} E(\|\boldsymbol{\theta}_{t+1}\|_2^2 | \mathcal{F}_t) &= \|\boldsymbol{\theta}_t\|_2^2 + \alpha_t^2 \|\mathbf{f}(\boldsymbol{\theta}_t)\|_2^2 + \alpha_t^2 E(\|\boldsymbol{\xi}_{t+1}\|_2^2 | \mathcal{F}_t) + 2\alpha_t \boldsymbol{\theta}_t^\top \mathbf{f}(\boldsymbol{\theta}_t) \\ &\leq \|\boldsymbol{\theta}_t\|_2^2 + \alpha_t^2 d(1 + \|\boldsymbol{\theta}_t\|_2^2) + 2\alpha_t \boldsymbol{\theta}_t^\top \mathbf{f}(\boldsymbol{\theta}_t) \\ &\leq (1 + d\alpha_t^2)\|\boldsymbol{\theta}_t\|_2^2 + d\alpha_t^2, \end{aligned} \tag{3.42}$$

where we use the fact that $\boldsymbol{\theta}_t^\top \mathbf{f}(\boldsymbol{\theta}_t) \leq 0$ from (3.36).

Next we define a new stochastic process

$$Z_t := a_t \|\boldsymbol{\theta}_t\|_2^2 + b_t, \tag{3.43}$$

and choose the constants a_t, b_t recursively so as to ensure that Z_t is a nonnegative supermartingale. For this purpose, note that

$$\begin{aligned} E(Z_{t+1} | \mathcal{F}_t) &= a_{t+1} E(\|\boldsymbol{\theta}_{t+1}\|_2^2 | \mathcal{F}_t) + b_{t+1} \\ &\leq a_{t+1}[(1 + d\alpha_t^2)\|\boldsymbol{\theta}_t\|_2^2 + d\alpha_t^2] + b_{t+1}. \end{aligned}$$

Now substitute

$$\|\boldsymbol{\theta}_t\|_2^2 = \frac{1}{a_t} Z_t - \frac{b_t}{a_t}.$$

This gives

$$E(Z_{t+1} | \mathcal{F}_t) \leq \frac{a_{t+1}(1 + d\alpha_t^2)}{a_t} Z_t - \frac{a_{t+1}(1 + d\alpha_t^2)}{a_t} b_t + a_{t+1} d\alpha_t^2 + b_{t+1}.$$

So we get the supermartingale property

$$E(Z_{t+1} | \mathcal{F}_t) \leq Z_t$$

provide we define a_{t+1} and b_{t+1} in such a way that

$$\frac{a_{t+1}(1 + d\alpha_t^2)}{a_t} = 1, \text{ or } a_{t+1} = \frac{a_t}{1 + d\alpha_t^2},$$

$$-b_t + a_{t+1} d\alpha_t^2 + b_{t+1} = 0, \text{ or } b_{t+1} = b_t - d\alpha_t^2 a_{t+1}.$$

The above recursive relationships define a_t and b_t completely once we specify a_0 and b_0. To ensure that $Z_t \geq 0$ for all t, we choose

$$a_0 = \prod_{t=0}^{\infty} (1 + d\alpha_t^2), b_t = \sum_{t=0}^{\infty} d\alpha_t^2 a_{t+1}.$$

The square-summability of the step size sequence $\{\alpha_t\}$ ensures that both a_0 and b_0 are well-defined. This gives the closed-form expressions

$$a_t = \prod_{k=t}^{\infty}(1+d\alpha_k^2), b_t = \sum_{k=t}^{\infty} d\alpha_k^2 a_{k+1} = \sum_{k=t}^{\infty} d\alpha_k^2 \prod_{m=k+1}^{\infty}(1+d\alpha_m^2). \tag{3.44}$$

Moreover, we have

$$1 < a_{t+1} < a_t < a_0, 0 < b_{t+1} < b_t < b_0, \ \forall t.$$

Therefore $a_t \downarrow a_\infty \geq 1$ and $b_t \downarrow b_\infty \geq 0$.

With the above choices for a_t, b_t, $\{Z_t\}$ is a nonnegative supermartingale. By Theorem 3.3, $\{Z_t\}$ converges almost surely to some some random variable. Because $a_t \to a_\infty > 1$ and $b_t \to b_\infty$, it follows from (3.43) that $\|\boldsymbol{\theta}_t\|_2$ also converges almost surely to some random variable ζ. Thus the proof is complete once it is shown that $\zeta = 0$ a.s..

Since $\{Z_t\}$ is a nonnegative supermartingale, it follows that

$$E[Z_t, P] \leq E[Z_0, P], \ \forall t \geq 0.$$

Moreover, since $a_t > 1$ for all t, we conclude that

$$E[\|\boldsymbol{\theta}_t\|_2^2, P] \leq c < \infty \ \forall t, \tag{3.45}$$

for some constant c. This is the first desired conclusion. Now let us return to (3.42). After taking the expected value with respect to P and noting that

$$E[E(\|\boldsymbol{\theta}_{t+1}\|_2^2|\mathcal{F}_t), P] = E[\|\boldsymbol{\theta}_{t+1}\|_2^2, P],$$

we get

$$E[\|\boldsymbol{\theta}_{t+1}\|_2^2, P] \leq E[\|\boldsymbol{\theta}_t\|_2^2, P] + \alpha_t^2 d(1 + E[\|\boldsymbol{\theta}_t\|_2^2, P]) + 2\alpha_t E[\boldsymbol{\theta}_t^\top \mathbf{f}(\boldsymbol{\theta}_t), P].$$

Applying this bound recursively leads to

$$\begin{aligned} E[\|\boldsymbol{\theta}_{t+1}\|_2^2, P] &\leq E[\|\boldsymbol{\theta}_0\|_2^2, P] + \sum_{k=0}^{t} d\alpha_k^2(1 + E[\|\boldsymbol{\theta}_k\|_2^2, P]) + 2\sum_{k=0}^{t}\alpha_k E[\boldsymbol{\theta}_k^\top \mathbf{f}(\boldsymbol{\theta}_k), P] \\ &\leq c + d(1+c)\sum_{k=0}^{\infty}\alpha_k^2 + 2\sum_{k=0}^{t}\alpha_k E[\boldsymbol{\theta}_k^\top \mathbf{f}(\boldsymbol{\theta}_k), P]. \end{aligned} \tag{3.46}$$

Recall that $\boldsymbol{\theta}_k^\top \mathbf{f}(\boldsymbol{\theta}_k) \leq 0$. Hence $-\boldsymbol{\theta}_k^\top \mathbf{f}(\boldsymbol{\theta}_k) = |\boldsymbol{\theta}_k^\top \mathbf{f}(\boldsymbol{\theta}_k)|$. Now we can rearrange (3.46) as

$$2\left|\sum_{k=0}^{t}\alpha_k E[\boldsymbol{\theta}_k^\top \mathbf{f}(\boldsymbol{\theta}_k), P]\right| \leq c + d(1+c)\sum_{k=0}^{\infty}\alpha_k^2 - E[\|\boldsymbol{\theta}_{t+1}\|_2^2, P] \leq c + d(1+c)\sum_{k=0}^{\infty}\alpha_k^2.$$

In short, we have

$$\left|\sum_{k=0}^{t}\alpha_k E[\boldsymbol{\theta}_k^\top \mathbf{f}(\boldsymbol{\theta}_k), P]\right| \leq \frac{1}{2}\left[c + d(1+c)\sum_{k=0}^{\infty}\alpha_k^2\right] =: c_1, \ \forall t.$$

Now let $t \to \infty$ and also, change the index of summation from k to t. This gives

$$\sum_{t=0}^{\infty}\alpha_t |E[\boldsymbol{\theta}_k^\top \mathbf{f}(\boldsymbol{\theta}_k), P]| \leq c_1 < \infty. \tag{3.47}$$

Now recall that $\sum_t \alpha_t = \infty$. This implies that

$$\liminf_{t\to\infty} |E[\boldsymbol{\theta}_t^\top \mathbf{f}(\boldsymbol{\theta}_t), P]| = 0. \tag{3.48}$$

Otherwise, there would exist an integer K and an $\epsilon > 0$ such that

$$|E[\boldsymbol{\theta}_t^\top \mathbf{f}(\boldsymbol{\theta}_t), P]| \geq \epsilon, \ \forall t \geq K.$$

Also

$$\sum_{t=K}^{\infty} \alpha_t = \infty,$$

because dropping a finite number of terms in the summation does not affect the divergence of the sum. These two observations taken together contradict (3.47). Therefore (3.48) holds. As a result, there is a subsequence of $\{\boldsymbol{\theta}_t\}$, call it $\{\boldsymbol{\theta}_{t_k}\}$, such that $\boldsymbol{\theta}_{t_k}^\top \mathbf{f}(\boldsymbol{\theta}_{t_k}) \to 0$ in probability. In turn this implies that there is yet another subsequence, which is again denoted by $\{\boldsymbol{\theta}_{t_k}\}$, such that $\boldsymbol{\theta}_{t_k}^\top \mathbf{f}(\boldsymbol{\theta}_{t_k}) \to 0$ almost surely as $k \to \infty$. Again, if

$$\liminf_{k\to\infty} \|\boldsymbol{\theta}_{t_k}\|_2^2 > 0,$$

then applying (3.35) leads to a contradiction. Hence there exists a subsequence of times $\{t_k\}$ such that $\|\boldsymbol{\theta}_{t_k}\|_2^2 \to 0$ almost surely as $k \to \infty$. Now $\|\boldsymbol{\theta}_{t_k}\|_2^2 \to \zeta$ a.s., and a subsequence converges almost surely to 0. This shows that $\zeta = 0$ a.s..

As a concrete application of Theorem 3.5, we consider the convergence of the iterations of a contraction map with noisy measurements. Though the result is a corollary of Theorem 3.5, it is stated separately as a theorem in view of its importance.

First we describe the set-up. Suppose $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^d$ is a global contraction with respect to the Euclidean norm. Specifically, suppose there exists a constant $\rho < 1$ such that

$$\|\mathbf{g}(\boldsymbol{\theta}) - \mathbf{g}(\boldsymbol{\phi})\|_2 \leq \rho \|\boldsymbol{\theta} - \boldsymbol{\phi}\|_2, \ \forall \boldsymbol{\theta}, \boldsymbol{\phi} \in \mathbb{R}^d. \tag{3.49}$$

To determine the unique fixed point $\boldsymbol{\theta}^*$ of $\mathbf{g}(\cdot)$, define the iterations as in (3.7), namely

$$\boldsymbol{\theta}_{t+1} = (1 - \alpha_t)\boldsymbol{\theta}_t + \alpha_t(\mathbf{g}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}), \tag{3.50}$$

where $\{\boldsymbol{\xi}_t\}$ is the measurement noise sequence.

Theorem 3.6. *Define $\mathcal{F}_t := \sigma(\boldsymbol{\theta}_0^t, \boldsymbol{\xi}_1^t)$, and suppose there exists a constant d such that* (3.39) *and* (3.40) *hold. Suppose further that the step size sequence $\{\alpha_t\}$ satisfies the RM conditions* (3.37)*. Then $\boldsymbol{\theta}_t \to \boldsymbol{\theta}^*$ a.s. as $t \to \infty$.*

Proof. Define $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ as $\mathbf{f} : \boldsymbol{\theta} \mapsto \mathbf{g}(\boldsymbol{\theta}) - \boldsymbol{\theta}$. Then, for all $\boldsymbol{\theta}, \boldsymbol{\phi} \in \mathbb{R}^d$, we have

$$\begin{aligned} \langle \boldsymbol{\theta} - \boldsymbol{\phi}, \mathbf{f}(\boldsymbol{\theta}) - \mathbf{f}(\boldsymbol{\phi}) \rangle &= -\|\boldsymbol{\theta} - \boldsymbol{\phi}\|_2^2 + \langle \boldsymbol{\theta} - \boldsymbol{\phi}, \mathbf{g}(\boldsymbol{\theta}) - \mathbf{g}(\boldsymbol{\phi}) \rangle \\ &\leq -\|\boldsymbol{\theta} - \boldsymbol{\phi}\|_2^2 + \rho \|\boldsymbol{\theta} - \boldsymbol{\phi}\|_2^2 = -(1 - \rho)\|\boldsymbol{\theta} - \boldsymbol{\phi}\|_2^2. \end{aligned}$$

In particular, with $\boldsymbol{\phi} = \boldsymbol{\theta}^*$, we have that

$$\langle \boldsymbol{\theta} - \boldsymbol{\theta}^*, \mathbf{f}(\boldsymbol{\theta}) \rangle \leq -(1 - \rho)\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2.$$

Hence $\mathbf{f}(\cdot)$ is passive, and moreover, $\boldsymbol{\theta}^*$ is the unique zero of $\mathbf{f}(\cdot)$. Also, we have that

$$\begin{aligned} \|\mathbf{f}(\boldsymbol{\theta})\|_2 &\leq \|g(\boldsymbol{\theta})\|_2 + \|\boldsymbol{\theta}\|_2 \leq \rho \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2 + \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2 + \|\boldsymbol{\theta}^*\|_2 \\ &= (1 + \rho)\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2 + \|\boldsymbol{\theta}^*\|_2 = a + b\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2, \end{aligned}$$

where the definitions of a and b are obvious. Now using the easily proven identity

$$(a + bx)^2 \leq (a^2 + 2ab) + (b^2 + 2ab)x^2 \ \forall x \geq 0,$$

we conclude that $\mathbf{f}(\cdot)$ satisfies (3.38). Then the desired result follows from Theorem 3.5. □

Another application of Theorem 3.5 is to minimizing a convex function $h : \mathbb{R}^d \to \mathbb{R}$ using noisy measurements of the gradient ∇h. Recall that a function $h : \mathbb{R}^d \to \mathbb{R}$ is said to be **convex** if

$$h[\lambda\boldsymbol{\theta} + (1-\lambda)\boldsymbol{\phi}] \leq \lambda h(\boldsymbol{\theta}) + (1-\lambda)h(\boldsymbol{\phi}),\ \forall \boldsymbol{\theta}, \boldsymbol{\phi} \in \mathbb{R}^d.$$

Moreover, if $h \in C^1$, then (see [17, Theorem B.4.1.1])

$$h(\boldsymbol{\phi}) \geq h(\boldsymbol{\theta}) + \langle \nabla h(\boldsymbol{\theta}), \boldsymbol{\phi} - \boldsymbol{\theta} \rangle,\ \forall \boldsymbol{\theta}, \boldsymbol{\phi} \in \mathbb{R}^d. \tag{3.51}$$

Moreover, $\boldsymbol{\theta}^*$ is a global minimum of h if and only if $\nabla h(\boldsymbol{\theta}^*) = \mathbf{0}$. Hence one can attempt to find $\boldsymbol{\theta}^*$ by finding a zero of the function $f(\boldsymbol{\theta}) = -\nabla h(\boldsymbol{\theta})$, using noisy measurements of the form

$$\mathbf{y}_{t+1} = -\nabla h(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1} = \mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1},$$

and the updating rule

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \mathbf{y}_{t+1} = \boldsymbol{\theta}_t + \alpha_t[-\nabla h(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}]. \tag{3.52}$$

This is sometimes referred to as "stochastic gradient descent." The next result establishes the convergence of this approach under very mild conditions. The theorem below is noteworthy because it is *not assumed* that the Hessian $\nabla^2 h$ is bounded. So the theorem applies to, for example, $h(\theta) = \theta^4$, whereas many existing results do not.

Theorem 3.7. *Define $\mathcal{F}_t = \sigma(\boldsymbol{\theta}_0^t, \boldsymbol{\xi}_1^t)$, and suppose that the conditions* (3.39) *and* (3.40) *hold. Suppose further that the convex, C^1 function h has a unique global minimum at $\boldsymbol{\theta}^*$, and that there exists a finite constant d_1 such that*

$$\|\nabla h(\boldsymbol{\theta})\|_2^2 \leq d_1(1 + \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2),\ \forall \boldsymbol{\theta} \in \mathbb{R}^d.$$

Then $\boldsymbol{\theta}_t \to \boldsymbol{\theta}^*$ almost surely as $t \to \infty$, provided the RM conditions (3.37) *hold.*

The proof consists of showing that the function $\mathbf{f}(\cdot) = -\nabla h(\cdot)$ satisfies (3.35) and is thus passive in the sense of Definition 3.1. Since $\boldsymbol{\theta}^*$ is the unique global minimum of h, we have that $h(\boldsymbol{\theta}) > h(\boldsymbol{\theta}^*)$ whenever $\boldsymbol{\theta} \neq \boldsymbol{\theta}^*$. Now apply (3.51) with $\boldsymbol{\phi} = \boldsymbol{\theta}^*$. This gives

$$\langle \boldsymbol{\theta} - \boldsymbol{\theta}^*, \mathbf{f}(\boldsymbol{\theta}) \rangle = \langle \nabla h(\boldsymbol{\theta}), \boldsymbol{\theta}^* - \boldsymbol{\theta} \rangle \leq h(\boldsymbol{\theta}^*) - h(\boldsymbol{\theta}) < 0 \text{ if } \boldsymbol{\theta} \neq \boldsymbol{\theta}^*.$$

Now the desired result follows from Theorem 3.5.

Corollary 3.1. *The conclusions of Theorems 3.6 and 3.7 continue to hold if $\{\boldsymbol{\xi}_t\}$ is an i.i.d. sequence with zero mean finite variance.*

3.3.2 Stochastic Approximation via Lyapunov Functions

In this subsection, we establish the convergencw of the stochastic approximation algorithm (3.34) under a different set of assumptions than passivity as defined in Definition 3.1. Specifically, the proof of Theorem 3.8 below is based on the existence of a "Lyapunov function" V that satisfies certain assumptions. At first glance these assumptions might appear to be rather restrictive. However, it can be shown using "converse" Lyapunov theory that if an associated differential equation $\dot{\boldsymbol{\theta}} = \mathbf{f}(\boldsymbol{\theta})$ possesses certain stability properties, then the existence of a suitable Lyapunov function V is guaranteed; see Section 8.4.

Though the proof of Theorem 3.8 below is broadly similar to that of Theorem 3.5, the assumptions themselves are quite distinct in that neither set implies the other.

Let us reprise the problem under study. Suppose $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ and that $\boldsymbol{\theta}^*$ is the unique solution to $\mathbf{f}(\boldsymbol{\theta}) = \mathbf{0}$. We begin with an initial guess $\boldsymbol{\theta}_0$ and update $\boldsymbol{\theta}_t$ via

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t(\mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}), \tag{3.53}$$

where $\{\boldsymbol{\xi}_t\}_{t \geq 1}$ is the sequence of measurement noises.

Now the following assumptions are made about the function $\mathbf{f}$. The first assumption is that the function f is globally Lipschitz continuous with constant L. Thus

$$\|\mathbf{f}(\boldsymbol{\theta}) - \mathbf{f}(\boldsymbol{\phi})\|_2 \leq L\|\boldsymbol{\theta} - \boldsymbol{\phi}\|_2, \ \forall \boldsymbol{\theta}, \boldsymbol{\phi} \in \mathbb{R}^d. \tag{3.54}$$

The next several assumptions concern the existence of a function V satisfying various assumptions. Though it is not immediately obvious, these are actually assumptions about the function $\mathbf{f}$. See Section 8.4 to see the connection. It is now assumed that there exists a $\mathcal{C}^2$ function $V : \mathbb{R}^d \to \mathbb{R}_+$, and constants $a, b, c > 0$ and $M < \infty$ such that

$$a\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2 \leq V(\boldsymbol{\theta}) \leq b\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2, \ \forall \boldsymbol{\theta} \in \mathbb{R}^d, \tag{3.55}$$

This implies, among other things, that $V(\boldsymbol{\theta}^*) = 0$ and that $V(\boldsymbol{\theta}) > 0$ for all $\boldsymbol{\theta} \neq \boldsymbol{\theta}^*$. Next

$$\dot{V}(\boldsymbol{\theta}) \leq -c\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2, \ \forall \boldsymbol{\theta} \in \mathbb{R}^d, \tag{3.56}$$

where $\dot{V} : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\dot{V}(\boldsymbol{\theta}) := \nabla V(\boldsymbol{\theta})\mathbf{f}(\boldsymbol{\theta}). \tag{3.57}$$

Note that the gradient $\nabla V(\boldsymbol{\theta})$ is taken as a row vector. The last assumption on V is that

$$\|\nabla^2 V(\boldsymbol{\theta})\|_S \leq M, \ \forall \boldsymbol{\theta} \in \mathbb{R}^d, \tag{3.58}$$

where $\nabla^2 V$ is the Hessian matrix of V, and $\|\cdot\|_S$ denotes the spectral norm of a matrix, that is, its largest singular value.

As before, let $\mathcal{F}_t := \sigma(\boldsymbol{\theta}_0^t, \boldsymbol{\xi}_1^t)$. Then the noise sequence $\{\xi_{t+1}\}$ is assumed to satisfy the same two conditions as before, reprised here for convenience: First,

$$E(\boldsymbol{\xi}_{t+1}|\mathcal{F}_t) = 0 \text{ a.s..} \tag{3.59}$$

Second, there exists a constant $d > 0$ such that

$$E(\|\boldsymbol{\xi}_{t+1}\|_2^2|\mathcal{F}_t) \leq d(1 + \|\boldsymbol{\theta}_t\|_2^2) \text{ a.s., } \forall t. \tag{3.60}$$

Finally, the step size sequence $\{\alpha_t\}$ satisfies the RM conditions (3.37), restated here:

$$\sum_{t=1}^{\infty} \alpha_t = \infty, \sum_{t=1}^{\infty} \alpha_t^2 < \infty. \tag{3.61}$$

Theorem 3.8. *Under the above set of assumptions, we have that*

1. *The sequence $\{\boldsymbol{\theta}_t\}$ is bounded almost surely.*
2. *Further,*

$$\boldsymbol{\theta}_t \to \boldsymbol{\theta}^* \text{ w.p. } 1 \text{ as } t \to \infty.$$

Proof. The proof is analogous to that of Theorem 3.5, with $\|\boldsymbol{\theta}_t\|_2^2$ replaced by $V(\boldsymbol{\theta}_t)$. To begin with, we translate coordinates so that $\boldsymbol{\theta}^* = \mathbf{0}$. This may cause the constant d in (3.60) to change, but it would still be finite. Note that

$$V(\boldsymbol{\theta}_{t+1}) = V(\boldsymbol{\theta}_t + \alpha_t(\mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1})).$$

Now by Taylor's expansion around $\boldsymbol{\theta}_t$, we get

$$V(\boldsymbol{\theta}_{t+1}) = V(\boldsymbol{\theta}_t) + \alpha_t \nabla V(\boldsymbol{\theta}_t)[\mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}] + r_{t+1},$$

where the remainder tems r_{t+1} satisfies

$$r_{t+1} = \alpha_t^2 [\mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}]^\top \nabla V(\mathbf{z}_t)[\mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}]$$

for some z_t belonging to the line segment from $\boldsymbol{\theta}_t$ to $\boldsymbol{\theta}_{t+1}$. Therefore

$$\begin{aligned} |r_{t+1}| &\leq \alpha_t^2 M \|\mathbf{f}(\boldsymbol{\theta}_t) + \boldsymbol{\xi}_{t+1}\|_2^2 \\ &= \alpha_t^2 M [\|\mathbf{f}(\boldsymbol{\theta}_t)\|_2^2 + \|\boldsymbol{\xi}_{t+1}^2\|_2^2 + 2\mathbf{f}(\boldsymbol{\theta}_t)^\top \boldsymbol{\xi}_{t+1}]. \end{aligned}$$

Now using Equations (3.56), (3.59) and (3.60) gives

$$\begin{aligned} E(V(\boldsymbol{\theta}_{t+1})|\mathcal{F}_t) &= V(\boldsymbol{\theta}_t) + \alpha_t \nabla V(\boldsymbol{\theta}_t)\mathbf{f}(\boldsymbol{\theta}_t) + E(r_{t+1}|\mathcal{F}_t) \\ &\leq V(\boldsymbol{\theta}_t) + \alpha_t^2 M[\|\mathbf{f}(\boldsymbol{\theta}_t)\|_2^2 + d(1 + \|\boldsymbol{\theta}_t\|_2^2). \end{aligned}$$

Now apply

$$\|\mathbf{f}(\boldsymbol{\theta}_t)\|_2^2 \leq L^2 \|\boldsymbol{\theta}_t\|_2^2 \leq \frac{L^2}{a} V(\boldsymbol{\theta}_t), \|\boldsymbol{\theta}_t\|_2^2 \leq \frac{1}{a} V(\boldsymbol{\theta}_t).$$

This leads to

$$E(V(\boldsymbol{\theta}_{t+1})|\mathcal{F}_t) \leq \left[1 + \frac{\alpha_t^2 M}{a}(L^2 + d)\right] V(\boldsymbol{\theta}_t) + dM\alpha_t^2.$$

This is entirely analogous to (3.42). By replicating earlier reasoning, we conclude that $V(\boldsymbol{\theta}_t)$ is bounded almost surely and converges to some random variable ζ. In the last part of the proof, we restore the term $\alpha_t \nabla V(\boldsymbol{\theta}_t)\mathbf{f}(\boldsymbol{\theta}_t)$ which is neglected earlier, and observe that

$$\nabla V(\boldsymbol{\theta}_t)\mathbf{f}(\boldsymbol{\theta}_t) \leq -c\|\boldsymbol{\theta}_t\|_2^2 \leq -(c/b)V(\boldsymbol{\theta}_t).$$

In fact, this part of the proof is simpler than that of Theorem 3.5, because in that case there is no bound on how small the product $|\boldsymbol{\theta}_t f(\boldsymbol{\theta}_t)|$ can be, whereas here the term $|\nabla V(\boldsymbol{\theta}_t)\mathbf{f}(\boldsymbol{\theta}_t)|$ is bounded below by the term $(c/b)V(\boldsymbol{\theta}_t)$. This allows us to conclude that $\zeta = 0$ a.s., so that $V(\boldsymbol{\theta}_t)$ is bounded almost surely and approaches zero at $t \to \infty$. Finally, using (3.55) gives that $\{\boldsymbol{\theta}_t\}$ is also bounded almost surely and approaches zero at $t \to \infty$. These details are simple and are left to the reader. □

Next we present two examples, one where convergence follows from Theorem 3.5 but Theorem 3.8 does not apply, and the other one is vice versa.

Example 3.2. As shown in Section 8.4, the hypotheses of Theorem 3.8 imply that the origin is a *globally exponentially stable (GES)* equilibrium of the ODE $\dot{\boldsymbol{\theta}} = \mathbf{f}(\boldsymbol{\theta})$. Now suppose $d = 1$ (scalar problem), and define $f(\theta) = -\tanh(\theta)$. Then $|f(\theta)|$ is bounded as a function of θ. Thus the ODE $\dot{\theta} = -\tanh(\theta)$ cannot be GES. Hence Theorem 3.8 does not apply. However, since $\theta \tanh(\theta) > 0$ for every $\theta \neq 0$, Theorem 3.5 applies.

Example 3.3. In the other direction, suppose we wish to solve the fixed point equation

$$\mathbf{v} = G\mathbf{v} + \mathbf{r}$$

for some vector $\mathbf{r} \in \mathbb{R}^d$. We can cast this into the standard stochastic approximation framework by defining

$$\mathbf{f}(\boldsymbol{\theta}) = \mathbf{r} + (G - I_d)\boldsymbol{\theta}.$$

This is like the standard relationship of a discounted reward Markov reward process. Now choose a matrix $G \in \mathbb{R}^{d \times d}$ that satisfies $\|G\|_S > 1$, and $\rho(G) < 1$, where $\rho(\cdot)$ denotes the spectral radius. Then choose a vector $\mathbf{x} \in \mathbb{R}^d$ such that $\mathbf{x}^\top \mathbf{x} < \mathbf{x}^\top G\mathbf{x}$. For example, let $d = 2$, $\lambda \in (0.5, 1)$, and

$$G = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}, \mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Then $\rho(G) = \lambda < 1$ whenever $\lambda < 1$. Moreover $\mathbf{x}^\top \mathbf{x} = 2$, whereas

$$\mathbf{x}^\top G\mathbf{x} = 2\lambda + 1 > 2 \text{ if } \lambda > 0.5.$$

Because $\rho(G) < 1$, the matrix $G - I_d$ is nonsingular, so that for each vector $\mathbf{r}$, there is a unique solution $\boldsymbol{\theta}^*$ to the equation $\mathbf{f}(\boldsymbol{\theta}) = \mathbf{0}$, namely $\boldsymbol{\theta}^* = (G - I_d)^{-1}\mathbf{r}$. Now let $\boldsymbol{\theta} = \boldsymbol{\theta}^* + \mathbf{x}$ where $\mathbf{x}$ is chosen as above. Then

$$(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top (\mathbf{f}(\boldsymbol{\theta}) - \mathbf{f}(\boldsymbol{\theta}^*))^\top = \mathbf{x}^\top (G - I_d)\mathbf{x} > 0.$$

Hence (3.5) does not hold, and Theorem 3.5 does not apply.

On the other hand, because $\rho(G) < 1$, the eigenvalues of the matrix $G - I_d$ all have negative real parts, and as a result, the Lyapunov matrix equation

$$G^\top P + PG = -I_d$$

has a unique solution P which is positive definite. Then, if we define $V(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top P \boldsymbol{\theta}$, then Theorem 3.8 applies, and implies that the iterates converge to $\boldsymbol{\theta}^*$.

3.4 Batch Asynchronous Stochastic Approximation

In this section, we state and prove a very general result that covers not only the asynchronous stochastic approximation (ASA) algorithms of (3.8) and (3.10), but also a much more general situation that we call "batch asynchronous stochastic approximation" (BASA). Specifically, in (3.8) and (3.10), *only one component* of $\boldsymbol{\theta}_t$ is updated at a given instant t. However, in BASA, it is possible to update multiple components at a given time. Naturally, these results apply also to the tradition asynchronous versions of (3.8) and (3.10).

Throughout we consider vectors $\boldsymbol{\theta}_t \in \mathbb{R}^d$ where d is fixed. We use $\theta_{t,i}$ to denote the i-th component of $\boldsymbol{\theta}$, which belongs to $\mathbb{R}$. $\|\boldsymbol{\theta}\|_\infty$ denotes the ℓ_∞-norm of $\boldsymbol{\theta}$. For $s \le t$, $\boldsymbol{\theta}_s^t$ denotes $(\boldsymbol{\theta}_s, \cdots, \boldsymbol{\theta}_t)$. Note that

$$\|\boldsymbol{\theta}_s^t\|_\infty = \max_{s \le \tau \le t} \|\boldsymbol{\theta}_\tau\|_\infty.$$

The symbol $\mathbb{N}$ denotes the set of natural numbers plus zero, so that $\mathbb{N} = \{0, 1, \cdots\}$. If $\{\mathcal{F}_t\}$ is a filtration with $t \in \mathbb{N}$, then $\mathcal{M}(\mathcal{F}_t)$ denotes the set of functions that are measurable with respect to $\mathcal{F}_t$. Recall that the symbol $[d]$ denotes the set $\{1, \cdots, d\}$.

In Section 3.3, the objective is to compute the zero of a function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$. Theorems 3.5 and 3.8 provide sufficient conditions under which "synchronous" stochastic approximation can be used to solve this problem. In particular, if $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^d$ is a contraction map with respect to the ℓ_2-norm, then Theorem 3.6 shows that a fixed point of $\mathbf{g}$ can be found using the iterative scheme (3.5). However, as seen in Chapter 2, often one has to determine the fixed point of a map $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^d$ which is a contraction in the ℓ_∞-norm. Theorem 3.5 does not apply to this case. Proving a version that works in this case is the main motivation for introducing "asynchronous" stochastic approximation in [39]. The treatment below basically extends those arguments to the case where more than one component of $\boldsymbol{\theta}_t$ is updated at any time. An alternate version of ASA is introduced in [7] using local clocks. However, the assumptions on the measurement noise process $\{\boldsymbol{\xi}_t\}$ are fairly restrictive, in that the process is assumed to be an i.i.d. sequence, an assumption that does not hold in RL problems. In contrast, in [39], the noise process is assumed only to be a martingale difference sequence, which is satisfied in RL problems. Thus, even if only one component of $\boldsymbol{\theta}_t$ is updated at any one time t, the treatment here combines the best features of both [39] (noise process is a martingale difference sequence) and [7] (updating can use a local clock). The possibility of "batch" updating is a bonus. One important difference between standard SA and BASA is that even the step sizes can now be random variables. In principle random step sizes can be permitted even in standard SA, but it is not common.

The proof is divided into two parts. In the first, it is shown that iterations using a very general updating rule are bounded almost surely. This result would appear to be of independent interest; it is referred to as "stability" by some authors. Then this result is applied to show convergence of the iterations to a fixed point of a contractive map. In contrast with both [39] and [7], for the moment we do not permit delayed information.

For the first theorem on almost sure boundedness, we set up the problem as follows: We consider a function $\mathbf{h} : \mathbb{N} \times (\mathbb{R}^d)^{\mathbb{N}} \to (\mathbb{R}^d)^{\mathbb{N}}$, and say that it is **nonanticipative** if, for each $t \in \mathbb{N}$,

$$\boldsymbol{\theta}_0^\infty, \boldsymbol{\phi}_0^\infty \in (\mathbb{R}^d)^{\mathbb{N}}, \boldsymbol{\theta}_0^t = \boldsymbol{\phi}_0^t \implies \mathbf{h}(\tau, \boldsymbol{\theta}_0^\infty) = \mathbf{h}(\tau, \boldsymbol{\phi}_0^\infty), 0 \leq \tau \leq t.$$

Note that such functions are also referred to as "causal" in control and system theory. There are also d distinct "update processes" $\{\nu_{t,i}\}_{t\geq 0}$ for each $i \in [d]$. Finally, there is a "step size" sequence $\{\beta_t\}_{t\geq 0}$, which for convenience we take to be deterministic, though this is not essential. Also, it is assumed that $\beta_t \in (0, 1)$ for all t.

The "core" stochastic processes are the parameter sequence $\{\boldsymbol{\theta}_t\}_{t\geq 0}$, and the noise sequence $\{\boldsymbol{\xi}_t\}_{t\geq 1}$. Note the mismatch in the initial values of t. Often it is assumed that $\boldsymbol{\theta}_0$ is deterministic, but this is not essential. We define the filtration

$$\mathcal{F}_0 = \sigma(\boldsymbol{\theta}_0), \mathcal{F}_t = \sigma(\boldsymbol{\theta}_0^t, \boldsymbol{\xi}_1^t) \text{ for } t \geq 1,$$

where $\sigma(\cdot)$ denotes the σ-algebra generated by the random variables inside the parentheses.

Now we can begin to state the problem set-up.

(U1). The update processes $\nu_{t,i} \in \mathcal{M}(\mathcal{F}_t)$ for all $t \geq 0, i \in [d]$.

(U2.) The update processes satisfy the conditions that $\nu_{0,i}$ equals either 0 or 1, and that $\nu_{t,i}$ equals either $\nu_{t-1,i}$ or $\nu_{t-1,i}+1$. In other words, the process can only increment by at most one at each time instant t, for each index i. This automatically guarantees that $\nu_{t,i} \leq t$ for all $i \in [d]$.

(S1). For "batch asynchronous updating" with a "global clock," the step size $\alpha_{t,i}$ for each index i is defined as

$$\alpha_{t,i} = \begin{cases} \beta_t & \text{if } \nu_{t,i} = \nu_{t-1,i} + 1, \\ 0, & \text{if } \nu_{t,i} = \nu_{t-1,i}. \end{cases} \tag{3.62}$$

Therefore $\alpha_{t,i}$ equals β_t for those indices i that get incremented at time t, and zero for other indices.

(S2.) For "batch asynchronous updating" with a "local clock," the step size $\alpha_{t,i}$ for each index i is defined as

$$\alpha_{t,i} = \begin{cases} \beta_{\nu_t,i} & \text{if } \nu_{t,i} = \nu_{t-1,i} + 1, \\ 0, & \text{if } \nu_{t,i} = \nu_{t-1,i}. \end{cases} \tag{3.63}$$

Note that, with a local clock, we can also write

$$\alpha_{t,i} = \nu_{0,i} + \sum_{\tau=1}^{t} I_{\{\nu_{t,i}=\nu_{t-1,i}+1\}}, \tag{3.64}$$

where I denotes the indicator function: It equals 1 if the subscripted statement is true, and equals 0 otherwise. Note that with a global clock, the step size for each index that has a nonzero $\alpha_{t,i}$ is the same, namely β_t. However, with a local clock, this is not necessarily true.

With this set-up, we can define the basic asynchronous iteration scheme.

$$\theta_{t+1,i} = \theta_{t,i} + \alpha_{t,i}(\eta_{t,i} - \theta_{t,i} + \xi_{t+1,i}), i \in [d], t \geq 0, \tag{3.65}$$

where

$$\boldsymbol{\eta}_t = \mathbf{h}(t, \boldsymbol{\theta}_0^t). \tag{3.66}$$

Note that in our view it makes no sense to define

$$\eta_{t,i} = h_i(\nu_{t,i}, \boldsymbol{\theta}_0^{\nu_{t,i}}).$$

We cannot think of any application for such an updating rule.

The question under study in the first part of the paper is the almost sure boundedness of the iterations $\{\boldsymbol{\theta}_t\}$. For this purpose we introduce a few assumptions.

(A1.) There exist constants $\gamma < 1$ and $c_0 \geq 0$ such that

$$\|\mathbf{h}(t, \boldsymbol{\theta}_0^t) - \mathbf{h}(t, \boldsymbol{\phi}_0^t)\|_\infty \leq \gamma \|\boldsymbol{\theta}_0^t - \boldsymbol{\phi}_0^t\|_\infty, \ \forall \boldsymbol{\theta}, \boldsymbol{\phi} \in (\mathbb{R}^d)^{\mathbb{N}}, t \geq 0, \tag{3.67}$$

$$\|\mathbf{h}(t, \mathbf{0})\|_\infty \leq c_0, \ \forall t \geq 0. \tag{3.68}$$

Note that in (3.70), we use the generic symbol $\mathbf{0}$ to denote a vector with all zero components, whose dimension is determined by the context. If we choose any $\rho \in (\gamma, 1)$ and define

$$c_1 := \frac{c_0}{\rho - \gamma}, \tag{3.69}$$

then it is easy to verify that

$$\|\mathbf{h}(t, \boldsymbol{\theta}_0^t)\|_\infty \leq \rho \max\{c_1, \|\boldsymbol{\theta}_0^t\|_\infty\}, \ \forall \boldsymbol{\theta}, t. \tag{3.70}$$

(A2.) The step sizes $\alpha_{t,i}$ satisfy the analogs of the Robbins-Monro conditions of [29], namely

$$\sum_{t=0}^{\infty} \alpha_{t,i} = \infty \text{ a.s., } \forall i \in [d], \tag{3.71}$$

$$\sum_{t=0}^{\infty} \alpha_{t,i}^2 < \infty \text{ a.s., } \forall i \in [d]. \tag{3.72}$$

Note that if $\sum_{t=0}^{\infty} \beta_t^2 < \infty$, then the square-summability of the $\alpha_{t,i}$ is automatic. However, the divergence of the summation of $\alpha_{t,i}$ requires additional conditions on both the step sizes β_t and the update processes $\nu_{t,i}$.

(A3.) The noise process $\{\boldsymbol{\xi}_t\}$ satisfies the following:

$$E(\xi_{t+1,i} | \mathcal{F}_t) = 0, \ \forall t \geq 0, i \in [d], \tag{3.73}$$

$$E(\xi_{t+1}^2 | \mathcal{F}_t) \leq c_2(1 + \|\boldsymbol{\theta}_t\|_2^2), \tag{3.74}$$

for some constant c_2.

Now we can state the result on the almost sure boundedness of the iterates.

Theorem 3.9. *With the set up above, and subject to assumptions (A1), (A2) and (A3), we have that the sequence $\{\boldsymbol{\theta}_t\}$ is bounded almost surely.*

Next, we state a result on convergence. For this purpose, we significantly reduce the generality of the function $\mathbf{h}(\cdot)$.

Theorem 3.10. *Suppose that, in addition to the conditions of Theorem 3.9, we have that the function $\mathbf{h}(\cdot)$ is defined by*

$$\mathbf{h}(t, \boldsymbol{\theta}_0^t) = \mathbf{g}(\boldsymbol{\theta}_t),$$

where the function $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^d$ satisfies

$$|g_i(\boldsymbol{\theta}) - g_i(\boldsymbol{\phi})| \leq \gamma \|\boldsymbol{\theta} - \boldsymbol{\phi}\|_\infty, \ \forall i \in [d], \ \forall \boldsymbol{\theta}, \boldsymbol{\phi} \in \mathbb{R}^d,$$

for some constant $g < 1$.[5] Let $\boldsymbol{\theta}^*$ denote the unique fixed point of the map $\mathbf{g}$. Then $\boldsymbol{\theta}_t \to \boldsymbol{\theta}^*$ almost surely as $t \to \infty$.

[5] In other words, the function $\mathbf{g}$ is a contraction with respect to the ℓ_∞-norm.

Now we prove the two theorems. The proof of Theorem 3.9 is very long and involves several auxiliary lemmas. Before proceeding to the lemmas, a matter of notation is cleared up. Throughout we are dealing with stochastic processes. So in principle we should be writing, for instance, $\boldsymbol{\theta}_t(\omega)$, where ω is the element of the probability space that captures the randomness. We do not do this in the interests of brevity, but the presence of the argument ω is implicit throughout.

Lemma 3.1. *Define a real-valued stochastic process $\{U_i(0;t)\}_{t\geq 0}$ by*

$$U_i(0;t+1) = (1-\alpha_{t,i})U_i(0;t) + \alpha_{t,i}\xi_{t+1,i}, \tag{3.75}$$

where $U_i(0,0) \in \mathcal{F}_0$, $\{\alpha_{t,i}\}$ satisfy (3.71) and (3.72), and $\{\boldsymbol{\xi}_t\}$ satisfies (3.73) and (3.74). Then $U_i(0;t) \to 0$ almost surely as $t \to \infty$.

This lemma is a ready consequence of Theorem 3.5. Note that (3.75) attempts to find a fixed-point of the function $f(U) = -U$, and that $Uf(U) < 0$ whenever $U \neq 0$. Now apply the theorem for $U = U_i$.

Lemma 3.2. *Define a doubly-indexed real-valued stochastic process $W_i(\tau;t)$ by*

$$W_i(t_0;t+1) = (1-\alpha_{t,i})W_i(t_0,t) + \alpha_{t,i}\xi_{t+1,i}, \tag{3.76}$$

where $W_i(t_0,t_0) = 0$, $\{\alpha_{t,i}\}$ satisfy (3.71) and (3.72), and $\{\boldsymbol{\xi}_t\}$ satisfies (3.73) and (3.74). Then, for each $\delta > 0$, there exists a t_0 such that $|W_i(s;t)| \leq \delta$, for all $t_0 \leq s \leq t$.

Proof. It is easy to prove using induction that $W_i(s;t)$ satisfies the recursion

$$U_i(0;t) = \left[\prod_{r=s}^{t-1}(1-\alpha_{r,i})\right] U_i(0;s) + W_i(s;t), 0 \leq s \leq t.$$

Note that the product in the square brackets is no larger than one. Hence

$$|W_i(s;t)| \leq |U_i(0;t)| + |U_i(0;s)|.$$

Now, given $\delta > 0$, choose a t_0 such that $|U_i(0;t)| \leq \delta/2$ whenever $t \geq t_0$. Then choosing $t_0 \leq s \leq t$ and applying the triangle inequality leads to the desired conclusion. □

Now we come to the proof of Theorem 3.9.

Proof. For $t \geq 0$, define

$$\Gamma_t := \max\{\|\boldsymbol{\theta}_0^t\|_\infty, c_1\}, \tag{3.77}$$

where $c_1 = (\rho-\gamma)/c_0$ as before. With this definition, it follows from (3.70) and (3.66) that

$$\|\eta_t\|_\infty \leq \rho\Gamma_t, \ \forall t. \tag{3.78}$$

Next, choose any $\epsilon \in (0,1)$ such that $\rho(1+\epsilon) < 1$.[6] Now we define a sequence of constants recursively. Let $\Lambda_0 = \Gamma_0$, and define

$$\Lambda_{t+1} = \begin{cases} \Lambda_t, & \text{if } \Gamma_{t+1} \leq \Lambda_t(1+\epsilon), \\ \Gamma_{t+1} & \text{if } \Gamma_{t+1} > \Lambda_t(1+\epsilon). \end{cases} \tag{3.79}$$

Define $\lambda_t = 1/\Lambda_t$. Then it is clear that $\{\Lambda_t\}$ is a nondecreasing sequence, starting at $\Gamma_0 = \|\boldsymbol{\theta}_0\|_\infty$. Consequently, $\{\lambda_t\}$ is a bounded, nonnegative, nonincreasing sequence. Further, $\lambda_t\Gamma_t \leq 1+\epsilon$ for all t. Moreover, either $\Lambda_{t+1} = \Lambda_t$, or else $\Lambda_{t+1} > \Lambda_t(1+\epsilon)$. Hence, saying that $\Lambda_{t+1} > \Lambda_t$ is the same as saying that $\Lambda_{t+1} > \Lambda_t(1+\epsilon)$. Let us refer to this as an "updating" of Λ_t at time $t+1$.

Next, observe that

[6] In the proof of Theorem 3.9, it is sufficient that $\rho(1+\epsilon) \leq 1$. However, in the proof of Theorem 3.10, we require that $\rho(1+\epsilon) < 1$. To avoid proliferating symbols, we use the same ϵ in both proofs.

$$\Gamma_t = \max\{\|\boldsymbol{\theta}_t\|_\infty, \Gamma_{t-1}\}. \tag{3.80}$$

It is a ready consequence of (3.79) that $\Gamma_t \leq \Lambda_t(1+\epsilon)$ for all t, whence $\|\boldsymbol{\theta}_t\|_\infty \lambda_t \leq 1+\epsilon$ for all t. Moreover, if Λ is updated at time t, then

$$\Gamma_t > \Lambda_{t-1}(1+\epsilon) \geq \Gamma_{t-1}.$$

This, coupled with (3.80), shows that $\|\boldsymbol{\theta}_t\|_\infty = \Gamma_t = \Lambda_t$. Hence $\|\boldsymbol{\theta}_t\|_\infty \lambda_t = 1$ at any time where Λ is updated (and at other times $\|\boldsymbol{\theta}_t\|_\infty \lambda_t \leq 1+\epsilon$). In the same vein, if $\lambda_t \|\boldsymbol{\theta}_{t+1}\|_\infty \leq 1+\epsilon$, or equivalently $\|\boldsymbol{\theta}_{t+1}\|_\infty \leq \Lambda_t(1+\epsilon)$, then (3.80) implies that $\Gamma_{t+1} \leq \Lambda_t(1+\epsilon)$, and there is no update at time t.

Now we make the following claim:

Claim 1: If $\{\boldsymbol{\theta}_t\}$ is unbounded, then Λ_t is updated infinitely often (i.e., for infinitely many values of t). To establish the claim, suppose $\{\boldsymbol{\theta}_t\}$ is unbounded, and let T be arbitrary. It is shown that there exists a $\tau \geq T$ such that $\Lambda_\tau > \Lambda_{\tau-1}(1+\epsilon)$, i.e., that Λ gets updated at time τ. Since this argument can again be repeated, it would show that Λ_t gets updated infinitely often if $\{\boldsymbol{\theta}_t\}$ is unbounded, establishing the claim. We prove the existence of such a τ as follows. Since $\{\boldsymbol{\theta}_t\}$ is unbounded, there exists a $t > T$ such that $\|\boldsymbol{\theta}_t\|_\infty > \Lambda_T(1+\epsilon)$. If there already exists a τ between T and $t-1$ such that Λ gets updated at time τ, then we are done. So suppose this is not the case, i.e., that

$$\Lambda_T = \Lambda_{T+1} = \cdots = \Lambda_{t-2} = \Lambda_{t-1}.$$

This implies in particular that Λ is not updated at time $t-1$, i.e., that

$$\Gamma_{t-1} \leq \Lambda_{t-2}(1+\epsilon) = \Lambda_{t-1}(1+\epsilon) = \Lambda_T(1+\epsilon).$$

On the other hand, by assumption $\|\boldsymbol{\theta}_t\|_\infty > \Lambda_T(1+\epsilon)$. Therefore

$$\Gamma_t = \|\boldsymbol{\theta}_t\|_\infty > \Lambda_{t-1}(1+\epsilon).$$

Therefore Λ is updated at time t, i.e.,

$$\Lambda_t = \Gamma_t = \|\boldsymbol{\theta}_t\|_\infty > \Lambda_{t-1}(1+\epsilon).$$

Hence we can take $\tau = t$.

The contrapositive of the claim just proven is the following: If there is a time T such that Λ is *not updated* after time T, then $\{\boldsymbol{\theta}_t\}$ is bounded. From the above discussion, a sufficient condition for this is the following: If there exists a T such that

$$\|\boldsymbol{\theta}_{t+1}\|_\infty \lambda_t \leq 1+\epsilon, \ \forall t \geq T, \tag{3.81}$$

then Λ is not updated after time t, and as a result, $\{\boldsymbol{\theta}_t\}$ is bounded. So henceforth our efforts are focused on establishing (3.81).

Now we derive a "closed form" expression for $\lambda_T \theta_{T+1,i}$. Towards this end, observe the following: A closed form solution to the recursion

$$a_{t+1} = (1-x_t)a_t + b_t \tag{3.82}$$

is given by

$$a_{T+1} = \left[\prod_{r=s}^{T}(1-x_r)\right] a_s + \sum_{t=s}^{T}\left[\prod_{r=t+1}^{T}(1-x_r)\right] b_t, \tag{3.83}$$

for every $0 \leq s \leq T$. The proof by induction is easy and is left to the reader. Observe that

$$\begin{aligned} \lambda_{T+1}\theta_{T+1,i} &= \theta_{T+1,i}(\lambda_{T+1}-\lambda_T) + \theta_{T+1,i}\lambda_T \\ &= \theta_{T+1,i}(\lambda_{T+1}-\lambda_T) + [(1-\alpha_{T,i})\theta_{T,i} + \alpha_{T,i}v_{T,i}]\lambda_T, \end{aligned} \tag{3.84}$$

where we use the shorthand

$$v_{T,i} = \eta_{T,i} + \xi_{T+1,i}.$$

Now we partition $\lambda_{T+1}\theta_{T+1,i}$ as $F_{T+1,i} + G_{T+1,i}$, and find recursive formulas for F and G such that (3.84) holds. Thus we must have

$$F_{T+1,i} + G_{T+1,i} = (1-\alpha_{T,i})(F_{T,i} + G_{T,i}) + \theta_{T+1,i}(\lambda_{T+1} - \lambda_T) + \alpha_{T,i}v_{T,i}\lambda_T.$$

This equation holds if

$$\begin{aligned} F_{T+1,i} &= (1-\alpha_{T,i})F_{T,i} + \theta_{T+1,i}(\lambda_{T+1}, \\ G_{T+1,i} &= (1-\alpha_{T,i})G_{T,i} + \alpha_{T,i}v_{T,i}\lambda_T. \end{aligned} \tag{3.85}$$

The solutions to (3.85) follow readily from (3.83), and are

$$F_{T+1,i} = \left[\prod_{r=s}^{T}(1-\alpha_{r,i})\right] F_s + \sum_{t=s}^{T}\left[\prod_{r=t+1}^{T}(1-\alpha_{r,i})\right](\theta_{t+1,i})(\lambda_{t+1} - \lambda_t), \tag{3.86}$$

$$G_{T+1,i} = \left[\prod_{r=s}^{T}(1-\alpha_{r,i})\right] G_s + \sum_{t=s}^{T}\left[\prod_{r=t+1}^{T}(1-\alpha_{r,i})\right]\alpha_{t,i}\lambda_t v_{t,i}. \tag{3.87}$$

Adding these two equations, noting that $F_s + G_s = \lambda_s\theta_{s,i}$, and expanding $v_{t,i}$ as $\eta_{t,i} + \xi_{t+1,i}$ gives

$$\begin{aligned} \lambda_{T+1}\theta^i_{T+1} &= \sum_{t=s}^{T}\Big[\prod_{r=t+1}^{T}(1-\alpha^i_r)\Big]\theta^i_{t+1}(\lambda_{t+1} - \lambda_t) \\ &+ \sum_{t=s}^{T}\Big[\prod_{r=t+1}^{T}(1-\alpha^i_r)\Big]\alpha^i_t(\lambda_t\eta^i_t + \lambda_t\xi^i_{t+1}) + \Big[\prod_{r=s}^{T}(1-\alpha^i_r)\Big]\lambda_s\theta^i_s. \end{aligned} \tag{3.88}$$

Next, let us expand the right side of (3.88) as

$$\begin{aligned} \lambda_{T+1}\theta^i_{T+1} &= \theta_{T+1,i}(\lambda_{T+1} - \lambda_T) + \sum_{t=s}^{T-1}\Big[\prod_{r=t+1}^{T}(1-\alpha^i_r)\Big]\theta^i_{t+1}(\lambda_{t+1} - \lambda_t) + \Big[\prod_{r=s}^{T}(1-\alpha^i_r)\Big]\lambda_s\theta^i_s \\ &+ \sum_{t=s}^{T}\Big[\prod_{r=t+1}^{T}(1-\alpha^i_r)\Big]\alpha^i_t\lambda_t\eta^i_t + \sum_{t=s}^{T}\Big[\prod_{r=t+1}^{T}(1-\alpha^i_r)\Big]\alpha^i_t\lambda_t\xi^i_{t+1}. \end{aligned} \tag{3.89}$$

Cancelling the common term $\lambda_{T+1}\theta_{T+1,i}$ and moving $\lambda_T\theta_{T+1,i}$ to the left side gives the desired expression, namely

$$\lambda_T\theta_{T+1,i} = A_i(s,T) + B_i(s,T) + C_i(s,T) + D_i(s,T), \tag{3.90}$$

where for $0 \le s < T$

$$A_i(s,T) = \sum_{t=s}^{T-1}\Big[\prod_{r=t+1}^{T}(1-\alpha^i_r)\Big]\theta^i_{t+1}(\lambda_{t+1} - \lambda_t) \tag{3.91}$$

$$B_i(s,T) = \Big[\prod_{r=s}^{T}(1-\alpha^i_r)\Big]\lambda_s\theta^i_s \tag{3.92}$$

$$C_i(s,T) = \sum_{t=s}^{T}\Big[\prod_{r=t+1}^{T}(1-\alpha^i_r)\Big]\alpha^i_t\lambda_t\eta^i_t \tag{3.93}$$

$$D_i(s,T) = \sum_{t=s}^{T}\Big[\prod_{r=t+1}^{T}(1-\alpha_r^i)\Big]\alpha_t^i\lambda_t\xi_{t+1}^i. \tag{3.94}$$

Now we carry on with the proof of Theorem 3.9. First, we find an upper bound for $C_i(s,T)$. Observe that, because $\eta(\cdot)$ is a contraction with constant ρ, and $\rho(1+\epsilon) \leq 1$, we get from (3.79) that $\Gamma_{t+1} \leq \Lambda_t(1+\epsilon)$, or $\Gamma_t\lambda_t \leq 1+\epsilon$. Since $\|\boldsymbol{\theta}_0^t\|_\infty \leq \Gamma_t$, all this implies that

$$\lambda_t\|\eta_t\|_\infty \leq \lambda_t\rho\Gamma_t(1+\epsilon) \leq 1,\ \forall t,$$

by the manner in which ϵ is chosen. Substituting this into (3.93) gives

$$\begin{aligned}
|C_i(s,T)| &\leq \sum_{t=s}^{T}\Big|\Big[\prod_{r=t+1}^{T}(1-\alpha_r^i)\Big]\alpha_t^i\lambda_t\eta_t^i\Big| &(3.95)\\
&\leq \sum_{t=s}^{T}\Big[\prod_{r=t+1}^{T}(1-\alpha_r^i)\Big]\alpha_t^i\\
&= \Big(1-\prod_{r=s}^{T}(1-\alpha_r^i)\Big) &(3.96)
\end{aligned}$$

Next, we derive a recursion for $D_i(s,T)$.

$$\begin{aligned}
D_i(s,T) &= \sum_{t=s}^{T}\Big[\prod_{r=t+1}^{T}(1-\alpha_r^i)\Big]\alpha_t^i\lambda_t\xi_{t+1}^i\\
&= \alpha_{T,i}\lambda_T\xi_{T+1,i} + \sum_{t=s}^{T-1}\Big[\prod_{r=t+1}^{T}(1-\alpha_r^i)\Big]\alpha_t^i\lambda_t\xi_{t+1}^i\\
&= \alpha_{T,i}\lambda_T\xi_{T+1,i} + (1-\alpha_{T,i})\sum_{t=s}^{T-1}\Big[\prod_{r=t+1}^{T-1}(1-\alpha_r^i)\Big]\alpha_t^i\lambda_t\xi_{t+1}^i\\
&= \alpha_{T,i}\lambda_T\xi_{T+1,i} + (1-\alpha_{T,i})D_i(s,T-1). &(3.97)
\end{aligned}$$

Now we make the following claim:

Claim 2: Define a stochastic process $\{M_i(0,t)\}_{t\geq 0}$ by the recursion

$$M_i(0,t+1) = (1-\alpha_{t,i})\lambda_t M_i(0,t) + \alpha_{t,i}\lambda_t\xi_{t+1,i},$$

with the initial condition $M_i(0,0) = 0$. Then $M_i(0,t) \to 0$ almost surely as $t \to \infty$.

The proof of Claim 2 follows readily from Lemma 3.2 by replacing $\xi_{t+1,i}$ by $\lambda_t\xi_{t+1,i}$, and observing that $\lambda_t\xi_{t+1,i}$ satisfies the analogs of (3.73) and (3.74), because $\lambda_t \in \mathcal{M}(\mathcal{F}_t)$ for all t and is bounded.

Next, we make another claim.

Claim 3: For every $\delta > 0$, there exists a t_0 such that the solution $D_i(s,T)$ of the recursion (3.97) with the initial condition $D_i(s,s-1) = 0$ satisfies $|D_i(s,T)| \leq \delta$ for all $t_0 \leq s \leq T$.

The proof follows readily from Lemma 3.2 with minor changes.

We continue with the proof of Theorem 3.9. Choose a t_0 such that

$$|D_i(s,T)| \leq \epsilon/2,\ \forall t_0 \leq s \leq T. \tag{3.98}$$

If Λ is not updated at any time after t_0, then it follows from earlier discussion that $\{\boldsymbol{\theta}_t\}$ is bounded. Otherwise, define s to be any time after t_0+1 at which Λ is updated. Thus $\|\boldsymbol{\theta}_s\|_\infty\lambda_s = 1$ due to the updating. Next, suppose that, for some $T \geq s$, we have that

$$\|\boldsymbol{\theta}_t\|_\infty\lambda_t \leq 1+\epsilon,\ \forall s \leq t \leq T. \tag{3.99}$$

Then it is shown that

$$\|\boldsymbol{\theta}_{T+1}\|_\infty \lambda_{T+1} \leq 1 + \epsilon.$$

The proof is by induction on T. Note that (3.99) holds with $T = s$ to start the induction. Now (3.99) implies that Λ is not updated between times s and T, so that $\lambda_s = \lambda_t = \lambda_T$ for $s \leq t \leq T$. Hence it follows from (3.91) that $A(s,T) = 0$. Next, (3.92) gives

$$|B(s,T)| \leq \prod_{r=s}^{T}(1-\alpha_{r,i})\lambda_s\|\boldsymbol{\theta}_s\|_\infty \leq \prod_{r=s}^{T}(1-\alpha_{r,i}).$$

Combining this bound with (3.96) gives

$$|B_i(s,T) + C_i(s,T)| \leq |B_i(s,T)| + |C_i(s,T)| \leq 1.$$

Finally, from Claim 3, we have that $|D_i(s,T)| \leq \epsilon$. Combining all these shows that

$$|\lambda_{T+1}\theta_{T+1,i}| \leq 1 + \epsilon \ \forall i \in [d], \text{ or } \lambda_{T+1}\|\boldsymbol{\theta}_{T+1}\|_\infty \leq 1 + \epsilon.$$

This completes the inductive step and proves the theorem. □

Proof. Now we come to the proof of Theorem 3.10. We study is the common situation where

$$\mathbf{h}(t, \boldsymbol{\theta}_0^t) = \mathbf{g}(\boldsymbol{\theta}_t),$$

where $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^d$ is a contraction with respect to the ℓ_∞-norm. In other words, there exists a $\gamma < 1$ such that

$$|g_i\boldsymbol{\theta}) - g_i(\boldsymbol{\phi})| \leq \gamma\|\boldsymbol{\theta} - \boldsymbol{\phi}\|_\infty, \ \forall i \in [d], \ \forall \boldsymbol{\theta}, \boldsymbol{\phi} \in \mathbb{R}^d.$$

In addition, define $c_0 = \|\mathbf{g}(\mathbf{0})\|_\infty$. This notation is consistent with (3.67) and (3.70). These hypotheses imply that there is a unique fixed point $\boldsymbol{\theta}^* \in \mathbb{R}^d$ of $\mathbf{g}(\cdot)$. Theorem 3.10 states that $\boldsymbol{\theta}_t \to \boldsymbol{\theta}^*$ almost surely as $t \to \infty$.

For notational convenience, it is assumed that $\boldsymbol{\theta}^* = \mathbf{0}$. The modifications required to handle the case where $\boldsymbol{\theta}^* \neq \mathbf{0}$ are obvious, and can be incorporated at the expense of more messy notation. If $\boldsymbol{\theta}^* = \mathbf{0}$, then $\mathbf{g}(\mathbf{0}) = \mathbf{0}$ so that $c_0 = 0$. Hence we can take $c_1 = 0$, $\rho \in (\gamma, 1)$, and (**??**) becomes

$$\|h(t, \boldsymbol{\theta}_0^t)\|_\infty \leq \rho\|\boldsymbol{\theta}_t\|_\infty \leq \rho\|\boldsymbol{\theta}_0^t\|_\infty, \|\boldsymbol{\eta}_t\|_\infty \leq \rho\|\boldsymbol{\theta}_t\|_\infty.$$

Let $\Omega_0 \subseteq \Omega$ be the subset of Ω such that $U_i(0,t)(\omega) \to 0$ as $t \to \infty$, for all $i \in [d]$. So for each $\omega \in \Omega_0$, there exists a bound $H_0 = H_0(\omega)$ such that

$$\|\boldsymbol{\theta}_0^\infty(\omega)\|_\infty \leq H_0(\omega), \text{ or } |\theta_{t,i}(\omega)| \leq H_0(\omega), \ \forall i \in [d], \ \forall t \geq 0.$$

Hereafter we suppress the dependence of ω. Now choose $\epsilon \in (0,1)$ such that $\rho(1+\epsilon) < 1$. Note that in the proof of Theorem 3.9, we required only that $\rho(1+\epsilon) \leq 1$. For such a choice of ϵ, Theorem 3.9 continues to apply. Therefore $\|\boldsymbol{\theta}_0^\infty\|_\infty \leq H_0$. Now define $H_{k+1} = \rho(1+\epsilon)H_k$ for each $k \geq 0$. Then clearly $H_k \to 0$ as $k \to \infty$.

Now we show that there exists a sequence of times $\{t_k\}$ such that $\|\boldsymbol{\theta}_{t_k}^\infty\|_\infty \leq H_k$, for each k. This is enough to show that $\boldsymbol{\theta}_t(\omega) \to \mathbf{0}$ for all $\omega\omega\Omega_0$, which is the claimed almost sure convergence to the fixed point of $\mathbf{g}$. The proof of this claim is by induction. It is shown that *if* there exists a t_k such that $\|\boldsymbol{\theta}_{t_k}^\infty\|_\infty \leq H_k$, then there exists a t_{k+1} such that $\|\boldsymbol{\theta}_{t_{k_1}}^\infty\|_\infty \leq H_{k_1}$. The statement holds for $k = 0$ – take $t_0 = 0$ because H_0 is a bound on $\|\boldsymbol{\theta}_0^\infty\|_\infty$. To prove the inductive step, suppose that $\|\boldsymbol{\theta}_{t_k}^\infty\|_\infty \leq H_k$. Choose a $\tau_k \geq t_0$ such that the solution to (3.76) satisfies

$$|W_i(\tau_k; t)| \leq \frac{\rho\epsilon}{2}H_k, \ \forall i \in [d], \ \forall t \geq \tau_k. \tag{3.100}$$

Define a sequence $\{Y_{t,i}\}$ by

$$Y_{t+1,i} = (1-\alpha_{t,i})Y_{t,i} + \alpha_{t,i}\rho H_k, Y_{\tau_k,i} = H_k, i \in [d], t \geq \tau_k. \tag{3.101}$$

Then clearly $Y_{t,i} \to \rho H_k$ as $t \to \infty$ for each $i \in [d]$. Now it is claimed that

$$-Y_{t,i} + W_i(\tau_k;t) \leq \theta_{t,i} \leq Y_{t,i} + W_i(\tau_k;t),\ \forall i \in [d], t \geq \tau_k. \tag{3.102}$$

The proof of (3.102) is also by induction on t. The bound (3.102) holds for $t = \tau_k$ because $W_i(\tau_k;\tau_k) = 0$, and the inductive assumption on k, which implies that

$$|\theta_{\tau_k,i}| \leq \|\boldsymbol{\theta}^\infty_{\tau_k}\|_\infty \leq \|\boldsymbol{\theta}^\infty_{t_k}\|_\infty \leq H_k = Y_{\tau_k,i}.$$

Now note that if (3.102) holds for a specific value of k, then for $t \geq \tau_k$ we have

$$\begin{aligned}
\theta_{t+1,i} &= (1-\alpha_{t,i})\theta_{t,i} + \alpha_{t,i}\eta_{t,i} + \alpha_{t,i}\xi_{t+1,i} \\
&\leq 1-\alpha_{t,i})\theta_{t,i} + \alpha_{t,i}\rho H_k + \alpha_{t,i}\xi_{t+1,i} \\
&\leq (1-\alpha_{t,i})[Y_{t,i} + W_i(\tau_k;t)] + \alpha_{t,i}\xi_{t+1,i} \\
&= (1-\alpha_{t,i})Y_{t,i} + \alpha_{t,i}\rho H_k + (1-\alpha_{t,i})W_i(\tau_k;t) + \alpha_{t,i}\xi_{t+1,i} \\
&= Y_{t+1,i} + W_{t+1,i}.
\end{aligned}$$

This proves the upper bound in (3.102) for $t+1$. So the inductive step is established, showing that the upper bound in (3.102) is true for all $t \geq \tau_k$. Now note that $Y_{t,i} \to \rho H_k$ as $t \to \infty$. Hence there exists a $t'_{k+1} \geq \tau_k$ such that

$$Y_{t,i} \leq \rho H_k + \frac{\rho\epsilon}{2} = \rho\left(1 + \frac{\epsilon}{2}\right) H_k,\ \forall t \geq t'_{k+1}.$$

This, combined with (3.100) shows that

$$Y_{t,i} + W_i(\tau_k;t) \leq \rho(1+\epsilon)H_k = H_{k+1},\ \forall t \geq t'_{k+1}.$$

Hence

$$\theta_{t,i} \leq H_{k+1}\ \forall i \in [d], t \geq t'_{k+1}.$$

A parallel argument gives a lower bound: There exists a t''_{k+1} such that

$$-H_{k+1} \leq \theta_{t,i}\ \forall i \in [d], t \geq t''_{k+1}.$$

If we define $t_{k+1} = \max\{t'_{k+1}, t''_{k+1}\}$, then

$$|\theta_{t,i}| \leq H_{k+1}\ \forall i \in [d], t \geq t_{k+1}.$$

This establishes the inductive step for k and completes the proof. □

3.5 Two Time Scale Stochastic Approximation

3.6 Finite-Time Stochastic Approximation

Chapter 4

Approximate Solution of MDPs via Simulation

The contents of Chapter 2 are based on the assumption that the parameters of the Markov Decision Process are all known. In other words, the $|\mathcal{U}|$ possible state transition matrices A^{u_k}, as well as the reward map $R : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ (or its random version), are all available to the agent to aid in the choice of an optimal policy. One can say that the distinction between MDP theory and reinforcement learning (RL) theory is that in the latter, it is *not* assumed that the parameters of the MDP are known. Thus, in RL, one attempts to learn these parameters based on observations.

In the RL literature, a couple of phrases are widely used without always being defined precisely. The first phrase is "tabular methods." As we will see, the methods presented in this chapter attempt to form estimates of the value function, or the action-value function, for a specific policy. These estimates are almost invariably *iterative*, in that the next estimate is based on the previous one. To elaborate this point, let $\hat{V}_t(x_i)$ denote the estimate, at step t, of the value of the state x_i. In many if not most iterative procedures, $\hat{V}_t(x_i)$ depends on the estimates for *all* states $\hat{V}_{t-1}(x_1)$ through $\hat{V}_{t-1}(x_n)$ at step $t-1$. When we attempt to estimate the action-value function, the estimate $Q_t(x_i, u_k)$ depends on *all* previous estimates of $Q_{t-1}(x_j, w_l)$. This requires that the problems under study be sufficiently small that these estimates will fit into the computer storage. The second phrase that is used is "on-policy" simulation and its twin, "off-policy" simulation. What this means is the following: Suppose we wish to approximate the value function V_π for a particular policy π, known as the **target** policy. For this purpose, we have available a sample path of the MDP under *some other* policy θ, which is known as the **behavior** policy. If the behavior policy is the same as the target policy, that is, if we are able to observe a sample path of the Markov process under the same policy that we wish to evaluate, then that situation is referred to as **on-policy**; otherwise the situation is called **off-policy**.

4.1 Monte-Carlo Methods

The phrase "Monte Carlo" methods is used nowadays to refer to almost any technique wherein an expected value of a random variable is approximated by its empirical average, that is, an average of its observed values. The main idea is that, as the number of observations increases, the empirical estimate converges (in probability and almost surely) to its true value. The original "Monte Carlo" method for estimating probabilities of discrete random variables, and for estimating the mean value of real-valued random variables, dates back to the 1940s. The contents of Section 6.1 contain the original Monte Carlo method. When it is desired to use a common set of samples to estimate, simultaneously, the probability of infinitely many sets, or the mean values of infinitely many random variables, that is known as "statistical learning theory." There are several book-length treatments of statistical learning theory, including [45]. This is among the problems studied in Chapter 6

4.1.1 MC Methods for Estimating the Value Function

One application of the Monte Carlo approach is to approximate the value of a policy for a MDP where the underlying parameters are unknown. While Monte Carlo methods are not always well-suited to address this situation, many of the philosophical approaches introduced here are applicable to other techniques as well.

Accordingly, it is assumed that the state space $\mathcal{X}$ and action space $\mathcal{U}$ are known, but everything else is unknown. This part of the discussion here assumes that some policy π has been chosen and implemented, and that we observe a time trajectory of triplets $\{(X_t, U_t, W_{t+1}\}_{t\geq 0}$ where $U_t = \pi(X_t)$ if the policy π is deterministic, $W_{t+1} = R_\pi(X_t)$ where the policy reward R_π is unknown and possibly random, and the state transtion matrix A^π resulting from the policy is also unknown. Because π is fixed throughout, we drop the subscript and superscript π. Note that we permit the reward R to be random; however, initially we restrict the study to the case where the policy is deterministic. Thus, if the same state x_i occurs more than once in the trajectory, so that $X_t = X_\tau = x_i$ for two different t, τ, then we will have $U_t = U_\tau = \pi(x_i)$; however, if the reward is random, we may not have $W_{t+1} = W_{\tau+1}$. At the end of the exercise, we will generate an estimate for the value vector $\mathbf{v}_\pi$ associated with the policy π. Note that, if the objective is to find an optimal policy (either exactly or approximately), then one would have to compute such estimates for each possible policy, of which there are $|\mathcal{U}|^{|\mathcal{X}|}$. This is one of the advantages of Q-learning, studied in Section 4.2.3, in which the approximations converge to the optimal policy.

It is also worth pointing out that, because the policy π is fixed throughout, one can actually think of the process under study as a Markov *reward* process, and not a Markov *decision* process. This is the viewpoint advocated in [35, Chapter 3].

The discussion in this section applies to the case where the underlying Markov process contains one or more absorbing, or terminal, states. Recall that a state x_i is said to be "absorbing" if

$$\Pr\{X_{t+1} = x_i | X_t = x_i\} = 1,$$

or equivalently, the row of the state transition matrix corresponding to the state x_i consists of a 1 in column i and zeros in other columns. The Markov process can have more than one absorbing state. While the dynamics of the MDP are otherwise assumed to be unknown, it is assumed that the learner knows which states are absorbing. By tradition, it is assued that the reward $R(x_i) = 0$ whenever x_i is an absorbing state. Observe too that the state transition matrix A^π depends on the policy under study. So different policies could lead to different absorbing states. However, since we study one policy at a time, it is acceptable to assume that the set of absorbing states under the policy being studied is known.

In this setting, an **episode** refers to any sample path $\{(X_t, U_t, W_{t+1}\}_{t\geq 0}$ that terminates in an absorbing state. Since the policy π is chosen by the learner and is deterministic, it is always the case that $U_t = \pi(X_t)$. Therefore U_t does not add any new information. Once X_t reaches an absorbing state, the episode terminates. The underlying assumption is that, once the Markov process reaches an absorbing state, it can be restarted with the initial state distributed according to its stationary (or some other) distribution. This assumption may not always hold in practice.

Now we discuss how to generate an estimate for the discounted future value, based on a single episode. The assumption mentioned above implies that we can repeat the estimation process over multiple episodes, and then average all of those estimates, to arrive finally at an overall estimate. Define

$$G_t = \sum_{i=0}^{\infty} \gamma^i W_{t+i}. \tag{4.1}$$

If an absorbing state is reached after a finite time, say T, then the summation can be truncated at time T, because $W_{t+1} = 0$ for $t \geq T$. In this connection, recall Theorem 8.12, which gives a formula for the average time needed to hit an absorbing state. Now by definition, for a state $x_i \in \mathcal{X}$, we have

$$V(x_i) = E[G_t | X_t = x_i].$$

Accordingly, suppose that an episode contains the state of interest x_i at time τ, that is, $X_\tau = x_i$. Let us also suppose that the episode terminates at time T. In such a case, we note that

$$E\left[\sum_{i=0}^{T-\tau}\gamma^i W_{\tau+i}\right] = E\left[\sum_{i=0}^{\infty}\gamma^i W_{\tau+i}\right] = V(x_i).$$

Therefore the quantity

$$G_\tau^T := \sum_{i=0}^{T-\tau}\gamma^i W_{\tau+i}$$

provides *an unbiased estimate* for $V(x_i)$. It is therefore *one method* of estimating $V(x_i)$. Now suppose we have L episodes, call them $\mathcal{E}_1, \cdots, \mathcal{E}_L$. Let k the number of these episodes in which the state of interest x_i occurs. Without loss of generality, renumber the episodes so that these are $\mathcal{E}_1$ through $\mathcal{E}_k$. It is of course possible that the state of interest x_i occurs *more than once* in the sample path. Therefore there are a couple of different ways of estimating $V(x_i)$ using this collection of episodes. For each such episode, let τ denote the *first* time at which x_i appears in the state sequence, and T the time at which the episode terminates.[1] Further, define

$$H_l := \sum_{i=0}^{T-\tau}\gamma^i W_{\tau+i}.$$

Then

$$\frac{1}{k}\sum_{l=1}^{k} H_l \tag{4.2}$$

provides an estimate for $V(x_i)$, known as the **first-time estimate**. If the state of interest x_i occurs multiple times within the same episode, then one can form multiple estimates H_l each time the state of interest x_i occurs in the trajectory, and then average them. This called the **everytime** estimate.

Example 4.1. The objective of this example is to illustrate the difference between a first-time estimate and an everytime estimate. Suppose $n = 3$, and for convenience label the states as A, B, C, where A is an absorbing state and B, C are nonabsorbing. Suppose further that $R(C) = 3$, $R(B) = 2$ and of course $R(A) = 0$. Suppose $L = 3$ and that the three episodes (all terminating at A) are:

$$\mathcal{E}_1 = CBCBBA, \mathcal{E}_2 = BBA, \mathcal{E}_3 = BCCBA.$$

Now suppose we wish to estimate the value $V(C)$. Then the episode $\mathcal{E}_2$ does not interest us because C does not occur in it. If the discount factor γ equals 0.9, then we can form the following quantities:

$$H_{11} = 3 + 2\cdot(0.9) + 3\cdot(0.9)^2 + 2\cdot(0.9)^3 + 2\cdot(0.9)^4,$$

$$H_{12} = 3 + 2\cdot(0.9) + 2\cdot(0.9)^2,$$

$$H_{31} = 3 + 3\cdot(0.9) + 2\cdot(0.9)^2, H_{32} = 3 + 2\cdot(0.9).$$

Then $(H_{11} + H_{31})/2$ is the first-time estimate for $V(C)$, while $(H_{11} + H_{12} + H_{31} + H_{32})/4$ is the everytime estimate for $V(C)$.

The convergence of the first-time estimate to the true value function depends on the following fact: In a Markov process with absorbing states, each episode starting from a specified state x_i is statistically independent of every other episode. Thus every first-time estimate for a value $V(x_i)$ can be thought of as providing an independent sample. By averaging these first-time estimates, it is possible to form an estimate of $V(x_i)$. Therefore, if the number of episodes in which the state of interest x_i occurs approaches infinity,

[1] Strictly speaking we should use the notation τ_1, T_1 etc., but we do not do this in the interests of clarity.

it can be stated that the first-time estimate converges to the true value $V(x_i)$. Moreover, one can invoke Hoeffding's inequality of Theorem 8.13 to generate a bound on just how reliable this estimate is, in terms of accuracy and confidence. The everytime estimates are *not* statistically independent. Therefore the analysis of their convergence is more delicate. It is shown in [32] that even the everytime estimate converges to the true value $V(x_i)$, if the number of episodes in which the state of interest x_i occurs approaches infinity. However, unlike the first-time estimate, the everytime estimate is biased, and has higher variance.

Example 4.2. This example, taken from [32] illustrates the bias of the everytime estimate. Consider a Markov process with just two states, called S for state and A for absorbing respectively. Suppose the state transition matrix is

	S	A
S	$1-p$	p
A	0	1

So all trajectories starting in S look like $SS\cdots SA$, where there are say l occurrences of S. If we were to attach a reward $R(S) = 1$, and set the discount factor γ to 1, then the first-time estimate for $V(S)$ would be l, the length of the sample path before hitting A. Moreover, the analysis of hitting times given in Theorem 8.12 shows that the average length of this sample path is $1/p$ (which may not be an integer, but which is the correct answer). On the other hand, there are l everytime estimates of V for such a trajectory, and their sum is $l(l+1)/2$. So the everytime estimate is $(l+1)/2$ for a trajectory that consists of l occurrences of S followed by A. Hence the expected value of the everytime estimate is $((1/p)+1)/2$, which is erroneous by a factor of 2 if p is very small.

The Monte Carlo method for estimating the value of a policy suffers from many drawbacks. In the case of first-time estimates, the total number of samples of a particular value $V(x_i)$ equals the total number of episodes that contain the state x_i. However, each episode can be quite long. Thus the total number of time steps elapsed can be far larger than the number of episodes. This makes the estimation process very slow, with a long observation leading to a small number of samples. Also, because the number of observations is very high, the variance of the estimate can also be very high. Another requirement is that, in order to be able to form an estimate for $V(x_i)$ for a particular state x_i, that state must occur in a large fraction of episodes. In turn this requires that when the Markov process with state transition matrix A^π is started from a randomly selected initial state, the resulting sample path must pass through the state x_i with high probability before hitting an absorbing state. Moreover, the method is based on the assumption that, once the Markov process reaches an absorbing state, it can be restarted with a specified probability distribution for the initial state. This assumption often does not hold in practice. Finally, note that "partial episodes," that is to say, trajectories of the Markov process that do not terminate in an absorbing state, are of no use in forming an estimate of $V(x_i)$. Everytime estimates provide a larger number of samples for $V(x_i)$ than first-time estimates. This is because every episode provides only one first-time estimate, but can provide multiple everytime estimates. The difficulty is that these everytime estimates are *not* statistically independent. Moreover, all of the above comments regarding the drawbacks of first-time estimates apply also to everytime estimates.

4.1.2 MC Methods for Estimating the Action-Value Function

The same Monte Carlo approach can also be used to construct estimates of the action-value function $Q(x_i, u_k)$ instead of the value function $V(x_i)$. The idea is the same as before: We follow several complete episodes, and in each episode, we keep track of how many times a particular pair (x_i, u_k) occurs. Note that in estimating $V(x_i)$, we just keep track of how many times a particular state x_i occurs. This raises a very specific issue. In the case of estimating the value function, it may be justifiable to assume that every state $x_i \in \mathcal{X}$ occurs in sufficiently many sample paths to permit a reasonable estimation of $V_\pi(x_i)$. On the other hand, when it comes to estimating an action-value function, *the only* pairs (x_i, u_k) that occur will be $(x_i, \pi(x_i))$. Therefore, if $\pi \in \Pi_d$, i.e., is a deterministic policy, then the vast majority of state-action pairs will *not* occur. One way

to overcome this problem is to restrict π to be a *probabilistic* policy, with the additional requirement that each state-action pair (x_i, u_k) has a positive probability under π. To put it another way, for each $x_i \in \mathcal{X}$, the probability distribution $\pi(x_i, \cdot)$ on $\mathcal{U}$ has all positive elements. Even in this case however, given that the number of state-action pairs is much larger than the number of states alone, the number of episodes required for reliably estimating the Q-function would be far larger than the number of episodes required for reliably estimating the V-function. Another possible approach is to use "off-policy" sampling, that is, generate samples using a policy that is different from the policy that we wish to evaluate. In this case, the policy π for which we wish to estimate Q_π is called the "target" policy, which can be deterministic, and/or have several missing pairs (x_i, u_k). In contrast, the policy ϕ that is used to generate the samples is called the "behavior" policy. One adjusts for the fact that ϕ may be different from π using a method called "importance sampling," which is described next.

Suppose we wish to estimate $\mathbf{v}_\pi$ for a policy π, but all we have is a sample path under another policy ϕ. Let the sequence of observations be $\{X_t, U_t, W_{t+1}\}_{t=0}^T$. It is assumed that the policy ϕ is probabilistic and "dominates" π. That is,

$$\Pr\{\pi(x_i) = u_k\} > 0 \implies \Pr\{\phi(x_i) = u_k\} > 0. \tag{4.3}$$

The easiest way to satisfy (4.3) is to choose $\phi \in \Pi_p$ (a probabilistic policy) such that the right side of the equation is always positive, that is

$$\Pr\{\phi(x_i) = u_k\} > 0 \; \forall x_i \in \mathcal{X}, u_k \in \mathcal{U}.$$

If we had a sample path under the policy π, then we could estimate $V_\pi(x_i)$ for each fixed state $x_i \in \mathcal{X}$ as follows: Take all episodes that contain x_i, and discard the initial part of the episode that happens before the first occurrence of x_i. For example, suppose a Markov process has the state space $\{B, C, D, A\}$ where A is an absorbing state, and B is the state of interest. Suppose there are three episodes, namely

$$\mathcal{E}_1 = CBDBCDA, \mathcal{E}_2 = CDDCA, \mathcal{E}_3 = BCDBCA.$$

Then we ignore $\mathcal{E}_2$ and discard the first part of $\mathcal{E}_1$. This gives

$$\bar{\mathcal{E}}_1 = BDBCDA, \bar{\mathcal{E}}_2 = BCDBCA.$$

Now back to the discussion. After discarding sample paths that do not contain x_i and the initial parts of the sample paths that do contain x_i, we concatenate them while keeping markers of where one sample path ends and the next one begins. Let $J(x_i)$ denote the number of distinct time instants when $X_t = x_i$. For first-time estimates, we count only the occasions when the sample path starts with x_i, whereas for everytime estimates, we count all occasions when $X_t = x_i$. For instance, in the example above, wth $x_i = B$, for first-time estimates we choose $J(B) = \{1, 7\}$, while for everytime estimates, we choose $J(B) = \{1, 3, 7, 10\}$. In either case, we compute

$$\hat{V}_\pi(x_i) = \frac{1}{|J(x_i)|} \sum_{t \in J(x_i)} G_t, \tag{4.4}$$

where

$$G_t = \sum_{l=0}^{T-t} \gamma^l W_{t+l}$$

is the discounted reward.

Now we describe importance sampling. In case π is a probabilistic policy, there is a likelihood of the state-action pairs

$$\Pr\{U_t, X_{t+1}, U_{t+1}, \cdots, X_T | X_t, U_t^{T-1} \sim \pi\},$$

where $U_t^{T-1} \sim \pi$ means that $\Pr\{U_\tau | X_\tau\}$ has the distribution $\pi(X_\tau)$, for $t \le \tau \le T-1$. This quantity can be expressed as

$$\mathcal{P}_\pi = \prod_{\tau=t}^{T-1} \pi(U_\tau | X_\tau) \Pr\{X_{\tau+1} | X_\tau, U_\tau\}.$$

However, because the sample path is generated using the policy ϕ, what we can actually measure is

$$\mathcal{P}_\phi = \prod_{\tau=t}^{T-1} \phi(U_\tau|X_\tau) \Pr\{X_{\tau+1}|X_\tau, U_\tau\}.$$

Now note there is a simple formula for the ratio of the two likelihoods. Specifically

$$\rho_{[t,T-1]} := \frac{\mathcal{P}_\pi}{\mathcal{P}_\phi} = \prod_{\tau=t}^{T-1} \frac{\pi(U_\tau|X_\tau)}{\phi(U_\tau|X_\tau)}. \tag{4.5}$$

In other words, the unknown transition probabilities $\Pr\{ X_{\tau+1}|X_\tau, U_\tau\}$ simply cancel out. Therefore the quantity $\rho_{[t,T-1]}$ can be computed because π, ϕ are known policies, and U_τ, X_τ can be observed.

With this background, we can modify the estimate in (4.4) in one of two possible ways. The estimate

$$\hat{V}_\pi(x_i) = \frac{1}{|J(x_i)|} \sum_{t \in J(x_i)} \rho_{[t,T-1]} G_t \tag{4.6}$$

is called the "ordinary importance sampling" estimate," whereas

$$\hat{V}_\pi(x_i) = \frac{\sum_{t \in J(x_i)} \rho_{[t,T-1]} G_t}{\sum_{t \in J(x_i)} \rho_{[t,T-1]}} \tag{4.7}$$

is called the "weighted importance sampling." In each case, the term $\rho_{[t,T-1]}$ compensates for off-policy sampling. The ordinary one is unbiased but can have very large variance, whereas the weighted one is biased but consistent, and has lower variance. For further details, see [33, p. 105].

This approach can also be used to estimate Q_π on the basis of a sample path run on another policy ϕ; see [33, p. 110].

4.1.3 MC Methods for Greedy Policy Optimization

Assume that the rather restrictive conditions required to estimate the action-value function for a specific policy π are satisfied, so that we have an estimate for $Q_\pi(x_i, u_k)$ for each state-action pair (x_i, u_k). In this subsection we show how such an estimate can be used to construct a "greedy" policy improvement procedure. This material is taken from [33, Sections 5.3 and 5.4]. Before presenting it, we state and prove the "policy improvement theorem," from [33, Section 4.2].

Recall the definition of the action-value function Q_π from (2.33), and its equivalent characterization in (2.36), namely

$$Q_\pi(x_i, u_k) = E[R(X_t, U_t) + \gamma V_\pi(X_{t+1})|X_t = x_i, U_t = u_k]. \tag{4.8}$$

Theorem 4.1. *(Policy Improvement Theorem) Suppose $\pi, \phi \in \Pi_d$, and moreover*

$$Q_\pi(x_i, \phi(x_i)) \geq Q_\pi(x_i, \pi(x_i)) = V_\pi(x_i), \ \forall x_i \in \mathcal{X}. \tag{4.9}$$

Then

$$V_\phi(x_i) \geq V_\pi(x_i), \ \forall x_i \in \mathcal{X}. \tag{4.10}$$

Moreover, suppose there is a state $x_i \in \mathcal{X}$ such that (4.9) holds with strict inequality, that is

$$Q_\pi(x_i, \phi(x_i)) > Q_\pi(x_i, \pi(x_i)).$$

Then there is a state $x_j \in \mathcal{X}$ such that (4.10) holds with strict inequality, that is,

$$V_\phi(x_j) > V_\pi(x_j).$$

Proof. We reason as follows:

$$\begin{aligned} V_\pi(x_i) &= Q_\pi(x_i, \pi(x_i)) \leq Q_\pi(x_i, \phi(x_i)) \\ &= E[R(X_t, U_t) + \gamma V_\pi(X_{t+1}) | X_t = x_i, U_t = \phi(x_i)] \\ &= E_\phi[R(X_t, U_t) + \gamma Q_\pi(X_{t+1}, \pi(X_{t+1}) | X_t = x_i, U_t = \phi(x_i)] \\ &\leq E_\phi[R(X_t, U_t) + \gamma Q_\pi(X_{t+1}, \phi(X_{t+1}) | X_t = x_i, U_t = \phi(x_i)]. \end{aligned} \tag{4.11}$$

Now look at the second term on the right side, without the γ. This equals

$$\begin{aligned} E_\phi[Q_\pi(X_{t+1}, \phi(X_{t+1}) | X_t = x_i] &= E_\phi[R(X_{t+1}, U_{t+1}) | X_t = X_t] \\ &+ \gamma E_\phi\{E_\phi[V_\pi(X_{t+2} | X_{t+1}, U_{t+1} = \phi(X_{t+1})] | X_t = x_i\}. \end{aligned}$$

Noting that

$$V_\pi(X_{t+2}) = Q_\pi(X_{t+2}, \pi(X_{t+2})) \leq Q_\pi(X_{t+2}, \phi(X_{t+2})),$$

we can repeat the above reasoning. This gives, for every integer l

$$V_\pi(x_i) \leq E_\phi \left[\sum_{i=0}^{l-1} \gamma^i R(X_{t+i}) | X_t = x_i \right] + \gamma^l E_\phi[Q_\pi(X_{t+l}, \phi(X_{t+l})) | X_t = x_i].$$

As $l \to \infty$, the second term approaches zero, while the first term approaches $V_\phi(x_i)$. Hence (4.10) follows. The statements about strict inequalities are easy to prove and are left as an exercise. □

The policy improvement theorem suggests the following "greedy" approach to finding an optimal policy. Suppose that we have an initial policy and corresponding action-value function Q_π. We can define a new "greedy" policy via

$$k^* := \arg\max_{u_k \in \mathcal{U}} Q_\pi(x_i, u_k), \phi(x_i) = u_{k^*}, \ \forall x_i \in \mathcal{X}. \tag{4.12}$$

Then (4.9) holds by construction, and it follows from Theorem 4.1 that $V_\phi(x_i) \geq V_\pi(x_i)$ for all $x_i \in \mathbf{x}$, or equivalently, $\mathbf{v}_\phi \geq \mathbf{v}_\pi$. Now we can compute the action-value function corresponding to ϕ and repeat. A consequence of Theorem 4.1 is that unless equality holds for all $x_i \in \mathcal{X}$ in (4.9), we have that at least one component of $\mathbf{v}_\phi$ exceeds that of $\mathbf{v}_\pi$. Now, it is easy to show that if the above greedy update policy terminates with (4.9) holding with equality for all $x_i \in \mathcal{X}$, then not only is $\mathbf{v}_\phi = \mathbf{v}_\pi$, but both are optimal policies. Another noteworthy point is that the incremental update in (4.12) can be implemented for just one index i; in other words, the update can be done asynchronously.

Now we show how to use Theorem 4.1 to construct "ϵ-greedy" policies. The update rule (4.12) can initially perform poorly when the current guess π is far from being optimal. Moreover, in reinforcement learning, there is always a trade-off between exploration and exploitation. One way to achieve both exploration and exploitation simultaneously is to use probabilistic policies, where for a given state x_k, *every* policy is applied with positive probability.

Suppose $\pi \in \Pi_p$ is a probabilistic policy. Then we use the notation

$$\pi(u_k | x_i) := \Pr\{U_t = u_k | X_t = x_i\}.$$

A policy $\pi \in \Pi_p$ is said to be **ϵ-soft** if

$$\pi(u_k | x_i) \geq \epsilon, \ \forall u_k \in \mathcal{U}, x_i \in \mathcal{X}.$$

Suppose $\pi \in \Pi_p$ is the current ϵ-soft policy. We can generate an updated ϵ-soft policy ϕ as follows: Define a *deterministic* policy $\psi \in \Pi_d$ by

$$k^* := \arg\max_{u_k \in \mathcal{U}} Q_\pi(x_i, u_k), \psi(x_i) = u_{k^*}, \ \forall x_i \in \mathcal{X}.$$

Now define the ϵ-soft policy $\phi \in \Pi_p$ by

$$\phi(u_k|x_i) = \begin{cases} \frac{\epsilon}{|\mathcal{U}|} & k \neq k^* \\ \frac{\epsilon}{|\mathcal{U}|} + (1-\epsilon) & k = k^*. \end{cases} \tag{4.13}$$

Theorem 4.2. *With ϕ defined as in* (4.13)*, the inequality* (4.9) *holds.*

Proof. Note that, for each fied $x_i \in \mathcal{X}$, we have

$$\begin{aligned} Q_\pi(x_i, \phi(x_i)) &= \sum_{u_k \in \mathcal{U}} \phi(u_k|x_i) Q_\pi(x_i, u_k) \\ &= \frac{\epsilon}{|\mathcal{U}|} \sum_{u_k \in \mathcal{U}} Q_\pi(x_i, u_k) + (1-\epsilon) \max_{u_k \in \mathcal{U}} Q_\pi(x_i, u_k) \\ &\geq \frac{\epsilon}{|\mathcal{U}|} \sum_{u_k \in \mathcal{U}} Q_\pi(x_i, u_k) + (1-\epsilon) \sum_{u_k \in \mathcal{U}} \frac{\pi(u_k|x_i) - \frac{\epsilon}{|\mathcal{U}|}}{1-\epsilon} Q_\pi(x_i, u_k) \\ &= \sum_{u_k \in \mathcal{U}} \pi(u_k|x_i) Q_\pi(x_i, u_k), \end{aligned}$$

which is (4.9). In the next to last equation, we reason as follows: Define nonnegative constants λ_k that add up to one, as follows:

$$\lambda_k := \frac{1}{1-\epsilon} \left[\pi(u_k|x_i) - \frac{\epsilon}{|\mathcal{U}|} \right].$$

Then

$$\sum_{u_k \in \mathcal{U}} \lambda_k Q_\pi(x_i, u_k) \leq \max_{u_k \in \mathcal{U}} Q_\pi(x_i, u_k).$$

Because (4.9) holds, we can apply the ϵ-greedy updating rule to improve the policy. □

The import of Theorem 4.2 is that the above ϵ-greedy updating rule will eventually converge to the optimal ϵ-soft policy. The ϵ-greedy policy and updating rule can be combined with a "schedule" for reducing ϵ to zero, which would presumably converge to the optimal policy.

4.2 Temporal Difference Methods

4.2.1 Basic Temporance Difference Method

Temporal difference (TD) methods are another way to approximate the value $\mathbf{v}_\pi$ corresponding to a specific policy π. Unlike Monte Carlo methods that wait until an entire episode is completed before computing an estimate for $\mathbf{v}_\pi$, Temporal Difference (TD) methods update various estimates at each time step. Since each estimate depends on an earlier estimate, this is known as "bootstrapping." As before it is assumed that a particular policy π has been chosen and implemented, and that a sample path $\{(X_t, U_t, W_{t+1})\}_{t \geq 0}$ is observed. So hereafter we drop the subscript π on V.

Suppose now that x_i is the state of interest, and that $X_t = x_i$. There are two equivalent formulas for the value function associated with a state x_i, one "explicit" and the other recursive. The explicit formula is

$$V(x_i) = E\left[\sum_{\tau=0}^{\infty} \gamma^\tau R_{t+\tau+1} | X_t = x_i \right], \tag{4.14}$$

where

$$R_{t+\tau+1} = R(X_{t+\tau}, U_{t+\tau})$$

is the reward, but paid at time $t+\tau+1$. If the reward is random, then the expectation includes this randomness also. Now (4.14) can be expanded as

$$V(x_i) = R_{t+1} + E\left[\sum_{\tau=1}^{\infty} \gamma^\tau R_{t+\tau+1} | X_t = x_i\right].$$

This can now be rewritten as the recursion (cf. (2.32))

$$V(x_i) = R_{t+1} + E\left[V(X_{t+1}) | X_t = x_i\right]. \tag{4.15}$$

With this background, we can think of Monte Carlo simulation as a way to approximate $V(x_i)$ using the formula (4.14). Specifically, suppose an episode passes through the state of interest x_i at time t and terminates in an absorbing state a tme T. Then the quantity

$$\hat{V}(x_i) = \sum_{\tau=0}^{T-t} \gamma^\tau R_{t_\tau+1}$$

provides an approximation to $V(x_i)$.

Temporal Difference Learning (TD) takes a different approach. One can express (4.15) as

$$V(X_t) \sim R_{t+1} + \gamma V(X_{t+1}), \tag{4.16}$$

where the symbol $\sim$ denotes that the random variables on both sides of the formula are the same. Check this statement, and if necessary, elaborate. So, if $\hat{\mathbf{v}} \in \mathbb{R}^n$ is a current guess for the value vector at time t, then we can take $\hat{V}(X_t)$ as a proxy for $V(X_t)$, W_{t+1} as a proxy for R_{t+1}, and $\hat{V}(X_{t+1})$ as a proxy for $V(X_{t+1})$. Therefore the error

$$\delta_{t+1} = W_{t+1} + \gamma \hat{V}_t(X_{t+1}) - \hat{V}_t(X_t), \tag{4.17}$$

gives a measure of how erroneous the estimate $\hat{V}(x_i)$ is. Call this the "Bellman gradient" and give an interpretation. But it does not tell us how far off the estimates $\hat{V}(x_j), x_j \neq x_i$ are. So we choose a predetermined sequence of step sizes $\{\alpha_t\}$, and adjust the estimate $\hat{\mathbf{v}}$ as follows:

$$\hat{V}_{t+1}(x_j) = \begin{cases} \hat{V}_t(x_j) + \alpha_t \delta_{t+1} & \text{if } X_{t+1} = x_j, \\ \hat{V}_t(x_j), & \text{otherwise.} \end{cases} \tag{4.18}$$

Note that only the component of $\hat{\mathbf{v}}$ corresponding to the current state at time t is adjusted, while the remainder are left unaltered.

Note that if $\hat{\mathbf{v}}_t = \mathbf{v}$, the true value vector, then

$$E[\delta_{t+1} | X_t = x_i] = E[W_{t+1} + \gamma \hat{V}(X_{t+1}) - \hat{V}(X_t) | X_t = x_i] = 0. \tag{4.19}$$

Thus δ_{t+1} is the "temporal difference" at time $t+1$ between what we expect to see and what we actually see. If that difference is not zero, we correct the corresponding component of $\hat{V}$ by moving in that direction.

Theorem 4.3. *Suppose*

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \sum_{t=0}^{\infty} \alpha_t^2 < \infty. \tag{4.20}$$

Then the estimated value vector $\hat{\mathbf{v}}_t$ converges to the true value vector $\mathbf{v}$ almost surely.

Proof. (See [35, Section 3.1.1].) To analyze the behavior of TD learning, define the map $F : \mathbb{R}^n \to \mathbb{R}^n$ as

$$F\mathbf{y} := \mathbf{r} + \gamma A\mathbf{y} - \mathbf{y} = \mathbf{r} + (\gamma A - I_n)\mathbf{y}, \tag{4.21}$$

where A is the state transition matrix of the unknown Markov process. Then (4.15) can be rewritten in vector notation as

$$\mathbf{v} = \mathbf{r} + \gamma A\mathbf{v},$$

where $\mathbf{v}$ is the unknown value vector. With this reformulation, it can be seen that TD learning is just asynchronous stochastic approximation with the function $f : \mathbb{R}^n \to \mathbb{R}^n$ equal to F. The conditions for the asynchronous version of stochastic approximation to converge are broadly similar to those in Theorem **??**. Specifically, it suffices to establish that the differential equation

$$\dot{\mathbf{y}} = F\mathbf{y} = \mathbf{r} + (\gamma A - I_n)\mathbf{y} \tag{4.22}$$

is globally asymptotically stable around the equilibrium $\mathbf{y} = \mathbf{v}$, the true value function, and moreover, there is a Lyapunov function that satisfies the conditions of (**??**) and (**??**). (Note that the V in those equations is the Lyapunov function, and not the value function.) But this is immediate. Note that the equation (4.22) is *linear*. Moreover, because $\rho(\gamma A) = \gamma < 1$, all eigenvalues of $\gamma A - I_n$ have negative real parts. So the global asymptotic stability of this equilibrium can be established using a *quadratic* Lyapunov function, so that (**??**) and (**??**) are automatically satisfied. Therefore $\hat{\mathbf{v}}_t \to \mathbf{v}$ as $t \to \infty$ almost surely. □

Note that the recursion formula for TD learning does not require the trajectory to be an episode. Therefore, for a given sample path of length T, the TD updates operate T times, whereas MC updates operate only as many times as the number of episodes contained in the sample path. Indeed, aside from the fact that $\hat{\mathbf{v}}$ is updated at every time instant, this is one more advantage, namely, that partial episodes are also useful. Moreover, TD learning can also (apparently) be used with Markov processes that do not have an absorbing state. However, in case the Markov process does have an absorbing state and the sample path corresponds to an episode, it is possible to derive a useful formula that can be used to relate Monte Carlo simulation to TD learning. Specifically, the Monte Carlo return over an episode can be expressed as a sum of temporal differences. Define, as before

$$G_t := \sum_{\tau=0}^{T-t} \gamma^\tau W_{t+\tau+1} \tag{4.23}$$

to be the total return over an episode starting at time t and ending at T. Then G_t satisfies the recursion

$$G_t = W_{t+1} + \gamma G_{t+1}.$$

So we can write

$$\begin{aligned} G_t - \hat{V}(X_t) &= W_{t+1} + \gamma G_{t+1} - \hat{V}(X_t) \\ &= W_{t+1} + \gamma G_{t+1} - \hat{V}(X_t) - \gamma\hat{V}(X_{t+1}) + \gamma\hat{V}(X_{t+1}) \\ &= \delta_t + \gamma(G_{t+1} - \hat{V}(X_{t+1})). \end{aligned} \tag{4.24}$$

Now repeat until the end of the episode, when both G_{T+1} and $\hat{V}(X_{T+1})$ are zero (because X_T is an absorbng state). This leads to

$$G_t - \hat{V}(X_t) = \sum_{\tau=0}^{T-t} \gamma^\tau \delta_{t+\tau}. \tag{4.25}$$

In TD learning, it is not necessary to look "only" one step ahead. It is possible to derive an "l-step look-ahead" TD predictor. Note that if we define

$$G_t^{t+l} := \sum_{\tau=0}^{l-1} \gamma^t R_{t+\tau+1}, \tag{4.26}$$

then with $l = 0$ we get

$$G_t^{t+l} = G_t = R_{t+1}.$$

(Note that the one-step look-ahead predictor corresponds to $l = 0$.) For $l > 0$, the analog of (4.16) is

$$V(X_t) \sim G_t^{t+l} + \gamma^l V(X_{t+l}). \tag{4.27}$$

So, given an estimated value vector $\hat{\mathbf{v}}$, we observe that if $\hat{\mathbf{v}} = \mathbf{v}$, the true value vector, then

$$E[G_t^{t+l} + \gamma^l \hat{V}(X_{t+l}) - \hat{V}(X_t)|X_t = x_i] = 0. \tag{4.28}$$

Let us define

$$\delta_{t+1}^{t+l+1} := G_t^{t+l} + \gamma^l \hat{V}(X_{t+l}) - \hat{V}(X_t). \tag{4.29}$$

The extra ones in the subscript and the superscript are to ensure that when $l = 0$, we have that $\delta_{t+1}^{t+l+1} = \delta_{t+1}$ as defined in (4.17). In view of (4.28), we can think of δ_{t+1}^{t+l+1} as an l-step temporal difference. The analogous updating rule for this l-step TD learning is

$$\hat{V}_{t+1}(x_j) = \begin{cases} \hat{V}_t(x_j) + \alpha_t \delta_{t+1}^{t+l+1} & \text{if } X_{t+1} = x_j, \\ \hat{V}_t(x_j), & \text{otherwise.} \end{cases} \tag{4.30}$$

The limiting behavior of first-time Monte Carlo and TD are quite different. Suppose that the duration of the data is T, which consists of K episodes, of duration $T_1, \cdots, T_K$ respectively For each episode k, form the first-time estimate $G_k(x_i)$ for each state $x_i \in \mathcal{X}$. If the state x_i does not occur in episode k, then this term is set equal to zero, which is functionally equivalent to omitting it. Then

$$\hat{V}_{\text{MC}}(x_i) \to \arg\min_{V(x_i)} \sum_{k=1} \sum_{k=1}^{T_k} (G_k(x_i) - V(x_i))^2. \tag{4.31}$$

In contrast, the TD estimate converges to the maximum likelihood estimate

$$\hat{a}_{ij}^{u_k} \frac{1}{n(x_i, u_k)} \sum_{k=1} \sum_{k=1}^{T_k} I_{(X_t, X_{t+1}, U_t) = (x_i, x_j, u_k)}. \tag{4.32}$$

As a result, TD methods converge more quickly than Monte Carlo methods. These statements are made on [33, p. 128].

4.2.2 SARSA: On-Policy Control

In this section we introduce SARSA which is an on-policy method for estimating the action-value function for a given policy. Then we show how the policy itself can be improved upon in an iterative fashion. Note that SARSA stands for State, Action, Reward, State, Action.

We begin with the observation that if a policy π is chosen, whether $\pi \in \Pi_d$ or $\pi \in \Pi_p$, the resulting process $\{X_t\}$ is Markovian. Equally, the joint state-action process $\{(X_t, U_t)\}$ is also Markovian. So, if our aim is to estimate the action-value function $Q_\pi : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, then Q_π can be viewed as a reward for this joint Markov process. Recall from Theorem 2.5 that if $\pi \in \Pi_d$, then Q_π satisfies the recursion (2.34), namely

$$Q_\pi(x_i, u_k) = R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} Q_\pi(x_j, \pi(x_j)).$$

However, for the case of probabilistic policies, it is advantageous to write the recursion as

$$Q_\pi(x_i, u_k) = R(x_i, u_k) + \gamma E[Q_\pi(X_{t+1}, \pi(X_{t+1}))|(X_t, U_t) = (x_i, u_k)]. \tag{4.33}$$

Because this is a recursion, it is amenable to learning via a temporal difference method.

Given a sample path $\{(X_t, U_t, W_{t+1})\}_{t\geq 0}$, start with some initial guess $\hat{Q}_0$ for the action-value function. Observe that, if at time t, the estimate $\hat{Q}_{t,\pi}$ were to be correct, then

$$E[R_{t+1} + \gamma \hat{Q}_{t,\pi}(X_{t+1}, \pi(X_{t+1})) - \hat{Q}_{t,\pi}(X_t, U_t)|X_t, U_t] = 0.$$

Therefore the term

$$\theta_{t+1} := W_{t+1} + \gamma \hat{Q}_{t,\pi}(X_{t+1}, U_{t+1}) - \hat{Q}_{t,\pi}(X_t, U_t) \tag{4.34}$$

can serve as a temporal difference. Hence we can update the guess $\hat{Q}$ as

$$\hat{Q}_{t+1,\pi}(x_i, u_k) = \begin{cases} \hat{Q}_{t,\pi}(X_t, U_t) + \alpha_{t+1}\theta_{t+1} & \text{if } (x_i, u_k) = (X_t, U_t), \\ \hat{Q}_{t,\pi}(X_t, U_t) & \text{if } (x_i, u_k) \neq (X_t, U_t). \end{cases} \tag{4.35}$$

It can be surmised that, if $\{\alpha_t\}$ approaches zero as per the conditions (4.20), then $Q_{t,\pi}$ converges almost surely to Q_π. However, this requires that every possible pair $(x_i, u_k) \in \mathcal{X} \times \mathcal{U}$ must be visited infinitely often by the sample path, which is impossible if π is a deterministic policy. Hence, in order to apply the above method for a deterministic policy, one should use an ϵ-soft version of π.

All this produces the action-value pair for *one fixed* policy. However, it is possible to combine this with an ϵ-greedy improvement approach to converge to an optimal policy. Specifically, we can update the policy π using an ϵ-greedy approach on the current policy, as before: Define

$$k^* = \arg\min_{u_k \in \mathcal{U}} \hat{Q}_\pi(x_i, u_k), \psi(x_i) = u_{k^*},$$

$$\phi(u_k|x_i) = \begin{cases} \frac{\epsilon}{|\mathcal{U}|} & k \neq k^* \\ \frac{\epsilon}{|\mathcal{U}|} + (1-\epsilon) & k = k^*, \end{cases}$$

where all policies are functions of t, but the dependence is not explicitly displayed in the interests of clarity. Now we can reduce ϵ to zero over time. Then perhaps $\phi \to \pi^*$ and $\hat{Q} \to Q^*$ as $t \to \infty$. However, this is just a hope. The Q-learning approach given next turns this hope into a reality by adopting a slightly different approach to updating Q.

4.2.3 Q-Learning

A substantial advance in RL came in the paper [47]. In this paper, the authors propose an iterative scheme for learning the optimal action-value function $Q^*(x_i, u_k)$, by starting with an arbitrary initial guess, and then updating the guess at each time instant.

Recall the following definitions and facts from Section 2.2. The **action-value function** is defined as follows, for a given policy π (cf. (2.33)):

$$Q_\pi(x_i, u_k) := E_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_\pi(X_t) | X_0 = x_i, U_0 = u_k \right].$$

As shown in (2.34), the function Q_π satisfies the recursion

$$Q_\pi(x_i, u_k) = R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} Q_\pi(x_j, \pi(x_j)).$$

If the return R is a random function of X_t, U_t, then the first term on the right side would be the expected value of the reward $R(X_t, U_t)$. The optimal action-value function Q^* satisfies the relationships

$$Q^*(x_i, u_k) = R(x_i, u_k) + \gamma \sum_{j=1}^{n} a_{ij}^{u_k} \max_{w_l \in \mathcal{U}} Q^*(x_j, w_l),$$

$$V^*(x_i) = \max_{u_k \in \mathcal{U}} Q^*(x_i, u_k).$$

The recursion relationship for Q^* suggests the following iterative scheme for estimating this function. Start with some arbitrary function $Q_0 : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$. Choose a sequence of positive step sizes $\{\alpha_t\}_{t\geq 0}$ satisfying

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \sum_{t=0}^{\infty} \alpha_t^2 < \infty. \tag{4.36}$$

As the time series $\{(X_t, U_t, W_{t+1})\}_{t\geq 0}$ is observed, update function Q_t as follows:

$$Q_{t+1}(x_j, w_j) = \begin{cases} Q_t(x_j, w_l) + \alpha_{t+1}[W_{t+1} + \gamma V_t(x_j) - Q_t(x_j, w_l)], & \text{if } (X_{t+1}, U_{t+1}) = (x_j, w_l), \\ Q_t(x_j, w_l) & \text{otherwise,} \end{cases} \tag{4.37}$$

where

$$V_t(x_j) = \max_{u_l \in \mathcal{U}} Q_t(x_j, u_l). \tag{4.38}$$

In other words, if $(X_{t+1}, U_{t+1}) = (x_j, w_l)$, then the corresponding estimate $Q_{t+1}(x_j, w_j)$ is updated, but the estimates of $Q(x_i, u_k)$ for $(X_{t+1}, U_{t+1}) \neq (x_j, w_l)$ are not updated.

The convergence of the above scheme is analyzed in the next theorem.

Theorem 4.4. *(See [47, p. 282].) If there exists a finite constant R_M such that $|R(X_t, U_t)| \leq R_M$, then*

$$Q_t(x_i, u_k) \to Q^*(x_i, u_k) \text{ as } t \to \infty, \ \forall x_i \in \mathcal{X}, u_k \in \mathcal{U}, \text{ w.p. } 1. \tag{4.39}$$

One of the main advantages of Q-learning over learning the value function V_π is the following: If $(X_t, U_t) = (x_i, u_j)$, then the quantity

$$R(x_i, u_k) + \gamma \max_{w_l \in \mathcal{U}} Q(X_{t+1}, w_l)$$

is an unbiased sample of the quantity

$$R(x_i, u_k) + \gamma \sum_{j \in [n]} a_{ij}^{u_k} \max_{w_l \in \mathcal{U}} Q(x_j, w_l).$$

In contrast, the quantity

$$\max_{u_k \in \mathcal{U}} [R(x_i, u_k) + \gamma V(X_{t+1})]$$

is *not* an unbiased sample of the quantity

$$\max_{u_k \in \mathcal{U}} \left[R(x_i, u_k) + \gamma \sum_{j \in [n]} a_{ij}^{u_k} v_j \right].$$

Problem 4.1. The objective of this problem is to analyze the l-step look-ahead Temporal Difference Learning rule proposed in (4.30). Suppose we have a Markov decision process with a prespecified policy π (deterministic, or probabilistic, it does not matter).

- Start with (2.32) for the value vector associated with the policy π, namely

  $$\mathbf{v} = \mathbf{r} + \gamma A \mathbf{v},$$

  where we havem omitted π as it is anyway fixed. Show that the following l-step look-ahead version of the above equation is also true:

  $$\mathbf{v} = \left(\sum_{i=0}^{l-1} \gamma^i A^i \right) \mathbf{r} + A^l \mathbf{v}.$$

- Using this fact, show that (4.27) is true.
- Show that (4.30) is a stochastic approximation approach to solving the above equation.
- Hence establish that if the conditions in (4.20) hold, then the l-step look-ahead TD learning converges almost surely to the true value function.

Hint: If A is a row-stochastic matrix, $\gamma < 1$, and l is an integer, show that all eigenvalues of the matrix

$$\sum_{i=0}^{l} \gamma^i A^i - I$$

have negative real parts.

Problem 4.2. Generate a 20×20 row-stochastic matrix A, and a 20×1 reward vector $\mathbf{r}$, in any manner you wish.

- Compute the actual value function for the A and $\mathbf{r}$ you have generated using (2.32).
- Generate a sample path of length 200 time steps starting from some arbitrary initial state. Compute the associated reward functions as a part of the sample path. (Note that the action is not required as the policy remains constant.)
- Choose the discount factor $\gamma = 0.95$ and the look-ahead step length $l = 5$ or $l = 10$. Using the sample path $\{(X_t, W_{t+1})\}$ generated above, estimate the value function using conventional TD learning, and l-step TD-learning with $l = 5$ and $l = 10$. Plot the Euclidean norm of the error (difference between the true and estimated value vector) as a function of the iteration number, for each of these three approaches. What conclusions do you draw, if any?

Chapter 5

Parametric Approximation Methods

Until now we have presented various iterative methods for computing the value function associated with a prespecified policy, and for recursively improving a starting policy towards an optimum. These methods converge asymptotically to the right answer, as the number of episodes (for Monte Carlo methods) or the number of samples (for Temporal Difference and Q-learning methods) approaches infinity. So in principle one could truncate these methods when some termination criterion is met, such as the change from one iteration to the next being smaller than some prespecified threshold. However, the emphasis is still finding the *exact* value function, or the *exact* optimal policy.

This approach can be used to find the value function if the size n of the state space $|\mathcal{X}|$ is sufficiently small. Similarly, this approach can be used to find the action-value function if the product nm of the size of the state space $|\mathcal{X}|$ and the size of the action space $|\mathcal{U}|$ is sufficiently small. However, in many realistic applications, these numbers are too large. Note that the value function $V : \mathcal{X} \to \mathbb{R}$ is an n-dimensional vector. Hence we can identify the "value space" with $\mathbb{R}^n$. Similarly, the policy π can be associated with a matrix $B \in \mathbb{R}^{n \times m}$, where $b_{ij} = \Pr\{\pi(x_i) = u_j\}$. The associated action-value function $Q : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, and can be thought of as an nm-dimentional vector. In principle, every vector in $\mathbb{R}^n$ can be the value vector of some Markov reward process, and every vector in $\mathbb{R}^{nm}$ can be the action-value function of some Markov Decision Process.

Therefore, when these numbers are too large, it becomes imperative to *approximate* the value or action-value functions using a smaller number of parameters (smaller than n for value functions and smaller than nm for action-value functions and policies). The present chapter is devoted to a study of such approximation methods. We begin with methods for approximating the value function. Then we can ask: Is it possible to find directly an approximation the optimal policy using a suitable set of basis functions? This question lies at the heart of "policy gradient" methods and is studied in Section 5.3. The solution is significantly facilitated by a result known as the "policy gradient theorem." Proceeding further, we can attempt simultaneously to approximate *both* the policy function and the value function over a common set of basis functions. This is known as the "actor-critic" approach and is studied in Section 5.3.

5.1 Value Approximation Methods

5.1.1 Preliminaries

The object of study in the present section is the approximation of the value function under a *fixed* policy π. Therefore, though we will start off with the notation $\mathbf{v}_\pi$, we will drop the subscript at after some time.

In Chapter 4, we proposed a few methods for estimating the value $V_\pi(x_i)$ of a state of interest x_i under a fixed policy π. In the Monte Carlo approach, multiple episodes of the process starting at a state of interest x_i could be used to estimate $V_\pi(x_i)$. In the Temporal Difference approach, it is possible to bootstrap and update the estimate $V_\pi(x_i)$ after each time instant, and not just after each episode. Now let $\mathbf{v}_\pi$ denote the

true n-dimensional value vector corrsponding to the policy π. Suppose we try to approximate $\mathbf{v}_\pi$ by another vector of the form $\hat{\mathbf{v}}_\pi(\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ is a vector of adjustable parameters, and $d \ll n$. Thus $\hat{\mathbf{v}}_\pi : \mathbb{R}^d \to \mathbb{R}^n$ where $n = |\mathcal{X}|$. Depending on the context, we can also think of $\hat{\mathbf{v}}$ as a collection of n maps, indexed by the state x_i, where each $\hat{V}(x_i, \cdot) : \mathbb{R}^d \to \mathbb{R}$. In general it is not possible to choose the parameter vector $\boldsymbol{\theta}$ so as to make $\hat{\mathbf{v}}_\pi(\boldsymbol{\theta})$ equal to $\mathbf{v}_\pi$. In the approaches suggested in Chapter 4, the estimate of $V_\pi(x_i)$ did not depend on the estimate of $V_\pi(x_j)$ for $x_j \neq x_i$. However, in the present instance, since our aim is to estimate all values $V_\pi(x_i)$ *simultaneously*, we need to define a suitable error criterion.

Suppose have a sample path $\{X_t\}_{t=0}^T$ that is used to estimate V_π. A natural choice is the least-squares criterion given by

$$E(\boldsymbol{\theta}) := \frac{1}{2(T+1)} \sum_{t=0}^{T} [V_\pi(X_t) - \hat{V}(X_t, \boldsymbol{\theta})]^2.$$

The above error criterion suggests that an error at any one time t is as significant as an error at any other time τ. The above error criterion can be rewritten as

$$E(\boldsymbol{\theta}, \boldsymbol{\mu}) := \frac{1}{2} \sum_{x_i \in \mathcal{X}} \mu_i [V_\pi(x_i) - \hat{V}_\pi(x_i, \boldsymbol{\theta})]^2, \tag{5.1}$$

where

$$\mu_i := \frac{1}{T+1} \sum_{t=0}^{T} I_{\{X_t = x_i\}}$$

is the fraction of times that $X_t = x_i$, the state of interest, in the sample path. With this definition of the coefficients μ_i, the error criterion can be further rewritten as

$$E(\boldsymbol{\theta}, \boldsymbol{\mu}) := \frac{1}{2} [\mathbf{v}_\pi - \hat{\mathbf{v}}_\pi(\boldsymbol{\theta})]^\top M [\mathbf{v}_\pi - \hat{\mathbf{v}}_\pi(\boldsymbol{\theta})], \tag{5.2}$$

where $M \in \mathbb{R}^{n \times n}$ is the diagonal matrix with the μ_i as its diagonal elements.

This raises the question as to what the coefficients μ_i should be. Clearly, in a Markov process, the seriousness of an error in estimating $V_\pi(x_i)$ should depend on how likey the state x_i is likely to occur in a sample path. Thus it would be reasonable to choose the coefficient vector $\boldsymbol{\mu}$ as the stationary distribution of the Markov process. However, there are two difficulties with this approach. The first difficulty is that we do not know what this stationary distribution is. So we try to construct an estimate of the stationary distribution, before we start the process of approximating the unknown value vector $\mathbf{v}_\pi$. The second difficulty is that if the Markov process has one or more absorbing states, then the above approach would be meaningless as shown below.

If we assume that, under the policy π, the resulting Markov process is irreducible (see Section 8.2), then there is a unique stationary distribution $\boldsymbol{\mu}$ of the process. Moreover, $\boldsymbol{\mu} > \mathbf{0}$, i.e., every component μ_i is positive. The question is how to estimate this stationary distribution. Theorem 8.8 and specifically (8.32) provide an approach. For *any* function $f : \mathcal{X} \to \mathbb{R}$, we have that

$$E[f, \boldsymbol{\mu}] = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} f(X_\tau). \tag{5.3}$$

Now define a function $f_i : \mathcal{X} \to \mathbb{R}$ by $f_i(x_i) = 1$, and $f_i(x_j) = 0$ if $j \neq i$. Then it is easy to see that the expected value $E[f_i, \boldsymbol{\mu}] = \mu_i$. Moreover, (5.3) implies that

$$\mu_i = E[f_i, \boldsymbol{\mu}] = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} I_{\{X_t = x_i\}}. \tag{5.4}$$

Thus, in a sample path, we count what fraction are equal to the state of interest x_i. As $t \to \infty$, this fraction converges almost surely to μ_i. Hence, if we take any sample path (or collection of sample paths), then the

fraction of occurrences of x_i in the sample path is a good approximation to μ_i. These estimates can be used in defining the error criterion in (5.1).

Much of reinforcement learning is based on Markov processes with absorbing states, and using episodes that terminate in an absorbing state. It is obvious that a Markov process with an absorbing state is *not* irreducible. Specifically, there is no path *from* an absorbing state *to* any other state. Therefore Theorem 8.8 does not apply. Let us change notation and suppose that a Markov process has nonabsorbing states $\mathcal{S}$ and absorbing stages $\mathcal{A}$. Then it is easy to see that the state transition matrix A looks like

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & I_{|\mathcal{A}|} \end{bmatrix}. \tag{5.5}$$

For example, if there is only one absorbing state, the bottom right corner is just a single element of 1. In this case it can be shown that $\rho(A_{11}) < 1$ (see [46, Chapter 4]. Therefore there is a unique stationary distribution, with a 1 in the last component and zeros elsewhere. If $|\mathcal{A}| > 1$, then it is again true that $\rho(A_{11}) < 1$, but now there are infinitely many stationary distributions; see Problem 5.1. Even if it is assumed that there is only one absorbing state (for the policy under study), it is obvious that trying to use the stationary distribution to assign the weights μ_i would be meaningless, because the weights would all equal zero for nonabsorbing states. A reasonble approach is to set μ_i equal to the fraction of states X_t in a "typical" sample path that equal x_i, *before* the sample path hits an absorbing state. For that purpose, the following theorem is useful.

Theorem 5.1. *Suppose a reducible Markov process with absorbing states has a state transition matrix of the form* (5.5)*. Suppose an initial state $X_0 \in \mathcal{S}$ is chosen in accordance with a probability distribution $\boldsymbol{\phi}$. Then the fraction of states in $\mathcal{S}$ along all nonterminal paths (that is, paths until the time when they hit an absorbing state) has the distribution $\boldsymbol{\mu}$ given by*

$$\boldsymbol{\mu}^\top = \boldsymbol{\phi}^\top (I_{|\mathcal{S}|} - A_{11})^{-1}. \tag{5.6}$$

Proof to be supplied later.

This theorem still leaves open the question of how to choose the *initial* distribution $\boldsymbol{\phi}$.

Either way, once the weight vector $\boldsymbol{\mu}$ has been chosen, the next step is to minimize the error function E defined in (5.1). This is known as "value approximation." Before discussing how value approximation is achieved, let us digress to discuss whether this is the "right" problem to solve.

Once we have minimized the error in (5.1), we have an approximate value function $\hat{\mathbf{v}}_\pi(\boldsymbol{\theta})$ *for a fixed policy* π. However, the objective in MDPs is to choose an optimal or nearly optimal policy. Therefore, in order to use the approximate value vectors $\hat{\mathbf{v}}_\pi$ to guide this choice, we must construct such approximations for *all* policies π. Moreover, we must ensure that $\hat{\mathbf{v}}_\pi$ is a uniformly good approximation *over all possible policies*. In other words, the error in (5.1) needs to be minimized for all policies π. Even if we restrict to deterministic policies, the cardinality of Π_d is $m^n = |\mathcal{U}|^{|\mathcal{X}|}$, which can become very large very quickly. Even otherwise, just because $\hat{\mathbf{v}}_\pi$ is uniformly close to $\mathbf{v}_\pi$ for all π, it does not follow that the minimizer of $\hat{\mathbf{v}}_\pi$ over π is anywhere close to the minimizer of $\mathbf{v}_\pi$ with respect to π. Therefore we can ask: Is it possible to find directly an approximation the optimal policy using a suitable set of basis functions? This question lies at the heart of "policy gradient" methods and is studied in Section 5.3. The solution is significantly facilitated by a result known as the "policy gradient theorem." Proceeding further, we can attempt simultaneously to approximate *both* the policy function and the value function over a common set of basis functions. This is known as the "actor-critic" approach and is studied in Section 5.3.

5.1.2 Stochastic Gradient and Semi-Gradient Methods

As discussed above, in the objective function of (5.1), the policy π is fixed, and can thus be omitted. Similarly, once a sample path (a set of episodes or otherwise) is given, the coefficient vector $\boldsymbol{\mu}$ is fixed, whence it too can be omitted from the argument, and we can just write $E(\boldsymbol{\theta})$. Thus the objective is to minimize the function

$$E(\boldsymbol{\theta}) := \frac{1}{2}[\mathbf{v} - \hat{\mathbf{v}}(\boldsymbol{\theta})]^\top M[\mathbf{v} - \hat{\mathbf{v}}(\boldsymbol{\theta})]. \tag{5.7}$$

To compute the gradient of E with respect to $\boldsymbol{\theta}$, which is the variable of optimization, set $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$. Then

$$\hat{\mathbf{v}}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}) \approx \hat{\mathbf{v}}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}}\hat{\mathbf{v}}(\boldsymbol{\theta})\Delta\boldsymbol{\theta}.$$

Substituting this into the expression for E and retaining only first-order terms shows that

$$\nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}) = [\nabla_{\boldsymbol{\theta}}\hat{\mathbf{v}}(\boldsymbol{\theta})]^\top M[\nabla_{\boldsymbol{\theta}}\hat{\mathbf{v}}(\boldsymbol{\theta})\boldsymbol{\theta} - \mathbf{v}]. \tag{5.8}$$

Now let us discuss what form the function $\hat{\mathbf{v}}(\boldsymbol{\theta})$ may take. There are two natural categories, namely: linear and nonlinear. In the linear case, there is a set of linearly independent vectors $\psi_l \in \mathbb{R}^n$ for $l = 1, \ldots, d$ where d is the number of weights. (Recall that $\boldsymbol{\theta} \in \mathbb{R}^d$.) Define

$$\Psi := [\psi_1 | \cdots | \psi_d] \in \mathbb{R}^{n \times d}, \tag{5.9}$$

and suppose that

$$\hat{\mathbf{v}}(\boldsymbol{\theta}) = \Psi\boldsymbol{\theta} \in \mathbb{R}^n. \tag{5.10}$$

The assumption that the columns of Ψ are linearly independent means that Ψ has full column rank. It means also that there are no redundant weights in $\boldsymbol{\theta}$.

Let $\mathcal{V} \subseteq \mathbb{R}^n$ denote the span of the d vectors that comprise the columns of Ψ, that is, the range of Ψ in $\mathbb{R}^n$. Then it is clear that, for every choice of the weight vector $\boldsymbol{\theta}$, the vector $\hat{\mathbf{v}}(\boldsymbol{\theta})$ belongs to $\mathcal{V}$. Hence minimizing $E(\boldsymbol{\theta})$ is equivalent to finding to the closest element of the true value vector $\mathbf{v}$ in $\mathcal{V}$, where the distance between two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ is measured by

$$d(\mathbf{a}, \mathbf{b}) = [(\mathbf{a} - \mathbf{b})^\top M(\mathbf{a} - \mathbf{b})]^{1/2}.$$

In this case (5.4) becomes

$$\nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}) = \Psi^\top M\Psi\boldsymbol{\theta} - \Psi^\top M\mathbf{v}. \tag{5.11}$$

As pointed out earlier, the function $\hat{\mathbf{v}}$ can also be viewed as a *family* of functions indexed by x_i. Thus we can define $\hat{V}_{x_i} : \mathbb{R}^d \to \mathbb{R}$ by

$$\hat{V}(x_i, \boldsymbol{\theta}) = [\hat{\mathbf{v}}(\boldsymbol{\theta})]_{x_i},$$

that is, the x_i-th component of the map $\hat{\mathbf{v}} : \mathbb{R}^d \to \mathbb{R}$. As an illustration, consider the linear map $\hat{\mathbf{v}}$ defined in (5.10). Then the map $\hat{V}_{x_i}$ is defined by

$$\hat{V}(x_i, \boldsymbol{\theta}) = \Psi^{x_i}\boldsymbol{\theta},$$

where Ψ^l denotes the l-th row of the matrix Ψ. With this in mind, let us define the error

$$E(x_i, \boldsymbol{\theta}) = \frac{1}{2}\mu_i[V(x_i) - \hat{V}(x_i, \boldsymbol{\theta})]^2. \tag{5.12}$$

Now suppose, as we have been doing, that a sample path $\{(X_t, U_t, W_{t+1})\}_{t \geq 0}$ is given. We wish to find the weight vector $\boldsymbol{\theta}$ethat (nearly) minimizes the error criterion in (5.1). In what follows, we make a distinction between so-called gradient methods (or updates), and so-called semi-gradient methods (or updates).

We begin with gradient methods, which are based on Monte Carlo simulation, and thus require the Markov process to have one or more absorbing states. The sample path is also required to terminate in an absorbing state at some time T. As before, define the cumulated discounted return

$$G_{t+1} = \sum_{\tau=0}^{T-t-1} \gamma^\tau W_{\tau+t+1}, t \leq T.$$

The iterations are started off at index $\tau = 0$ with some initial weight vector $\boldsymbol{\theta}_0$. Then for each time τ in $0, \cdots, T - t - 1$, we adjust $\boldsymbol{\theta}_\tau$ as follows:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_{t+1}[G_{t+1} - \hat{V}(X_t, \boldsymbol{\theta}_t)]\nabla_{\boldsymbol{\theta}}\hat{V}(X_t, \boldsymbol{\theta}_t). \tag{5.13}$$

Here $\{\alpha_t\}$ is a predetermined sequence of time steps that can either be constant or decrease slowly to zero. However, if we are to interpret (5.13) as moving $\boldsymbol{\theta}_t$ in the direction of the gradient of the error function $E(X_t, \boldsymbol{\theta})$ defined in (5.12), then the presence of the coefficient μ_{X_t} is essential.

The semi-gradient updating method is reminiscent of the Temporal Difference method. We proceed as follows: Start at time $t = 0$ at any initial state X_0. Generate a sample path $\{(X_t, U_t, W_{t+1})\}$ by choosing $U_t \sim \pi(\cdot|X_t)$ in case the policy is probabilistic, and $U_t = \pi(X_t)$ if the policy is deterministic. Observe W_{t+1} and X_{t+1}. Repeat. Start with any initial vector $\boldsymbol{\theta}_0$. On the basis of this sample path, update the weight vector $\boldsymbol{\theta}_t$ at each time instant as follows:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_{t+1}[R_{t+1} + \gamma \hat{V}(X_{t+1}, \boldsymbol{\theta}_t) - \hat{V}(X_t, \boldsymbol{\theta}_t)]\nabla_{\boldsymbol{\theta}} \hat{V}(X_t, \boldsymbol{\theta}_t). \tag{5.14}$$

If the value approximation function $\hat{\mathbf{v}}(\boldsymbol{\theta})$ is linear as in (5.10), then it is easy to see that

$$\hat{V}(X_t, \boldsymbol{\theta}) = \Psi^{X_t} \boldsymbol{\theta}.$$

Therefore

$$\nabla_{\boldsymbol{\theta}} \hat{V}(X_t, \boldsymbol{\theta}) = (\Psi^{X_t})^\top,$$

where Ψ^{X_t} denotes the X_t-th row of the matrix Ψ. If we define

$$\mathbf{y}_t = (\Psi^{X_t})^\top, \tag{5.15}$$

that is, the transpose of row X_t of the matrix Ψ, then we can write

$$\hat{V}(X_t, \boldsymbol{\theta}) = \langle \mathbf{y}_t, \boldsymbol{\theta} \rangle = \mathbf{y}_t^\top \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} \hat{V}(X_t, \boldsymbol{\theta}) = \mathbf{y}_t.$$

Therefore the updating rule (5.14) becomes

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_{t+1}(W_{t+1} + \gamma \mathbf{y}_{t+1}^\top \boldsymbol{\theta}_t - \mathbf{y}_t^\top \boldsymbol{\theta}_t)\mathbf{y}_t. \tag{5.16}$$

If the approximation function $\hat{\mathbf{v}}(\boldsymbol{\theta})$ is a linear map of the form $\Psi\boldsymbol{\theta}$, we can establish the convergence of the semi-gradient method by viewing it as an implementation of stochastic approximation.

Theorem 5.2. *Suppose the value approximation function $\hat{\mathbf{v}}(\boldsymbol{\theta})$ is as in* (5.10)*. Suppose the state transition matrix A is irreducible, and that $\boldsymbol{\mu}$ is its unique stationary distribution. Define the error function $E(\boldsymbol{\theta})$ as in* (5.1)*, and use the update rule* (5.16)*. Finally, suppose the usual conditions on $\{\alpha_t\}$ hold, namely*

$$\sum_{t=1}^{\infty} \alpha_t = \infty, \sum_{t=1}^{\infty} \alpha_t^2 < \infty. \tag{5.17}$$

Then the sequence of estimates $\{\boldsymbol{\theta}_t\}$ converges almost surely to $\boldsymbol{\theta}^*$, which is the unique solution of the equation

$$C\boldsymbol{\theta}^* - \mathbf{b} = \mathbf{0}, \tag{5.18}$$

where

$$C = \Psi^\top M(\gamma A - I_n)\Psi, \mathbf{b} = E[R_{t+1}\mathbf{y}_t], M = \text{Diag}(\mu_{x_i}). \tag{5.19}$$

To prove this theorem, we state and prove a couple of preliminary results.

Definition 5.1. A matrix $F \in \mathbb{R}^{d \times d}$ is said to be **row diagonally dominant** if

$$f_{ii} > \sum_{j \neq i} |f_{ij}|, \tag{5.20}$$

which automatically implies that all diagonal elements of F are strictly positive. The matrix F is said to be **column diagonally dominant** if $F^\top$ is row diagonally dominant, or equivalently

$$f_{ii} > \sum_{j \neq i} |f_{ji}|. \tag{5.21}$$

Finally, the matrix F is said to be **diagonally dominant** if F is both row and column diagonally dominant.

Lemma 5.1. *Suppose F is diagonally dominant. Then so is $(F + F^\top)/2$, the symmetric part of F.*

The proof is omitted as it is obvious.

Lemma 5.2. *Suppose F is diagonally dominant. Then the symmetric matrix $F + F^\top$ is positive definite.*

Proof. The proof is a ready consequence of Lemma 5.1 and the Gerschgorin circle theorem. Recall that, for *any* matrix $G \in \mathbb{R}^{d\times d}$, the eigenvalues of G are contained in the union of the d Girschgorin circles

$$\mathcal{C}_i := \{z \in \mathbb{C} : |z - g_{ii}| \leq \sum_{j\neq i} |g_{ij}|\}, i = 1, \cdots, d.$$

Now let $G = F + F^\top$. Then G is symmetric and thus has only real eigenvalues. So the Girschgorin circles turn into the Girshgorin *intervals*. The diagonal dominance of G implies that each interval is a subset of $\mathbb{R}_+$. Therefore all eigenvalues of G are positive, and G is positive definite. □

Lemma 5.3. *(Lyapunov Matrix Equation) Suppose $C \in \mathbb{R}^{d\times d}$. Then all eigenvalues of C have negative real parts if there is a positive definite matrix P such that*

$$-(C^\top P + PC) =: Q$$

is positive definite.

This is a standard result in linear control theory and the proof can be found in many places, for example [44, Theorem 5.4.42].

At last we come to the proof of the main theorem

Proof. (Of Theorem 5.2.) Note that $\mathbf{y}_{t+1}^\top$ is $\Psi^{X_{t+1}}$, which is row X_{t+1} of the matrix Ψ. Thus the the conditional expectation

$$E[\mathbf{y}_{t+1}^\top | \mathbf{y}_t] = \mathbf{y}_t^\top A,$$

where A is the state transition matrix. Therefore

$$E[(W_{t+1} + \gamma \mathbf{y}_{t+1}^\top \boldsymbol{\theta}_t - \mathbf{y}_t^\top \boldsymbol{\theta}_t)\mathbf{y}_t | \mathbf{y}_t] = \mathbf{b} - C\boldsymbol{\theta}.$$

Now the update rule (5.16) is just stochastic approximation used to solve the *linear* equation

$$C\boldsymbol{\theta}^* = \mathbf{b}.$$

If all eigenvalues of the matrix C have negative real parts, then the differential equation

$$\dot{\boldsymbol{\theta}} = C\boldsymbol{\theta} - b$$

is globally asymptotically stable around its unique equilibrium $\boldsymbol{\theta}^*$. Thus the proof is completed once it is established that all eigenvalues of C have negative real parts.

From Lemma 5.3, a sufficient condition for this is that $-(C + C^\top)$ is positive definite. Now note that

$$-(C + C^\top) = \Psi^\top [H + H^\top]\Psi,$$

where

$$H = M(I_n - \gamma A). \tag{5.22}$$

It is now shown that H is both row and column diagonally dominant. Observe first that H has $1 - \gamma a_{ii}$ on the diagonal (which are all positive because $a_{ii} \leq 1$ and $\gamma < 1$), and has $-\gamma a_{ij}$ on the off-diagonal elements. Therefore, to establish the diagonal dominance of H, it is enough to show that

$$H\mathbf{1}_n > \mathbf{0}, \mathbf{1}_n^\top H > \mathbf{0}.$$

Let us begin with the first inequality. We have that

$$H\mathbf{1}_n = M(I_n - \gamma A)\mathbf{1}_n = M(1-\gamma)\mathbf{1}_n > \mathbf{0}.$$

Here we make use of the fact that $A\mathbf{1}_n = \mathbf{1}_n$ due to the row-stochasticity of A. For the second inequality, note that

$$\mathbf{1}_n^\top M = \boldsymbol{\mu}^\top,$$

where $\boldsymbol{\mu}$ is the stationary distribution and thus satisfies $\boldsymbol{\mu}^\top A = \boldsymbol{\mu}^\top$. Therefore

$$\mathbf{1}_n^\top H = \mathbf{1}_n^\top M(I - \gamma A) = \boldsymbol{\mu}^\top(I_n - \gamma A) = \boldsymbol{\mu}^\top(1-\gamma) > \mathbf{0}.$$

Now, by Lemma 5.2, the diagonal dominance of H implies that $H + H^\top$ is positive definite. The fact that Ψ has full row rank of d implies that

$$-(C + C^\top) = \Psi^\top[H + H^\top]\Psi$$

is also positive definite. Finally, it follows from Lemma 5.3 that all eigenvalues of C have negative real parts. □

Problem 5.1. Consider a Markov process with two or more absorbing states, so that its state transition matrix looks like

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & I_s \end{bmatrix},$$

where $s = |\mathcal{A}|$. Show that the set of *all* stationary distributions of this Markov process consists of

$$\begin{bmatrix} \mathbf{0}_{|\mathcal{S}|} \\ \boldsymbol{\phi} \end{bmatrix},$$

where

$$\phi_i \geq 0, i = 1, \cdots, s, \sum_{i=1}^{s} \phi_i = 1.$$

Problem 5.2. Give an example of a Markov process that is reducible but does not have any absorbing states.

Problem 5.3. Show that the proof of Theorem 5.2 remains valid if $\boldsymbol{\mu}$ is *sufficiently close* to the stationary distribution of A. State and prove a theorem to this effect.

5.2 Value Approximation via TD$^{(\lambda)}$-Methods

Until now we have studied the application of Monte Carlo methods for episodic Markov processes, and Temporal Difference (TD) methods for not necessarily episodic Markov processes. In this section we introduce a variant of TD-learning, known as TD$^{(\lambda)}$-learning. In this model, $\lambda \in [0, 1)$ is an adjustable parameter. The choice $\lambda = 0$ leads to conventional TD-learning.

In this section we examine two classes of value approximation problems: The case where the value is the discounted future reward (which is the case we have been studying until now), and the case where the value is the average of future rewards. While the two cases are broadly similar, there are a few complicating factors in the case of average reward processes.

5.2.1 Discounted Returns

The reference for the contents of this section is [42].

Suppose as before that we have a Markov reward process with state process $\{X_t\}$ over a finite state space $\mathcal{X}$, and reward process $\{R_{t+1}\}$, where the reward could be a random (but bounded) function of the state. Choose a discount factor $\gamma \in (0,1)$. Define

$$V(x_i) := E\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | X_0 = x_i\right], \ \forall x_i \in \mathcal{X}. \tag{5.23}$$

Note that $V : \mathcal{X} \to \mathbb{R}^n$ where $n = |\mathcal{X}|$. If n is too large, we approximate V by another function $\hat{V} : \mathcal{X} \times \mathbb{R}^d \to \mathbb{R}$. Therefore, for each parameter $\boldsymbol{\theta} \in \mathbb{R}^d$, the quantity $\hat{V}(x_i, \boldsymbol{\theta})$ is an approximation for the true value $V(x_i)$.

Suppose have a sample path $\{(X_t, W_{t+1})\}_{t \geq 0}$. Using this sample path we wish to estimate the value function V. For this purpose, define the temporal difference

$$\delta_{t+1} := W_{t+1} + \gamma \hat{V}(X_{t+1}, \boldsymbol{\theta}) - \hat{V}(X_t, \boldsymbol{\theta}).$$

This is the same as δ_{t+1} defined in (5.14). The conventional TD updating rule is

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_{t+1} \delta_{t+1} \nabla_{\boldsymbol{\theta}} \hat{V}(X_t, \boldsymbol{\theta}_t),$$

which is the same as (5.16). To define TD$^{(\lambda)}$-updating, choose a number $\lambda \in [0,1]$, and define

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_{t+1} \delta_{t+1} \left(\sum_{\tau=0}^{t} (\gamma\lambda)^{t-\tau} \nabla_{\boldsymbol{\theta}} \hat{V}(X_\tau, \boldsymbol{\theta}_\tau) \right). \tag{5.24}$$

If we choose $\lambda = 0$ and set $(\gamma\lambda)^0 = 1$ if $\lambda = 0$, then it is obvious that (5.24) becomes (5.16), the standard TD updating rule.

In principle the TD$^{(\lambda)}$-updating rule (5.24) can be applied with *any* function $\hat{V}$. However, in the remainder of this section, we focus on the case of *linear* approximation, where there is a matrix $\Psi \in \mathbb{R}^{n \times d}$, and $\hat{V}(x_i) = \Psi^{x_i} \boldsymbol{\theta}$, for each $x_i \in \mathcal{X}$. Assume that Ψ has full column rank, so that none of the components of $\boldsymbol{\theta}$ is redundant. Also, define as before the vector

$$y_\tau := [\Psi^{X_\tau}]^\top \in \mathbb{R}^d.$$

Then the **eligibility vector** $\mathbf{z}_t \in \mathbb{R}^d$ can be defined as

$$\mathbf{z}_t = \sum_{\tau=0}^{t} (\gamma\lambda)^{t-\tau} \mathbf{y}_\tau. \tag{5.25}$$

With this new notation, the TD$^{(\lambda)}$-updating rule (5.24) can be written as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_{t+1} \delta_{t+1} \mathbf{z}_t, \tag{5.26}$$

with $\mathbf{z}_0 = \mathbf{0}$. Note that if $\lambda = 0$ then $\mathbf{z}_t = \mathbf{y}_t$ and (5.26) becomes (5.16). Also note that $\mathbf{z}_t$ satisfies the recursion

$$\mathbf{z}_t = \gamma\lambda \mathbf{z}_{t-1} + \mathbf{y}_t. \tag{5.27}$$

In [42], the convergence properties of TD$^{(\lambda)}$-updating are studied. As one might expect, the updating rule is interpreted as the stochastic approximation algorithm applied to solve a linear equation of the form

$$C\boldsymbol{\theta}^* = \mathbf{b}.$$

If the ODE

$$\dot{\boldsymbol{\theta}} = C\boldsymbol{\theta} - \mathbf{b}$$

is globally asymptotically stable, then the stochastic approximation converges to a solution of the algebraic equation. However, the matrix C is more complicated than in Section 5.2. We give a brief description of the results and direct the reader to [42] for full details.

We begin with an alternative to Theorem 2.2. That theorem states that, whenever the discount factor $\gamma < 1$, the map $\mathbf{y} \mapsto T\mathbf{y} := \mathbf{r} + \gamma A\mathbf{y}$ is a contraction with respect to the ℓ_∞-norm, with contraction constant γ. The key to the proof is the fact that if A is row-stochastic, then the induced norm $\|A\|_{\infty\to\infty} = 1$. Now it is shown that a similar statement holds for an entirely different norm.

Suppose A is row-stochastic and irreducible, and let $\boldsymbol{\mu}$ denote its stationary distribution. Note that, by Theorem 8.6, $\boldsymbol{\mu}$ is uniquely defined and has all positive elements. Define $M = \mathrm{Diag}(\mu_i)$ and define a norm $\|\cdot\|_M$ on $\mathbb{R}^d$ by

$$\|\mathbf{v}\|_M = (\mathbf{v}^\top M\mathbf{v})^{1/2}. \tag{5.28}$$

Then the corresponding distance between two vectors $\mathbf{v}_1, \mathbf{v}_2$ is given by

$$\|\mathbf{v}_1 - \mathbf{v}_2\|_M = ((\mathbf{v}_1 - \mathbf{v}_2)^\top M(\mathbf{v}_1 - \mathbf{v}_2))^{1/2}.$$

Lemma 5.4. *Suppose $A \in [0,1]^{n\times n}$, is row-stochastic, and irreducible. Let $\boldsymbol{\mu}$ be the stationary distribution of A. Then*

$$\|A\mathbf{v}\|_M \le \|\mathbf{v}\|_M, \ \forall \mathbf{v} \in \mathbb{R}^n. \tag{5.29}$$

Consequently, the map $\mathbf{v} \mapsto \mathbf{r}\gamma A\mathbf{v}$ is a contraction with respect to $\|\cdot\|_M$.

In order to prove Lemma 5.4, we make use of a result known as "Jensen's inequality," of which only a very simple special case is presented here.

Lemma 5.5. *(Jensen's Inequality) Suppose Y is a real-valued random variable taking values in a finite set $\mathcal{Y} = \{y_1, \ldots, y_n\}$, and let $\mathbf{p}$ denote the probability distribution of $\mathcal{Y}$. Suppose $f : \mathbb{R} \to \mathbb{R}$ is a convex function. Then*

$$f(E[Y, \mathbf{p}]) \le E[f(Y), \mathbf{p}]. \tag{5.30}$$

It can be shown that Jensen's inequality holds for arbitrary real-valued random variables. But what is stated above is adequate for present purposes.

Proof. (Of Lemma 5.5.) Note that, because $f(\cdot)$ is convex, we have

$$f(E[Y, \mathbf{p}]) = f\left(\sum_{i=1}^n p_i y_i\right) \le \sum_{i=1}^n p_i f(y_i) = E[f(Y), \mathbf{p}].$$

This is the desired bound. □

Proof. (Of Lemma 5.4.) We will show that

$$\|A\mathbf{v}\|_M^2 \le \|\mathbf{v}\|_M^2, \ \forall \mathbf{v} \in \mathbb{R}^n,$$

which is clearly equivalent to (5.29). Now

$$\|A\mathbf{v}\|_M^2 = \sum_{i=1}^n \mu_i (A\mathbf{v})_i^2 = \sum_{i=1}^n \mu_i \left(\sum_{j=1}^n A_{ij} v_j\right)^2.$$

However, for each fixed index i, the row A^i is a probability distribution, and the function $f(Y) = Y^2$ is convex. If we apply Jensen's inequality with $f(Y) = Y^2$, we see that

$$\left(\sum_{j=1}^{n} A_{ij} v_j\right)^2 \leq \sum_{j=1}^{n} A_{ij} v_j^2, \ \forall i.$$

Therefore

$$\|A\mathbf{v}\|_M^2 \leq \sum_{i=1}^{n} \mu_i \left(\sum_{j=1}^{n} A_{ij} v_j^2\right) = \sum_{j=1}^{n} \left(\sum_{i=1}^{n} \mu_i A_{ij}\right) v_j^2 = \sum_{j=1}^{n} \mu_j v_j^2 = \|\mathbf{v}\|_M^2,$$

where in the last step we use the fact that $\boldsymbol{\mu} A = \boldsymbol{\mu}$. □

Next, to analyze the behavior of the TD$^{(\lambda)}$-updating, we define a map $T^{(\lambda)} : \mathbb{R}^n \to \mathbb{R}^n$ via

$$[T^{(\lambda)}\mathbf{f}]_i := (1-\lambda) \sum_{l=0}^{\infty} \lambda^l E\left[\sum_{\tau=0}^{l} \gamma^\tau R(X_{\tau+1}) + \gamma^{l+1} f_{X_{l+1}} | X_0 = x_i\right]. \tag{5.31}$$

Note that $T^{(\lambda)}\mathbf{f}$ can be written explicitly as

$$T^{(\lambda)}\mathbf{f} = (1-\lambda) \sum_{l=0}^{\infty} \lambda^l \left[\sum_{\tau=0}^{l} \gamma^\tau A^\tau \mathbf{r} + \gamma^{l+1} A^{l+1} \mathbf{f}\right], \tag{5.32}$$

where, as before

$$\mathbf{r} = [\ R(x_1) \quad \cdots \quad R(x_n)\ ]^\top,$$

and

$$R(x_j) = E[R_1 | X_0 = x_i].$$

Lemma 5.6. *The map $T^{(\lambda)}$ is a contraction with respect to $\|\cdot\|_M$, with contraction constant $[\gamma(1-\lambda)]/(1-\gamma\lambda)$.*

Proof. Note that the first term on the right side of (5.32) does not depend on $\mathbf{f}$. Therefore

$$T^{(\lambda)}(\mathbf{f}_1 - \mathbf{f}_2) = \gamma(1-\lambda) \sum_{l=0}^{\infty} (\gamma\lambda)^l A^{l+1} (\mathbf{f}_1 - \mathbf{f}_2).$$

However, it is already known from Lemma 5.4 that

$$\|A(\mathbf{f}_1 - \mathbf{f}_2)\|_M \leq \|\mathbf{f}_1 - \mathbf{f}_2\|_M.$$

By repeatedly applying the above, it follows that

$$\|A^l(\mathbf{f}_1 - \mathbf{f}_2)\|_M \leq \|\mathbf{f}_1 - \mathbf{f}_2\|_M, \ \forall l.$$

Therefore

$$\|T^{(\lambda)}(\mathbf{f}_1 - \mathbf{f}_2)\|_M \leq \gamma(1-\lambda) \sum_{l=0}^{\infty} (\gamma\lambda)^l \|\mathbf{f}_1 - \mathbf{f}_2\|_M = \frac{\gamma(1-\lambda)}{1-\gamma\lambda} \|\mathbf{f}_1 - \mathbf{f}_2\|_M.$$

This is the desired bound. □

Let us define $\mathbf{v} \in \mathbb{R}^n$ as the unique solution of

$$T^{(\lambda)}\mathbf{v} = \mathbf{v}.$$

Then it is easy to verify that $\mathbf{v}$ is in fact the value function, because the value function also satisfies the above equation, and the equation has a unique solution because $T^{(\lambda)}$ is a contraction. Presumably we could find $\mathbf{v}$ by running a stochastic approximation type of iteration, *without* value approximation. However, our interest is in what happens *with* value approximation. Towards this end, define a projection $\Pi : \mathbb{R}^n \to \mathbb{R}^n$ by

$$\Pi\mathbf{a} := \Psi(\Psi^\top M\Psi)^{-1}\Psi^\top M\mathbf{a}. \tag{5.33}$$

Then

$$\Pi\mathbf{a} = \underset{\mathbf{b}\in\Psi(\mathbb{R}^d)}{\arg\min} \|\mathbf{a}-\mathbf{b}\|_M. \tag{5.34}$$

Thus $\Pi\mathbf{a}$ is the closest point to $\mathbf{a}$ in the subspace $\Psi(\mathbb{R}^n)$. With this definition, we can state the following:

Theorem 5.3. *Suppose the sequence $\{\boldsymbol{\theta}_t\}$ is defined by (5.24). Suppose further that the standard conditions*

$$\sum_{t=0}^{\infty}\alpha_t = \infty, \sum_{t=0}^{\infty}\alpha_t^2 < \infty \tag{5.35}$$

hold. Then the sequence $\{\boldsymbol{\theta}_t\}$ converges almost surely to $\boldsymbol{\theta}^*$, where $\boldsymbol{\theta}^*$ is the unique solution of

$$\Pi T^{(\lambda)}(\Psi\boldsymbol{\theta}^*) = \Psi\boldsymbol{\theta}^*. \tag{5.36}$$

Moreover

$$\|\Psi\boldsymbol{\theta}^* - \mathbf{v}\|_M \le \frac{1-\gamma\lambda}{1-\gamma}\|\Pi\mathbf{v}-\mathbf{v}\|_M. \tag{5.37}$$

Let us understand what the above bound states. Because we are approximating the true value vector $\mathbf{v}$ by a vector of the form $\Psi\boldsymbol{\theta}$, the best possible approximator in terms of the distance $\|\cdot\|_M$ is given by $\Pi\mathbf{v}$, while $\|\Pi\mathbf{v}-\mathbf{v}\|_M$ is the smallest possible approximation error. Now (5.37) states that the limit of TD$^{(\lambda)}$ iterations satisfies an error bound that is larger than the lowest possible error by an "expansion"factor of $(1-\gamma)/(1-\gamma\lambda)$. In order to make this expansion factor smaller, we should choose λ closer to one. However, the closer λ is to one, the more slowly the infinite series in (5.32) will converge. These tradeoffs are still not well-understood.

The proof as usual consists of constructing a matrix C whose eigenvalues all have negative real parts, and showing that $\boldsymbol{\theta}^*$ converges to the solution of $C\boldsymbol{\theta}^* = \mathbf{b}$. The details can be found in the paper.

But let us discuss what the bound (5.37) means. Note that, since $\Psi\boldsymbol{\theta} \in \Psi(\mathbb{R}^d)$, the quantity $\|\Pi\mathbf{v}^*-\mathbf{v}^*\|_M$ is the best that we could hope to achieve. The distance between the limit $\Psi\boldsymbol{\theta}^*$ and the true value vector $\mathbf{v}^*$ is bounded by a factor $(1-\gamma\lambda)/(1-\gamma$ times this minimum. Note that $(1-\gamma\lambda)/(1-\gamma > 1$. So this is the extent to which the TD$^{(\lambda)}$ iterations miss the optimal approximation.

5.2.2 Average Returns

Until now we have studied discounted Markov Decision Processes. But we can also study *average reward* processes. The key reference for this subsection is [43].

Suppose $\{X_t\}$ is a Markov process on a finite set $\mathcal{X}$, with state transition matrix A (which is unknown). Suppose further that A is irreducible and aperiodic, and let $\boldsymbol{\mu}$ denote the stationary distribution of A. Suppose $R : \mathcal{X} \to \mathbb{R}$ is the reward function. Define

$$c^* := \lim_{T\to\infty}\frac{1}{T+1}\sum_{t=0}^{T}R(X_t). \tag{5.38}$$

Then by the assumptions on A, it follows that

$$c^* = \sum_{x_i \in \mathcal{X}} \mu_i R(x_i) = E[R, \boldsymbol{\mu}]. \tag{5.39}$$

Note that c^* is a constant that is independent of the initial state. Now define

$$J(x_i) := \lim_{T \to \infty} E\left[\frac{1}{T+1} \sum_{t=0}^{T} R(X_t) | X_0 = x_i\right] - c^* \tag{5.40}$$

to be the average reward *starting in state* x_i, minus the overall average reward c^*. So we can think of $J(x_i)$ as the relative advantage or disadvantage of starting in state x_i. Define $\mathbf{J}$ as the vector $J(x_i)$ as x_i varies over $\mathcal{X}$. Then $\mathbf{J}$ satisfies the so-called **Poisson equation**

$$\mathbf{J} = \mathbf{r} - c^* \mathbf{1}_n + A\mathbf{J}. \tag{5.41}$$

The vector $\mathbf{J}$ is called the **differential reward**. Note that if we replace $\mathbf{J}$ by $\mathbf{J} + \alpha \mathbf{1}_n$ for some constant α, then because $A\mathbf{1}_n = \mathbf{1}_n$, the vector $\mathbf{J} + \alpha \mathbf{1}_n$ also satisfies (5.41). Therefore there is a unique vector $\mathbf{J}^* \in \mathbb{R}^n$ that satisfies (5.41) and also $\boldsymbol{\mu}^\top \mathbf{J}^* = 0$. Then the set of *all* solutions to (5.41) is given by $\{\mathbf{J}^* + \alpha \mathbf{1}_n : \alpha \in \mathbb{R}\}$. Further, $\mathbf{J}^*$ satisfies

$$\mathbf{J}^* = \sum_{t=0}^{\infty} A^t (\mathbf{r} - c^* \mathbf{1}_n). \tag{5.42}$$

Now the problem studied in this subsection is how to approximate $\mathbf{J}^*$. Choose a matrix $\Psi \in \mathbb{R}^{n \times d}$, and approximate $\mathbf{J}^*$ by $\Psi \boldsymbol{\theta}$. A key assumption is that $\mathbf{1}_n \notin \Psi(\mathbb{R}^d)$. In other words, the vector of all ones *does not belong* to the range of Ψ.

Suppose we have a sample path $\{(X_t, W_{t+1})\}_{t \geq 0}$. As before, for $X_t \in \mathcal{X}$, define

$$\mathbf{y}_t = [\Psi^{X_t}]^\top \in \mathbb{R}^d. \tag{5.43}$$

In this case the approximating function is defined by

$$\hat{J}(X_t, \boldsymbol{\theta}) := \langle \mathbf{y}_t, \boldsymbol{\theta} \rangle = \mathbf{y}_t^\top \boldsymbol{\theta} = \boldsymbol{\theta}^\top \mathbf{y}_t. \tag{5.44}$$

There is also another sequence $\{c_t\}$ in $\mathbb{R}$ that tries to approximate the constant c^*. At time t, define the temporal difference

$$\delta_{t+1} := W_{t+1} - c_t + \hat{J}(X_{t+1}, \boldsymbol{\theta}_t) - \hat{J}(X_t, \boldsymbol{\theta}_t). \tag{5.45}$$

Next, choose a constant $\lambda \in [0, 1)$ and use a TD$^{(\lambda)}$ update

$$c_{t+1} = c_t + \eta_{t+1}(W_{t+1} - c_t), \tag{5.46}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_{t+1} \delta_{t+1} \left(\sum_{\tau=0}^{t} \lambda^{t-\tau} y_\tau \right), \tag{5.47}$$

where η_t, α_t are step sizes. We will choose these step sizes in tandem, so that $\eta_t = \beta \alpha_t$ for each time t; we can also just choose $\beta = 1$ so that $\eta_t = \alpha_t$. Also, $\mathbf{y}_\tau$ is defined as per (5.43). Define the **eligibility vector**

$$\mathbf{z}_t := \sum_{\tau=0}^{t} \lambda^{t-\tau} y_\tau \in \mathbb{R}^d, \tag{5.48}$$

and observe that $\mathbf{x}_t$ satisfies the recursion

$$\mathbf{z}_t = \lambda \mathbf{z}_{t-1} + \mathbf{y}_t. \tag{5.49}$$

Next, define the projection matrix

$$\Pi := \Psi(\Psi^\top M\Psi)^{-1}\Psi^\top M, \tag{5.50}$$

where $M = \mathrm{Diag}(\mu_i)$. Then, as we have seen before, for every $\mathbf{a} \in \mathbb{R}^n$, we have that

$$\Pi\mathbf{a} = \arg\min_{\mathbf{b}\in\Psi(\mathbb{R}^d)} \|\mathbf{a}-\mathbf{b}\|_M,$$

where $\|\cdot\|_M$ is the weighted Euclidean distance as before. For each $\lambda \in [0,1)$, define $T^{(\lambda)} : \mathbb{R}^n \to \mathbb{R}^n$ by

$$T^{(\lambda)}\mathbf{v} := (1-\lambda)\sum_{l=0}^{\infty}\lambda^l\left(\sum_{\tau=0}^{l}A^\tau(\mathbf{r}-c^*\mathbf{1}_n)+A^{l+1}\mathbf{v}\right). \tag{5.51}$$

Just as before, the map $T^{(\lambda)}$ is a contraction with respect to $\|\cdot\|_M$, with constant $1-\lambda$. Note that, for each finite l, we have

$$\sum_{\tau=0}^{l}A^\tau(\mathbf{r}-c^*\mathbf{1}_n)+A^{l+1}\mathbf{J}^* = \mathbf{J}^*. \tag{5.52}$$

With these observations, we can state the following theorem:

Theorem 5.4. *Suppose $\eta_t = \beta\alpha_t$ for some fixed constant β. Suppose further that*

$$\sum_{t=0}^{\infty}\alpha_t = \infty, \sum_{t=0}^{\infty}\alpha_t^2 < \infty \tag{5.53}$$

Then, as $t \to \infty$, almost surely we have that (i) $c_t \to c^*$, and (ii) $\boldsymbol{\theta}_t \to \boldsymbol{\theta}^*$, where $\boldsymbol{\theta}^*$ is the unique solution of

$$\Pi T^{(\lambda)}(\Psi\boldsymbol{\theta}^*) = \Psi\boldsymbol{\theta}^*. \tag{5.54}$$

Define

$$\boldsymbol{\phi} = \begin{bmatrix} c_t \\ \boldsymbol{\theta}_t \end{bmatrix} \in \mathbb{R}^{d+1}. \tag{5.55}$$

The idea behind the proof is the standard one, namely to use stochastic approximation to solve a linear equation of the form

$$C\boldsymbol{\phi} = \mathbf{b}.$$

However, there are a few wrinkles. The matrix C turns out to be

$$C = \begin{bmatrix} -\beta & \mathbf{0} \\ -\frac{1}{1-\lambda}\Psi^\top M\mathbf{1}_n & \Psi^\top M(A^{(\lambda)}-I_n)\Psi \end{bmatrix}, \tag{5.56}$$

where

$$A^{(\lambda)} := (1-\lambda)\sum_{l=0}^{\infty}\lambda^l A^{l+1}. \tag{5.57}$$

Now note that, since $\lambda < 1$, we have that $\rho(\lambda A) = \lambda < 1$. Therefore

$$\sum_{l=0}^{\infty}\lambda^l A^l = (I_n - \lambda A)^{-1}.$$

Therefore

$$A^{(\lambda)} = (1-\lambda)(I_n-\lambda A)^{-1}A. \tag{5.58}$$

It is easy to verify that $A^{(\lambda)}\mathbf{1}_n = \mathbf{1}_n$. Therefore, without additional assumptions, the matrix C is *not* asymptotically stable in the sense of having its eigenvalues in the open left half-plane. However, $A^{(\lambda)}\mathbf{v} \neq \mathbf{0}$ if $\mathbf{v}$ is not a multiple of $\mathbf{1}_n$. Now the assumption that $\mathbf{1}_n \notin \Psi(\mathbb{R}^d)$ guarantees that successive approximations $\Psi\boldsymbol{\theta}_t$ are not multiples of $\mathbf{1}_n$. In the complement of the one-dimensional subspace generated by $\mathbf{1}_n$, the matrix C is asymptotically stable.

5.3 Policy Gradient and Actor-Critic Methods

The contents of Sections 5.1 and 5.2 are aimed at *value approximation*. Thus, given a policy π, the objective is to compute an approximation $\hat{\mathbf{v}}_\pi$ for the true value function $\mathbf{v}_\pi$ corresponding to this policy. Since the true value vector $\mathbf{v}_\pi \in \mathbb{R}^{|\mathcal{X}|}$, if the size of the state space $|\mathcal{X}| =: n$ is too large, we choose $\hat{\mathbf{v}}_\pi = \hat{\mathbf{v}}_\pi(\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ and $d \ll n$. However, ultimately our objective is to choose good *policies*. So for this purpose we can choose some suitable objective function, and directly choose π so as to optimize that objective function. The question then arises as to what this objective function should be. For technical reasons, in policy optimization, one studies the *average* reward, as opposed to the *discounted cumulative* reward that has been the main focus until now. Recall that we briefly introduced average reward Markov processes in Section 5.2. We shall build on that in the present section to go beyong just *reward* processes to *decision* processes.

5.3.1 Preliminaries

The set-up is the standard Markov Decision Process (MDP) with a finite state space $\mathcal{X}$ of cardinality n, a finite action space $\mathcal{U}$ of cardinality m, and state transition matrices A^{u_k} for each $u_k \in \mathcal{U}$. Until now, much of the focus has been on *deterministic* policies $\pi : \mathcal{X} \to \mathcal{U}$, where the current state X_t determines the current action U_t uniquely as $\pi(X_t)$. However, policy approximation methods that have been widely studied do not work very well with deterministic policies. Instead, the focus is on *probabilistic* policies. Given the action set $\mathcal{U}$, let $\mathbb{S}(\mathcal{U})$ denote the set of probability distributions on $\mathcal{U}$. Since $\mathcal{U}$ is a finite set, $\mathbb{S}(\mathcal{U})$ consists of m-dimensional nonnegative vectors whose components add up to one, that is, the simplex in $\mathbb{R}^m_+$. A probabilistic policy $\pi : \mathcal{X} \to \mathbb{S}(\mathcal{U})$ is a map from the current state X_t to a corresponding probability distribution on $\mathcal{U}$. Until now there is no difference with earlier notation. Now we introduce a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$, and denote the policy as $\pi(x_i, u_k, \boldsymbol{\theta})$. Thus

$$\pi(x_i, u_k, \boldsymbol{\theta}) = \Pr\{U_t = u_k | X_t = x_i, \boldsymbol{\theta}\}. \tag{5.59}$$

In order for policy approximation theory to work, it is usually assumed that some or all of the following assumptions hold:

P1. We have that

$$\pi(x_i, u_k, \boldsymbol{\theta}) > 0, \ \forall x_i \in \mathcal{X}, u_k \in \mathcal{U}, \boldsymbol{\theta} \in \mathbb{R}^d. \tag{5.60}$$

Note that this assumption rules out deterministic policies.

P2. The quantity $\pi(x_i, u_k, \boldsymbol{\theta})$ is twice continuously differentiable with respect to $\boldsymbol{\theta}$. Moreover, there exists a finite constant M such that

$$\left| \frac{\partial \ln \pi(x_i, u_k, \boldsymbol{\theta})}{\partial \theta_l} \right| \leq M, \ \forall x_i \in \mathcal{X}, u_k \in \mathcal{U}, \boldsymbol{\theta} \in \mathbb{R}^d. \tag{5.61}$$

Note that a ready consequence of (5.61) is that

$$\left| \frac{\partial \pi(x_i, u_k, \boldsymbol{\theta})}{\partial \theta_l} \right| \leq M, \ \forall x_i \in \mathcal{X}, u_k \in \mathcal{U}, \boldsymbol{\theta} \in \mathbb{R}^d. \tag{5.62}$$

See Problem 5.4.

In policy approximation, the objective is to identify an optimal choice of $\boldsymbol{\theta}$ that would maximize a chosen objective function. As stated above, this objective function is usually the average reward under the policy. In principle, the value approximation methods studied earlier in this chapter could be used for this purpose, as follows: For each possible policy π, compute an approximate value $\hat{\mathbf{v}}_\pi$. These approximations are a function of an auxiliary parameter vector $\boldsymbol{\theta}$ which is suppressed in the interests of clarity. Ensure that these approximations are *uniformly close* in the sense that

$$\|\mathbf{v}_\pi - \hat{\mathbf{v}}_\pi\|_\infty \leq \epsilon, \ \forall \pi \in \Pi, \tag{5.63}$$

where Π is the class of policies under study. Note that (5.63) is equivalent to

$$|V_\pi(x_i) - \hat{V}_\pi(x_i)| \leq \epsilon, \ \forall x_i \in \mathcal{X}, \pi \in \Pi.$$

Now fix a state $\mathbf{x}_i \in \mathcal{X}$, and choose a policy $\hat{\pi}_i^* \in \Pi$ such that

$$\hat{\pi}_i^* = \arg\min_{\pi \in \Pi} \hat{V}_\pi(x_i). \tag{5.64}$$

Then $\hat{\pi}_i^*$ is an approximately optimal policy when starting from the state x_i. Define π_i^* to be the "true" optimal policy, that is

$$\pi_i^* = \arg\min_{\pi \in \Pi} V(x_i). \tag{5.65}$$

Then it is a ready consequence of (5.63) that

$$|\hat{V}_{\hat{\pi}_i^*}(x_i) - V_{\pi_i^*}(x_i)| \leq \epsilon. \tag{5.66}$$

In other words, the optimum of the approximate value function (for a given starting state) is within ϵ of the true optimal value starting from x_i. The proof is left as an exercise; see Problem 5.5. Thus it is possible, at least in principle, to compute a good approximation to the *optimal value*. However, there is no reason to assume that the policy $\hat{\pi}_i^*$ is in any way "close" to the true optimal policy π_i^*.

To address this issue, in policy optimization one directly parametrizes the policy π as $\pi(x_i, u_k, \boldsymbol{\theta})$, and then chooses $\boldsymbol{\theta}$ so as to optimize an appropriate objective function. As mentioned above, the preferred choice is the average reward. In order to carry out this process, it is highly desirable to have an expression for the *gradient* of this objective function with respect to the parameter vector $\boldsymbol{\theta}$. This is provided by the "policy gradient theorem" given below. However, there are other considerations that need to be taken into account.

Numerical examples have shown that if a nearly optimal policy is chosen on the basis of value approximation, the resulting set of policies may not converge; see for example [41]. To quote from the article:

> This analysis is particularly interesting, since the algorithm is closely related to Q-learning (Watkins and Dayan, 1992) and temporal-difference learning ($TD(\lambda)$) (Sutton, 1988), with λ set to 0. The counter-example discussed demonstrates the short-comings of some (but not all) variants of Q-learning and temporal-difference learning that are employed in practice.

This means the following: For a fixed tolerance ϵ, choose an approximately optimal policy $\hat{\pi}_i^*(\epsilon)$ as in (5.64). As the tolerance ϵ is reduced towards zero, we will get a sequence of nearly optimal policies, one for each ϵ. The question is: As $\epsilon \to 0^+$, does the corresponding policy $\hat{\pi}_i^*(\epsilon)$ approach the true optimal policy π_i^*? In general, the answer is no. So this would seem to rule out value approximation as a way of determining nearly optimal *policies*, though it does allow us to determine nearly optimal *values*. On the other side, choosing nearly optimal policies using policy parametrization leads to slow convergence and high variance. A class of algorithms known as "actor-critic" consist of simultaneously carrying out policy optimization (approximately) coupled with value approximation. Numerical experiments show that actor-critic algorithms lead to faster convergence and lower variance than pure policy optimization alone. It must be emphasized that "actor-critic" refers to a *class* of algorithms (or a philosophy), and not one specific algorithm. Thus, within the umbrella actor-critic, there are multiple variants currently in use.

As stated above, actor-critic algorithms update the policy and the value in parallel. Here "actor" refers to policy updating, while "critic" refers to value updating. Some assumptions about actor-critic algorithms are often understated or even unstated, so we mention them now.

- The behavior of actor-critic algorithms has been analyzed for the most part only when the objective function to be maximized is the average reward.
- In order to *prove* the convergence of actor-critic algorithms, it is usually assumed that the actor (policy) is updated more slowly than the critic (value).

5.3.2 Policy Gradient Formula

In this section we present an important theorem on evaluating the gradient of a cost function with respect to the parameter that determines a policy. The formula for the gradient is not entirely "expicit," but has the ingredients that permit one to "learn" this gradient and to use it in order to maximize an objective function.

We begin with the problem set-up. As before, we have a finite state space $\mathcal{X}$, a finite action space $\mathcal{U}$, and a family of probabilistic policies $\pi(\boldsymbol{\theta})$, where each $\pi(\boldsymbol{\theta}) : \mathcal{X} \to \mathbb{S}(\mathcal{U})$, where $\mathbb{S}(\mathcal{U})$ is the set of probability distributions on $\mathcal{U}$. Thus, as stated in (5.59), we define

$$\pi(x_i, u_k, \boldsymbol{\theta}) = \Pr\{U_t = u_k | X_t = x_i, \boldsymbol{\theta}\}.$$

Assume that conditions (P1) and (P2) hold regarding the differentiability of the policy with respect to the parameter $\boldsymbol{\theta}$. There is also a reward function $R : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$. Under the policy $\pi(\boldsymbol{\theta})$, the state transition matrix of the corresponding Markov process is given by

$$\Pr\{X_{t+1} = x_j | X_t = x_i, \pi(\boldsymbol{\theta})\} = [A^{\pi(\boldsymbol{\theta})}]_{ij} =: [A^{\boldsymbol{\theta}}]_{ij} = \sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) a_{ij}^{u_k}.$$

In what follows, to simplify notation, we denote various quantities with the subscript or superscript $\boldsymbol{\theta}$, instead of $\pi(\boldsymbol{\theta})$. To illustrate, above we have used $A^{\boldsymbol{\theta}}$ to denote $A^{\pi(\boldsymbol{\theta})}$.

Assume that $A^{\boldsymbol{\theta}}$ is irreducible and aperiodic for each $\boldsymbol{\theta}$, and let $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ denote the corresponding stationary distribution. Therefore $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ satisfies

$$\sum_{x_i \in \mathcal{X}} \mu_{\boldsymbol{\theta}}(x_i) \left[\sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) a_{ij}^{u_k} \right] = \mu_{\boldsymbol{\theta}}(x_j) \tag{5.67}$$

The above equation illustrates another notational convention. For a vector such as $\boldsymbol{\mu}_{\boldsymbol{\theta}}$, we use bold-faced letters, while for a component of the vector such as $\mu_{\boldsymbol{\theta}}(x_i)$, we don't use bold-faced letters.

Associated with each policy π is an average reward c^*, defined in analogy with (5.38) and (5.39), namely

$$c^*(\boldsymbol{\theta}) := \lim_{T \to \infty} \frac{1}{T+1} \sum_{t=0}^{T} R_{\boldsymbol{\theta}}(X_t) = E[R_{\boldsymbol{\theta}}, \boldsymbol{\mu}_{\boldsymbol{\theta}}]. \tag{5.68}$$

Because the policy π is probabilistic, we have from (2.27) that[1]

$$R_{\boldsymbol{\theta}}(x_i) = \sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) R(x_i, u_k). \tag{5.69}$$

Therefore an equivalent characterization of $c^*(\boldsymbol{\theta})$ is

$$c^*(\boldsymbol{\theta}) = \sum_{x_i \in \mathcal{X}} \mu_{\boldsymbol{\theta}}(x_i) \sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) R(x_i, u_k). \tag{5.70}$$

In addition to the constant c^* which depends only on $\boldsymbol{\theta}$ and nothing else, we also define the function $Q_{\boldsymbol{\theta}} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ by

$$Q_{\boldsymbol{\theta}}(x_i, u_k) = \sum_{t=1}^{\infty} E[R_{\boldsymbol{\theta}}(X_t) - c^*(\boldsymbol{\theta}) | X_0 = x_i, U_0 = u_k, \pi(\boldsymbol{\theta})]. \tag{5.71}$$

Note that, because $C^*(\boldsymbol{\theta})$ is the "weighted average" of the various rewards, the summation in (5.71) is well-defined even though there is no discount factor. Also, $Q_{\boldsymbol{\theta}}(x_i, u_k)$ satisfies the recursion

$$Q_{\boldsymbol{\theta}}(x_i, u_k) = R(x_i, u_k) - c^*(\boldsymbol{\theta}) + \sum_{u_k \in \mathcal{U}} a_{ij}^{u_k} V_{\boldsymbol{\theta}}(x_j), \tag{5.72}$$

[1] As per our convention, if $X_t = x_i$, the reward $R(x_i)$ is paid at time $t+1$.

where the value $V_{\boldsymbol{\theta}}(x_i)$ is defined as

$$V_{\boldsymbol{\theta}}(x_i) = \sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) Q_{\boldsymbol{\theta}}(x_i, u_k). \tag{5.73}$$

Now the objective of policy approximation is to choose the parameter $\boldsymbol{\theta}$ so as to maximize $c^*(\boldsymbol{\theta})$. For this purpose, it is highly desirable to have an expression for the gradient $\nabla_{\boldsymbol{\theta}} c^*(\boldsymbol{\theta})$. This is precisely what the policy gradient theorem gives us. The expression for the gradient can be combined with various methods to look for a minimizing choice of $\boldsymbol{\theta}$.

With all this notation, the policy gradient theorem can be stated very simply. The source for this proof is [34]. Note that in that paper, the policy gradient theorem is proved not only for the average reward but for another type of reward as well. We do not present that result here and the interested reader can consult the original paper.

Theorem 5.5. *(Policy Gradient Theorem) We have that*

$$\nabla_{\boldsymbol{\theta}} c^*(\boldsymbol{\theta}) = \sum_{x_i \in \mathcal{X}} \mu_{\boldsymbol{\theta}}(x_i) \sum_{u_k \in \mathcal{U}} \nabla_{\boldsymbol{\theta}} \pi(x_i, u_k, \boldsymbol{\theta}) Q_{\boldsymbol{\theta}}(x_i, u_k). \tag{5.74}$$

Remark: As we change the parameter $\boldsymbol{\theta}$, the corresponding stationary distribution $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ also changes. Moreover, it is difficult to compute this distribution, and even more difficult to compute the gradient with respect to $\boldsymbol{\theta}$. Therefore the main benefit of the policy theorem is that *the only* function whose gradient appears is $\pi(x_i, u_k, \boldsymbol{\theta})$, and this is a function that is chosen by the learner; therefore it is easy to compute the gradient $\nabla_{\boldsymbol{\theta}} \pi(x_i, u_k, \boldsymbol{\theta})$.

For convenience we will write $\partial / \partial \boldsymbol{\theta}$ instead of $\nabla_{\boldsymbol{\theta}}$, even though $\boldsymbol{\theta}$ is a vector. Recall the expression (5.73) for $V_{\boldsymbol{\theta}}(x_i)$. This leads to

$$\begin{aligned}
\frac{\partial V_{\boldsymbol{\theta}}(x_i)}{\partial \boldsymbol{\theta}} &= \sum_{u_k \in \mathcal{U}} \left[\frac{\partial \pi(x_i, u_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(x_i, u_k) + \pi(x_i, u_k, \boldsymbol{\theta}) \frac{\partial Q_{\boldsymbol{\theta}}(x_i, u_k)}{\partial \boldsymbol{\theta}} \right] \\
&= \sum_{u_k \in \mathcal{U}} \frac{\partial \pi(x_i, u_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(x_i, u_k) + \sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) \frac{\partial}{\partial \boldsymbol{\theta}} \left[R(x_i, u_k) - c^*(\boldsymbol{\theta}) + \sum_{x_j \in \mathcal{X}} a_{ij}^{u_k} V_{\boldsymbol{\theta}}(x_j) \right].
\end{aligned}$$

However, $R(x_i, u_k)$ does not depend on $\boldsymbol{\theta}$ so its gradient is zero. Therefore the above equation can be expressed as

$$\begin{aligned}
\frac{\partial V_{\boldsymbol{\theta}}(x_i)}{\partial \boldsymbol{\theta}} &= \sum_{u_k \in \mathcal{U}} \frac{\partial \pi(x_i, u_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(x_i, u_k) + \sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) \left[-\frac{\partial c^*(\theta)}{\partial \boldsymbol{\theta}} + \sum_{x_j \in \mathcal{X}} a_{ij}^{u_k} \frac{\partial V_{\boldsymbol{\theta}}(x_j)}{\partial \boldsymbol{\theta}} \right] \\
&= \sum_{u_k \in \mathcal{U}} \frac{\partial \pi(x_i, u_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(x_i, u_k) - \frac{\partial c^*(\theta)}{\partial \boldsymbol{\theta}} + \sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) \sum_{x_j \in \mathcal{X}} a_{ij}^{u_k} \frac{\partial V_{\boldsymbol{\theta}}(x_j)}{\partial \boldsymbol{\theta}},
\end{aligned} \tag{5.75}$$

because $\partial c^*(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}$ does not depend on u_k, and

$$\sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) = 1.$$

Now multiply both sides of (5.75) by $\mu_{\boldsymbol{\theta}}(x_i)$ and sum over $x_i \in \mathcal{X}$. This gives

$$\begin{aligned}
\sum_{x_i \in X} \mu_{\boldsymbol{\theta}}(x_i) \frac{\partial V_{\boldsymbol{\theta}}(x_i)}{\partial \boldsymbol{\theta}} &= \sum_{x_i \in X} \mu_{\boldsymbol{\theta}}(x_i) \sum_{u_k \in \mathcal{U}} \frac{\partial \pi(x_i, u_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(x_i, u_k) - \frac{\partial c^*(\theta)}{\partial \boldsymbol{\theta}} \\
&+ \sum_{x_i \in X} \mu_{\boldsymbol{\theta}}(x_i) \sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) \sum_{x_j \in \mathcal{X}} a_{ij}^{u_k} \frac{\partial V_{\boldsymbol{\theta}}(x_j)}{\partial \boldsymbol{\theta}}.
\end{aligned} \tag{5.76}$$

Here again we use the fact that $c^*(\boldsymbol{\theta})$ does not depend on x_i, so that

$$\sum_{x_i \in X} \mu_{\boldsymbol{\theta}}(x_i) \frac{\partial c^*(\theta)}{\partial \boldsymbol{\theta}} = \frac{\partial c^*(\theta)}{\partial \boldsymbol{\theta}}.$$

Now we invoke (5.67) which involves $\boldsymbol{\mu}(\boldsymbol{\theta})$. This shows that the last term on the right side of (5.76) can be expressed as

$$\sum_{x_i \in X} \mu_{\boldsymbol{\theta}}(x_i) \sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) \sum_{x_j \in \mathcal{X}} a_{ij}^{u_k} \frac{\partial V_{\boldsymbol{\theta}}(x_j)}{\partial \boldsymbol{\theta}} = \sum_{x_j \in X} \mu_{\boldsymbol{\theta}}(x_j) \frac{\partial V_{\boldsymbol{\theta}}(x_j)}{\partial \boldsymbol{\theta}}.$$

Therefore (5.76) can be expressed as

$$\sum_{x_i \in X} \mu_{\boldsymbol{\theta}}(x_i) \frac{\partial V_{\boldsymbol{\theta}}(x_i)}{\partial \boldsymbol{\theta}} = \sum_{x_i \in X} \mu_{\boldsymbol{\theta}}(x_i) \sum_{u_k \in \mathcal{U}} \frac{\partial \pi(x_i, u_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(x_i, u_k) - \frac{\partial c^*(\theta)}{\partial \boldsymbol{\theta}} + \sum_{x_j \in X} \mu_{\boldsymbol{\theta}}(x_j) \frac{\partial V_{\boldsymbol{\theta}}(x_j)}{\partial \boldsymbol{\theta}}.$$

Obviously the left side is the same as the last term on the right side. Cancelling these two terms and rearranging gives

$$\frac{\partial c^*(\theta)}{\partial \boldsymbol{\theta}} = \sum_{x_i \in X} \mu_{\boldsymbol{\theta}}(x_i) \sum_{u_k \in \mathcal{U}} \frac{\partial \pi(x_i, u_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(x_i, u_k),$$

which is the desired result.

Now we present a related result known as the "policy gradient theorem with functional approximation." While the formula (5.74) is very neat, its applicability is limited by the fact that it involves the unknown function $Q_{\boldsymbol{\theta}}$. To get around this difficulty, we replace the true function $Q_{\boldsymbol{\theta}}$ by an approximation $\hat{Q}_{\boldsymbol{\theta}}$. Specifically, suppose $\hat{Q}_{\boldsymbol{\theta}}(X_t, U_t)$ is an unbiased estimator of $Q_{\boldsymbol{\theta}}(X_t, U_t)$, for example $R(X_t, U_t)$. In turn suppose that $f : \mathcal{X} \times \mathcal{U} \times \Omega \to \mathbb{R}$ is an approximator to $\hat{Q}_{\boldsymbol{\theta}}$. Note that $\hat{Q}_{\boldsymbol{\theta}}(X_t, U_t)$ can be observed at each time instant. So we try to choose the parameter $\boldsymbol{\omega} \in \Omega$ so as to minimize the average error

$$J(\boldsymbol{\omega}) := \lim_{T \to \infty} \frac{1}{2(T+1)} \sum_{t=0}^{T} [f(X_t, U_t, \boldsymbol{\omega}) - \hat{Q}(X_t, U_t)]^2.$$

So at time t, we can update the current guess via

$$(\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t) \propto \nabla_{\boldsymbol{\omega}} J(\boldsymbol{\omega}_t) = [f(X_t, U_t, \boldsymbol{\omega}) - \hat{Q}(X_t, U_t)] \nabla_{\boldsymbol{\omega}} f(X_t, U_t, \boldsymbol{\omega}_t),$$

where we use only the last term in the summation (as it is the only one that depends on ω_t. Note that we use the symbol $\propto$ because for the moment we are not worried about the choice of the step size.

As is by now familiar practice, we can write $J(\omega)$ equivalently as

$$J(\boldsymbol{\omega}) = \frac{1}{2} \sum_{x_i \in \mathcal{X}} \sum_{u_k \in \mathcal{U}} \eta(x_i, u_k) [f(x_i, u_k, \boldsymbol{\omega}) - \hat{Q}(x_i, u_k)]^2,$$

where $\boldsymbol{\eta}$ is the stationary distribution of the joint Markov process $\{(X_t, U_t)\}$. It is easy to see that

$$\Pr\{X_t = x_i, U_t = u_k | \pi(\boldsymbol{\theta})\} = \Pr\{X_t = x_i | \pi(\boldsymbol{\theta})\} \cdot \Pr\{U_t = u_k | X_t = x_i, \pi(\boldsymbol{\theta})\} = \mu_{\boldsymbol{\theta}}(x_i) \pi(x_i, u_k, \boldsymbol{\theta}).$$

We will encounter the above formula again. Also, since $\hat{Q}_{\boldsymbol{\theta}}$ is an unbiased estimate of $Q_{\boldsymbol{\theta}}$, we can replace $\hat{Q}_{\boldsymbol{\theta}}$ by $Q_{\boldsymbol{\theta}}$ when we take the expectation (i.e., sum over x_i and u_k). So when $\boldsymbol{\omega}$ is chosen such that $\nabla_{\boldsymbol{\omega}} J(\omega) = \mathbf{0}$ (i.e., at a stationary point of $J(\cdot)$), we have that

$$\sum_{x_i \in \mathcal{X}} \sum_{u_k \in \mathcal{U}} \mu_{\boldsymbol{\theta}}(x_i) \pi(x_i, u_k, \boldsymbol{\theta}) [f(x_i, u_k, \boldsymbol{\omega}) - Q_{\boldsymbol{\theta}}(x_i, u_k)] \nabla_{\boldsymbol{\omega}} f(x_i, u_k, \boldsymbol{\omega}) = \mathbf{0}. \tag{5.77}$$

Now suppose that the $\hat{Q}_{\boldsymbol{\theta}}$-approximating function f is "compatible" with the policy-approximating function π in the sense that

$$\nabla_{\boldsymbol{\omega}} f(x_i, u_k, \boldsymbol{\omega}) = \frac{1}{\pi(x_i, u_k, \boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \pi(x_i, u_k, \boldsymbol{\theta}). \tag{5.78}$$

One possibility is simply to choose $f(\cdot, \cdot, \boldsymbol{\omega})$ to be linear in $\boldsymbol{\omega}$, with the gradient given by the right side of (5.78). Now (5.78) can be rewritten as

$$\pi(x_i, u_k, \boldsymbol{\theta}) \nabla_{\boldsymbol{\omega}} f(x_i, u_k, \boldsymbol{\omega}) = \nabla_{\boldsymbol{\theta}} \pi(x_i, u_k, \boldsymbol{\theta}).$$

Hence, under (5.78), (5.77) implies that

$$\sum_{x_i \in \mathcal{X}} \sum_{u_k \in \mathcal{U}} \mu_{\boldsymbol{\theta}}(x_i) \nabla_{\boldsymbol{\theta}} \pi(x_i, u_k, \boldsymbol{\theta}) [f(x_i, u_k, \boldsymbol{\omega}) - Q_{\boldsymbol{\theta}}(x_i, u_k)] = \mathbf{0}.$$

Combining the above with (5.74) gives an alternate formula for the gradient of the value function $c^*(\boldsymbol{\theta})$, namely

$$\nabla_{\boldsymbol{\theta}} c^*(\boldsymbol{\theta}) = \sum_{x_i \in \mathcal{X}} \sum_{u_k \in \mathcal{U}} \mu_{\boldsymbol{\theta}}(x_i) \nabla_{\boldsymbol{\theta}} \pi(x_i, u_k, \boldsymbol{\theta}) f(x_i, u_k, \boldsymbol{\omega}). \tag{5.79}$$

The significance of this formula is that the unknown function $Q_{\boldsymbol{\theta}}$ can be replaced by the approximating function $f(x_i, u_k, \boldsymbol{\omega})$. So if we use a gradient search method (for example) to update $\boldsymbol{\theta}_t$, we can use (5.79) instead of (5.74). Note that this introduces a "coupling" between $\boldsymbol{\theta}_t$ and $\boldsymbol{\omega}_t$. In other words, as the policy approximation gets updated, so does the value approximation. This is a precursor to actor-critic methods which we study next.

5.3.3 Actor-Critic Methods

In this subsection, we elaborate upon the policy gradient theorem to study simultaneous approximation of both policy and value, often known as actor-critic methods. The actor is the policy, which is controlled by a parameter $\boldsymbol{\theta} \in \mathbb{R}^d$, while the critic is an approximator of the value corresponding to the policy, and is controlled by a parameter $\boldsymbol{\zeta} \in \mathbb{R}^l$. In [23, 22], which is the main reference for this subsection, it is assumed that $l > d$, that is, there are mor adjustable parameters in the critic than there are in the actor. Morever, the actor is of the form $\Psi\boldsymbol{\theta}$ while the critic is of the form $\Phi\boldsymbol{\zeta}$. Not only does Φ have more columns than Ψ, but the range of Ψ is a subset of the range of Ψ. In other words, the critic's parameterization contains the actor's parametrization as a subset.

The initial ideas in [23] are similar to those in [34], but with different notation. So we begin by restating the first part of [23] in the current notation. If $\pi_{\boldsymbol{\theta}}$ is a policy, then under this policy not only is $\{X_t\}$ a Markov process, but so is the joint process $\{(X_t, U_t)\}$. If $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ denotes the stationary distribution of X_t, then the joint distribution of (X_t, U_t) is given by

$$\Pr\{X_t = x_i, U_t = u_k\} = \Pr\{X_t = x_i\} \Pr\{U_t = u_k | X_t = x_i, \pi_\theta\} = \mu_{\boldsymbol{\theta}}(x_i) \pi(x_i, u_k, \boldsymbol{\theta}) =: \eta_{\boldsymbol{\theta}}(x_i, u_k). \tag{5.80}$$

With this definition, (5.70) can also be written a

$$c^*(\pi(\boldsymbol{\theta})) = \sum_{x_i \in \mathcal{X}} \sum_{u_k \in \mathcal{U}} \eta_{\boldsymbol{\theta}}(x_i, u_k) R(x_i, u_k) = E[R, \boldsymbol{\eta}]. \tag{5.81}$$

Thus $c^*(\boldsymbol{\theta})$ is the expected value of the reward R under the joint stationary distribution of (X_t, U_t). Moreover, $V_{\boldsymbol{\theta}}$ as defined in (5.73) satisfies the Poisson equation

$$c^*(\boldsymbol{\theta}) + V_{\boldsymbol{\theta}}(x_i) = \sum_{u_k \in \mathcal{U}} \pi(x_i, u_k, \boldsymbol{\theta}) \left[R(x_i, u_k) + \sum_{x_j \in \mathcal{X}} a_{ij}^{u_k} V_{\boldsymbol{\theta}}(x_j) \right]. \tag{5.82}$$

We continue to have the characterization (5.72) for $Q_{\boldsymbol{\theta}}$.

Now we make a few assumptions regarding the policy. Define the vector

$$\boldsymbol{\omega}(x_i, u_k, \boldsymbol{\theta}) = \frac{1}{\pi(x_i, u_k, \boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \pi(x_i, u_k, \boldsymbol{\theta}). \tag{5.83}$$

Note that

$$\omega(x_i, u_k, \theta_l) = \frac{\partial \ln \pi(x_i, u_k, \boldsymbol{\theta})}{\partial \theta_i}.$$

We can also write

$$\nabla_{\boldsymbol{\theta}} \pi(x_i, u_k, \boldsymbol{\theta}) = \pi(x_i, u_k, \boldsymbol{\theta}) \omega(x_i, u_k, \boldsymbol{\theta}).$$

So (5.74) becomes

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} c^*(\boldsymbol{\theta}) &= \sum_{u_k \in \mathcal{U}} \sum_{x_i \in \mathcal{X}} \mu_{\boldsymbol{\theta}}(x_i) \pi(x_i, u_k, \boldsymbol{\theta}) \omega(x_i, u_k, \boldsymbol{\theta}) Q_{\boldsymbol{\theta}}(x_i, u_k) \\ &= \sum_{u_k \in \mathcal{U}} \sum_{x_i \in \mathcal{X}} \eta_{\boldsymbol{\theta}}(x_i, u_k) \omega(x_i, u_k, \boldsymbol{\theta}) Q_{\boldsymbol{\theta}}(x_i, u_k). \end{aligned} \tag{5.84}$$

Now we give a very brief description of the actor-critic methods in [23]. Suppose $\mathbf{q}_1, \mathbf{q}_2$ are two functions that map $\mathcal{X} \times \mathcal{U}$ into $\mathbb{R}$. In other words, $\mathbf{q}_1, \mathbf{q}_2$ are $|\mathcal{X}| \cdot |\mathcal{U}|$-dimensional vectors. Suppose we define an inner product $\langle \mathbf{q}_1, \mathbf{q}_2 \rangle_{\boldsymbol{\eta}}$ as

$$\langle \mathbf{q}_1, \mathbf{q}_2 \rangle_{\boldsymbol{\eta}} = \sum_{u_k \in \mathcal{U}} \sum_{x_i \in \mathcal{X}} \eta_{\boldsymbol{\theta}}(x_i, u_k) q_1(x_i, u_k) q_2(x_i, u_k).$$

This is similar to Lemma 5.4, where we defined an inner product making use of a stationary distribution. Then

$$\frac{\partial c^*(\boldsymbol{\theta})}{\partial \theta_i} = \langle \mathbf{q}_{\boldsymbol{\theta}}, \boldsymbol{\omega}_{\boldsymbol{\theta}} \rangle_{\boldsymbol{\eta}},$$

where $\mathbf{q}_{\boldsymbol{\theta}}$ is the function $Q_{\boldsymbol{\theta}}$ written out as a vector of dimension nm, and $\boldsymbol{\omega}_{\boldsymbol{\theta}} \in \mathbb{R}^{nm}$ is the vector defined in (5.83). Hence, in order to compute this partial derivative, it is not necessary to know the vector $Q_{\boldsymbol{\theta}}$ – it is enough to know a projection of it on the space spanned by the $\boldsymbol{\omega}_{\boldsymbol{\theta}}$ vectors. Now suppose we use a "linear" parametrization of the type

$$\pi(\boldsymbol{\theta}) = \sum_{l=1}^{d} \boldsymbol{\psi}_l \theta_l,$$

where each $\boldsymbol{\psi}_l \in \mathbb{R}^{nm}$ and the components can be thought of as $\psi_l(x_i, u_k)$. Recall that $\pi(\boldsymbol{\theta})$ is a probability distribution on $\mathcal{U}$, so the set of policies consists of summations of the above form that are in $\mathbb{S}(\mathcal{U})$. One possibility is to choose the columns to correspond to policies, and to restrict $\boldsymbol{\theta}$ to vary over the unit simplex in $\mathbb{R}^d$, so that the policy $\pi_{\boldsymbol{\theta}}$ is a *convex combination* of the columns of Ψ, as opposed to a *linear combination* of the columns. In any case, we have

$$\frac{\partial \pi(\boldsymbol{\theta})}{\partial \theta_l} = \boldsymbol{\psi}_l,$$

and as a result

$$\boldsymbol{\omega}_l = \frac{1}{\pi(\boldsymbol{\theta})} \boldsymbol{\psi}_l \propto \boldsymbol{\psi}_l.$$

Therefore the span of the vectors $\{\boldsymbol{\omega}_l\}$ is the same as the span of the vectors $\boldsymbol{\psi}_l$, that is, $\Psi(\mathbb{R}^d)$. So all we need to find is a projection of the nm-dimensional vector $\mathbf{q}_{\boldsymbol{\theta}}$ onto $\Psi(\mathbb{R}^d)$. The projection is computed using the inner product $\langle \cdot, \cdot \rangle_{\boldsymbol{\eta}}$ defined above. Nevertheless, the projection belongs to $\Psi(\mathbb{R}^d)$.

In [23], the authors slightly over-parametrize by taking

$$\hat{\mathbf{q}}_{\boldsymbol{\theta}} = \sum_{l=1}^{s} \boldsymbol{\phi}_l \zeta_l = \Phi \boldsymbol{\zeta},$$

as an approximation for the *projection* of $\mathbf{q}$, where $\Phi \in \mathbb{R}^{nm\times s}$ and $\boldsymbol{\zeta} \in \mathbb{R}^s$. Here $s \ll nm$ so that the above approximation results in a reduction in the dimension of $\mathbf{q}$. At the same time, one chooses $s > d$, so that $\boldsymbol{\zeta}$ has more parameters than $\boldsymbol{\theta}$. Moreover, one chooses the vectors in such a way that

$$\text{span}\{\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_d\} \subseteq \text{span}\{\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_s\}.$$

Therefore, the approximant $\hat{\mathbf{q}}$ may not belong to $\Psi(\mathbb{R}^d)$. However, the projection of $\hat{\mathbf{q}}$ onto $\Psi(\mathbb{R}^d)$ can be used to determine the gradient of $c^*(\boldsymbol{\theta})$.

Now we present the actor-critic methods in [23]. Unlike in earlier situations where we had a fixed policy, here the policy also varies with t. Nevertheless, we will get a sample path $\{(X_t, U_t, W_{t+1})\}$. That sample path is used below.

The critic tries to form an estimate of the average reward process. The equations are analogous to (5.45), (5.46) and (5.47). There are two estimates, one for the value of the process $c^*(\boldsymbol{\theta})$, and another for the parameter ζ_t that gives an approximation $\hat{\mathbf{q}}_{\boldsymbol{\zeta}_t} = \Phi\boldsymbol{\zeta}_t$. These updates are as follows:

$$c_{t+1} = c_t + \beta_{t+1}(W_{t+1} - c_t),$$

$$\boldsymbol{\zeta}_{t+1} = \boldsymbol{\zeta}_t + \beta_{t+1}\delta_{t+1}\mathbf{z}_t,$$

where β_t is a step size,

$$\delta_{t+1} = W_{t+1} - c_t + \hat{Q}_{\boldsymbol{\zeta}_t}(X_{t+1}, U_{t+1}) - \hat{Q}_{\boldsymbol{\zeta}_t}(X_t, U_t)$$

is the temporal difference. The eligibility vector $\mathbf{z}_t$ is defined as follows: Let

$$\mathbf{y}_\tau := [\Phi^{(X_t,U_t)}]^\top$$

denote the (X_t, U_t)-th row of the matrix Φ, transposed. Then the TD$^{(\lambda)}$ update is defined via

$$\mathbf{z}_t = \sum_{\tau=0}^{t} \lambda^{t-\tau}\mathbf{y}_\tau$$

As before $\mathbf{z}_t$ satisfies the recursion

$$\mathbf{z}_t = \lambda\mathbf{z}_{t-1} + \mathbf{y}_t.$$

Konda and Tsitsiklis study the case $\lambda = 1$, so that

$$\mathbf{z}_t = \mathbf{z}_{t-1} + \mathbf{y}_t.$$

This is at the opposite end of the spectrum from $TD(0)$ which is the standard TD-learning.

For updating the policy, we choose a globally Lipscitz-continuous function $\Gamma : \mathbb{R}^s \to \mathbb{R}_+$ with the property that, for some finite constant $C > 0$, we have

$$\Gamma(\boldsymbol{\zeta}) \leq \frac{C}{1 + \|\boldsymbol{\zeta}\|}, \ \forall \boldsymbol{\zeta} \in \mathbb{R}^s.$$

Then the policy update is just a gradient update, given by

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_{t+1}\Gamma(\boldsymbol{\zeta}_t)(\hat{Q}_{\boldsymbol{\zeta}_t}(X_{t+1}, U_{t+1})\boldsymbol{\omega}_{\boldsymbol{\theta}_t}(X_{t+1}, U_{t+1}).$$

Note that the update of $\boldsymbol{\theta}_t$ depends also on $\boldsymbol{\zeta}_t$.

To study the convergence properties of the above actor-critic method, we make the following assumption called "uniform positive definiteness": For each $\boldsymbol{\theta} \in \mathbb{R}^d$, the $s \times s$ matrix $G(\boldsymbol{\theta})$ satisfies

$$G(\boldsymbol{\theta}) = \sum_{x_i \in \mathcal{X}} \sum_{u_k \in \mathcal{U}} \eta(x_i, u_k, \boldsymbol{\theta})[\Phi^{(x_i,u_k)}]^\top \Phi^{(x_i,u_k)} \geq \epsilon I_s,$$

for some positive constant ϵ. With this assumption we can state the following theorem:

Theorem 5.6. *Suppose that*

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \sum_{t=0}^{\infty} \alpha_t^2 < \infty, \sum_{t=0}^{\infty} \beta_t = \infty, \sum_{t=0}^{\infty} \beta_t^2 < \infty,$$

and in addition

$$\frac{\alpha_t}{\beta_t} \to 0 \text{ as } t \to \infty.$$

If $\{\boldsymbol{\theta}_t\}$ *is bounded almost surely, then*

$$\lim_{t\to\infty} \|\nabla_{\boldsymbol{\theta}} c^*(\boldsymbol{\theta})\| = 0.$$

Note that the policy parameter $\boldsymbol{\theta}_t$ is update more slowly than the value parameter $\boldsymbol{\zeta}_t$.

Problem 5.4. Show that (5.62) is a consequence of (5.61).

Problem 5.5. Prove (5.66).

5.4 Zap Q-Learning

Chapter 6

Introduction to Empirical Processes

Chapter 7

Finite-Time Bounds

In previous chapters, the convrgence results presented are mostly asymptotic in nature. By and large, they do not tell us what happens after a particular learning algorithm has been run a finite number of times. In contrast, in the present chapter, several results are presented that are "finite time" in nature.

7.1 Finite Time Bounds on Regret

Suppose there are K machines ("bandits") with random payoffs in $[0,1]$, with unknown means $\mu_1, \cdots, \mu_K$, and unknown probability distributions. Each time a machine is played, it returns a payoff distributed according to its unknown probability distribution. Moreover, each return is independent of all other returns of the same machine. It is evident that, as a machine is played more and more times, the learner knows more and more about the unknown probability distribution. In turn, this information can be used to make quasi-informed decisions about which machine to play next. The objective is to develop a strategy for choosing the next machine to play in such a way that an appropriate objective function is optimized. The objective function studied in this section is the "regret" associated with a strategy. Define

$$\mu^* := \max_{1 \le i \le K} \mu_i.$$

Thus μ^* is the best possible return. If learner knew which machine has the return μ^*, then s/he would just keep playing that machine, which would lead to the maximum possible expected return per play. On the other hand, there is some amount of "exploration" of all the machines, during the course of which many suboptimal choices will be made. The regret attempts to capture the return foregone, which is seen as the cost of exploration. Specifically, suppose that some strategy for choosing the machine to be played at each time generates a sequence of indices $\{M_t\}_{t \ge 1}$ where each M_t lies between 1 and K. After n plays, let $T_i(n)$ denote the number of times that machine i is played. Where convenient, we also denote this as $n_i(n)$ or just n_i if n is obvious from the context. Now, the *actual* returns of machine i at these n_i time instants are random, and can be denoted by $\{X_{i,1}, \cdots, X_{i,n_i}\}$. By playing machine i a total of n_i times until time n, the reward foregone is

$$\mu^* n - \sum_{i=1}^{K} \mu_i n_i.$$

However, this is a random number, because the number of plays n_i is random. Therefore, corresponding to a policy for choosing the next machine to be played, the **regret** is defined as

$$C = \mu^* n - \sum_{i=1}^{K} \mu_i E[n_i]. \tag{7.1}$$

There are several equivalent ways to express the regret. For example, define the **empirical return** of machine i after n plays as

$$\hat{\mu}_i := \frac{1}{n_i} \sum_{t=1}^{n_i} X_{i,t}. \tag{7.2}$$

Due to the assumption that the returns are i.i.d., it is obvious that $E[\hat{\mu}_i] = \mu_i$. We can write

$$C = E\left[\sum_{i=1}^{K} n_i(\mu^* - \hat{\mu}_i)\right].$$

In this section we follow [2]. We study two strategies, and derive upper bounds for the regret of each strategy. In the second case, our result is a slight generalization of the cotents of [2].

7.2 Finite Time Bounds for Reinforcement Learning

7.3 Probably Approximately Correct Markov Decision Processes

7.4 Unification of Regret and RL Bounds

7.5 Empirical Dynamic Programming

Chapter 8

Background Material

The objective of this chapter is to collect in one place the background material required to understand the main body of the notes. While the exposition is rigorous, and several references are given throughout, these notes are not, by themselves, sufficient to gain a mastery over these topics. A reader who is encountering these topics for the first time is strongly encouraged to consult the various references in order to understand the present discussion thoroughly. To avoid tedious repetition, we do not include the phrase "Introduction to" in the title of each section, but that should be assumed.

8.1 Random Variables and Stochastic Processes

In this section we give a very cursory introduction to the topics of measure, probability, random variables, and allied concepts. The topic can be said to have been started by Kolmogorov, and his very brief monograph [21] gives a good motivation for the subject. For concepts from measure theory, the reader can consult [4] which is a thorough treatment of these topics. Another good source with greater emphasis on probability theory is [27]. Topics such as conditional expectation and martingales are briefly discussed in [6, 11]. However, a more detailed introduction can be found in [48].

8.1.1 Random Variables

Definition 8.1. Suppose Ω is a set and that $\mathcal{F}$ is a collection of subsets of X. Then $\mathcal{F}$ is said to be a **σ-algebra**[1] if $\mathcal{F}$ satisfies the following axioms:

S1. $\Omega \in \mathcal{F}$.

S2. If $A \in \mathcal{F}$ then $A^c \in \mathcal{F}$, where A^c denotes the complement of A in Ω.[2]

S3. If $\{A_i\}_{i\geq 1}$ is any countable sequence of sets belonging to $\mathcal{F}$, then

$$\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}. \tag{8.1}$$

The pair $(\Omega, \mathcal{F})$ is called a **measurable space**.

Definition 8.2. Suppose $(\Omega, \mathcal{F})$ is a measurable space. A function $P : \mathcal{F} \to [0, 1]$ is called a **probability measure** if it satisfies the following axioms:

[1] The term σ-field is more popular, but this terminology is preferred here.
[2] Note that [S1] and [S2] together imply that $\emptyset \in \mathcal{F}$.

P1. $P(\Omega) = 1$.

P2. P is *countably additive*, that is: Whenever $\{A_i\}$ are pairwise disjoint sets from $\mathcal{F}$, we have that

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i). \tag{8.2}$$

The triple $(\Omega, \mathcal{F}, P)$ is called a **probability space**.

Note that if Ω is a finite or countable set, it is customary to take $\mathcal{F}$ to be the "power set" of Ω, that is, the collection of all subsets of Ω, often denoted by 2^{Ω}. If a nonnegative weight p_i is assigned to each element $i \in \Omega$, and if P is defined as

$$P(A) = \sum_{i \in A} p_i = \sum_{i \in \Omega} p_i I_{\{i \in A\}}, \tag{8.3}$$

then it is easy to verify that $(\Omega, 2^{\Omega}, P)$ is a probability space. However, if Ω is an uncountable set, e.g., the real numbers, then the above approach of assigning weights to individual elements does not work. Note that in (8.3), $I_{\{i \in A\}}$ is the indicator function that equals 1 if $i \in A$ and 0 if $i \notin A$.

Definition 8.3. Suppose $(\Omega, \mathcal{F})$ and $(\mathcal{X}, \mathcal{G})$ are measurable spaces. Then a map $f : \Omega \to \mathcal{X}$ is said to be **measurable** if $f^{-1}(S) \in \mathcal{F}$ for all $S \in \mathcal{G}$.

Thus a map from Ω into $\mathcal{X}$ is measurable if the preimage of every set in $\mathcal{G}$ under f belongs to $\mathcal{F}$.

Definition 8.4. Suppose $(\Omega, \mathcal{F}, P)$ is a probability space, and $(\mathcal{X}, \mathcal{G})$ is a measurable space. A function $X : \Omega \to \mathcal{X}$ is said to be a **random variable** if it is measurable, that is,

$$X^{-1}(S) \in \mathcal{F}, \ \forall S \in \mathcal{G}. \tag{8.4}$$

In such a case, for each set $S \in \mathcal{G}$, the quantity

$$P(X^{-1}(S)) =: P_X(S)$$

is called the **probability that** $X \in S$.

In the above definition, $(\mathcal{X}, \mathcal{G})$ is called the "event space," and the sets belonging to the σ-algebra $\mathcal{G}$ are called "events," because each such set has a probability associated with it (via (8.4))). The triple $(\Omega, \mathcal{F}, P)$ is called the "sample space."

Example 8.1. Suppose we wish to capture the notion of a two-sided coin that comes up H for heads 60% of the time, and T for tails 40% of the time. In such a case, the event space (the set of possible outcomes) is just $\mathcal{X} = \{H, T\}$. Because the set $\mathcal{X}$ is finite, the corresponding σ-algebra $\mathcal{G}$ can be just $2^{\mathcal{X}} = \{\emptyset, \{H\}, \{T\}, \mathcal{X}\}$. The sample space $(\Omega, \mathcal{F}, P)$ can be anything, as can the map $f : \Omega \to \mathcal{X}$, provided only that two conditions hold: First,

$$f^{-1}(\{H\}) = \{\omega \in \Omega : f(\omega) = H\} \in \mathcal{F}, f^{-1}(\{T\}) = \{\omega \in \Omega : f(\omega) = T\} \in \mathcal{F}.$$

(Actually, either one of the conditions would imply the other.) Second,

$$P(f^{-1}(\{H\}) = 0.6, P(f^{-1}(\{T\}) = 0.4.$$

Definition 8.5. Suppose X is a random variable defined on the sample space $(\Omega, \mathcal{F}, P)$ taking values in $(\mathcal{X}, \mathcal{G})$. Then the **$\sigma$-algebra generated by** X is defined as the smallest σ-algebra contained in $\mathcal{F}$ with respect to which X is measurable, and is denoted by $\sigma(X)$.

Example 8.2. Consider again the random variable studied in Example 8.1. Thus $\mathcal{X} = \{H, T\}$ and $\mathcal{G} = 2^{\mathcal{X}}$. Now suppose X is a measurable map from some $(\Omega, \mathcal{F}, P)$ into $(\mathcal{X}, \mathcal{G})$. Then all possible preimages of sets in $\mathcal{G}$ are:

$$\emptyset = X^{-1}(\emptyset), \Omega = X^{-1}(\mathcal{X}), A := X^{-1}(\{H\}), B := X^{-1}(\{T\}) = A^c = \Omega \setminus A.$$

Thus the *smallest possible* σ-algebra on Ω with respect to which X is measurable consists of $\{\emptyset, A, A^c, \Omega\}$. Therefore this is the σ-algebra of Ω generated by X. Any other sets in $\mathcal{F}$ are basically superfluous. We can carry this argument further and simply take the sample space Ω to be the same as the event space $\mathcal{X}$, and X to be the identity operator on $(\Omega, \mathcal{F}, P)$ into $(\Omega, \mathcal{F})$. Thus $\Omega = \{H, T\}$, and $\mathcal{F} = \{\emptyset, \{H\}, \{T\}, \Omega\}$. Further, we can define $P(\{H\}) = 0.6$, $P(\{T\}) = 0.4$. This is sometimes called the **canonical representation** of the random variable X. Usually we can do this whenever the event space is finite or countable.

Originally, the phrase "random variable" was used only for the case where the event space $\mathcal{X} = \mathbb{R}$, and the σ-algebra is the so-called **Borel σ-algebra**, which is defined as the smallest σ-algebra of subsets of $\mathbb{R}$ that contains all closed subsets of $\mathbb{R}$. Random quantities such as the outcomes of coin-toss experiments were called something else (depending on the author). Subsequently, the phrase "random variable" came to be used for *any* situation where the outcome is uncertain, as defined above. In the context of Markov Decision Processes, often the Markov process evolves over a finite set, and the action space is also finite. So a lot of the heavy machinery above is not needed to describe the evolution of an MDP. However, in reinforcement learning, the parameters of the MDP need to be estimated, including the reward–and these are real-valued quantities. So it is desirable to deal with random variables that assume values in a continuum. When we do that, the sample set Ω equals $\mathbb{R}$ or some subset thereof, and $\mathcal{F}$ equals the Borel σ-algebra. Finally, the probability measure P is defined on Ω.

8.1.2 Independence, Joint and Conditional Probabilities

Kolmogorov, who laid down the foundations of probability theory, remarks on [21, p. 8] (in English translation) that

> Historically, the independence of experiments and random variables represents the very mathematical concept that has given the theory of probability its peculiar stamp.

This statement, together with the text that precedes it, can be paraphrased as: Without the concept of independence, there is essentially no difference between measure theory and probability theory. Thus the concept of independence is fundamental (and unique) to probability theory.

Definition 8.6. Suppose $(\Omega, \mathcal{F}, P)$ is a probability space. Then two events $S, T \in \mathcal{F}$ are said to be **independent** if

$$P(S \cap T) = P(S)P(T).$$

Suppose now that $\mathcal{F}_1, \mathcal{F}_2$ are sub-σ-algebras of $\mathcal{F}$. Then $\mathcal{F}_1$ and $\mathcal{F}_2$ are said to be **independent** if

$$P(S \cap T) = P(S)P(T), \ \forall S \in \mathcal{F}_1, T \in \mathcal{F}_2. \tag{8.5}$$

Two random variables X_1, X_2 defined on $(\Omega, \mathcal{F}, P)$ are said to be **independent** if the corresponding σ-algebras $\sigma(X_1), \sigma(X_2)$ are independent.

It is easy to show that two events S, T are independent if and only if the corresponding σ-algebras $\mathcal{F}_1 = \{\emptyset, S, S^c, \Omega\}$ and $\mathcal{F}_2 = \{\emptyset, T, T^c, \Omega\}$ are independent.

The extension of the above definition to any finite number of events, or σ-algebras, or random variables, is quite obvious. For more details, see [11, Section 3.1] or [48, Chapter 4].

Until now we have discussed what might be called "individual" random variables. Now we discuss the concept of joint random variables, and the associated notion of joint probability. The definition below is for two joint variables, but it is obvious that a similar definition can be made for any finite number of joint random variables. In turn this reads to the concept of conditional probability.

Definition 8.7. Suppose $(\mathcal{X}, \mathcal{G})$ and $(\mathcal{Y}, \mathcal{H})$ are measurable spaces. Then the **product** of these two spaces is $(\mathcal{X} \times \mathcal{Y}, \mathcal{G} \otimes \mathcal{H})$ where $\mathcal{G} \otimes \mathcal{H}$ is the smallest σ-algebra of subsets of $\mathcal{X} \times \mathcal{Y}$ that contains all products of the form $S \times T, S \in \mathcal{G}, T \in \mathcal{H}$.

Note that $\mathcal{G} \otimes \mathcal{H}$ is called the "product" σ-algebra, and is not to be confused with $\mathcal{G} \times \mathcal{H}$, the Cartesian product of the two collections $\mathcal{G}$ and $\mathcal{H}$. In fact one can write $\mathcal{G} \otimes \mathcal{H} = \sigma(\mathcal{G} \times \mathcal{H})$, where $\sigma(\mathcal{S})$ denotes the smallest σ-algebra containing all sets in the collection $\mathcal{S}$. Previously we had defined $\sigma(X)$, the σ-algebra generated by a random variable X. The two usages are consistent. Suppose X is a random variable on $(\Omega, \mathcal{F}, P)$ mapping Ω into $(\mathcal{X}, \mathcal{G})$, and let $\mathcal{S}$ consist of all preimages in Ω of sets in $\mathcal{G}$. Then $\sigma(X)$ and $\sigma(\mathcal{S})$ are the same.

Suppose $(\Omega, \mathcal{F}, P)$ is a probability space, and that $(\mathcal{X}, \mathcal{G})$ and $(\mathcal{Y}, \mathcal{H})$ are measurable spaces. Let $(\mathcal{X} \times \mathcal{Y}, \mathcal{G} \otimes \mathcal{H})$ denote their product. Suppose further that $Z : \Omega \to \mathcal{X} \times \mathcal{Y}$ is measurable and thus a random variable taking values in $\mathcal{X} \times \mathcal{Y}$. Express Z as (X, Y) where X, Y are the components of Z, so that $X : \Omega \to \mathcal{X}, Y : \Omega \to \mathcal{Y}$. Then it can be shown that $\mathcal{X}$ and $\mathcal{Y}$ are themselves measurable and are thus random variables in their own right. The probability measures associated with these two random variables are as follows:

$$P_X(S) := P^{-1}(Z \in (S \times \mathcal{Y})),\ \forall S \in \mathcal{G}, P_Y(T) := P^{-1}(Z \in (\mathcal{X} \times T)),\ \forall T \in \mathcal{H}. \tag{8.6}$$

We refer to $Z = (X, Y)$ as a joint random variable with **joint probability measure** P_Z, and to P_X and P_Y as the **marginal probability measures** (or just marginal probabilities) of P_Z for X and Y respectively.

Definition 8.8. Suppose $(\Omega, \mathcal{F}, P)$ is a probability space. Suppose $(\mathcal{X}, \mathcal{G})$ and $(\mathcal{Y}, \mathcal{H})$ are measurable spaces, and let $Z = (X, Z) : \Omega \to \mathcal{X} \times \mathcal{Y}$ be a joint random variable. Finally, suppose $S \in \mathcal{G}, T \in \mathcal{H}$ are events involving X and Y respectively. Then the **conditional probability** $\Pr\{X \in S | Y \in T\}$ is defined as

$$\Pr\{X \in S | Y \in T\} = \frac{\Pr\{Z = (X, Y) \in S \times T\}}{\Pr\{Y \in T\}} = \frac{P_Z(S \times T)}{P_Y(T)}. \tag{8.7}$$

Further, X and Y are said to be **independent** if

$$P_Z(S \times T) = P_X(S) \times P_Y(T),\ \forall S \in \mathcal{G}, T \in \mathcal{H}. \tag{8.8}$$

In the definition of the conditional probability (8.7), it is assumed that $P_Y(T) > 0$.

A common application of conditional probabilities arises when both $\mathcal{X}$ and $\mathcal{Y}$ are finite sets. In this case X, Y, Z are random variables assuming values in finite sets $\mathcal{X}, \mathcal{Y}, \mathcal{X} \times \mathcal{Y}$. Suppose to be specific that $\mathcal{X} = \{x_1, \cdots, x_n\}$ and $\mathcal{Y} = \{y_1, \cdots, y_m\}$. Then it is convenient to represent the joint probability distribution of $Z = (X, Y)$ as an $n \times m$ matrix Θ, where

$$\theta_{ij} = \Pr\{Z = (x_i, y_j)\} = \Pr\{X = x_i \& Y = y_j\}.$$

Let us denote the marginal probabilities as

$$\phi_i = \Pr\{X = x_i\}, \psi_j = \Pr\{Y = y_j\}.$$

Then it is easy to infer that

$$\boldsymbol{\phi}^\top = \Theta \mathbf{1}_n, \boldsymbol{\psi} = \mathbf{1}_n^\top \Theta,$$

where $\mathbf{1}_k$ denotes a column vector of k ones. Note that we follow the convention that a probability distribution is a row vector. Also, in this simple situation, it can be assumed without any loss of generality that $\phi_i > 0$ for all i, and $\psi_j > 0$ for all j. If $\phi_i = 0$ for some index i, it means that $\theta_{ij} = 0$ for all j; therefore the element x_i can be deleted from the $\mathcal{X}$ without affecting anything. Similar remarks apply to $\boldsymbol{\psi}$ as well. With these notational conventions, it is easy to see that

$$\Pr\{X \in S | Y \in T\} = \frac{\Pr\{Z = (X, Y) \in S \times T\}}{\Pr\{Y \in T\}} = \frac{\sum_{x_i \in S, y_j \in T} \theta_{ij}}{\sum_{y_j \in T} \psi_j}.$$

All of the above definitions can be extended to more than two random variables.

8.1.3 Conditional Expectations

The concept of conditional *probability* discussed above can be applied to even "abstract" random variables, that is, random variables assuming values in some abstract set. In contrast, concepts such as expected value (both unconditional and conditional) are meant to be used with real-valued random variables. The ideas extend readily to vector-valued random variables by applying them componentwise. The objective of this subsection is to introduce these concepts. The discussion below requires an understanding of *integration* with respect to a probability measure. We do not go into too many details regarding the abstract concept of integration with respect to a measure, because that would be rather tangential to the main discussion. Instead we refer interested reader to [4] for details.

Throughout this subsection, we deal with real-valued random variables. For this purpose, on the set $\mathbb{R}$ of real numbers, we define the **Borel σ-algebra**, which is denoted by $\mathcal{B}$ and consists of the smallest σ-algebra of subsets of $\mathbb{R}$ that contains all closed sets.[3] Thus, when we say that X is a real random variable on $(\Omega, \mathcal{F}, P)$, we mean that X is a measurable map from $(\Omega, \mathcal{F}, P)$ to $(\mathbb{R}, \mathcal{B})$.

Note that a probability space can be thought of as a measure space where the underlying set (Ω) has measure one. So in principle we can attempt to integrate the function $X(\omega), \omega \in \Omega$ using the measure P. Therefore, if it exists, the quantity

$$E[X, P] := \int_\Omega X(\omega) P(d\omega) \tag{8.9}$$

is called the **mean** or the **expected value** of the random variable X.[4] Next, for $1 \leq p < \infty$, we define the function space $L_p(\Omega, P)$ as the set of functions whose p-th powers are absolutely integrable, that is

$$L_p(\Omega, P) := \left\{ f : \Omega \to \mathbb{R} \text{ s.t. } \int_\Omega |f(\omega)|^p P(d\omega) < \infty \right\}. \tag{8.10}$$

The L_p-norm of a function $f \in L_p(\Omega, P)$ is defined as

$$\|f\|_p := \left[\int_\Omega |f(\omega)|^p P(d\omega) \right]^{1/p}. \tag{8.11}$$

If $p = \infty$, we define $L_\infty(\Omega, P)$ to be the set of functions that are **essentially bounded**, that is, bounded except on a set of measure zero, and define the corresponding norm as the "essential supremum" of $f(\cdot)$, that is

$$\|f\|_\infty = \inf\{c : P\{|f(\omega)| \geq c\} = 0\}. \tag{8.12}$$

Next, for a given $p \in [1, \infty]$, and define the **conjugate index** $q \in [1, \infty]$ as the unique solution of the equation

$$\frac{1}{p} + \frac{1}{q} = 1. \tag{8.13}$$

In particular, if $p \in (1, \infty)$, then $q = p/(p-1)$. If $p = 1$, then $q = \infty$ and vice versa. Then Hölder's inequality [4, p. 113] states that if $f \in L_p(\Omega, P)$ and $g \in L_q(\Omega, P)$ where p and q are conjugate indices, then the product $fg \in L_1(\Omega, P)$, and

$$\|fg\|_1 \leq \|f\|_p \cdot \|g\|_q, \text{ or } \int_\Omega |f(\omega)g(\omega)| P(d\omega) \leq \left[\int_\Omega |f(\omega)|^p P(d\omega) \right]^{1/p} \cdot \left[\int_\Omega |g(\omega)|^q P(d\omega) \right]^{1/q}. \tag{8.14}$$

In particular, choosing $p = q = 2$ leads to Schwarz' inequality, namely, if $f, g \in L_2(\Omega, P)$, then $fg \in L_1(\Omega, P)$, and

$$\|fg\|_1 \leq \|f\|_2 \cdot \|g\|_2, \text{ or } \int_\Omega |f(\omega)g(\omega)| P(d\omega) \leq \left[\int_\Omega |f(\omega)|^2 P(d\omega) \right]^{1/2} \cdot \left[\int_\Omega |g(\omega)|^2 P(d\omega) \right]^{1/2}. \tag{8.15}$$

[3] Or open sets, or semi-open sets–they all generate the same σ-algebra.

[4] Note that some authors also use the phrase "expectation" to mean "expected value." In such a case, this phrase will be doing double duty, first to denote the *real number* defined above, and second to denote the *random variable* defined in Definition 8.9 below. This dual usage is by now pervasive in the probability theory literature.

By using Hölder's inequality and the fact that $P(\Omega) = 1$, it is easy to show that

$$L_q(\Omega, P) \subseteq L_p(\Omega, P) \text{ whenever } p < q. \tag{8.16}$$

In particular, if a real random variable X is square-integrable, it is also absolutely integrable, and thus has a well-defined mean. Moreover, with $\mu := E[X, P]$, we can write

$$\int_\Omega [X(\omega) - \mu]^2 P(d\omega) = \int_\Omega X^2(\omega) P(d\omega) - \mu^2 =: V(X, P),$$

often called the "variance" of X.

In the discussion below, we often deal with two random variables X and X' that differ only on a set of measure zero, that is,

$$P\{\omega : X(\omega) \neq X'(\omega)\} = 0.$$

In such a case, we write $X = X'$ a.e., or $X = X'$ a.s..

The concept of a conditional expectation is defined next.

Definition 8.9. (See [11, Definition 4.16] or [48, Section 9.2].) Suppose $(\Omega, \mathcal{F}, P)$ is a probability space, and that X is a real random variable with the additional property that $X \in L_1(\Omega, \mathcal{F}, P)$. Suppose that $\mathcal{G} \subseteq \mathcal{F}$ is another σ-algebra on Ω. Then the **conditional expectation** of X with respect to $\mathcal{G}$, denoted by $E(X|\mathcal{G})$, is any random variable Y such that (i) Y is measurable with respect to $(\Omega, \mathcal{G})$, and (ii)

$$\int_D X(\omega) P(d\omega) = \int_D Y(\omega) P(d\omega), \ \forall D \in \mathcal{G} \tag{8.17}$$

Any two conditional expectations $E(X|\mathcal{G})$ agree almost surely, and each is called a "version" of the conditional expectation.

Note that $E(X|\mathcal{G})$ is a $(\Omega, \mathcal{G})$-measurable approximation to X such that, when restricted to sets in $\mathcal{G}$, $E(X|\mathcal{G})$ is functionally equivalent to X, as stated in (8.17). Note that (8.17) can also be expressed as

$$\int_\Omega X(\omega) I_D(\omega) P(d\omega) = \int_\Omega Y(\omega) I_D(\omega) P(d\omega), \ \forall D \in \mathcal{G}$$

where $I_D(\cdot)$ is the indicator function of the set D.

To make the discussion below easier to follow, we employ the notation $Y \in \mathcal{M}(\mathcal{G})$ to indicate that Y maps Ω into $\mathbb{R}$, and is measurable with respect to $(\Omega, \mathcal{G})$ and $(\mathbb{R}, \mathcal{B})$.

In the above definition, it is not clear that such a conditional expectation exists. Any Y that satisfies (8.17) is called a "version" in [48, 14]. The next theorem summarizes, without proof, some key properties of the conditional expectation. These details can be found in [48, Chapter 9] and/or [14, Section 4.1].

Theorem 8.1. *Suppose $X \in L_1(\Omega, \mathcal{F}, P)$ and that $\mathcal{G} \subseteq \mathcal{F}$ is another σ-algebra on Ω. Then*

1. *(Existence) There is at least one $Y \in \mathcal{M}(\mathcal{G})$ such that (8.17) holds.*
2. *(Uniqueness) If $Y, Y' \in \mathcal{M}(\mathcal{G})$ both satisfy (8.17), then $Y(\omega) = Y'(\omega)$ a.s..*
3. *(Expected Value Preservation) Every conditional expectation $Y = E(X|\mathcal{G})$ belongs to $L_1(\Omega, \mathcal{F}, P)$. Moreover*
$$E[Y, P] = E[X, P], \text{ or } \int_\Omega Y(\omega) P(d\omega) = \int_\Omega X(\omega) P(d\omega). \tag{8.18}$$
4. *(Self-Replication) If $X \in \mathcal{M}(\mathcal{G})$, then $E(X|\mathcal{G}) = X$ a.s..*
5. *(Idempotency) If p, q are conjugate indices, and $Z \in L_q(\Omega, \mathcal{G}, P)$, $X \in L_p(\Omega, \mathcal{F}, P)$, then*
$$E[(ZX)|\mathcal{G}] = ZE(X|\mathcal{G}) \text{ a.s..} \tag{8.19}$$

6. *(Iterated Conditioning) If $\mathcal{H} \subseteq \mathcal{G} \subseteq \mathcal{F}$ are σ-algebras, then*

$$E[E(X|\mathcal{G})|\mathcal{H}] = E(X|\mathcal{H}). \tag{8.20}$$

7. *(Linearity) If $X_1, X_2 \in L_1(\Omega, \mathcal{F}, P)$ and $a_1, a_2 \in \mathbb{R}$, then*

$$E[(a_1X_1 + a_2X_2)|\mathcal{G}] = a_1E(X_1|\mathcal{G}) + a_2E(X_2|\mathcal{G}) \text{ a.s..} \tag{8.21}$$

8. *(Nonnegativity) If $X(\omega) \geq 0$ a.s., then $E(X|\mathcal{G})(\omega) \geq 0$ a.s..*

9. *(Projection Property) If $X \in L_2(\Omega, \mathcal{F}, P)$ (and not just $L_1(\Omega, \mathcal{F}, P)$), then*

$$E(X|\mathcal{G}) = \underset{Y \in L_2(\Omega, \mathcal{G}, P)}{\arg\min} \|Y - X\|_2^2 \text{ a.s..} \tag{8.22}$$

Now we interpret some of the statements in the theorem. The obvious ones are not discussed. Item 3 states that the expected value of the conditional expectation is the same as the expected value of the original random variable.[5] Item 5 states that if X is multiplied by a *bounded* random variable $Z \in \mathcal{M}(\mathcal{G})$,[6] then the multiplier Z just passes through the conditional expectation operation. Item 6 states that if we were to first take the conditional expectation of X with respect to $\mathcal{G}$, and then take the conditional expectation of the resulting random variable with respect to a smaller σ-algebra $\mathcal{H}$, then the answer would be the same as if we had directly taken the conditional expectation with respect to $\mathcal{H}$. Note that this property is called the "tower property" on [48, p. 88]. A ready consequence of Items 7 and 8 is that, if $X_1 \geq X_2$ almost surely, then $E(X_1|G) \geq E(X_2|G)$ almost surely. Finally, Item 9 states that if X belongs to the smaller space $L_2(\Omega, \mathcal{F}, P)$ which is an inner product space, then its conditional expection also belongs to $L_2(\Omega, \mathcal{F}, P)$, and can be computed using the projection theorem.

Example 8.3. In this example, we illustrate the concept of a conditional expectation in a very simple case, namely, that of a random variable assuming only finitely many values. Suppose $\mathcal{X} = \{x_1, \cdots, x_n\}$ and $\mathcal{V} = \{v_1, \cdots, v_m\}$ are finite sets, and that $Z = (X, V)$ is a joint random variable assuming values in $\mathcal{X} \times \mathcal{V}$. Let $\Theta \in [0,1]^{n \times m}$ denote the joint probability distribution of Z written out as a matrix, and let $\boldsymbol{\phi}, \boldsymbol{\psi}$ denote the marginal probability distributions of X and V respectively, written out as row vectors. Finally, suppose $f : \mathcal{X} \times \mathcal{V} \to \mathbb{R}$ is a given function. Then $f(Z)$ is a real-valued random variable assuming values in some finite set.

Because both X and V are finite-valued, we can use the canonical representation, and choose $\Omega = \mathcal{X} \times \mathcal{V}$, $\mathcal{F} = 2^\Omega$, and $P = \Theta$. Now suppose we define $\mathcal{G}$ to be the σ-algebra generated by V alone. Thus $\mathcal{G} = \{\emptyset, \mathcal{X}\} \otimes 2^\mathcal{V}$. Again, because $f(X, V)$ assumes only finitely many values over a finite set, it is a bounded random variable. Therefore $E(f|\mathcal{G})$ is the best approximation to $f(X, V)$ using a function of V alone. From Item 9 of the theorem this conditional expectation can be determined using projections.

As pointed out after Definition 8.8, it can be assumed without loss of generality that every component of $\boldsymbol{\psi}$ is positive. Therefore the ratio

$$\frac{\theta_{ij}}{\psi_j} = \Pr\{X = x_i | V = v_j\}$$

is well-defined, though it could be zero.

In order to determine $E(f|\mathcal{G})$, we should find a function $g : \mathcal{V} \to \mathbb{R}$ such that the error $E[(f-g)^2, \Theta]$ is minimized. Let $g_1, \cdots, g_m$ denote the values of $g(\cdot)$, and define the objective function

$$J = \frac{1}{2} \sum_{j=1}^{m} \sum_{i=1}^{n} (g_j - f_{ij})^2 \theta_{ij}.$$

[5] To streamline the notation wherever possible, we write $E[X, P]$ to denote the expected value, which is a real number, and $E(X|\mathcal{G})$ to denote the conditional expectation, which is a random variable.

[6] Hereafter we follow the probabilists' convention and say "bounded" when we actually mean "bounded except on a set of measure zero," that is, "essentially bounded."

Then the objective is to choose the constants $g_1, \cdots, g_m$ so as to minimize J. This happens when

$$0 = \frac{\partial J}{\partial g_j} = \sum_{i=1}^{n} (g_j - f_{ij})\theta_{ij}.$$

This expression can be rewritten as

$$0 = g_j \sum_{i=1}^{n} \theta_{ij} - \sum_{i=1}^{n} f_{ij}\theta_{ij} = g_j \psi_j - \sum_{i=1}^{n} f_{ij}\theta_{ij},$$

or

$$g_j = \sum_{i=1}^{n} f_{ij} \frac{\theta_{ij}}{\psi_j} = E[f(X,V)|V = v_j].$$

This formula explains the terminology "conditional expectation." g_j equals the expected value of f conditioned on the event that $V = v_j$. The same expression also shows that

$$E[g, \boldsymbol{\psi}] = \sum_{j=1}^{m} g_j \psi_j = \sum_{j=1}^{m} \sum_{i=1}^{n} f_{ij}\theta_{ij} = E[f, \Theta].$$

For future use, we introduce definitions of what it means for a sequence of real-valued random variables to converge. Three commonly used notions of convergence are convergence probability, almost sure convergence, and convergence in the mean. All are defined here.

Definition 8.10. Suppose $\{X_n\}_{n \geq 0}$ is a sequence of real-valued random variables, and X^* is a real-valued random variable, on a common probability space $(\Omega, \mathcal{F}, P)$. Then the sequence $\{X_n\}_{n \geq 0}$ is said to converge to X^* **in probability** if

$$P(\{\omega \in \Omega : |X_n(\omega) - X^*(\omega)| > \epsilon\}) \to 0 \text{ as } n \to \infty, \ \forall \epsilon > 0. \tag{8.23}$$

The sequence $\{X_n\}_{n \geq 0}$ is said to converge to X^* **almost surely** (or almost everywhere) if

$$P(\{\omega \in \Omega : X_n(\omega) \to X^*(\omega) \text{ as } n \to \infty\}) = 1. \tag{8.24}$$

Now suppose that $X_n, X^* \in L_1(\Omega, P)$. Then the sequence $\{X_n\}_{n \geq 0}$ is said to converge to X^* **in the mean** if

$$\|X_n - X^*\|_1 \to 0 \text{ as } n \to \infty. \tag{8.25}$$

Note that, for any $p \in (1, \infty)$, "convergence in the p-th mean" can be defined in the space $L_p(\Omega, P)$, as $\|X_n - X^*\|_p \to 0$ as $n \to \infty$. Also, the extension of Definition 8.10 to random variables assuming values in a vector space $\mathbb{R}^d$ is obvious and is left to the reader.

The relationship between the various types of convergence is as follows:

Theorem 8.2. *Suppose $\{X_n\}, X^*$ are random variables defined on some probability space $(\Omega, \mathcal{F}, P)$. Suppose $X_n \to X^*$ in probability as $n \to \infty$. Then every subsequence of $\{X_n\}$ contains a subsequence that converges almost surely to X^*.*

The converse of Theorem 8.2 is also true. See [4, Section 21, Problem 10(c)].

Theorem 8.3. *Suppose $\{X_n\}, X^* \in L_1(\Omega, P)$. Then*

1. *$X_n \to X^*$ a.s. implies that $X_n \to X^*$ in probability.*
2. *$X_n \to X^*$ in the mean implies that $X_n \to X^*$ in probability*
3. *Suppose there is a nonnegative random variable $Z \in L_1(\Omega, P)$ such that $|X_n| \leq Z$ a.e., and suppose that $X_n \to X^*$ a.s.. Then $X_n \to X^*$ in the mean.*

These statements also apply to $\mathbb{R}^d$-valued random variables.

Problem 8.1. Show that a consequence of Definition 8.1 is that $\emptyset \in \mathcal{F}$.

Problem 8.2. Show that a consequence of Definition 8.2 is that $P(\emptyset) = 0$.

8.2 Markov processses

In this section, we introduce the concept of Markov processes, which plays a central role in Reinforcement Learning. As a prelude, we introduce the concept of a stochastic process.

Definition 8.11. Suppose $(\Omega, \mathcal{F}, P)$ is a probability space, and that $(\mathcal{X}, \mathcal{G})$ is a measure space. A **stochastic process** on $(\mathcal{X}, \mathcal{G})$ is a sequence of random variables $\{X_t\}_{t\geq 0}$ where each X_t takes values in $\mathcal{X}$.

For each finite index T, let X_0^T denote the tuple $(X_0, \ldots, X_T)$. We can view this is as a random variable taking values in the $(T+1)$-fold product $(\prod_{i=0}^T \mathcal{X}, \otimes_{i=0}^T \mathcal{G})$, with its own probability distribution $P_{X_0^T}$. In principle, in order to talk about stochastic processes precisely, we should define the *infinite* product, but this leads to too many technicalities. So we avoid that. As a result, there are minor imprecisions in the discussion below. Two types of stochastic processes make their appearence in Reinforcement Learning, namely: Those where X_t takes its values in some finite set, which could be an abstract set of labels, and those where $X_t \in \mathbb{R}^d$ for some integer d. In this section, we deal only with stochastic processes where the "alphabet" $\mathcal{X}$, which is the set to which X_t belongs, is some finite set.

8.2.1 Markov Processes: Basic Properties

Suppose $\mathcal{X}$ is a set of finite cardinality, say $\mathcal{X} = \{x_1, \cdots, x_n\}$, and suppose that $\{X_t\}_{t\geq 0}$ is a stochastic process assuming values in $\mathcal{X}$, that is, $\{X_t\}_{t\geq 0}$ is a sequence of random variables assuming values in $\mathcal{X}$. Let the symbol X_0^t denote the (finite) collection of random variables $(X_0, \cdots, X_t)$.

Definition 8.12. The process $\{X_t\}_{t\geq 0}$ is said to **possess the Markov property**, or to be a **Markov process**, if

$$\Pr\{X_{t+1}|X_0^t\} = \Pr\{X_{t+1}|X_t\},\ \forall t \geq 0. \tag{8.26}$$

Because all random variables assume values in the finite set $\mathcal{X}$, we can make the abstract equation (8.26) more explicit. Equation (8.26) is a shorthand for the following statement: Suppose $u \in \mathcal{X}$ and $(y_0, \cdots, y_t) \in \mathcal{X}^{t+1}$ are arbitrary. Then (8.26) is equivalent to

$$\Pr\{X_{t+1} = u|X_0^t = (y_0, \cdots, y_t)\} = \Pr\{X_{t+1} = u|X_t = y_t\},\ \forall u \in \mathcal{X}, (y_0, \cdots, y_t) \in \mathcal{X}^{t+1}.$$

In other words, the conditional probability of the state X_{t+1} depends only on the most recent value of X_t; adding information about the past values of X_τ for $\tau < t$ does not change the conditional probability. One can also say that X_{t+1} is independent of X_0^{t-1} given X_t. This property is sometimes paraphrased as "the future is conditionally independent of the past, given the present."

A Markov process over a finite set $\mathcal{X}$ is completely characterized by its **state transition matrix** A, where

$$a_{ij} := \Pr\{X_{t+1} = x_j|X_t = x_i\},\ \forall x_i, x_j \in \mathcal{X}.$$

Thus in a_{ij}, i denotes the current state and j the future state. The reader is cautioned that some authors interchange the roles of i and j in the above definition. If the transition probability does not depend on t, then the Markov process is said to be **stationary**; otherwise it is said to be **nonstationary**. We do not deal with nonstationary Markov processes in these notes.

Note that $a_{ij} \in (0, 1)$ for all i, j. Also, at any time $t+1$, it must be the case that $X_{t+1} \in \mathcal{X}$, no matter what X_t is. Therefore, the sum of each row of A equals one, i.e.,

$$\sum_{j=1}^n a_{ij} = 1, i = 1, \ldots n. \tag{8.27}$$

The above equation can be expressed compactly as

$$A\mathbf{1}_n = \mathbf{1}_n, \tag{8.28}$$

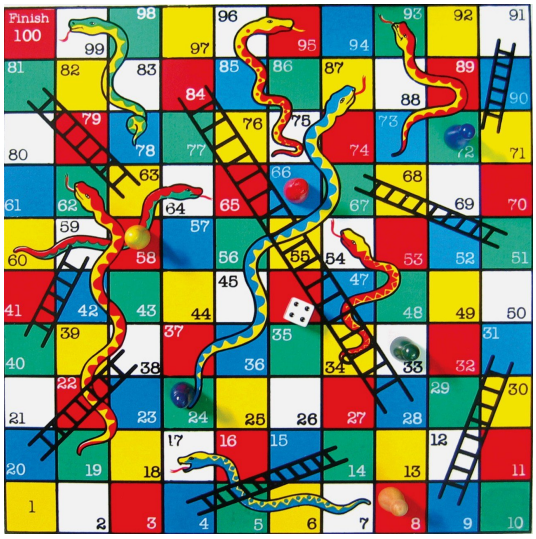


Figure 8.1: Snakes and Ladders Game

where $\mathbf{1}_n$ denotes the column vector consisting of n ones. For future purposes, let us refer to a matrix $A \in [0,1]^{n\times n}$ that satisfies (8.27) as a **row-stochastic matrix**, and denote by $\mathbb{S}_{n\times n}$ the set of all row-stochastic matrices of dimension $n \times n$.

The matrix A is often called the "one-step" transition matrix, because row i of A gives the probablity distribution of X_{t+1} if $X_t = x_i$. So we can ask: What is the k-step transition matrix? In other words, what is the probability distribution of X_{t+k} if $X_t = x_i$? It is not difficult to show that this conditional probability is just the i-row of A^k. Thus the k-step transition matrix is just A^k.

Example 8.4. A good example of a Markov process is the "snakes and ladders" game. Take for example the board shown in Figure 8.1. In this case, we can let X_t denote an integer between 1 and 100, corresponding to the square on which the player is. Thus $\mathcal{X} = \{1, \cdots, 100\}$. Suppose the player throws a four-sided die with each of the outcomes $(1, 2, 3, 4)$ being equally probable. Then the resulting sequence of positions $\{X_t\}_{t\geq 0}$ is a stochastic process. Suppose for example that the player is on square 60. Note that what happens next after a player has reached square 60 (or any other square) *does not depend on how the player reached that square.* That is why the sequence of positions is a Markov process. Now, if the player is on square 60 so that $X_t = 60$, then with probability of $1/4$, the position at time $t+1$ will be 61, 19 (snake on 62), 81 (ladder on 63) and 60 (snake on 64). Hence, in row 60 of the 100×100 state transition matrix, there are elements of $1/4$ in columns $19, 60, 61, 81$ and zeros in the remaining 96 columns. In the same manner, the entire 100×100 state transition matrix can be determined.

Let us suppose that the snakes and ladders game always starts with the player being in square 1. Thus X_0 is not random, but is deterministic, and the "probability distribution" of X_0, viewed as a row vector, has a 1 in column 1 and zeros elsewhere. If we multiply this row vector by A^k for any integer k, we get the probability distribution of the player's position after k moves.

An application of the Gerschgorin circle theorem [18, Theorem 6.1.1] shows that, whenever A is row-stochastic, the spectral radius $\rho(A) \leq 1$. Moreover, the relationship (8.27) shows that $\lambda = 1$ is an eigenvalue of A with column eigenvector $\mathbf{1}_n$, so that in fact $\rho(A) = 1$. Thus one can ask: What does the row eigenvector corresponding to $\lambda = 1$ look like? If there is a *nonnegative* row eigenvector $\boldsymbol{\mu} \in \mathbb{R}^n_+$, then it can be scaled so that $\boldsymbol{\mu}\mathbf{1}_n = 1$. Such a $\boldsymbol{\mu}$ is called a **stationary distribution** of the Markov process, because if X_t has the probability distribution $\boldsymbol{\mu}$, then so does X_{t+1}. More generally, if X_0 has the probability distribution $\boldsymbol{\mu}$, then so does X_t for all $t \geq 0$.

Theorem 8.4. *(See [5, Theorem 3.2, p. 8].) Every row-stochastic matrix A has a nonnegative row eigenvector corresponding to the eigenvalue $\lambda = 1$.*

Note that Theorem 8.4 is a very weak statement. It states only that there exists *a* stationary distribution; nothing is said about whether this is unique or not. To proceed further, it is helpful to make some assumptions about A.

Definition 8.13. A row-stochastic matrix A is said to be **irreducible** if it is not possible to partition the permute the rows and columns symmetrically (via a permutation matrix Π) such that

$$\Pi^{-1} A \Pi = \begin{bmatrix} B_{11} & 0 \\ B_{21} & B_{22} \end{bmatrix}.$$

Thus a row-stochastic matrix is irreducible if it is not possible to turn it into a block-triangular matrix through symmetric row and column permutations. The notion of irreducibility plays a crucial role in the theory of Markov processes. So it is worthwhile to give an alternate characterization of irreducibility.

Lemma 8.1. *A row-stochastic matrix A is irreducible if and only if, for any pair of states $y_s, y_f \in \mathcal{X}$, there exists a sequence of states $y_1, \cdots y_l \in \mathcal{X}$ such that, with $y_0 = y_s$ and $y_{l+1} = y_f$, we have that*

$$a_{y_k y_{k+1}} > 0, k = 0, \ldots, l.$$

Thus the matrix A is irreducible if and only if, for every pair of states y_s and y_f, there is a path from y_s to y_f such that every step in the path has a positive probability. In such a case we can say that y_f is reachable from y_s. There are several equivalent characterizations of irreducibility, and for nonnegative matrices in general, not necessarily satisfying (8.27); see [46, Chapter 3]. Another useful reference is [5], which is devoted entirely to the study of nonnegative matrices. One such characterization is given next.

Theorem 8.5. *(See [46, Corollary 3.8].) A row-stochastic matrix A is irreducible if and only if*

$$\sum_{l=0}^{n-1} A^l > 0,$$

where $A^0 = I$ and the inequality is componentwise.

So we can start with $M_0 = I$ and define recursively $M_{l+1} = I + AM_l$. If $M_l > 0$ for any l, then A is irreducible. If we get up to M_{n-1} and this matrix is not strictly positive, then A is not irreducible.

Theorem 8.6. *(See [46, Theorem 3.25].) Suppose A is an irreducible row-stochastic matrix. Then*

1. *$\lambda = 1$ is a simple eigenvalue of A.*
2. *The corresponding row eigenvector of A has all positive elements.*
3. *Thus A has a unique stationary distribution, whose elements are all positive.*
4. *There is an integer p, called the **period** of A, such that the spectrum of A is invariant under rotation by $\exp(\mathbf{i}2\pi/p)$.*
5. *In particular, $\exp(\mathbf{i}2l\pi/p)$, $l = 0, \cdots, p-1$ are all eigenvalues of A.*

Now we introduce a concept that is stronger than irreducibility.

Definition 8.14. A row-stochastic matrix A is said to be **primitive** if there exists an integer l such that $A^l > 0$.

Definition 8.15. An irreducible row-stochastic matrix A is said to be **aperiodic** if $\lambda = 1$ is the only eigenvalue of A with magnitude one.

Theorem 8.7. *(See [46, Theorem 3.15].) A row-stochastic matrix A is primitive if and only if it is irreducible and aperiodic.*

Example 8.5. Suppose

$$A_1 = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Then A_1 is primitive, while A_2 is irreducible but not primitive; it has a period $p = 3$.

In some situations, the following result is useful.

Theorem 8.8. *(See [46, Lemma 4.12].) Suppose A is an irreducible row-stochastic matrix, and let $\boldsymbol{\mu}$ denote the corresponding stationary distribution. Then*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} A^t = \mathbf{1}_n \boldsymbol{\mu}. \tag{8.29}$$

Therefore, the average of $I, A, \cdots, A^{T-1}$ approaches the rank one matrix $\mathbf{1}_n\boldsymbol{\mu}$. So, if $\boldsymbol{\phi}$ is *any* probability distribution on $\mathcal{X}$, and the Markov process is started off with the initial distribution $\boldsymbol{\phi}$, then the distribution of the state X_t is $\boldsymbol{\phi}A^t$. Note that, because $\boldsymbol{\phi}$ is a probability distribution, we have that $\boldsymbol{\phi}\mathbf{1}_n = 1$. Therefore (8.45) implies that

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\phi}A^t = \boldsymbol{\phi}\mathbf{1}_n\boldsymbol{\mu} = \boldsymbol{\mu}, \ \forall \boldsymbol{\phi}. \tag{8.30}$$

The above relationship holds for *every* $\boldsymbol{\phi}$ and forms the basis for the so-called **Markov chain Monte Carlo (MCMC)** algorithm. Suppose $\{X_t\}_{t\geq 0}$ is a Markov process evolving over the state space $\mathcal{X}$, with an irredcible state transition matrix A and stationary distribution $\boldsymbol{\mu}$. Suppose further that $f : \mathcal{X} \to \mathbb{R}$ is a real-valued function defined on the state space $\mathcal{X}$. We wish to compute the expected value of the random variable $f(X_t)$ with respect to the stationary distribution $\boldsymbol{\mu}$, namely

$$E[f(X), \boldsymbol{\mu}] = \sum_{x_i \in X} f(x_i)\mu_i. \tag{8.31}$$

While we may know A, often we may not know $\boldsymbol{\mu}$ or may not wish to spend the effort to compute it due to the high dimension of A. In such a case, we start off the Markov process with an arbitrary initial probability distribution $\boldsymbol{\phi}$, let it run for some time t_0, and then compute the quantity

$$\hat{f}_T = \frac{1}{T} \sum_{t=t_0+1}^{t_0+T} f(X_t). \tag{8.32}$$

Because this quantity is based on the observed state X_t which is random, $\hat{f}_T$ is also random. However, the expected value of $\hat{f}_T$ is precisely $E[f(X), \boldsymbol{\mu}]$. Moreover, its sample-path average $\hat{f}_T$ converges to $E[f(X), \boldsymbol{\mu}]$ as $T \to \infty$, and is a good approximation for the expected value for finite T.

The next result is analogous to Theorem 8.8 for primitive matrices.

Theorem 8.9. *(See [46, Corollary 4.13].) Suppose A is a primitive row-stochastic matrix, and let $\boldsymbol{\mu}$ denote the corresponding stationary distribution. Then*

$$A^l \to \mathbf{1}_n^\top \boldsymbol{\mu} \text{ as } l \to \infty. \tag{8.33}$$

Now we prove a couple of useful lemmas about irreducible and primitive matrices respectively. These are useful when we study so-called Markov Decision Processes.

Theorem 8.10. *Suppose A is a nontrivial convex combination of row stochastic matrices $A_1, \cdots, A_k$, and that at least one A_i is irreducible. Then A is irreducible.*

Without loss of generality, write

$$A = \sum_{i=1}^{k} \gamma_i A_i,$$

where $\gamma_1 > 0$ and A_1 is irreducible. Then

$$A^l \geq \gamma_1^l A_1^l \ \forall l,$$

where the inequality holds componentwise, because all other "cross-product" terms in the expansion of A^l are nonnegative matrices. Because A_1 is irreducible, it follows from Theorem 8.5 that

$$\sum_{l=0}^{n-1} A_1^{n-1} > 0,$$

where again the inequality is componentwise. Combining this with the above inequality shows that

$$\sum_{l=0}^{n-1} A^l \geq \sum_{l=0}^{n-1} \gamma_1^l A_1^{n-1} \geq \gamma_1^{n-1} \sum_{l=0}^{n-1} A_1^{n-1} > 0.$$

Therefore A is irreducible.

Corollary 8.1. *The set of irreducible matrices is convex.*

Theorem 8.11. *Suppose A is a nontrivial convex combination of row stochastic matrices $A_1, \cdots, A_k$, and that at least one A_i is primitive. Then A is primitive.*

The proof is similar to that of Theorem 8.9, except that Theorem 8.5 is replaced by Definition 8.14.

Corollary 8.2. *The set of primitive matrices is convex.*

8.2.2 Stopping Times and Hitting Probabilities

The contents of this subsection are very useful in Section 4.1.1, when we study reinforcement learning using "episodes."

Definition 8.16. A state $x_i \in \mathcal{X}$ is said to be an **absorbing state** if $X_t = x_i$ implies that $X_{t+1} = x_i$, or equivalently, that $X_\tau = x_i$ for all $\tau \geq t$. Another equivalent defintion is that row i of the state transtion matrix A consists of a 1 in column i and zeros elsewhere. More generally, a subset $\mathcal{S} \subseteq \mathcal{X}$ is said to be a **set of absorbing states** if $X_t \in \mathcal{S} \implies X_\tau \in \mathcal{S}$ for all $\tau > t$.

Now we illustrate the concepts of absorbing states, and of absorbing sets. For convenience, we change notation slightly. Assume that the state space $\mathcal{X}$ of a Markov process can be partitioned as $\mathcal{T} \cup \mathcal{S}$, where $\mathcal{T}$ denotes the set of "transient" states, and $\mathcal{S}$ is an absorbing set. Suppose further that $\mathcal{T} = \{x_1, \cdots, x_m\}$, and $\mathcal{S} = \{a_1, \ldots, a_s\}$. It is a ready consequence of Definition 8.16 that the state transition matrix M of the Markov process has the form (note the change in notation):

$$M = \begin{bmatrix} A & B \\ 0 & C \end{bmatrix}, \tag{8.34}$$

where $C \in \mathbb{S}_{s \times s}$ is a row stochastic matrix in itself, and the matrix B has at least one nonzero element. Note too that the set S can be absorbing, even if no individual state in S is absorbing. For example, suppose C

is a permutation matrix over s indices. However, if $C = I_s$, the identity matrix, then not only is the set S absorbing, but every individual state in S is absorbing. In this case the matrix M looks like

$$M = \begin{bmatrix} A & B \\ 0 & I_s \end{bmatrix}. \tag{8.35}$$

An illustration of an absorbing state is provided by the snakes and ladders game. If the player's position hits 100, then the game is over. So 100 is an absorbing state. In other games like Blackjack, there are *two* absorbing states, namely W and L (for win and lose). In the Markov process literature, any sample path X_0^l such that X_l is an absorbing state is called an **episode**.

It can be shown that if the state X_t of the Markov process enters the absorbing set S with probability one as $t \to \infty$, then $B \neq 0$, that is, B contains at least one nonzero element, and further, $\rho(A) < 1$. See specifically Items 3 and 6 of [46, Theorem 4.7]. More details can be found in [46, Section 4.2.2]. (Note that notation in [46] is different.) For the purposes of RL, it is useful to go beyond these facts, and to compute the probability distribution of the time at which the state trajectory enters S. In turn this gives the average number of time steps needed to reach the absorbing set. In case there are multiple absorbing states, it is also possible to compute the probability of hitting an individual absorbing state a_i within the overall absorbing set $\mathcal{S}$. To be specific, define θ_{iS} to be the *first time* that a sample path $\{X_0^\infty\}$ hits the set S, starting at $X_0 = x_i$. Further, if M is of the form (8.35) so that each set in S is absorbing, define θ_{ik} to be the first time that a sample path $\{X_0^\infty\}$ hits the absorbing state a_k, starting at $X_0 = x_i$. Then we have the following result:

Theorem 8.12. *With the above notation, we have that*

$$\Pr\{\theta_{iS} = l\} = \mathbf{e}_i^\top A^{l-1} B \mathbf{1}_s \ \forall l \geq 1, \tag{8.36}$$

where $\mathbf{e}_i$ denotes the i-th elementary column vector with a 1 in row i and zeros elsewhere. If M has the form (8.35), the for each $k \in [s]$, we have

$$\Pr\{\theta_{ik} = l\} = \mathbf{e}_i^\top A^{l-1} \mathbf{b}_k \ \forall l \geq 1 \tag{8.37}$$

where $\mathbf{b}_k$ denotes the k-th column of B. The probability that a sample path X_0^∞ with $X_0 = x_i$ terminates in the absorbing state a_k is given by

$$p_{ik} = \mathbf{e}_i^\top (I - A)^{-1} \mathbf{b}_k. \tag{8.38}$$

Moreover,

$$\sum_{k=1}^{s} p_{ik} = 1, \ \forall i \in [m].$$

The vector of probabilities that a sample path X_0^∞ terminates in the absorbing state a_k is given by

$$\mathbf{p}_k = (I - A)^{-1} \mathbf{b}_k. \tag{8.39}$$

For each transient initial state $x_i \in \mathcal{T}$, define the average hitting time to reach the absorbing set S starting from the initial state x_i to be the expected value of θ_{iS}, that is

$$\bar{\theta}_{iS} = \sum_{l=1}^{\infty} l \Pr\{\theta_{iS} = l\},$$

and the vector of average hitting times as $\bar{\boldsymbol{\theta}}_S \in \mathbb{R}^m$. Then

$$\bar{\boldsymbol{\theta}}_S = (I - A)^{-1} B \mathbf{1}_s. \tag{8.40}$$

Proof. We begin by deriving the expressions for the probability distributions. For each pair of indices $i, j \in [m]$ and each integer l, the value $(A^l)_{ij}$ is the probability that, starting in state x_i at time $t = 0$, the state at time l equals x_j, while staying within the set $\mathcal{X}$. Thus the probability that $\theta_{iS} = l$ is given by

$$\Pr\{\theta_{iS} = l\} = \sum_{j=1}^{m} (A^{l-1})_{ij} (B\mathbf{1}_s)_j = \mathbf{e}_i^\top A^{l-1} B\mathbf{1}_s.$$

This is (8.36). If S consists of individual absorbing states, and we wish to determine the probability distribution that $X_l = a_k$ given that $X_0 = x_i$, then we simply replace $B\mathbf{1}_s$ by the corresponding k-th column of B. This is (8.37). Equation (8.38) is obtained by observing that, since $\rho(A) < 1$, we have that

$$\sum_{l=1}^{\infty} A^{l-1} = (I - A)^{-1}.$$

Therefore the probability that a trajectory starting at x_i terminates in state a_k is given by

$$\sum_{l=1}^{\infty} \mathbf{e}_i^\top A^{l-1} \mathbf{b}_k = \mathbf{e}_i \left[\sum_{l=1}^{\infty} A^{l-1} \right] \mathbf{b}_k = \mathbf{e}_i (I - A)^{-1} \mathbf{b}_k.$$

This is (8.38). Stacking these probabilities as i varies over $[m]$ gives (8.39).

Next we deal with the hitting times. Define the vector $\mathbf{b} = B\mathbf{1}_s$, and consider the modified Markov process with the state transition matrix

$$M = \begin{bmatrix} A & \mathbf{b} \\ 0 & 1 \end{bmatrix}.$$

In effect, we have aggregated the set of absorbing states into one "virtual state." From the standpoint of computing $\bar{\boldsymbol{\theta}}$, this is permissible, because once the trajectory hits the set S, or the virtual "last state" in the modified formulation, the time counter stops. To prove (8.40), suppose the Markov process starts in state x_i. Then there are two possibilities: First, with probability b_i, the trajectory hits the last virtual state. In this case the counter stops, and we can say that the hitting time is 1. Second, with probability a_{ij} for each j, the trajectory hits the state x_j. In this case, the hitting time is now $1 + \bar{\theta}_j$. Therefore we have

$$\bar{\theta}_i = b_i + \sum_{j=1}^{n} a_{ij}(1 + \bar{\theta}_j).$$

Observe however that

$$b_i = 1 - \sum_{j=1}^{n} a_{ij}.$$

Substituting in the previous equation gives

$$\bar{\theta}_i = 1 + \sum_{j=1}^{n} a_{ij} \bar{\theta}_j,$$

or in matrix form

$$(I - A)\bar{\boldsymbol{\theta}} = \mathbf{1}_m.$$

Clearly this is equivalent to (8.40). □

Example 8.6. Consider the "toy" snakes and ladders game with two extra states, called W and L for win and lose respectively. The rules of the game are as follows:

- Initial state is S.

- A four-sided, fair die is thrown at each stage.
- Player must land exactly on W to win and exactly on L to lose.
- If implementing a move causes crossing of W and L, then the move is not implemented.

There are twelve possible states in all: S, 1, ... , 9 , W, L. However, 2, 3, 9 can be omitted, leaving nine states, namely S, 1, 4, 5, 6, 7, 8, W, L. At each step, there are at most four possible outcomes. For example, from the state S, the four outcomes are 1, 7, 5, 4. From state 6, the four outcomes are 7, 8, 1, and W. From state 7, the four outcomes are 8, 1, W, 7. From state 8, there four possible outcomes are 1, W, L and 8 with probability 1/4 each, because if the die comes up with 4, then the move cannot be implemented. It is time-consuming but straight-forward to compute the state transition matrix as

	S	1	4	5	6	7	8	W	L
S	0	0.25	0.25	0.25	0	0.25	0	0	0
1	0	0	0.25	0.50	0	0.25	0	0	0
4	0	0	0	0.25	0.25	0.25	0.25	0	0
5	0	0.25	0	0	0.25	0.25	0.25	0	0
6	0	0.25	0	0	0	0.25	0.25	0.25	0
7	0	0.25	0	0	0	0	0.25	0.25	0.25
8	0	0.25	0	0	0	0	0.25	0.25	0.25
W	0	0	0	0	0	0	0	1	0
L	0	0	0	0	0	0	0	0	1

The average duration of a game, which is the expected time before hitting one of the two absorbing states W or L, is given by (8.40), and is

$$\boldsymbol{\theta} = \begin{bmatrix} 5.5738 \\ 5.4426 \\ 4.7869 \\ 4.9180 \\ 3.9344 \\ 3.1475 \\ 3.1475 \end{bmatrix}.$$

To compute the probabilities of reaching the absorbing states W or L from any nonabsorbing state, define A to be the 7×7 submatrix on the top left, and B to be the 7×2 submatrix on the top left. Then the probabilities of hitting W and L are given by (8.39), and are given by

$$[P_W \;\; P_L] = (I - A)^{-1} B = \begin{bmatrix} 0.5433 & 0.4567 \\ 0.5457 & 0.4543 \\ 0.5574 & 0.4426 \\ 0.5550 & 0.4450 \\ 0.6440 & 0.3560 \\ 0.5152 & 0.4848 \\ 0.5152 & 0.4848 \end{bmatrix}.$$

Not surprisingly, the two columns add up to one in each row, showing that, irrespective of the starting state, the sample path with surely hit either W or L. Also not surprisngly, the probability of hitting W is maximum in state 6, because it is possible to win in one throw of the die, but impossible to lose in one throw.

Problem 8.3. Suppose the last rule of the toy snakes and ladders game is modified as follows: If implementing a move causes the player to go past L, then the player moves to L (and loses the game). With this modification, compute the average length of a game and the probability of winning / losing from each state.

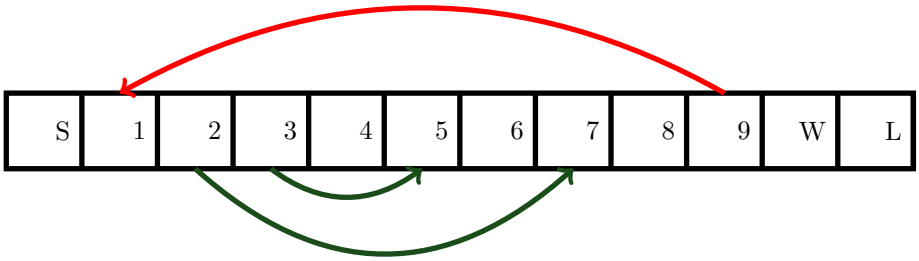


Figure 8.2: Toy Snakes and Ladders Game

8.2.3 Maximum Likelihood Estimate of Markov Processes

Suppose $\{X_t\}_{t\geq 0}$ is a Markov process evolving over a finite state space (or alphabet) $\mathcal{X} = \{x_1, \ldots, x_n\}$, with an unknown state transition matrix. We are able to observe a sample path $y_0^l := \{y_0, y_1, \ldots, y_l\}$ of the process, where each $y_i \in \mathcal{X}$. From this observation, we wish to determine *the most likely* state transition matrix A, that is, the matrix A that maximizes the likelihood of the observed sample path. As it turns out, the solution is very simple.

Suppose A is a row-stochastic matrix. In other words, $A \in [0,1]^{n\times n}$ and satisfies

$$\sum_{j=1}^{n} a_{ij} = 1, i = 1, \ldots, n. \tag{8.41}$$

For a given sample path y_0^l, the likelihood that this sample path is generated by a Markov process with state transition matrix A is given by

$$\begin{aligned} L(y_0^l|A) &= \Pr\{y_0\} \prod_{t=1}^{l} Pr\{X_t = y_t | X_{t-1} = y_{t-1}, A\} \\ &= \Pr\{y_0\} \prod_{t=1}^{l} a_{y_t - 1 y_t}. \end{aligned} \tag{8.42}$$

The formula becomes simpler if we take the logarithm of the above. Clearly, maximizing the log-likelihood of observing y_0^l is equivalent to maximizing the likelihood of observing y_0^l. Thus

$$LL(y_0^l|A) = \log \Pr\{y_0\} + \sum_{t=1}^{l} \log a_{y_t - 1 y_t}. \tag{8.43}$$

A further simplification is possible. For each pair $(x_i, x_j) \in \mathcal{X}^2$, let ν_{ij} denote the number of times that the string $x_i x_j$ occurs (in that order) in the sample path y_0^l. Next, define

$$\bar{\nu}_i := \sum_{j=1}^{n} \nu_{ij}. \tag{8.44}$$

It is easy to see that, instead of summing over strings $y_{t-1}y_t$, we can sum over strings $x_i x_j$. Thus $y_{t-1}y_t = x_i x_j$ precisely ν_{ij} times. Therefore

$$LL(y_0^l|A) = \log \Pr\{y_0\} + \sum_{i=1}^{n} \sum_{j=1}^{n} \nu_{ij} \log a_{ij}. \tag{8.45}$$

We can ignore the first term as it does not depend on A. Also, A needs to satisfy the stochasticity constraint (8.28). So we want to maximize the right side of (8.45) (without the term $\log \Pr\{y_0\}$) subject to (8.41). For this purpose we form the Lagrangian

$$J = \sum_{i=1}^{n}\sum_{j=1}^{n} \nu_{ij} \log a_{ij} + \sum_{i=1}^{n} \lambda_i \left(1 - \sum_{j=1}^{n} a_{ij}\right),$$

where $\lambda_1, \ldots, \lambda_n$ are the Lagrange multipliers. Next, observe that

$$\frac{\partial J}{\partial a_{ij}} = \frac{\nu_{ij}}{a_{ij}} - \lambda_i.$$

Setting the partial derivatives to zero gives

$$\lambda_i = \frac{\nu_{ij}}{a_{ij}}, \text{ or } a_{ij} = \frac{\nu_{ij}}{\lambda_i}.$$

The value of λ_i can be determined from (8.41), which gives

$$\sum_{j=1}^{n} a_{ij} = \frac{1}{\lambda_i} \sum_{j=1}^{n} \nu_{ij} = \frac{\bar{\nu}_i}{\lambda_i} = 1 \implies \lambda_i = \bar{\nu}_i.$$

Therefore the maximum likelihood estimate for the state transition matrix of a Markov process, based on the sample path y_0^l, is given by

$$a_{ij} = \frac{\nu_{ij}}{\bar{\nu}_i}. \tag{8.46}$$

8.3 Contraction Mapping Theorem

In this section we introduce a very powerful theorem known as the contraction mapping theorem (also known as the Banach fixed point theorem), which provides an iterative technique for solving noninear equations. It holds in extremely general settings. We present a version that is sufficient for the present purposes.

Theorem 8.13. *Suppose $f : \mathbb{R}^n \to \mathbb{R}^n$ and that there exists a constant $\rho < 1$ such that*

$$\|f(x) - f(y)\| \leq \rho \|x - y\|, \ \forall x, y \in \mathbb{R}^n, \tag{8.47}$$

where $\|\cdot\|$ on $\mathbb{R}^n$. Then there is a unique $x^* \in \mathbb{R}^n$ such that

$$f(x^*) = x^*. \tag{8.48}$$

To find x^*, choose an arbitrary $x_0 \in \mathbb{R}^n$ and define $x_{l+1} = f(x_l)$. Then $\{x_l\} \to x^*$ as $l \to \infty$. Moreover, we have the explicit estimate

$$\|x^* - x_l\| \leq \frac{\rho^l}{1-\rho} \|x_1 - x_0\|. \tag{8.49}$$

Proof. By definition, we have that

$$\|x_{l+1} - x_l\| \leq \rho \|x_- x_{l-1}\| \leq \cdots \leq \rho^l \|x_1 - x_0\|. \tag{8.50}$$

Suppose $m > l$, say $m = l + r$ with $r > 0$. Then

$$\begin{aligned} \|x_m - x_l\| &= \|x_{l+r} - x_l\| \leq \sum_{i=0}^{r-1} \|x_{l+i+1} - x_{l+i}\| \\ &\leq \sum_{i=0}^{r-1} \rho^{l+i} \|x_1 - x_0\| \\ &\leq \sum_{i=0}^{\infty} \rho^{l+i} \|x_1 - x_0\| \\ &= \frac{\rho^l}{1-\rho} \|x_1 - x_0\|. \end{aligned} \tag{8.51}$$

Therefore $\|x_m - x_l\| \to 0$ as $\min\{m, l\} \to \infty$. Such a sequence is called a **Cauchy sequence**. In $\mathbb{R}^n$, a Cauchy sequence always converges to a limit. Denote this limit by x^*. Then $x^* = \lim_{l\to\infty} x_l$. Now (8.47) makes it clear that the function f is continuous. Therefore

$$f(x^*) = \lim_{l\to\infty} f(x_l) = \lim_{l\to\infty} x_{l+1} = x^*.$$

Therefore x^* satisfies (8.48). To show that x^* is unique, suppose $f(y^*) = y^*$. Then it follows from (8.47) that

$$\|x^* - y^*\| = \|f(x^*) - f(y^*)\| \leq \rho \|x^* - y^*\|.$$

Since $\rho < 1$, the only way in which the above inequality can hold is if $\|x^* - y^*\| = 0$, i.e., if $x^* = y^*$. Finally, let $m \to \infty$ in (8.51) so that $x_m \to x^*$ and $\|x_m - x_l\| \to \|x^* - x_l\|$. Then (8.51) becomes (8.49). □

The bound (8.49) is extremely useful. Note that $\|x_1 - x_0\| = \|f(x_0) - x_0\|$. Therefore $\|x_1 - x_0\|$ is a measure of how far off the initial guess x_0 is from being a fixed point of f. Then (8.49) gives an *explicit* estimate of how far x^* is from x_l, for each iteration x_l. Note that the bound on the right side of (8.49) decreases by a factor of ρ at each iteration.

8.4 Some Elements of Lyapunov Stability Theory

Bibliography

[1] A. Arapostathis, V. S. Borkar, E. Fernández-Gaucherand, M. K. Ghosh, and S. I. Marcus. Discrete-time controlled Markov processes with average cost criterion: A survey. *SIAM Journal of Control and Optimization*, 31(2):282–344, 1993.

[2] P. Auer, N. Cesi-Bianchi, and F. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.

[3] A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximation*. Springer-Verlag, 1990.

[4] S. K. Berbarian. *Measure and Integration*. Chelsea, 1965.

[5] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, 1979.

[6] V. S. Borkar. *Probability Theory: An Advanced Course*. Springer-Verlag, 1995.

[7] V. S. Borkar. Asynchronous stochastic approximations. *SIAM Journal on Control and Optimization*, 36(3):840–851, 1998.

[8] V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

[9] V. S. Borkar and S. P. Meyn. The O.D.E. method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38:447–469, 2000.

[10] R. J. Boucherie and N. M. van Dijk, editors. *Markov Decision Processes in Practice*. Springer Nature, 2017.

[11] L. Breiman. *Probability*. SIAM: Society for Industrial and Applied Mathematics, 1992.

[12] Deep Mind. Alphazero: Shedding new light on chess, shogi, and go. https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go.

[13] C. Derman and J. Sacks. On dvoretzky's stochastic approximation theorem. *Annals of Mathematical Statistics*, 30(2):601–606, 1959.

[14] R. Durrett. *Probability: Theory and Examples (5th Edition)*. Cambridge University Press, 2019.

[15] A. Dvoretzky. On stochastic approximation. In *Proceedings of the Third Berkeley Symposium on Mathematical Statististics and Probability*, volume 1, pages 39–56. University of California Press, 1956.

[16] E. G. Gladyshev. On stochastic approximation. *Theory of Probability and Its Applications*, X(2):275–278, 1965.

[17] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer-Verlag, Berlin and Heidelberg, 2001.

[18] R. A. Horn and C. R. Johnson. *Matrix Analysis (Second Edition)*. Cambridge University Press, 2013.

[19] ImageNet. http://image-net.org/about-stats, 2010.

[20] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23(3):462–466, 1952.

[21] A. N. Kolmogorov. *Foundations of Probability (Second English Edition)*. Chelsea, 1950. (English translation by Kai Lai Chung).

[22] V. Konda and J. Tsitsiklis. On actor-critic algorithms. *SIAM Journal on Control and Optimization*, 42(4):1143–1166, 2003.

[23] V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *Neural Information Processing Systems (NIPS1999)*, pages 1008–1014, 1999.

[24] H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Applied Mathematical Sciences. Springer-Verlag, 1978.

[25] H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, 1997.

[26] T. L. Lai. Stochastic approximation (invited paper). *The Annals of Statistics*, 31(2):391–406, 2003.

[27] K. R. Parthasarathy. *Probability Measures on Metric Spaces*. Academic Press, 1967.

[28] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley, 2005.

[29] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[30] C. E. Shannon. Programming a computer for playing chess. *Philosophical Magazine, Ser.7*, 41(314), March 1950.

[31] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, December 2018.

[32] S. P. Singh and R. S. Sutton. Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22(1–3):123–158, 1996.

[33] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction (Second Edition)*. MIT Press, 2018.

[34] R. S. Sutton, D. McAllester, S. Singh, and Y.Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12 (Proceedings of the 1999 conference)*, pages 1057–1063. MIT Press, 2000.

[35] C. Szepesvári. *Algorithms for Reinforcement Learning*. Morgan and Claypool, 2010.

[36] G. Tesauro. Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6(2):215–219, 1994.

[37] G. Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[38] G. Tesauro. Programming backgammon using self-teaching neural nets. *Artificial Intelligence*, 134(1-2):181–199, 2002.

[39] J. N. Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine Learning*, 16:185–202, 1994.

[40] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, September 1986.

[41] J. N. Tsitsiklis and B. V. Roy. Feature-based methods for large-scale dynamic programming. *Machine Learning*, 22:59–94, 1996.

[42] J. N. Tsitsiklis and B. V. Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, May 1997.

[43] J. N. Tsitsiklis and B. V. Roy. Average cost temporal-difference learning. *Automatica*, 35:1799–1808, 1999.

[44] M. Vidyasagar. *Nonlinear Systems Analysis (SIAM Classics Series)*. Society for Industrial and Applied Mathematics (SIAM), 2002.

[45] M. Vidyasagar. *Learning and Generalization: With Applications to Neural Networks and Control Systems*. Springer-Verlag, 2003.

[46] M. Vidyasagar. *Hidden Markov Processes: Theory and Applications to Biology*. Princeton University Press, 2014.

[47] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

[48] D. Williams. *Probability with Martingagles*. Cambridge University Press, 1991.

[49] J. Wolfowitz. On the stochastic approximation method of Robbins and Monro. *Annals of Mathematical Statistics*, 23(3):457–461, 1952.

[50] M. Zastow. I'm in shock!: How an AI beat the world's best human at go, 2016.