CS234 Notes - Lecture 2
Making Good Decisions Given a Model of the World

Rahul Sarkar, Emma Brunskill

March 20, 2018

3 Acting in a Markov decision process

We begin this lecture by recalling the definitions of a **model**, **policy** and **value function** for an agent. Let the agent's state and action spaces be denoted by S and A respectively. We then have the following definitions:

- **Model** : A model is the mathematical description of the dynamics and rewards of the agent's environment, which includes the transition probabilities $P(s'|s,a)$ of being in a successor state $s' \in S$ when starting from a state $s \in S$ and taking an action $a \in A$, and the rewards $R(s,a)$ (either deterministic or stochastic) obtained by taking an action $a \in A$ when in a state $s \in S$.
- **Policy** : A policy is a function $\pi : S \to A$ that maps the agent's states to actions. Policies can be stochastic or deterministic.
- **Value function** : The value function V^π corresponding to a particular policy π and for a state $s \in S$, is the cumulative sum of future (discounted) rewards obtained by the agent, by starting from the state s and following the policy.

We also recall the notion of **Markov property** from the last lecture. Consider a stochastic process $(s_0, s_1, s_2, \dots)$ evolving according to some transition dynamics. We say that the stochastic process has the Markov property if and only if $P(s_i|s_0, \dots, s_{i-1}) = P(s_i|s_{i-1})$, $\forall\ i = 1, 2, \dots$, i.e. the transition probability of the next state conditioned on the history including the current state is equal to the transition probability of the next state conditioned only on the current state. In such a scenario, the current state is a sufficient statistic of history of the stochastic process, and we say that *"the future is independent of the past given present."*

In this lecture, we will build on these definitions and proceed in order by first defining a **Markov process (MP)**, followed by the definition of a **Markov reward process (MRP)** and finally build on both of them to define a **Markov decision process (MDP)**. We will finish this lecture by discussing some algorithms which enable us to make good decisions when a MDP is completely known.

3.1 Markov process

In its most generality, a Markov process is a stochastic process that satisfies the Markov property, because of which we say that a Markov process is *"memoryless"*. For the purpose of this lecture, we will make two additional assumptions that are very common in the reinforcement learning setting:

- *Finite state space* : The state space of the Markov process is finite. This means that for the Markov process $(s_0, s_1, s_2, \dots)$, there is a state space S with $|S| < \infty$, such that for all realizations of the Markov process, we have $s_i \in S$ for all $i = 1, 2, \dots$.

- *Stationary transition probabilities* : The transition probabilities are time independent. Mathematically, this means the following:

$$P(s_i = s' | s_{i-1} = s) = P(s_j = s' | s_{j-1} = s) \ , \ \forall \, s, s' \in S \ , \ \forall \, i, j = 1, 2, \dots \ . \tag{1}$$

Unless otherwise specified, we will always assume that these two properties hold for any Markov process that we will encounter in this lecture, including for any Markov reward process and any Markov decision process to be defined later by adding progressively extra structure to the Markov process. Note that a Markov process satisfying these assumptions is also sometimes called a *"Markov chain"*, although the precise definition of a Markov chain varies.

For the Markov process, these assumptions lead to a nice characterization of the transition dynamics in terms of a *transition probability matrix* $\mathbf{P}$ of size $|S| \times |S|$, whose (i, j) entry is given by $P_{ij} = P(j|i)$, with i, j referring to the states of S ordered arbitrarily. It should be noted that the matrix $\mathbf{P}$ is a non-negative row-stochastic matrix, i.e. the sum of each row equals 1.

Henceforth, we will thus define a Markov process by the tuple $(S, \mathbf{P})$, which consists of the following:

- S : A finite state space.
- $\mathbf{P}$: A transition probability model that specifies $P(s'|s)$.

Exercise 3.1. (a) Prove that $\mathbf{P}$ is a row-stochastic matrix. (b) Show that 1 is an eigenvalue of any row-stochastic matrix, and find a corresponding eigenvector. (c) Show that any eigenvalue of a row-stochastic matrix has maximum absolute value 1.

Exercise 3.2. The *max-norm* or *infinity-norm* of a vector $x \in \mathbb{R}^n$ is denoted by $||x||_\infty$, and defined as $||x||_\infty = \max_i |x_i|$, i.e. it is the component of x with the maximum absolute value. For any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, define the following quantity

$$||\mathbf{A}||_\infty = \sup_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \frac{||\mathbf{A}x||_\infty}{||x||_\infty} \ . \tag{2}$$

(a) Prove that $||\mathbf{A}||_\infty$ satisfies all the properties of a norm. The quantity so defined is called the *"induced infinity norm"* of a matrix.

(b) Prove that

$$||\mathbf{A}||_\infty = \max_{i=1,\dots,m} \left(\sum_{j=1}^{n} |A_{ij}| \right) \ . \tag{3}$$

(c) Conclude that if $\mathbf{A}$ is row-stochastic, then $||\mathbf{A}||_\infty = 1$.

(d) Prove that for every $x \in \mathbb{R}^n$, $||\mathbf{A}x||_\infty \leq ||\mathbf{A}||_\infty ||x||_\infty$.

3.1.1 Example of a Markov process : Mars Rover

To practice our understanding, consider the Markov process shown in Figure 1. Our agent is a Mars rover whose state space is given by $S = \{S1, \ S2, \ S3, \ S4, \ S5, \ S6, \ S7\}$. The transition probabilities of the states are indicated in the figure with arrows. So for example if the rover is in the state $S4$ at

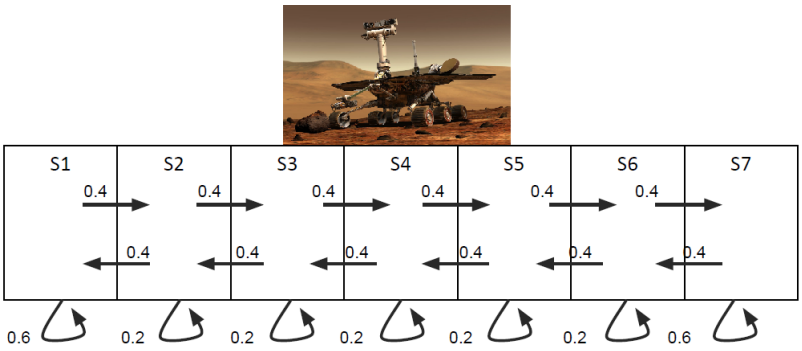

Figure 1: Mars Rover Markov process.

the current time step, in the next time step it can go to the states $S3,\ S4,\ S5$ with probabilities given by $0.4,\ 0.2,\ 0.4$ respectively.

Assuming that the rover starts out in state $S4$, some possible episodes of the Markov process could look as follows:

- $S4,\ S5,\ S6,\ S7,\ S7,\ S7,\ \ldots$
- $S4,\ S4,\ S5,\ S4,\ S5,\ S6,\ \ldots$
- $S4,\ S3,\ S2,\ S1,\ \ldots$

Exercise 3.3. Consider the example of a Markov process given in Figure 1. (a) Write down the transition probability matrix for the Markov process.

3.2 Markov reward process

A Markov reward process is a Markov process, together with the specification of a reward function and a discount factor. It is formally represented using the tuple $(S, \mathbf{P}, R, \gamma)$ which are listed below:

- S : A finite state space.
- $\mathbf{P}$: A transition probability model that specifies $P(s'|s)$.
- R : A reward function that maps states to rewards (real numbers), i.e $R : S \to \mathbb{R}$.
- γ: Discount factor between 0 and 1.

We have already explained the roles played by S and $\mathbf{P}$ in the context of a Markov process. We will next explain the concept of the reward function R and the discount factor γ, which are specific to the Markov reward process. Additionally, we will also define and explain a few quantities which are important in this context, such as the horizon, return and state value function of a Markov reward process.

3.2.1 Reward function

In a Markov reward process, whenever a transition happens from a current state s to a successor state s', a reward is obtained depending on the current state s. Thus for the Markov process $(s_0, s_1, s_2, \ldots)$, each transition $s_i \to s_{i+1}$ is accompanied by a reward r_i for all $i = 0, 1, \ldots$, and so a particular episode

of the Markov reward process is represented as $(s_0, r_0, s_1, r_1, s_2, r_2, \dots)$. We should note that these rewards can be either deterministic or stochastic. For a state $s \in S$, we define the expected reward $R(s)$ by:

$$R(s) = \mathbb{E}[r_0 | s_0 = s], \tag{4}$$

that is $R(s)$ is the expected reward obtained during the first transition, when the Markov process starts in state s. Just like the assumption of stationary transition probabilities, going forward we will also assume the following:

- *Stationary rewards* : The rewards in a Markov reward process are stationary which means that they are time independent. In the deterministic case, mathematically this means that for all realizations of the process we must have that:

$$r_i = r_j \text{ , whenever } s_i = s_j \ \ \forall\, i, j = 0, 1, \dots \text{ ,} \tag{5}$$

  while in the case of stochastic rewards we require that the cumulative distribution functions (cdf) of the rewards conditioned on the current state be time independent. This is written mathematically as:

$$F(r_i | s_i = s) = F(r_j | s_j = s) \ \ , \ \ \forall\, s \in S \ \ , \ \ \forall\, i, j = 0, 1, \dots \text{ ,} \tag{6}$$

  where $F(r_i | s_i = s)$ denotes the cdf of r_i conditioned on the state $s_i = s$. Notice that as a consequence of (5) and (6), we furthermore have the following result about the expected rewards:

$$R(s) = \mathbb{E}[r_i | s_i = s] \ \ , \ \ \forall\, i = 0, 1, \dots \text{ .} \tag{7}$$

We will see that as long as the "stationary rewards" assumption is true about a Markov reward process, only the expected reward R matters in the things that we will be interested in, and we can depose of the quantities r_i entirely. Hence going forward, the word "reward" will be used interchangeably to mean both R and r_i, and should be easily understood from context. Finally notice that R can be represented as a vector of dimension $|S|$, in the case of a finite state space S.

Exercise 3.4. (a) Under the assumptions of stationary transition probabilities and rewards, prove equation (7).

3.2.2 Horizon, Return and Value function

We next define the notions of the horizon, return and value function for a Markov reward process.

- **Horizon** : The horizon H of a Markov reward process is defined as the number of time steps in each episode (realization) of the process. The horizon can be finite or infinite. If the horizon is finite, then the process is also called a *finite Markov reward process*.
- **Return** : The return G_t of a Markov reward process is defined as the discounted sum of rewards starting at time t up to the horizon H, and is given by the following mathematical formula:

$$G_t = \sum_{i=t}^{H-1} \gamma^{i-t} r_i \ \ , \ \ \forall\, 0 \leq t \leq H-1. \tag{8}$$

- **State value function** : The state value function $V_t(s)$ for a Markov reward process and a state $s \in S$ is defined as the expected return starting from state s at time t, and is given by the following expression:

$$V_t(s) = \mathbb{E}[G_t | s_t = s]. \tag{9}$$

  Notice that when the horizon H is infinite, this definition (9) together with the stationary assumptions of the rewards and transition probabilities imply that $V_i(s) = V_j(s)$ for all $i, j = 0, 1, \dots$, and thus in this case we will define:

$$V(s) = V_0(s) \text{ .} \tag{10}$$

Exercise 3.5. (a) If the assumptions of stationary transition probabilities and stationary rewards hold, and if the horizon H is infinite, then using the definitions in (8) and (9) prove that $V_i(s) = V_j(s)$ for all $i, j = 0, 1, \ldots$.

3.2.3 Discount factor

Notice that in the definition of return G_t in (8), if the horizon is infinite and $\gamma = 1$, then the return can become infinite even if the rewards are all bounded. If this happens, then the value function $V(s)$ can also become infinite. Such problems cannot then be solved using a computer. To avoid such mathematical difficulties and make the problems computationally tractable we set $\gamma < 1$, which exponentially weighs down the contribution of rewards at future times, in the calculation of the return in (8). This quantity γ is called the *discount factor*. Other than for purely computational reasons, it should be noted that humans behave in much the same way - we tend to put more importance in immediate rewards over rewards obtained at a later time. The interpretation of γ is that when $\gamma = 0$, we only care about the immediate reward, while when $\gamma = 1$, we put as much importance on future rewards as compared the present. Finally, notice that if the horizon of the Markov reward process is finite, i.e. $H < \infty$, then we can set $\gamma = 1$, as the returns and value functions are always finite.

Exercise 3.6. Consider a finite horizon Markov reward process, with bounded rewards. Specifically assume that $\exists\ M \in (0, \infty)$ such that $|r_i| \leq M\ \ \forall\ i$ and across all episodes (realizations). (a) Show that the return for any episode G_t as defined in (8) is bounded. (b) Can you suggest a bound? Specifically can you find $C(M, \gamma, t, H)$ such that $|G_t| \leq C$ for any episode?

Exercise 3.7. Consider an infinite horizon Markov reward process, with bounded rewards and $\gamma < 1$. (a) Prove that the return for any episode G_t as defined in (8) converges to a finite limit. *Hint: Consider the partial sums* $S_N = \sum_{i=t}^{N} \gamma^{i-t} r_i$ *for* $N \geq t$. *Show that* $\{S_N\}_{N \geq t}$ *is a Cauchy sequence.*

3.2.4 Example of a Markov reward process : Mars Rover

As an example, consider the Markov reward process in Figure 2. The states and the transition probabilities of this process are exactly the same as in the Mars rover Markov process example of Exercise 3.3. The rewards obtained by executing an action from any of the states $\{S2,\ S3,\ S4,\ S5,\ S6\}$ is 0, while any moves from states $S1,\ S7$ yield rewards 1, 10 respectively. The rewards are stationary and deterministic. Assume $\gamma = 0.5$ in this example.

For illustration, let us again assume that the rover is initially in state $S4$. Consider the case when the horizon is finite : $H = 4$. A few possible episodes in this case with the return G_0 in each case are given below:

- $S4,\ S5,\ S6,\ S7,\ S7\ : G_0 = 0 + 0.5 * 0 + 0.5^2 * 0 + 0.5^3 * 10 = 1.25$
- $S4,\ S4,\ S5,\ S4,\ S5\ : G_0 = 0 + 0.5 * 0 + 0.5^2 * 0 + 0.5^3 * 0 = 0$
- $S4,\ S3,\ S2,\ S1,\ S2\ : G_0 = 0 + 0.5 * 0 + 0.5^2 * 0 + 0.5^3 * 1 = 0.125$

3.3 Computing the value function of a Markov reward process

In this section we give three different ways to compute the value function of a Markov reward process:

- Simulation
- Analytic solution
- Iterative solution

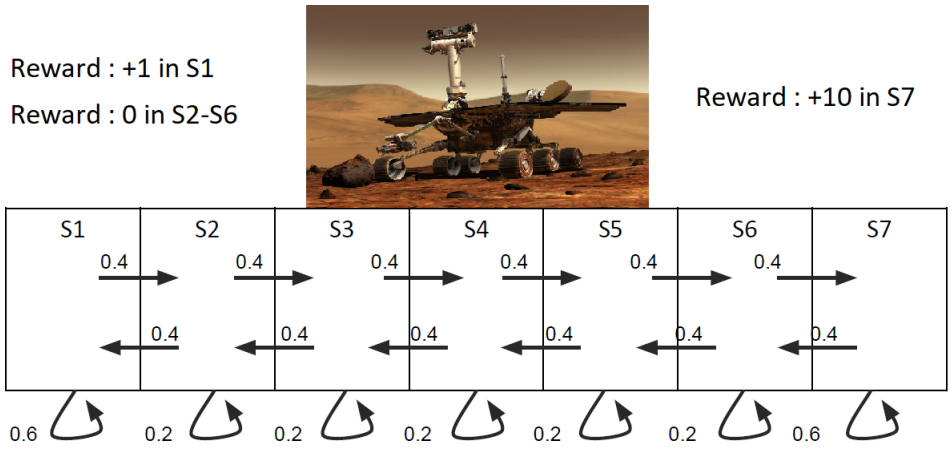

Figure 2: Mars Rover Markov reward process.

3.3.1 Monte Carlo simulation

The first method involves generating a large number of episodes using the transition probability model and rewards of the Markov reward process. For each episode, the returns can be calculated which can then be averaged to give the average returns. Concentration inequalities bound how quickly the averages concentrate to the mean value. For a Markov reward process $M = (S, \mathbf{P}, R, \gamma)$, state s, time t, and the number of simulation episodes N, the pseudo-code of the simulation algorithm is given in Algorithm 1.

Algorithm 1 Monte Carlo simulation to calculate MRP value function

procedure MONTE CARLO EVALUATION(M, s, t, N)
  $i \leftarrow 0$
  $G_t \leftarrow 0$
  while $i \neq N$ **do**
    Generate an episode, starting from state s and time t
    Using the generated episode, calculate return $g \leftarrow \sum_{i=t}^{H-1} \gamma^{i-t} r_i$
    $G_t \leftarrow G_t + g$
    $i \leftarrow i + 1$
  $V_t(s) \leftarrow G_t / N$
  return $V_t(s)$

3.3.2 Analytic solution

This method works only for an infinite horizon Markov reward processes with $\gamma < 1$. Using (9), the fact that the horizon is inifnite, and using the stationary Markov property we have for any state $s \in S$:

$$
\begin{aligned}
V(s) \overset{(a)}{=} V_0(s) = \mathbb{E}[G_0 | s_0 = s] = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r_i \middle| s_0 = s\right] = \mathbb{E}[r_0 | s_0 = s] + \sum_{i=1}^{\infty} \gamma^i \mathbb{E}[r_i | s_0 = s] \\
\overset{(b)}{=} \mathbb{E}[r_0 | s_0 = s] + \sum_{i=1}^{\infty} \gamma^i \left(\sum_{s' \in S} P(s_1 = s' | s_0 = s) \mathbb{E}[r_i | s_0 = s, s_1 = s'] \right) \\
\overset{(c)}{=} \mathbb{E}[r_0 | s_0 = s] + \gamma \sum_{s' \in S} P(s'|s) \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r_i \middle| s_0 = s'\right] \overset{(d)}{=} R(s) + \gamma \sum_{s' \in S} P(s'|s) V(s') \ ,
\end{aligned}
\tag{11}
$$

where (a) follows from (8), (9), and (10), (b) follows by the law of total expectation, (c) follows from the Markov property and due to stationarity, and (d) follows from (4). There is a nice interpretation of the final result of (11), namely that the first term $R(s)$ is the immediate reward while the second term $\gamma \sum_{s' \in S} P(s'|s)V(s')$ is the discounted sum of future rewards. The value function $V(s)$ is the sum of these two quantities. As $|S| < \infty$, it is possible to write this equation in matrix form as:

$$V = R + \gamma \mathbf{P} V \ , \tag{12}$$

where $\mathbf{P}$ is the transition probability matrix introduced earlier, and R and V are column vectors of dimension $|S|$ formed by stacking all the values $R(s)$ and $V(s)$ respectively, for all $s \in S$. Equation (12) can be rearranged to give $(\mathbf{I} - \gamma \mathbf{P})V = R$, which has an analytical solution $V = (\mathbf{I} - \gamma \mathbf{P})^{-1} R$. Notice that as $\gamma < 1$ and $\mathbf{P}$ is row-stochastic, $(\mathbf{I} - \gamma \mathbf{P})$ is non-singular and hence can be inverted. Thus (12) always has a solution and the solution is unique. However, the computational cost of the analytical method is $O(|S|^3)$, as it involves a matrix inverse and hence it is completely unsuitable for cases where the state space is very large.

Exercise 3.8. Consider the matrix $(\mathbf{I} - \gamma \mathbf{P})$. (a) Show that $1 - \gamma$ is an eigenvalue of this matrix, and find a corresponding eigenvector. (b) For $0 < \gamma < 1$, use the result of Exercise 3.1 to conclude that $(\mathbf{I} - \gamma \mathbf{P})$ is non-singular, and thus invertible.

Exercise 3.9. Consider the Markov reward process introduced in the example in section 3.2.4. (a) If the horizon H is infinite, calculate the value function for all the states.

3.3.3 Iterative solution

We now give an iterative solution to evaluate the value function in the infinite horizon case (with $\gamma < 1$) and a dynamic programming based solution for the finite horizon case. The surprising thing is that both the algorithms look surprisingly similar, to the point that it is hard to tell the difference. We first consider the finite horizon case. It is easy to prove (by following almost exactly the same proof of (11)) that the analog of equation (11) in the finite horizon case is given by:

$$\begin{aligned} V_t(s) &= R(s) + \gamma \sum_{s' \in S} P(s'|s) V_{t+1}(s') \ \ , \ \ \forall \, t = 0, \ldots, H-1, \\ V_H(s) &= 0 \ \ . \end{aligned} \tag{13}$$

Exercise 3.10. Prove equations (13) for a finite horizon Markov reward process.

These equations immediately lend themselves to a dynamic programming solution whose pseudo-code is outlined in Algorithm 2. The algorithm takes as input a finite horizon Markov reward process $M = (S, \mathbf{P}, R, \gamma)$, and computes the value function for all states and at all times.

Algorithm 2 Dynamic programming algorithm to calculate finite MRP value function

procedure DYNAMIC PROGRAMMING VALUE FUNCTION EVALUATION(M)
    For all states $s \in S$, $V_H(s) \leftarrow 0$
    $t \leftarrow H - 1$
    while $t \geq 0$ **do**
        For all states $s \in S$, $V_t(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s) V_{t+1}(s')$
        $t \leftarrow t - 1$
    return $V_t(s)$ for all $s \in S$ and $t = 0, \ldots, H$

Let us now look at the iterative algorithm for the infinite horizon case with $\gamma < 1$. The pseudo-code for this algorithm is presented in Algorithm 3. The algorithm takes as input a Markov reward process $M = (S, \mathbf{P}, R, \gamma)$, and a tolerance ϵ, and computes the value function for all states.

Algorithm 3 Iterative algorithm to calculate MRP value function

procedure ITERATIVE VALUE FUNCTION EVALUATION(M, ϵ)
  For all states $s \in S$, $V'(s) \leftarrow 0$, $V(s) \leftarrow \infty$
  while $||V - V'||_\infty > \epsilon$ **do**
    $V \leftarrow V'$
    For all states $s \in S$, $V'(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s)V(s')$
  return $V'(s)$ for all $s \in S$

For both these algorithms 2 and 3, the computational cost of each loop is $O(|S|^2)$. This is an improvement over the $O(|S|^3)$ cost of the analytical method in the inifinite horizon case, however one may need quite a few iterations to converge depending on the tolerance level ϵ.

While the proof of correctness of algorithm 2 in the finite horizon case is obvious, for the infinite horizon case it is not so clear if algorithm 3 always converges, and if it does whether it converges to the correct solution $(\mathbf{I} - \gamma \mathbf{P})^{-1} R$. The answers to both these questions are affirmative as is shown by the following theorem.

Theorem 3.1. *Algorithm 3 always terminates. Moreover, if the output of the algorithm is V' and we denote the true solution as $V = (\mathbf{I} - \gamma \mathbf{P})^{-1} R$, then we have the error estimate $||V' - V||_\infty \leq \frac{\epsilon\gamma}{1-\gamma}$.*

Proof. We consider the vector space $\mathbb{R}^{|S|}$ equipped with the $||\cdot||_\infty$ norm (see Exercise 3.2), and recall that $\mathbb{R}^{|S|}$ so constructed is a Banach space (see Section A for a discussion on normed vector spaces). We start by noticing that both V and all the iterates of algorithm 3 are elements of $\mathbb{R}^{|S|}$.

Define the operator $B : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ *(also known as the "Bellman backup" operator)* that acts on an element $U \in \mathbb{R}^{|S|}$ as follows

$$(BU)(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s)U(s') \quad , \quad \forall\, s \in S, \tag{14}$$

which can be written in compact matrix-vector notation as

$$BU = R + \gamma \mathbf{P} U \ . \tag{15}$$

We first prove that the operator B is a strict contraction (defined in Definition A.3). For every $U_1, U_2 \in \mathbb{R}^{|S|}$, using (15) we have

$$\begin{aligned} ||BU_1 - BU_2||_\infty &= \gamma ||\mathbf{P}U_1 - \mathbf{P}U_2||_\infty = \gamma ||\mathbf{P}(U_1 - U_2)||_\infty \\ &\leq \gamma ||\mathbf{P}||_\infty ||U_1 - U_2||_\infty = \gamma ||U_1 - U_2||_\infty \ , \end{aligned} \tag{16}$$

where the second step follows by Exercise 3.2, and thus as $0 < \gamma < 1$, we conclude that B is a strict contraction on $\mathbb{R}^{|S|}$. Thus by the contraction mapping theorem (Theorem A.5), we conclude that B has a unique fixed point. From (15) and (12) it also follows that $BV = R + \gamma \mathbf{P} V = V$, and hence V is a fixed point of B, and hence by uniqueness it must also be the only fixed point.

We next consider the iterates produced by algorithm 3 (if it is not allowed to terminate) and denote them by $\{V_k\}_{k \geq 1}$. Notice that these iterates satisfy the following relations

$$V_k = \begin{cases} 0 & \text{if } \ k = 1, \\ BV_{k-1} & \text{if } \ k > 1 \end{cases} \ . \tag{17}$$

By Theorem A.5, we further conclude that $\{V_k\}_{k \geq 1}$ is a Cauchy sequence, and hence by Definition A.1 we conclude that $\exists\ N \geq 1$, such that $||V_m - V_n||_\infty < \epsilon$ for all $m, n > N$. This completes the proof that algorithm 3 terminates. Notice that the contraction mapping theorem (Theorem A.5) also implies that $V_k \to V$ (see Definition A.2 for exact notion of convergence).

To prove the error bound when the algorithm terminates, let the algorithm terminate after k iterations, and so the last iterate is V_{k+1}. We then have $||V_{k+1} - V_k||_\infty \leq \epsilon$. Then using the triangle inequality and the fact that $V_{k+1} = BV_k$ we get,

$$\begin{aligned} ||V_k - V||_\infty &\leq ||V_k - V_{k+1}||_\infty + ||V_{k+1} - V||_\infty = ||V_k - V_{k+1}||_\infty + ||BV_k - BV||_\infty \\ &\leq ||V_k - V_{k+1}||_\infty + \gamma||V_k - V||_\infty = \epsilon + \gamma||V_k - V||_\infty \ , \end{aligned} \tag{18}$$

and so $||V_k - V||_\infty \leq \frac{\epsilon}{1-\gamma}$. This finally allows us to conclude that

$$||V_{k+1} - V||_\infty = ||BV_k - BV||_\infty \leq \gamma||V_k - V||_\infty \leq \frac{\epsilon\gamma}{1-\gamma} \ . \tag{19}$$

□

Exercise 3.11. Suppose that in algorithm 3, the initialization step is changed so V' is set randomly (all entries finite), instead of $V' \leftarrow 0$. (a) Will the algorithm still converge? (b) Does the algorithm still retain the same error estimate of Theorem 3.1 ?

Exercise 3.12. Suppose the assumptions of Theorem 3.1 hold. Using the same notations as in the theorem prove the following:
(a) For all $k \geq 1$, $||V_k - V||_\infty \leq \gamma^{k-1}||V||_\infty$.
(b) $||V_2||_\infty \leq (1+\gamma)||V||_\infty$.
(c) For all $m, n \geq 1$, $||V_m - V_n||_\infty \leq (\gamma^{m-1} + \gamma^{n-1})||V||_\infty$.

3.4 Markov decision process

We are now in a position to define a Markov decision process (MDP). A MDP inherits the basic structure of a Markov reward process with some important key differences, together with the specification of a set of actions that an agent can take from each state. It is formally represented using the tuple (S, A, P, R, γ) which are listed below:

- S : A finite state space.
- A : A finite set of actions which are available from each state s.
- P : A transition probability model that specifies $P(s'|s, a)$.
- R : A reward function that maps a state-action pair to rewards (real numbers), i.e. $R : S \times A \to \mathbb{R}$.
- γ: Discount factor $\gamma \in [0, 1]$.

Some of these quantities have been explained in the context of a Markov reward process. However in the context of a MDP, there are important differences that we need to mention. The basic model of the dynamics is that there is a state space S, and an action space A, both of which we will consider to be finite. The agent starts from a state s_i at time i, chooses an action a_i from the action space, obtains a reward r_i and then reaches a successor state s_{i+1}. An episode of a MDP is thus represented as $(s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, \dots)$.

Unlike in the case of a Markov process or a Markov reward process where the transition probability was only a function of the successor state and the current state, in the case of a MDP the transition probabilities at time i are a function of the successor state s_{i+1} along with both the current state s_i and the action a_i, written as $P(s_{i+1}|s_i, a_i)$. We still assume the principle of stationary transition probabilities which in the context of a MDP is written mathematically as

$$P(s_i = s'|s_{i-1} = s, a_{i-1} = a) = P(s_j = s'|s_{j-1} = s, a_{j-1} = a), \tag{20}$$

for all $s, s' \in S$, for all $a \in A$, and for all $i, j = 1, 2, \ldots$.

The reward r_i at time i depends on both s_i and a_i in the case of a MDP, in contrast to a Markov reward process where it depended only on the current state. These rewards can be stochastic or deterministic, but just like in the case of a Markov reward process, we will assume that the rewards are stationary and the only relevant quantity will be the expected reward which we will denote by $R(s, a)$ for a fixed state s and action a, and defined below:

$$R(s, a) = \mathbb{E}[r_i | s_i = s, a_i = a] \ , \ \ \forall \ i = 0, 1, \ldots \ . \tag{21}$$

The notions of the **discount factor** γ, **horizon** H and **return** G_t for a MDP are exactly equivalent to those in the case of a Markov reward process. However the notion of a **state value function** is slightly modified for a MDP as explained next.

3.4.1 MDP policies and policy evaluation

Given a MDP, a policy for the MDP specifies what action to take in each state. A policy can either be deterministic or stochastic. To cover both these cases, we will consider a policy to be a probability distribution over actions given the current state. It is important to note that the policy may be varying with time, which is especially true in the case of finite horizon MDPs. We will denote a generic policy by the boldface symbol $\boldsymbol{\pi}$, defined as the infinite dimensional tuple $\boldsymbol{\pi} = (\pi_0, \pi_1, \ldots)$, where π_t refers to the policy at time t. We will call policies that do not vary with time *"stationary policies"*, and indicate them as π, i.e. in this case $\boldsymbol{\pi} = (\pi, \pi, \ldots)$. For a stationary policy π, if at time t the agent is in state s, it will choose an action a with probability given by $\pi(a|s)$ and this probability does not depend on t, while for a non-stationary policy the probability will depend on time t and we will be denoted by $\pi_t(a|s)$.

Given a policy $\boldsymbol{\pi}$ one can define two quantities : *the state value function* and *the state-action value function* for the MDP corresponding to the policy $\boldsymbol{\pi}$, as shown below:

- **State value function** : The state value function $V_t^{\boldsymbol{\pi}}(s)$ for a state $s \in S$ is defined as the expected return starting from the state $s_t = s$ at time t and following policy $\boldsymbol{\pi}$, and is given by the expression $V_t^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\boldsymbol{\pi}}[G_t | s_t = s]$, where $\mathbb{E}_{\boldsymbol{\pi}}$ denotes that the expectation is taken with respect to the policy $\boldsymbol{\pi}$. Frequently we will drop the subscript $\boldsymbol{\pi}$ in the expectation to simplify notation going forward. Thus $\mathbb{E}$ will mean expectation with respect to the policy unless specified otherwise, and so we can write

  $$V_t^{\boldsymbol{\pi}}(s) = \mathbb{E}[G_t | s_t = s] \ . \tag{22}$$

  Notice that when the horizon H is infinite, this definition (22) together with the stationary assumptions of the rewards, transition probabilities and policy imply that for all $s \in S$, $V_i^{\boldsymbol{\pi}}(s) = V_j^{\boldsymbol{\pi}}(s)$ for all $i, j = 0, 1, \ldots$, and thus in this case we will define in a manner analogous to the case of a Markov reward process:

  $$V^{\pi}(s) = V_0^{\boldsymbol{\pi}}(s) \ . \tag{23}$$

- **State-action value function** : The state-action value function $Q_t^{\boldsymbol{\pi}}(s, a)$ for a state s and action a is defined as the expected return starting from the state $s_t = s$ at time t and taking the action $a_t = a$, and then subsequently following the policy $\boldsymbol{\pi}$. It is written mathematically as

  $$Q_t^{\boldsymbol{\pi}}(s, a) = \mathbb{E}[G_t | s_t = s, a_t = a] \ . \tag{24}$$

  In the infinite horizon case, similar to the state value function, the stationary assumptions about the rewards, transition probabilities and policy imply that for all $s \in S$ and $a \in A$, $Q_i^{\boldsymbol{\pi}}(s, a) = Q_j^{\boldsymbol{\pi}}(s, a)$ for all $i, j = 0, 1, \ldots$, which motivates the following definition

  $$Q^{\pi}(s, a) = Q_0^{\boldsymbol{\pi}}(s, a) \ . \tag{25}$$

Exercise 3.13. Consider a stationary policy $\boldsymbol{\pi} = (\pi, \pi, \dots)$. If the assumptions of stationary transition probabilities and stationary rewards hold, and if the horizon H is infinite, then using the definitions in (22) and (24) prove that for all $s \in S$ and $a \in A$, (a) $V_i^{\boldsymbol{\pi}}(s) = V_j^{\boldsymbol{\pi}}(s)$, and (b) $Q_i^{\boldsymbol{\pi}}(s,a) = Q_j^{\boldsymbol{\pi}}(s,a)$ for all $i, j = 0, 1, \dots$.

In the infinite horizon case, the assumptions about stationary transition probabilities and rewards lead to the following important identity connecting the state value function and the state-action value function for a stationary policy π :

$$\begin{aligned} Q^\pi(s,a) &\overset{(a)}{=} Q_0^\pi(s,a) = \mathbb{E}[G_0 | s_0 = s, a_0 = a] = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r_i \middle| s_0 = s, a_0 = a\right] \\ &= \mathbb{E}[r_0 | s_0 = s, a_0 = a] + \sum_{i=1}^{\infty} \gamma^i \mathbb{E}[r_i | s_0 = s, a_0 = a] \\ &\overset{(b)}{=} R(s,a) + \sum_{i=1}^{\infty} \gamma^i \left(\sum_{s' \in S} P(s_1 = s' | s_0 = s, a_0 = a) \mathbb{E}[r_i | s_0 = s, a_0 = a, s_1 = s'] \right) \\ &\overset{(c)}{=} R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) \left(\sum_{i=1}^{\infty} \gamma^{i-1} \mathbb{E}[r_i | s_1 = s'] \right) \\ &\overset{(d)}{=} R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^\pi(s') \ , \end{aligned} \tag{26}$$

for all $s \in S$, $a \in A$, where (a) follows from (24) and (25), (b) is due to the law of total expectation, (c) follows from the Markov property, and (d) follows from Exercise 3.13 and linearity of expectation.

Exercise 3.14. Consider a policy $\boldsymbol{\pi}$, not necessarily stationary. (a) Prove that in this case the analog of equation (26) is given by $Q_t^{\boldsymbol{\pi}}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_{t+1}^{\boldsymbol{\pi}}(s')$, for all $s \in S$, $a \in A$ and for all $t = 0, 1, \dots$.

An interesting aspect of specifying a stationary policy π on a MDP is that evaluating the value function for the policy is equivalent to evaluating the value function on an equivalent Markov reward process. Specifically we define the Markov reward process $M'(S, \mathbf{P}^\pi, R^\pi, \gamma)$, where $\mathbf{P}^\pi$ and R^π are given by:

$$\begin{aligned} R^\pi(s) &= \sum_{a \in A} \pi(a|s) R(s,a) \ , \\ P^\pi(s'|s) &= \sum_{a \in A} \pi(a|s) P(s'|s,a) \ . \end{aligned} \tag{27}$$

Exercise 3.15. Consider a stationary policy π for a MDP. (a) Prove that the value function of the policy V^π satisfies the identity $V^\pi(s) = R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s'|s) V^\pi(s')$ for all states $s \in S$, with R^π and $\mathbf{P}^\pi$ defined by (27).

The evaluation of the value function corresponding to the policy can then be carried out using the techniques introduced in the context of Markov reward processes. For example, in the infinite horizon case with $\gamma < 1$, the iterative algorithm to calculate the value function corresponding to a stationary policy π is given in algorithm 4. The algorithm takes as input a Markov decision process $M = (S, A, P, R, \gamma)$, a stationary policy π, and a tolerance ϵ, and computes the value function for all the states.

Exercise 3.16. (a) Prove that when $\gamma < 1$, algorithm 4 always converges. *Hint: Use Theorem 3.1.* (b) Consider a positive sequence of real numbers $\{\epsilon_i\}_{i \geq 1}$ such that $\epsilon_i \to 0$. Suppose algorithm 4 is run to termination for each ϵ_i, and denote each corresponding output of the algorithm as V_i^π. Prove that the sequence $V_i^\pi \to V^\pi$, where V^π is the value of the policy.

Algorithm 4 Iterative algorithm to calculate MDP value function for a stationary policy π

procedure POLICY EVALUATION(M, π, ϵ)
  For all states $s \in S$, define $R^\pi(s) = \sum_{a \in A} \pi(a|s) R(s,a)$
  For all states $s, s' \in S$, define $P^\pi(s'|s) = \sum_{a \in A} \pi(a|s) P(s'|s,a)$
  For all states $s \in S$, $V'(s) \leftarrow 0$, $V(s) \leftarrow \infty$
  while $||V - V'||_\infty > \epsilon$ **do**
    $V \leftarrow V'$
    For all states $s \in S$, $V'(s) = R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s'|s) V(s')$
  return $V'(s)$ for all $s \in S$

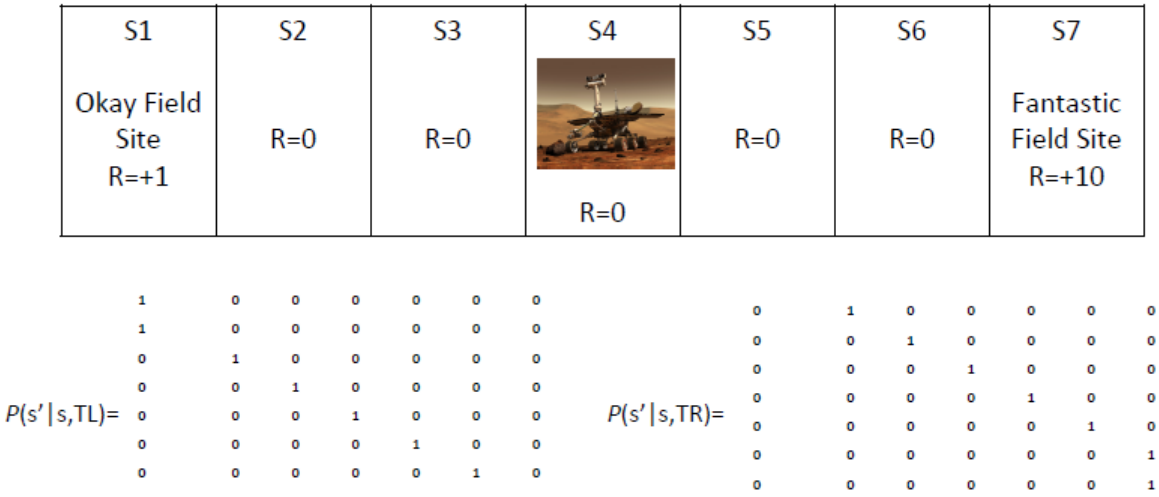

Figure 3: Mars Rover Markov decision process.

3.4.2 Example of a Markov decision process : Mars Rover

As an example of a MDP, consider the example given in Figure 3. The agent is again a Mars rover whose state space is given by $S = \{S1,\ S2,\ S3,\ S4,\ S5,\ S6,\ S7\}$. The agent has two actions in each state called *"try left"* and *"try right"*, and so the action space is given by $A = \{TL, TR\}$. Taking an action always succeeds, unless we hit an edge in which case we stay in the same state. This leads to the two transition probability matrices for each of the two actions as shown in Figure 3. The rewards from each state are the same for all actions, and is 0 in the states $\{S2,\ S3,\ S4,\ S5,\ S6\}$, while for the states $S1,\ S7$ the rewards are 1, 10 respectively. The discount factor for this MDP is some $\gamma \in [0, 1]$.

Exercise 3.17. Consider the MDP discussed above in Figure 3. Let $\gamma = 0$, and consider a stationary policy π which always involves taking the action TL from any state. (a) Calculate the value function of the policy for all states if the horizon is finite. (b) Calculate the value function of the policy when the horizon is infinite. *Hint: Use Theorem A.3.*

3.5 Bellman backup operators

In this section, we introduce the concept of the Bellman backup operators and prove some of their properties which will turn out to be extremely useful in the next section when we discuss MDP control. We have already encountered one Bellman backup operator in (14), (15) in the proof of Theorem 3.1. We will now define two other closely related (but not same!) Bellman backup operators : *the Bellman*

expectation backup operator and *the Bellman optimality backup operator.*

3.5.1 Bellman expectation backup operator

Suppose we are given a MDP $M = (S, A, P, R, \gamma)$, and a stationary policy π which can be deterministic or stochastic. We have already seen in section 3.4.1 that this is equivalent to a MRP $M' = (S, P^\pi, R^\pi, \gamma)$, where P^π and R^π are defined in (27). The value function of policy π evaluated on M, and denoted by V^π, is the same as the value function evaluated on M', where we have used the corresponding definitions of the value function for a MDP and MRP respectively. Note that V^π lives in the finite dimensional Banach space $\mathbb{R}^{|S|}$, which we will equip with the infinity norm $||\cdot||_\infty$ introduced in Exercise 3.2.

Then for element $U \in \mathbb{R}^{|S|}$ the Bellman expectation backup operator B^π for the policy π is defined as

$$(B^\pi U)(s) = R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s'|s) U(s') \quad , \quad \forall\, s \in S \,. \tag{28}$$

We should note that we have already seen this operator appear once before in algorithm 4. We now prove some properties of this operator.

Theorem 3.2. *The operator B^π defined in (28) is a contraction map. If $\gamma < 1$ then it is a strict contraction and has a unique fixed point.*

Proof. Consider $U_1, U_2 \in \mathbb{R}^{|S|}$. Then for a state $s \in S$, we have from (28) and triangle inequality

$$\begin{aligned} |(B^\pi U_1)(s) - (B^\pi U_2)(s)| &= \gamma \left| \sum_{s' \in S} P^\pi(s'|s)(U_1(s') - U_2(s')) \right| \leq \gamma \sum_{s' \in S} P^\pi(s'|s) |U_1(s') - U_2(s')| \\ &\leq \gamma \sum_{s' \in S} P^\pi(s'|s) \max_{s'' \in S} |U_1(s'') - U_2(s'')| = \gamma \sum_{s' \in S} P^\pi(s'|s)\, ||U_1 - U_2||_\infty \\ &= \gamma\, ||U_1 - U_2||_\infty \quad . \end{aligned} \tag{29}$$

As (29) is true for every $s \in S$ we conclude that $||B^\pi U_1 - B^\pi U_2||_\infty \leq \gamma\, ||U_1 - U_2||_\infty$, and hence B^π is a contraction map as $\gamma \in [0, 1]$.

Considering $\gamma < 1$ in (29), we conclude that in this case B^π is a strict contraction, and hence by applying Theorem A.5 it has a unique fixed point. □

Corollary 3.2.1. *Let $\gamma < 1$. Then for any $U \in \mathbb{R}^{|S|}$ the sequence $\{(B^\pi)^k U\}_{k \geq 0}$ is a Cauchy sequence and converges to the fixed point of B^π.*

Proof. The proof follows directly by applying Theorem 3.2, followed by Theorem A.4 and the contraction mapping theorem (TheoremA.5). □

This also implies that for a stationary policy π, the value function of the policy V^π is a fixed point of B^π, as shown by the following corollary.

Corollary 3.2.2. *Let π be a policy for an infinite horizon MDP with $\gamma < 1$. Then the value function of the policy V^π is a fixed point of B^π.*

Proof. The fact that $(B^\pi V^\pi)(s) = V^\pi(s)$ for all states $s \in S$, follows from the definition (28) of B^π and Exercise 3.15. □

The next theorem proves the *"monotonicity"* property of the Bellman expectation backup operator.

Theorem 3.3. *Suppose we have $U_1, U_2 \in \mathbb{R}^{|S|}$ such that for all $s \in S$, $U_1(s) \geq U_2(s)$. Then for every stationary policy π, we have $(B^\pi U_1)(s) \geq (B^\pi U_2)(s)$ for all $s \in S$. If instead the inequality is strict, i.e. $U_1(s) > U_2(s)$ for all $s \in S$, then we have $(B^\pi U_1)(s) > (B^\pi U_2)(s)$ for all $s \in S$.*

Proof. When $U_1(s) \geq U_2(s)$ for all $s \in S$, using definition (28) of B^π we obtain,

$$(B^\pi U_1)(s) - (B^\pi U_2)(s) = \sum_{s' \in S} P^\pi(s'|s)(U_1(s') - U_2(s')) \geq 0 \ , \tag{30}$$

ans when $U_1(s) > U_2(s)$ for all $s \in S$, the same steps give $(B^\pi U_1)(s) - (B^\pi U_2)(s) > 0$, for all states $s \in S$. □

3.5.2 Bellman optimality backup operator

Suppose we are now given a MDP $M = (S, A, P, R, \gamma)$. We again consider the finite dimensional Banach space $\mathbb{R}^{|S|}$ equipped with the infinity norm $||\cdot||_\infty$. Then for every element $U \in \mathbb{R}^{|S|}$ the Bellman optimality backup operator B^* is defined as

$$(B^* U)(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) U(s') \right] \ , \ \forall \ s \in S \ . \tag{31}$$

We next prove analogous properties for this operator which are similar to the ones for the Bellman expectation backup operator.

Theorem 3.4. *For every $U_1, U_2 \in \mathbb{R}^{|S|}$, and for all states $s \in S$ the following inequalities are true:*

(a)

$$\begin{aligned} (B^* U_1)(s) - (B^* U_2)(s) &\leq \gamma \max_{a \in A} \left[\sum_{s' \in S} P(s'|s,a) \ (U_1(s') - U_2(s')) \right] \\ &\leq \gamma \max_{a \in A} \left[\sum_{s' \in S} P(s'|s,a) \ |U_1(s') - U_2(s')| \right] \ , \end{aligned} \tag{32}$$

(b)

$$|(B^* U_1)(s) - (B^* U_2)(s)| \leq \gamma \max_{a \in A} \left[\sum_{s' \in S} P(s'|s,a) \ |U_1(s') - U_2(s')| \right] \leq \gamma \ ||U_1 - U_2||_\infty \ . \tag{33}$$

Proof. We first prove part *(a)*. Fix a state $s \in S$. Using (31) and as the action space A is finite, we conclude that there exists $a_1, a_2 \in A$, not necessarily different, such that the following holds:

$$\begin{aligned} (B^* U_1)(s) &= R(s,a_1) + \gamma \sum_{s' \in S} P(s'|s,a_1) U_1(s') \ , \\ (B^* U_2)(s) &= R(s,a_2) + \gamma \sum_{s' \in S} P(s'|s,a_2) U_2(s') \ . \end{aligned} \tag{34}$$

Then by the definition of maximum in (31), we also have for the action a_1 that

$$(B^* U_2)(s) \geq R(s,a_1) + \gamma \sum_{s' \in S} P(s'|s,a_1) U_2(s') \ . \tag{35}$$

Thus from (34) and (35) we deduce the following

$$\begin{aligned}(B^*U_1)(s) - (B^*U_2)(s) &\leq \gamma \sum_{s' \in S} P(s'|s, a_1)\ (U_1(s') - U_2(s')) \\ &\leq \gamma \max_{a \in A} \left[\sum_{s' \in S} P(s'|s, a)\ (U_1(s') - U_2(s')) \right] \quad ,\end{aligned} \tag{36}$$

which proves the first inequality of *(a)*. For the second inequality notice that we have for all states $s' \in S$, $U_1(s') - U_2(s') \leq |U_1(s') - U_2(s')|$, and so multiplying each of these inequalities by positive numbers $P(s'|s, a)$ for some $a \in A$, and summing over all s' gives

$$\sum_{s' \in S} P(s'|s, a)\ (U_1(s') - U_2(s')) \leq \sum_{s' \in S} P(s'|s, a)\ |(U_1(s') - U_2(s'))| \quad . \tag{37}$$

The result is proved by taking the max over all $a \in A$, by using monotonicity of the max function.

To prove part *(b)*, notice that by interchanging the roles of U_1, U_2, we have from part *(a)*

$$(B^*U_2)(s) - (B^*U_1)(s) \leq \gamma \max_{a \in A} \left[\sum_{s' \in S} P(s'|s, a)\ |U_1(s') - U_2(s')| \right] \quad , \tag{38}$$

and thus combining (38) and (32) we obtain

$$\begin{aligned}|(B^*U_1)(s) - (B^*U_2)(s)| &\leq \gamma \max_{a \in A} \left[\sum_{s' \in S} P(s'|s, a)\ |U_1(s') - U_2(s')| \right] \\ &\leq \gamma \max_{a \in A} \left[\sum_{s' \in S} P(s'|s, a) \max_{s'' \in S} |U_1(s'') - U_2(s'')| \right] \\ &= \gamma \max_{a \in A} \left[\sum_{s' \in S} P(s'|s, a)\ ||U_1 - U_2||_\infty \right] \\ &= \gamma \max_{a \in A} ||U_1 - U_2||_\infty = \gamma\ ||U_1 - U_2||_\infty \quad ,\end{aligned} \tag{39}$$

which proves *(b)*. □

Theorem 3.5. *The operator B^* defined in (31) is a contraction map. If $\gamma < 1$ then it is a strict contraction and has a unique fixed point.*

Proof. The fact that B^* is a contraction follows from Theorem 3.4 by observing that (32) is true for all $s \in S$, and so must be true in particular for $\arg\max_{s \in S} |(B^*U_1)(s) - (B^*U_2)(s)|$, for every $U_1, U_2 \in \mathbb{R}^{|S|}$. Thus $||B^*U_1 - B^*U_2||_\infty \leq \gamma\ ||U_1 - U_2||_\infty$, proving that B^* is a contraction map as $\gamma \in [0, 1]$. Setting $\gamma < 1$ in this inequality proves that B^* is a strict contraction for $\gamma \in [0, 1)$ and thus has a unique fixed point by Theorem A.5. □

Corollary 3.5.1. *Let $\gamma < 1$. Then for any $U \in \mathbb{R}^{|S|}$ the sequence $\{(B^*)^k U\}_{k \geq 0}$ is a Cauchy sequence and converges to the fixed point of B^*.*

Proof. The proof follows directly by applying Theorem 3.5, followed by Theorem A.4 and the contraction mapping theorem (TheoremA.5). □

The next theorem compares the result of the application of B^π versus B^* to some $U \in \mathbb{R}^{|S|}$.

Theorem 3.6. *For every stationary policy π, for every $U \in \mathbb{R}^{|S|}$ and for all $s \in S$, $(B^*U)(s) \geq (B^\pi U)(s)$.*

Proof. Fix a stationary policy π, and let B^π be the corresponding Bellman expectation backup operator. Fix some $U \in \mathbb{R}^{|S|}$. Let us also fix some $s \in S$. Then from definition (31) of B^* we have

$$(B^*U)(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a)U(s') \right] \geq R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a)U(s') \ , \ \forall \ a \in A \ . \quad (40)$$

Multiplying (40) by $\pi(a|s)$ and summing over all $a \in A$ gives

$$\begin{aligned} (B^*U)(s) &= \sum_{a \in A} \pi(a|s)(B^*U)(s) \geq \sum_{a \in A} \pi(a|s) \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a)U(s') \right] \\ &= \sum_{a \in A} \pi(a|s)R(s,a) + \gamma \sum_{s' \in S} \left(\sum_{a \in A} \pi(a|s)P(s'|s,a) \right) U(s') \\ &= R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s'|s)U(s') = (B^\pi U)(s) \ , \end{aligned} \quad (41)$$

where the last equality follows from definitions (27) and (28) of R^π, P^π and B^π, thus proving the theorem. □

3.6 MDP control in the infinite horizon setting

We now have all the background necessary to discuss the problem of *"MDP control"*, where we seek to find the best policy (often a policy), that achieves the greatest value function among the set of all possible policies. In the context of reinforcement learning, this is precisely the objective of the agent. We are going to first discuss the infinite horizon case in this section, and the finite horizon case will be mentioned in the next section. We do it this way because the infinite horizon case is a much harder problem, that presents quite a few mathematical challenges which will need to be resolved.

To get started, we need to address the question *"what do we exactly mean by finding an optimal policy ?"*. Precisely we want to know whether a policy always exists, which we will denote by $\boldsymbol{\pi}^*$, whose value function is at least as good as the value function of any other policy. In other words, we need to ensure that the supremum of the value function is actually attained for some policy ! To appreciate the subtlety of this point, consider the example of maximizing the function $f : \mathbb{R} \to \mathbb{R}$ on $(0,1)$ defined as $f(x) = x$, and note that this problem does not have a solution. But $\sup f(x) = 1$, although $\nexists \ x \in (0,1)$ for which this is attained.

We first define precisely what it means for a policy, not necessarily stationary, to be an **optimal policy**.

Definition 3.1. A policy $\boldsymbol{\pi}^*$ is an *optimal policy* iff for every policy $\boldsymbol{\pi}$, for all $t = 0, 1, \ldots$, and for all states $s \in S$, $V_t^{\boldsymbol{\pi}^*}(s) \geq V_t^{\boldsymbol{\pi}}(s)$.

The next result that we leave for the reader to prove states that for an infinite horizon MDP, existence of an optimal policy also implies the existence of a stationary optimal policy. This result is intuitively obvious, and is a very important result as it significantly reduces the universe of policies to consider when searching for an optimal policy, if it exists. In particular, it states that we need only consider policies that are stationary.

Exercise 3.18. (a) Consider an infinite horizon MDP. Let $\boldsymbol{\pi}^*$ be an optimal policy for the MDP. Prove that there exists a stationary policy π, that is $\boldsymbol{\pi} = (\pi, \pi, \ldots)$, which is also optimal.

The next two theorems improve on the conclusion of Exercise 3.18 and show us that we may restrict the search to a finite set of deterministic stationary policies.

Theorem 3.7. *The number of deterministic stationary policies is finite, and equals* $|A|^{|S|}$.

Proof. Since the policies are stationary and deterministic, each policy can be represented as a function $\pi : S \to A$. The number of such distinct functions is given by $|A|^{|S|}$. This also proves that the set of deterministic stationary policies is finite. □

Theorem 3.8. *If* π *is a stationary policy for an infinite horizon MDP with* $\gamma < 1$*, then there exists a deterministic stationary policy* $\hat{\pi}$ *such that* $V^{\hat{\pi}}(s) \geq V^{\pi}(s)$ *for all states* $s \in S$*. One such policy is given by the stationary policy*

$$\hat{\pi}(s) = \arg\max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi}(s') \right] \ , \ \forall \ s \in S \ , \tag{42}$$

which satisfies the equality $(B^{\hat{\pi}} V^{\pi})(s) = (B^* V^{\pi})(s) \geq V^{\pi}(s)$ *for all* s*. Moreover* $V^{\hat{\pi}}(s) = V^{\pi}(s)$ *for all* s*, iff* $(B^* V^{\pi})(s) = V^{\pi}(s)$ *for all* s*.*

Proof. We first notice that the policy $\hat{\pi}$ defined in (42) is a stationary policy (by definition), and is also deterministic for every $s \in S$, by the definition of arg max with ties broken randomly.

As $\hat{\pi}$ is deterministic, we can conclude using (27) that $R^{\hat{\pi}}(s) = R(s, \hat{\pi}(s))$ and $P^{\hat{\pi}}(s'|s) = P(s'|s, \hat{\pi}(s))$ for all $s \in S$ and $a \in A$, and thus we have

$$(B^{\hat{\pi}} V^{\pi})(s) = R(s, \hat{\pi}(s)) + \gamma \sum_{s' \in S} P(s'|s, \hat{\pi}(s)) V^{\pi}(s) = (B^* V^{\pi})(s) \ , \tag{43}$$

for all states $s \in S$ using (42), and the definitions of the Bellman backup operators in (28) and (31). Next, by Corollary 3.2.2 we have $B^{\pi} V^{\pi} = V^{\pi}$, and by Theorem 3.6 we have $B^* V^{\pi} \geq B^{\pi} V^{\pi}$, and so combining these with (43) we obtain

$$(B^{\hat{\pi}} V^{\pi})(s) = (B^* V^{\pi})(s) \geq V^{\pi}(s) \ , \ \forall \ s \in S \ . \tag{44}$$

Next using Theorem 3.3, the monotonicity property of $B^{\hat{\pi}}$ allows us to conclude by repeatedly applying $B^{\hat{\pi}}$ to both sides of (44) that $((B^{\hat{\pi}})^k V^{\pi})(s) \geq V^{\pi}(s)$ for all $k \geq 1$, and for all states $s \in S$. Then using Corollary 3.2.1, and noticing that $V^{\hat{\pi}}$ is the unique fixed point of $B^{\hat{\pi}}$ we obtain by taking limits

$$V^{\hat{\pi}}(s) = (B^{\hat{\pi}} V^{\hat{\pi}})(s) = \lim_{k \to \infty} ((B^{\hat{\pi}})^k V^{\pi})(s) \geq V^{\pi}(s) \ , \ \forall \ s \in S \ . \tag{45}$$

To prove the second part of the theorem, first assume that $B^* V^{\pi} = V^{\pi}$. Then by (44) we have $B^{\hat{\pi}} V^{\pi} = B^* V^{\pi} = V^{\pi}$, and so by uniqueness of the fixed point of $B^{\hat{\pi}}$ we get $V^{\hat{\pi}} = B^{\hat{\pi}} V^{\hat{\pi}} = V^{\pi}$. Next assume that $V^{\hat{\pi}} = V^{\pi}$. Then again by (44) we have $V^{\pi} = V^{\hat{\pi}} = B^{\hat{\pi}} V^{\hat{\pi}} = B^{\hat{\pi}} V^{\pi} = B^* V^{\pi} \geq V^{\pi}$, implying that $B^* V^{\pi} = V^{\pi}$, thus completing the proof. □

Corollary 3.8.1. *In the notation of Theorem 3.8, if* $\exists \ s \in S$ *such that* $(B^* V^{\pi})(s) > V^{\pi}(s)$*, then* $V^{\hat{\pi}}(s) > V^{\pi}(s)$*. In this case, we say that* $\hat{\pi}$ *is "strictly better" than* π *as a policy.*

Proof. The proof follows immediately by noting that the inequality in (44) becomes a strict inequality, and then applying Theorem 3.3. □

The consequences of Theorems 3.7 and 3.8 is spectacular, because now the search for an optimal policy has been reduced to the set of only the deterministic stationary policies which is a finite set, if such a policy exists. The reader is to prove that this is actually the case in the following exercise.

Exercise 3.19. Consider an infinite horizon MDP with $\gamma < 1$. Denote Π to be the set of all deterministic stationary policies. (a) Prove that $\exists \ \pi^* \in \Pi$, such that for all $\pi \in \Pi$, and for all states $s \in S$, $V^{\pi^*}(s) \geq V^{\pi}(s)$. (b) Conclude that $\boldsymbol{\pi^*} = (\pi^*, \pi^*, \dots)$ is an optimal policy. *Hint : See Theorem 3.10.*

We have thus established the existence of an optimal policy and moreover concluded that a deterministic stationary policy suffices. This then allows us to make the following definition:

Definition 3.2. The *optimal value function* for an infinite horizon MDP is defined as

$$V^*(s) = \max_{\pi \in \Pi} V^\pi(s) \ , \tag{46}$$

and there exists a stationary deterministic policy $\pi^* \in \Pi$, which is an optimal policy, such that $V^*(s) = V^{\pi^*}(s)$ for all states $s \in S$, where Π is the set of all stationary deterministic policies.

We next look at a few algorithms to compute the optimal value function and an optimal policy.

3.6.1 Policy search

Definition 3.2 immediately renders itself to a brute force algorithm called **policy search** to find the optimal value function V^* and an optimal policy π^*, as described in pseudo-code in algorithm 5. The algorithm takes as input an infinite horizon MDP $M = (S, A, P, R, \gamma)$ and a tolerance ϵ for accuracy of policy evaluation, and returns the optimal value function and an optimal policy.

Algorithm 5 Policy search algorithm to calculate optimal value function and find an optimal policy

```
procedure POLICY SEARCH(M, ε)
    Π ← All stationary deterministic policies of M
    π* ← Randomly choose a policy π ∈ Π
    V* ← POLICY EVALUATION (M, π*, ε)
    for π ∈ Π do
        V^π ← POLICY EVALUATION (M, π, ε)
        if V^π(s) ≥ V*(s) for all s ∈ S, then
            V* ← V^π
            π* ← π
    return V*(s), π*(s) for all s ∈ S
```

It is clear that algorithm 5 always terminates as it checks all $|A|^{|S|}$ deterministic stationary policies. Thus the run-time complexity of this algorithm is $O(|A|^{|S|})$. It is possible to prove correctness of the algorithm when $\epsilon = 0$, i.e. when in each iteration the policy evaluation is done exactly. In practice ϵ is set to a small number such as 10^{-9} to 10^{-12}.

Theorem 3.9. *Algorithm 5 returns the optimal value function and an optimal policy when $\epsilon = 0$.*

Proof. Let π^* be an optimal policy, and thus $V^{\pi^*}(s) = V^*(s)$ for all states $s \in S$. Since the algorithm checks every policy in Π, it means that π^* must get selected at some iteration of the algorithm. Thus for the policies considered in future iterations the value function can no longer strictly increase. Future iterations may select a different policy with the same optimal value function, thus completing the proof. □

Exercise 3.20. Consider the MDP discussed in section 3.4.2, shown in Figure 3. Consider the horizon to be infinite. (a) How many deterministic stationary policies does the agent have ? (b) If $\gamma < 1$, is the optimal policy unique ? (c) If $\gamma = 1$, is the optimal policy unique ?

3.6.2 Policy iteration

We now discuss a more efficient algorithm than policy search called **policy iteration**. The algorithm is a straightforward application of Theorem 3.8, which states that given any stationary policy π, we can find a deterministic stationary policy that is no worse than the existing policy. In particular the

Algorithm 6 Policy improvement algorithm to improve an input policy

procedure POLICY IMPROVEMENT(M, V^π)
  $\hat{\pi}(s) \leftarrow \arg\max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^\pi(s') \right] \ , \ \forall \ s \in S$
  return $\hat{\pi}(s)$ for all $s \in S$

theorem also applies to deterministic policies. This simple step has a special name called **"policy improvement"**, whose pseudo-code is presented in algorithm 6.

The output of algorithm 6 is always guaranteed to be at least as good as the policy π corresponding to the input value function V^π, and represents a *"greedy"* attempt to improve the policy. When performed iteratively with the policy evaluation algorithm (algorithm 4), this gives rise to the policy iteration algorithm. The pseudo-code of policy iteration is outlined in algorithm 7.

Algorithm 7 Policy iteration algorithm to calculate optimal value function and find an optimal policy

procedure POLICY ITERATION(M, ϵ)
  $\pi \leftarrow$ Randomly choose a policy $\pi \in \Pi$
  while true **do**
    $V^\pi \leftarrow$ POLICY EVALUATION (M, π, ϵ)
    $\pi^* \leftarrow$ POLICY IMPROVEMENT (M, V^π)
    if $\pi^*(s) = \pi(s)$ **then**
      break
    else
      $\pi \leftarrow \pi^*$
  $V^* \leftarrow V^\pi$
  return $V^*(s)$, $\pi^*(s)$ for all $s \in S$

The proof of correctness of algorithm 7 is left to the reader as the next exercise. Note that the algorithm will always terminate as there are a finite number of stationary deterministic policies by Theorem 3.7.

Exercise 3.21. Consider an infinite horizon MDP with $\gamma < 1$. (a) Show that when algorithm 7 is run with $\epsilon = 0$, it finds the optimal value function and an optimal policy. *Hint : See Theorem 3.10.* (b) Prove that the termination criteria used in the algorithm makes sense: precisely show that if the policy does not change during a policy improvement step, then the policy cannot improve in future iterations. (c) Show that the value functions corresponding to the policies in each iteration of the algorithm form a non-decreasing sequence for every $s \in S$. (d) What is the worst case run-time complexity of this algorithm ?

3.6.3 Value iteration

We now discuss **value iteration** which is yet another technique that can be used to compute the optimal value function and an optimal policy, given a MDP. To motivate this method we will need the following theorem:

Theorem 3.10. *For a MDP with $\gamma < 1$, let the fixed point of the Bellman optimality backup operator B^* be denoted by $V^* \in \mathbb{R}^{|S|}$. Then the policy given by*

$$\pi^*(s) = \arg\max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s') \right] \ , \ \forall \ s \in S \ , \tag{47}$$

is a stationary deterministic policy. The value function of this policy V^{π^*} satisfies the identity $V^{\pi^*} = V^*$, and thus V^* is also the fixed point of the operator B^{π^*}. In particular this implies that there exists a stationary deterministic policy π^* whose value function is the fixed point of B^*. Moreover, π^* is an optimal policy.

Proof. We start by noting that π^* as defined in (47) is a stationary deterministic policy, and so we can conclude using (27) that $R^{\pi^*}(s) = R(s, \pi^*(s))$ and $P^{\pi^*}(s'|s) = P(s'|s, \pi^*(s))$ for all $s \in S$ and $a \in A$.

As V^* is the fixed point of B^*, we have $B^*V^* = V^*$. So using definition (31) of B^*, and (47) we can write

$$\begin{aligned} V^*(s) &= \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s') \right] \\ &= R(s, \pi^*(s)) + \gamma \sum_{s' \in S} P(s'|s, \pi^*(s)) V^*(s') \\ &= R^{\pi^*}(s) + \gamma \sum_{s' \in S} P^{\pi^*}(s'|s) V^*(s') \\ &= V^{\pi^*}(s) \end{aligned} \tag{48}$$

for all $s \in S$, completing the proof of the first part of the theorem.

To prove that π^* is an optimal policy, we show that if an optimal policy exists then its value function must be a fixed point of the operator B^*. So assume that an optimal policy exists, which by Theorem 3.8 we can take to be a stationary deterministic policy, and let us denote it as μ and the corresponding optimal value function as V^μ. Now for the sake of contradiction, suppose V^μ is not a fixed point of B^*. Then there exists $s \in S$ such that $V^\mu(s) \neq (B^*V^\mu)(s)$, which upon combining with Theorem 3.8 implies that $V^\mu(s) > (B^*V^\mu)(s)$. Then application of Corollary 3.8.1 implies that there exists a policy $\hat{\pi}$ which is strictly better than μ, and so we have a contradiction. This proves that V^μ must be the unique fixed point of B^*. Combining this fact with the first part implies that V^* must be the optimal value function and π^* is an optimal policy. This completes the proof. □

Theorem 3.10 suggests a straightforward way to calculate the optimal value function V^* and an optimal policy π^*. The idea is to run fixed point iterations to find the fixed point of B^* using Corollary 3.5.1. Once we have V^*, an optimal policy π^* can be extracted using (47). The pseudo-code of this algorithm is given in algorithm 8, which takes as input an infinite horizon MDP $M = (S, A, P, R, \gamma)$ and a tolerance ϵ, and returns the optimal value function and an optimal policy.

Algorithm 8 Value iteration algorithm to calculate optimal value function and find an optimal policy

procedure VALUE ITERATION(M, ϵ)
  For all states $s \in S$, $V'(s) \leftarrow 0$, $V(s) \leftarrow \infty$
  while $||V - V'||_\infty > \epsilon$ **do**
    $V \leftarrow V'$
    For all states $s \in S$, $V'(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V(s') \right]$
  $V^* \leftarrow V$ for all $s \in S$
  $\pi^* \leftarrow \arg\max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s') \right] \quad , \forall\, s \in S$
  return $V^*(s)$, $\pi^*(s)$ for all $s \in S$

If algorithm 8 is run with $\epsilon = 0$, we can recover the optimal value function and an optimal policy exactly. However in practice, ϵ is set to be a small number such as 10^{-9}-10^{-12}.

3.7 MDP control for a finite horizon MDP

We now briefly discuss the MDP control problem for a finite horizon MDP. Having already discussed the control problem for infinite horizon MDPs, we simply state that in the finite horizon case, a deterministic policy can be obtained that is optimal. But the policy is no longer stationary, and so at each time t the policy is different. The proof is not too difficult and the reader is asked to derive these facts in the following exercise.

Exercise 3.22. Consider a MDP with finite horizon H and finite rewards. A typical episode of the MDP will look like $(s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}, s_H)$. Let a policy for the MDP be denoted by $\boldsymbol{\pi} = (\pi_0, \pi_1, \ldots, \pi_{H-1})$. Then prove the following statements:

(a) Show that the number of deterministic policies for the MDP is given by $H|A|^{|S|}$.

(b) Assuming that an optimal policy $\boldsymbol{\pi^*}$ exists, derive a recurrence relation for the optimal value function $V^{\boldsymbol{\pi}^*} = (V_0^{\boldsymbol{\pi}^*}, \ldots, V_H^{\boldsymbol{\pi}^*})$, with $V_H^{\boldsymbol{\pi}^*}(s) = 0$ for all states $s \in S$. Precisely, derive a relationship between $V_t^{\boldsymbol{\pi}^*}$ and $V_{t+1}^{\boldsymbol{\pi}^*}$.

(c) Let $\boldsymbol{\Pi}$ be the set of all deterministic policies, i.e. for every $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, π_t is a deterministic policy at time t and for all times $t = 0, \ldots, H-1$. Show that for every policy, deterministic or stochastic, there exists a $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ which is no worse.

(b) Show that $\boldsymbol{\Pi}$ contains a policy that is optimal.

Because of the conclusion of Exercise 3.22, just like in the infinite horizon case we can restrict our search for an optimal policy to the set of deterministic policies. We present an algorithm, namely **value iteration** for this purpose, which is analogous to its counterpart in the infinite horizon case.

Algorithm 9 Value iteration algorithm for finite horizon MDPs

procedure FINITE VALUE ITERATION(M)
  For all states $s \in S$, $V_H^*(s) \leftarrow 0$
  $t \leftarrow H - 1$
  while $t \geq 0$ **do**
    For all states $s \in S$, $V_t^*(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_{t+1}^*(s')\right]$
    For all states $s \in S$, $\pi_t^*(s) = \arg\max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_{t+1}^*(s')\right]$
    $t \leftarrow t - 1$
  return For all states $s \in S$, $V_t^*(s)$ for $t = 0, \ldots, H$, $\pi_t^*(s)$ for $t = 0, \ldots, H-1$

The proof of correctness of the algorithm is left to the reader as the next exercise.

Exercise 3.23. (a) Prove the correctness of algorithm 9. *Hint : Use results of Exercise 3.22 (b).*

The next exercise, which is also not too difficult to prove, establishes a correspondence between value iteration in the finite and infinite horizon cases.

Exercise 3.24. Consider a MDP $M = (S, A, P, R, \gamma)$ with infinite horizon and $\gamma < 1$. Let V^* be the optimal value function of M. Define a sequence of finite horizon MDPs M_k with horizon H_k, such that $M_k = M$ and $H_k = k$, for all $k = 1, 2, \ldots$. Let $\{(V_k)^*\}_{k \geq 1}$ be the sequence of optimal value functions returned by algorithm 9 when run with the input M_k, and corresponding to $t = 0$. (a) Prove that $(V_k)^* \to V^*$ as $k \to \infty$.

Appendices

A Contraction mapping theorem [1]

In this section, we introduce the notion of contraction maps in a Banach space setting, that we have heavily relied on in the previous section to prove many of our important theorems. The notation used in this section will be completely independent of what was introduced before, and so the reader should read this section in a self-contained fashion.

Let $(V, ||\cdot||)$ be a Banach space, where V is a vector space and $||\cdot||$ is the norm defined on the vector space. V may be finite or infinite dimensional. As it is a Banach space, we remind the reader that the space is complete, meaning that all Cauchy sequences (Definition A.1) converge (Definition A.2). We first give a few definitions:

Definition A.1. A sequence $\{v_k\}_{k\geq 1}$ of elements $v_k \in V$, $\forall\ k = 1, 2, \ldots$, is called a *Cauchy sequence* iff for every real number $\epsilon > 0$ there exists an integer $N \geq 1$, such that $||v_m - v_n|| < \epsilon$ for all $m, n > N$.

Definition A.2. Let $\{v_k\}_{k\geq 1}$ be a sequence of elements of V. We say that the sequence *converges* to an element $v \in V$, iff for every real number $\epsilon > 0$ there exists an integer $N \geq 1$, such that $||v_k - v|| < \epsilon$ for all $k \geq N$. We write this as $v_k \to v$.

Our first theorem of this section shows that any sequence that is eventually constant is Cauchy.

Theorem A.1. *A sequence $\{v_k\}_{k\geq 1}$ in a normed vector space that is eventually constant is Cauchy.*

Proof. As the sequence is eventually constant, there exists a positive integer r and $v \in V$ such that for all $k \geq r$, $v_k = v$. Then for any $\epsilon > 0$, one can choose $N = r$ in Definition A.1, giving $0 = ||v_m - v_n|| < \epsilon$ for all $m, n > N$, thus completing the proof. □

We can now prove that the limit of a Cauchy sequence is unique.

Theorem A.2. *A Cauchy sequence $\{v_k\}_{k\geq 1}$ in a Banach space converges to a unique limit.*

Proof. The fact that the Cauchy sequence converges to a limit is true by the definition of a Banach space. We need to show that this limit is unique. We prove it by contradiction.

Suppose $\exists v, w \in V$, $v \neq w$, such that $v_k \to v$ and $v_k \to w$. Let $\delta = ||v - w||$, and note that $\delta > 0$ as $v \neq w$. By Definition A.2, there exist positive integers M, N such that $||v_m - v|| < \delta/2 \quad ,\forall\ m \geq M$ and $||v_n - w|| < \delta/2 \quad ,\forall\ n \geq N$. Let $l = \max(M, N)$. Then by triangle inequality we have, $||v - w|| \leq ||v - v_l|| + ||v_l - w|| < \delta$, which is a contradiction. □

We next define the notion of a *"contraction map"* on a Banach space, and the notion of a *"fixed point"* of an operator that maps V to itself.

Definition A.3. A function $T : V \to V$ is called a *contraction* on V iff for every $v, w \in V$, $||Tv - Tw|| \leq ||v - w||$. The map is called a *strict contraction* iff there exists a real number $0 \leq \gamma < 1$, such that for every $v, w \in V$, $||Tv - Tw|| \leq \gamma||v - w||$. The constant γ is called the *contraction factor* of T.

Definition A.4. Consider a function $T : V \to V$. We say that $v \in V$ is a *fixed point* of T in V, iff $Tv = v$.

[1] Additional material that was not covered in class.

We should note that a map $T : V \to V$ may have many fixed points or none. For example, the contraction map $T : \mathbb{R} \to \mathbb{R}$ given by $T(x) = x + 1$ has no fixed points in $\mathbb{R}$. On the other hand the map $T : \mathbb{R} \to \mathbb{R}$ given by $T(x) = x$, which is also a contraction, has infinitely many fixed points in $\mathbb{R}$. Similarly, any linear map from V to itself has 0 as a fixed point, but may not be a contraction.

The $\gamma = 0$ case is special, as shown by the following theorem.

Theorem A.3. *Suppose T is a strict contraction on a normed vector space V (not necessarily Banach) with contraction factor $\gamma = 0$. Then T is a constant map.*

Proof. Consider an element $v \in V$, and let $c = Tv$. Now for every element $w \in V$, we have $||Tv - Tw|| \leq 0$, which implies $||Tv - Tw|| = 0$. By property of norms this implies that $Tw = Tv = c$. □

We next prove a theorem involving repeated application of a strict contraction map.

Theorem A.4. *Suppose T is a strict contraction on a normed vector space V (not necessarily Banach) with contraction factor γ. Then for every element $v \in V$, the sequence $\{v, Tv, T^2v, \dots\}$ is a Cauchy sequence.*

Proof. If $\gamma = 0$, Theorem A.3 implies that the sequence $\{v, Tv, T^2v, \dots\}$ is a constant sequence, except for the first term, and hence Cauchy by Theorem A.1.

So assume that $\gamma \neq 0$. Let $\alpha = ||Tv - v||$. By repeated application of the contraction map we have for all $n \geq 0$,

$$||T^{n+1}v - T^n v|| \leq \gamma ||T^n v - T^{n-1}v|| \leq \cdots \leq \gamma^n ||Tv - v|| = \gamma^n \alpha. \tag{49}$$

Then by the triangle inequality and (49) we additionally have for all m, n satisfying $0 \leq n \leq m$,

$$\begin{aligned} ||T^m v - T^n v|| &= \left\| \sum_{k=n}^{m-1} (T^{k+1}v - T^k v) \right\| \leq \sum_{k=n}^{m-1} ||T^{k+1}v - T^k v|| \\ &\leq \sum_{k=n}^{m-1} \gamma^k \alpha = \alpha \left(\frac{\gamma^n - \gamma^m}{1 - \gamma} \right) < \frac{\alpha \gamma^n}{1 - \gamma} \,. \end{aligned} \tag{50}$$

To prove the sequence is Cauchy, we fix an $\epsilon > 0$, and set $N = \max\left(1, \left\lceil \log\left(\frac{\epsilon(1-\gamma)}{\alpha}\right) \Big/ \log \gamma \right\rceil\right)$. Then for all m, n satisfying $m \geq n > N$, and as a consequence of (50), we have

$$||T^m v - T^n v|| \leq \frac{\alpha \gamma^n}{1 - \gamma} < \frac{\alpha \gamma^N}{1 - \gamma} \leq \epsilon \,, \tag{51}$$

which completes the proof. □

We can now prove the main result of this section : *"the contraction mapping theorem".*

Theorem A.5. *Suppose the function $T : V \to V$ is a strict contraction on a Banach space V. Then T has a unique fixed point in V. Moreover, for every element $v \in V$, the sequence $\{v, Tv, T^2v, \dots\}$ is Cauchy and converges to the fixed point.*

Proof. As T is a strict contraction, let $\gamma \in [0, 1)$ be the contraction factor of T.

We first prove the uniqueness part by contradiction. Let $v, w \in V$ be fixed points of T and $v \neq w$, so $||v - w|| > 0$. Then we have that $||Tv - Tw|| = ||v - w||$. By the contraction property we also have $||Tv - Tw|| \leq \gamma ||v - w|| < ||v - w||$. But then this implies $||v - w|| < ||v - w||$, a contradiction.

We now prove the existence part. Take any element $v \in V$ and consider the sequence $\{v_k\}_{k\geq 1}$ defined as follows:

$$v_k = \begin{cases} v & \text{if } k = 1, \\ Tv_{k-1} & \text{if } k > 1 \end{cases} . \tag{52}$$

Then by Theorem A.4, $\{v_k\}_{k\geq 1}$ is a Cauchy sequence, and hence as V is a Banach space, the sequence converges to a unique limit $v^* \in V$ by Theorem A.2. We claim that v^* is a fixed point of T. To prove this, choose any $\epsilon > 0$ and define $\delta = \epsilon/(1+\gamma)$. As $v_k \to v^*$, by Definition A.2, $\exists\ N \geq 1$ such that $||v_k - v^*|| < \delta\ ,\ \forall\ k \geq N$. Then by triangle inequality we have:

$$\begin{aligned} ||Tv^* - v^*|| &\leq ||Tv^* - v_{N+1}|| + ||v_{N+1} - v^*|| \\ &= ||Tv^* - Tv_N|| + ||v_{N+1} - v^*|| \\ &\leq \gamma||v^* - v_N|| + ||v_{N+1} - v^*|| \\ &< \gamma\delta + \delta = \epsilon\ . \end{aligned} \tag{53}$$

Thus we have proved that $||Tv^* - v^*|| < \epsilon$ for all $\epsilon > 0$, which implies that $||Tv^* - v^*|| = 0$. As V is a normed vector space, this finally implies that $Tv^* = v^*$, thus completing the existence proof and also proving the second part of the theorem. □

B Solutions to selected exercises

Exercise 3.3

Solution. The transition probability matrix is given by:

$$\mathbf{P} = \begin{array}{c} \\ S1 \\ S2 \\ S3 \\ S4 \\ S5 \\ S6 \\ S7 \end{array} \begin{array}{c} \begin{array}{ccccccc} S1 & S2 & S3 & S4 & S5 & S6 & S7 \end{array} \\ \begin{pmatrix} 0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.2 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0.2 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0.2 & 0.4 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0.2 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & 0.6 \end{pmatrix} \end{array}$$

Exercise 3.9

Solution. If the states are ordered as $\{S1,\ S2,\ S3,\ S4,\ S5,\ S6,\ S7\}$, the value function vector can be found by solving (12). The result is $V = [1.53,\ 0.37,\ 0.13,\ 0.22,\ 0.85,\ 3.59,\ 15.31]^T$.

Exercise 3.17

Solution. In both cases the value function of the policy is given by the vector $V^\pi = [1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 10]^T$.

Exercise 3.20

Solution. The agent has 2^7 deterministic stationary policies available to it. When $\gamma < 1$, the optimal policy is unique and the action in each state is to *"try right"*. If $\gamma = 1$, the optimal policy is not unique. All policies lead to infinite reward and are hence optimal.