Reinforcement Learning Lecture Notes: Theory of Discounted MDPs

Mohammad Sadegh Talebi
University of Copenhagen
sadegh.talebi@di.ku.dk

Initial Draft: April 28, 2022
This Version: February 6, 2024

1 Introduction

In this note, we study infinite-horizon discounted Markov Decision Processes (or discounted MDPs, for short). [1, 2] provide an elegant treatment of this subject. Discounted MDPs constitute the most well-studied models in the theory of MDPs. They admit a complete theory and are by now very well-understood. In terms of applications, they are appealing as a plethora of decision making problems arising in economics and operations research can be modeled using discounted MDPs.

2 Discounted MDPs: Definition

An **infinite-horizon discounted MDP** M is a tuple $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where:

- $\mathcal{S}$ denotes the **state-space**, which is the set of all possible states the agent may occupy;
- $\mathcal{A} = \cup_{s\in\mathcal{S}}\mathcal{A}_s$ denotes the **action-space**, where $\mathcal{A}_s$ is the set of actions available to the agent at state $s \in \mathcal{S}$;
- $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{A})$ denotes the **transition function** such that $P(\cdot|s, a)$ is the probability distribution of next-state when action $a \in \mathcal{A}_s$ in state $s \in \mathcal{S}$ is selected;
- $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denotes the **reward function** such that $R(s, a)$ denotes the probability distribution of a (possibly random) reward obtained when choosing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$; and
- $\gamma \in (0, 1)$ is a **discount factor**.

The state-space $\mathcal{S}$ and action-space $\mathcal{A}$ maybe finite, countably infinite, or even continuous. For example, consider a decision making scenario where the temperature of a certain physical location defines the state. Assuming that the temperature of the location ranges between $-30°\mathrm{C}$ and $+70°\mathrm{C}$, then $\mathcal{S} = [-30, +70]$, i.e., a continuous state-space. On the other hand, for practical purposes, we might be interested in $0.1°\mathrm{C}$ resolution of the temperature –for instance, our thermometer's reading could support up to this resolution. Then, we may consider $\mathcal{S} = \{-30, -29.90, \ldots, 69.90, 70\}$, which is a finite state-space.

Remark 1 *In the case of finite or countably infinite states (resp. actions), the state-space (resp. action-space) is generally termed* **discrete***. An MDP is said to be discrete if both state-space and action-space are discrete. this note, we restrict the attention to discrete MDPs. It is worth noting, however, that some results we present hold in greater generality.*

In a discrete MDP M, the transition function P is a collection of discrete probability distributions or probability vectors. $P(\cdot|s, a)$ denotes the probability distribution of next-states for $s \in \mathcal{S}$ and $a \in \mathcal{A}_s$. Hence, $\sum_{x\in\mathcal{S}} P(x|s, a) = 1$. We may also treat $P(\cdot|s, a)$ as a probability vector of dimension $S := |\mathcal{S}|$.

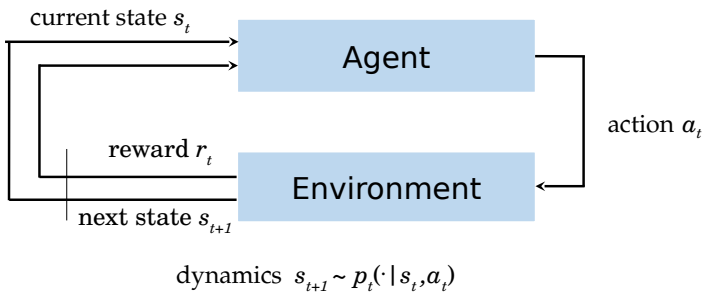

Figure 1: The interaction between the agent and the environment (MDP)

The Interaction between MDP and the Agent. Consider a time slotted system, where time is divided into slots (or periods) of identical lengths indexed by $t \in \mathbb{N}$.[1] The interaction between the agent and the MDP (environment) proceeds as follows. The agent starts in some initial state s_1, which is determined by Nature, according to some distribution, which might be known or unknown to the agent. At each time period $t \in \mathbb{N}$, the agent is in state s_t. At the beginning of time period t, she chooses an action $a_t \in \mathcal{A}_{s_t}$ according to some (possibly randomized) action selection rule and executes it. Upon executing a_t in s_t, the agent receives a (possibly random) reward r_t from the environment, which is sampled from the reward distribution $R(s_t, a_t)$, namely, $r_t \sim R(s_t, a_t)$. The agent may receive r_t immediately, or later during the period –in one shot or multiple phases. However, we do assume that she will earn the entire r_t until the end of the t-th period and before deciding a_{t+1}. Also, upon executing a_t in s_t, the environment generates a next-state s_{t+1} according to $P(\cdot|s_t, a_t)$, i.e., $s_{t+1} \sim P(\cdot|s_t, a_t)$. Then, once the new slot begins, the environment transits to s_{t+1} and a new round of decision step begins. This process continues indefinitely. This interaction is depicted in Figure 2.

This interaction generates a **history** (or **trajectory**) h_t at each time t:

$$h_t = (s_1, a_1, s_2, a_2, \ldots, s_{t-1}, a_{t-1}, s_t)\,.$$

The goal of the agent is to maximize the total discounted rewards collected during her interaction with the environment. More formally, she seeks to solve:

$$\max_{\text{strategies}} \lim_{N\to\infty} \mathbb{E}\Big[\sum_{t=1}^{N} \gamma^{t-1} r_t\Big]$$

for any choice of initial state where the expectation is taken with respect to possible randomness in the initial states, the randomness in the received rewards and state transitions, and possible randomization in the action selection rule. Here, maximization is done over all possible decision making (action selection) strategies. This defines a notion of optimality, which is called **the expected total discounted reward optimality criterion**.

The Markov Property. MDPs adhere to the **Markov property** (hence, the name Markov decision process). More concretely, at each time t, the probability distribution of next-state s_{t+1} is fully determined by the current state s_t and action a_t, and conditionally independent of the past history of the process. Precisely speaking, given (or conditioned on) the current state-action pair (s_t, a_t), s_{t+1} is *independent of the past* state-action pairs. Formally,

$$\mathbb{P}\Big(s_{t+1} = s' \Big| s_1, a_1, \ldots, s_{t-1}, a_{t-1}, s_t, a_t\Big) = \mathbb{P}\Big(s_{t+1} = s' \Big| s_t, a_t\Big) = P(s'|s_t, a_t)\,.$$

[1] If the slots are of different lengths, the underlying decision process becomes an instance of a **semi-Markov decision process (SMDP)**. In an SMDP, the time it takes to complete the execution of an action – termed holding time – could be stochastic, but whose distribution is determined by the current state-action pair.

Similarly, the distribution of r_t at time t is fully determined by the current state-action pair (s_t, a_t). In other words, letting $f_{s,a}$ denote the density of $R(s,a)$, we have for $X \in \mathbb{R}$,

$$\mathbb{P}\Big(r_t \in X \Big| s_1, a_1, \ldots, s_{t-1}, a_{t-1}, s_t, a_t\Big) = \mathbb{P}\Big(r_t \in X \Big| s_t, a_t\Big) = \int_X f_{s_t,a_t}(x)\mathrm{d}x\,.$$

The Markov property implies that the notion of state (and action) is rich enough to contain all the necessary information one needs to predict the next-state and (one-step) reward.

Remark 2 *In general, the reward and transition functions might be time-varying making the MDP* **non-stationary**. *Non-stationary MDPs could be useful in modeling several applications. However, in the rest of this note, we assume stationary MDPs, that is those with stationary reward and transition functions.*

Assumption 1 *In the rest of this note, we assume* **deterministic** *reward functions. That is, we assume choosing a in s yields $r = r(s,a) = R(s,a)$. We further assume that (deterministic) rewards are uniformly bounded, namely there exists a constant $R_{\max} < \infty$ such that*

$$\sup_{s\in\mathcal{S}} \sup_{a\in\mathcal{A}_s} |r(s,a)| < R_{\max}\,.$$

Considering deterministic rewards may sound restrictive — that's true. We stress however that almost all the presented results below will hold for the case of stochastic rewards under very mild assumptions. In the case of stochastic rewards, it suffices to replace $R(s,a)$ in the results with its mean $\mu(s,a) := \mathbb{E}_{r\sim R(s,a)}[r]$.

2.1 Examples

We provide a few examples of MDPs.

Example 1. *A manufacturer at each time period receives an order for a given product with probability α. At any period, she has the choice of processing all the unfilled orders in a batch, or process no order at all. The cost per unfilled order at any period is $c > 0$, and the setup cost to process unfilled order is $K > 0$. Assume that the total number of orders that can remain unfilled is n, and that there is a discount factor $\gamma < 1$. The goal of the manufacturer is to find an order processing strategy that has minimal expected cost.*

We can model this problem as a discounted MDP. Define the state as the number of unfilled orders at the beginning of each period. Hence, the state-space is $\mathcal{S} = \{0, 1, \ldots, n\}$. For $s \neq 0, n$, we have $\mathcal{A}_s = \{C, \overline{C}\}$, where C (resp. $\overline{C}$) corresponds to processing unfilled orders (resp. processing no order). We have $\mathcal{A}_0 = \{\overline{C}\}$ and $\mathcal{A}_n = \{C\}$. The rewards are given by:

$$\begin{aligned} &r(i,C) = -K, \quad r(i,\overline{C}) = -ci, \quad i = 1,\ldots,n-1,\\ &r(0,\overline{C}) = 0, \quad r(n,C) = -K. \end{aligned}$$

Transition probabilities are given by:

$$\begin{aligned} &P(0|i,C) = 1-\alpha, \quad P(1|i,C) = \alpha, \quad i = 1,2,\ldots,n-1,\\ &P(i|i,\overline{C}) = 1-\alpha, \quad P(i+1|i,\overline{C}) = \alpha, \quad i = 1,2,\ldots,n-1,\\ &P(0|n,C) = 1-\alpha, \quad P(1|n,C) = \alpha,\\ &P(0|0,\overline{C}) = 1-\alpha, \quad P(1|0,\overline{C}) = \alpha. \end{aligned}$$

Example 2. *A job seeker receives a job offer at each time period, which she may accept or reject. The offered policy takes one of n possible values $w_1, \ldots, w_n$ with given probabilities independently of preceding offers. If she accepts the offer, she must keep the job for the rest of her life. If she rejects the offer, she receives unemployment compensation c for the current period and is eligible to accept future offers. Assume that income is discounted by a factor $\gamma < 1$. The job seeker is interested in a strategy maximizing her income. We can model this task as a discounted MDP. The state-space is $\mathcal{S} = \{s_1, s_2, \ldots, s_n, s'_1, \ldots, s'_n\}$, where for each i, s_i corresponds to the case where the job seeker is unemployed and being offered a salary w_i, and s'_i corresponds to the case where she is employed at a salary level w_i. Let q_i be the probability of an offer at salary level w_i at any one period. We have $\mathcal{A}_{s_i} = \{C, \overline{C}\}$ for all i, where C denotes the action corresponding to accepting an offer ($\overline{C}$ rejecting the offer). Furthermore, $\mathcal{A}_{s'_i} = \{X\}$ for all i, where X indicates continuation of the job. Rewards are given by: For all i, $r(s_i, C) = w_i$, $r(s_i, \overline{C}) = c$, and $r(s'_i, X) = w_i$. Transition probabilities are given by:*

$$P(s'_i|s'_i, X) = 1, \quad P(s'_i|s_i, C) = 1, \quad P(s_j|s_i, \overline{C}) = q_j, \quad \forall i.$$

Example 3. *Consider the RiverSwim environment, originally presented in [3] (see Figure 2.1). This MDP models a situation where an agent is swimming against a current. This MDP has a state-space $\mathcal{S} = \{s_1, s_2, \ldots, s_L\}$, where in each state, there are two actions: $\mathcal{A} = \{$left, right$\}$. In each state $s_i, (i \neq 1)$, taking* left *deterministically brings the agent to s_{i-1} and gives no reward. Taking* left *in s_1 leaves the state unchanged and results in a (deterministic) reward of 0.05. Taking* right *in $s_i, (i \neq 1, L)$, the agent either moves to s_{i+1} (w.p. 0.4), remains in the current state (w.p. 0.55), or ends up in s_{i-1} (w.p. 0.05). The corresponding reward is 0, except in state s_L, where the agents receives a (deterministic) reward of 1.*

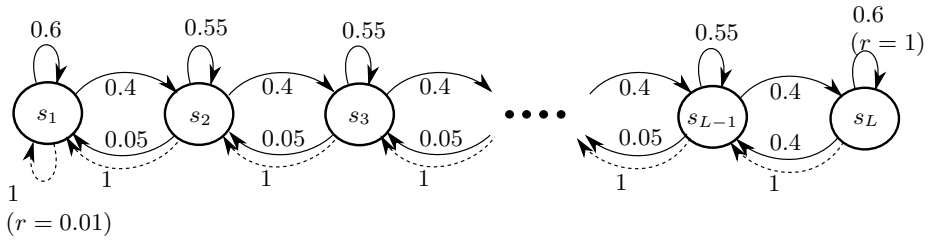

Figure 2: The RiverSwim MDP

3 Policy and Value Function

3.1 Notions of Policy

In decision making, the agent chooses actions based on a **policy**. In other words, a policy determines the decision making strategy of the agent. In simple words, a policy is a mapping that prescribes what action to choose at various states, and in doing so, it may possibly use all the currently available information as input. We categorize policies based on the amount of information they use to choose an action, and based on the way they output an action (i.e., deterministically or in a randmized way). A policy can be:

- *Randomized or deterministic:* A **randomized** policy[2] outputs a probability distribution over actions whereas a **deterministic** policy outputs a single action.
- *History-dependent or stationary:* A **history-dependent** policy determines the action based on the history (hence, it is necessarily time-varying, or non-stationary), whereas a **stationary** policy uses the current state only and does not vary over time.

[2]Some texts call it stochastic policy.

Hence, we identify *four classes* of policies as follows. A **history-dependent randomized** policy is a mapping of the form $\pi : \mathcal{H} \to \Delta(\mathcal{A})$, where $\mathcal{H}$ is the set of all possible histories and $\Delta(\mathcal{A})$ denotes the set of probability distributions over $\mathcal{A}$. Here, π maps a history $h_t \in \mathcal{H}$ of observations up to time t (i.e., a sequence $(s_1, a_1, r_1, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t)$) to a probability distribution over $\mathcal{A}$. Let Π^{HR} denote the set of all history-dependent randomized policies. (Obviously, Π^{HR} depends on the MDP at hand.) We write $a_t \sim \pi(h_t)$ to indicate that a_t is sampled from $\pi(h_t)$. Such a policy could be complicated to use in practice since it depends on the history and it prescribes choosing actions in a randomized manner. A **history-dependent deterministic policy** has the form $\pi : \mathcal{H} \to \mathcal{A}$. Hence, it prescribes $a_t = \pi(h_t)$. Let Π^{HD} denote the set of all history-dependent deterministic policies. A **stationary randomized policy** is a mapping $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ and it uses the current state to choose an action; hence $a_t \sim \pi(s_t)$. Let Π^{SR} denote the set of all stationary randomized policies. Finally, a **stationary deterministic policy** is a mapping $\pi : \mathcal{S} \to \mathcal{A}$, and $a_t = \pi(s_t)$. Let Π^{SD} denote the set of all stationary deterministic policies.

We have $\Pi^{\text{SD}} \subset \Pi^{\text{SR}} \subset \Pi^{\text{HR}}$ and $\Pi^{\text{SD}} \subset \Pi^{\text{HD}} \subset \Pi^{\text{HR}}$. Table 3.1 summarizes the notions of policies above.

	deterministic	ramdomized
stationary	$\pi : \mathcal{S} \to \mathcal{A}$	$\pi : \mathcal{S} \to \Delta(\mathcal{A})$
history-dependent	$\pi : \mathcal{H} \to \mathcal{A}$	$\pi : \mathcal{H} \to \Delta(\mathcal{A})$

Table 1: Various notions of policies in a discounted MDP

Induced Markov Chains and Markov Reward Processes. A run of a policy π on a discounted MDP M (with stationary reward and transition functions) yields a sequence $\big((X_t, r(X_t, Y_t))\big)_{t\in\mathbb{N}}$, where X_t denotes the state at time t, Y_t denotes the action at time t, and $r(X_t, Y_t)$ represents the reward received in (X_t, Y_t). The sequence $\big((X_t, r(X_t, Y_t))\big)_{t\in\mathbb{N}}$ is, in general, called a discrete time **reward process**[3]. In the case $\pi \in \Pi^{\text{SR}}$, $\big((X_t, r(X_t, Y_t))\big)_{t\in\mathbb{N}}$ is called a **Markov Reward Process (MRP)**. Further, the sequence $(X_t)_{t\in\mathbb{N}}$ is a Markov chain.

Precisely speaking, let $\pi \in \Pi^{\text{SR}}$. We denote by P^π the S-by-S transition probability of the Markov chain induced by π on M. The elements of P^π are given by:

$$P^\pi_{s,s'} = \sum_{a\in\mathcal{A}_s} P(s'|s,a)\pi(a|s)\,, \quad s, s' \in \mathcal{S}.$$

Also for $\pi \in \Pi^{\text{SR}}$, we define $r^\pi \in \mathbb{R}^S$ to be the reward vector induced by π on M, defined by

$$r^\pi(s) = \sum_{a\in\mathcal{A}_s} r(s,a)\pi(a|s)\,, \quad s \in \mathcal{S}.$$

If π is stationary deterministic, then $P^\pi_{s,s'} = P(s'|s,\pi(s))$ and $r^\pi(s) = r(s,\pi(s))$.

In summary, every policy $\pi \in \Pi^{\text{SR}}$ induces an MRP on M. Note, however, that the reward process induced by a non-stationary policy *may not* adhere to the Markov property.

3.2 Value and Q-Value Functions

Let $\pi \in \Pi^{\text{HR}}$ in M. The **state value function** of π (or for short, the **value function** or simply the **value** of π), denoted by V^π_M, is a mapping from the state-space $\mathcal{S}$ to reals; formally, $V^\pi_M : \mathcal{S} \to \mathbb{R}$. It measures the expected total discounted reward of π when the process starts from its argument. For $s \in \mathcal{S}$, $V^\pi_M(s)$ is defined as the expected sum of discounted rewards obtained by always following π, when starting from s. Formally:

$$V^\pi_M(s) := \lim_{N\to\infty} \mathbb{E}^\pi\Big[\sum_{t=1}^{N} \gamma^{t-1} r(s_t, a_t)\Big| s_1 = s\Big]\,,$$

[3]This terminology is also valid in the case of a generic (i.e., non-Markovian) decision process.

where $\mathbb{E}^\pi$ indicates that the expectation is taken over trajectories generated by π. Here, the subscript M indicates that the underlying MDP is M.

In the case of bounded rewards (i.e., $\sup_{s,a} |r(s,a)| < \infty$), the above limit exists and also interchanging the expectation and the limit is allowed. Hence, under this assumption, we write

$$V_M^\pi(s) := \mathbb{E}^\pi \Big[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t) \Big| s_1 = s \Big] .$$

We often use the following more concise form:

$$V_M^\pi(s) := \mathbb{E}_s^\pi \Big[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, s_t) \Big] . \tag{1}$$

Further note that under the bounded reward assumption, it is straightforward to verify that

$$V_M^\pi(s) \leq \frac{R_{\max}}{1-\gamma}, \qquad \forall s \in \mathcal{S},$$

or alternatively, $\|V_M^\pi\|_\infty \leq R_{\max}/(1-\gamma)$.

The **state-action value function** (a.k.a. **action-value function** and **Q-value**) of a policy π, denoted by Q^π, is a mapping from the state-action space to reals. It is defined as:

$$Q_M^\pi(s,a) := \lim_{N \to \infty} \mathbb{E}^\pi \Big[\sum_{t=1}^{N} \gamma^{t-1} r(s_t, a_t) \Big| s_1 = s, a_1 = a \Big] .$$

Under the bounded reward assumption as above, we can write:

$$Q_M^\pi(s,a) := \mathbb{E}^\pi \Big[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t) \Big| s_1 = s, a_1 = a \Big] . \tag{2}$$

Intuitively, $V_M^\pi(s)$ measures the sum of future discounted rewards (in expectation) when the agent starts in state s and follows policy π. Similarly, $Q_M^\pi(s,a)$ measures the sum of future discounted rewards (in expectation) when the agent starts in state s and takes action a in the first step (with possibly $a \neq \pi(s)$), and then follows policy π afterwards.

Remark 3 *Whenever it is clear from the context that the underlying MDP is M, we drop the dependence of various quantities and functions on M (e.g., write V^π to denote V_M^π).*

3.3 Optimal Policy and Values

Solving a discounted MDP M amounts to solving the following optimization problem:

$$V^\star(s) = \sup_{\pi \in \Pi^{\text{HR}}} V^\pi(s) ,$$

for all $s \in \mathcal{S}$. $V^\star : \mathcal{S} \to \mathbb{R}$ is called the **optimal value** function. If there exists a policy $\pi^\star$ such that $V^{\pi^\star}(s) = V^\star(s)$ for all $s \in \mathcal{S}$, then $\pi^\star$ is called an **optimal policy** in M. A policy π is **ε-optimal** for $\varepsilon > 0$ if $V^\pi(s) \geq V^\star(s) - \varepsilon$ for all $s \in \mathcal{S}$.

Optimization over the space of all history-dependent randomized policies could be cumbersome, if possible at all. However, as we will see, we can restrict attention only to stationary deterministic policies. In other words, for any discrete MDP, there always exists a stationary deterministic optimal policy. Further, it can be shown that in any discrete MDP, there exists an ε-optimal deterministic stationary policy for all $\varepsilon > 0$.

4 Equivalence between Discounted MDPs and MDPs with Random Horizon

The above formulation indicates the use of discounted MDPs for decision making problems where the horizon is infinite and rewards are discounted (with rate γ). However, discounted MDPs admit another nice interpretation (and use) in problems where rewards are not discounted and the problem horizon is finite. Consider a scenario where the problem horizon ν is random, *but independent of agent's actions*. In particular, we assume that the random horizon ν has a geometric distribution with success probability $\gamma \in [0, 1)$, i.e., $\nu \sim \text{Geo}(\gamma)$. In other words,

$$\mathbb{P}(\nu = n) = \gamma^{n-1}(1-\gamma)\,, \quad n \in \mathbb{N}.$$

(Examples of this scenario abound.) Let $V^{\pi}_{M,\nu}$ denote the value of policy π in this model defined by:

$$V^{\pi}_{M,\nu}(s) = \mathbb{E}^{\pi}_{s}\Big[\mathbb{E}_{\nu}\Big[\sum_{t=1}^{\nu} r(s_t, a_t)\Big]\Big]\,,$$

where $\mathbb{E}_{\nu}$ denotes the expectation w.r.t. the randomness in the horizon ν. We can model this scenario using a discounted MDP $M' = (\mathcal{S}', \mathcal{A}', P', R')$ as follows. Let's augment a terminal state T to the original state-space $\mathcal{S}$, i.e., $\mathcal{S}' = \mathcal{S} \cup \{\mathsf{T}\}$. There will be a single absorbing action a_{T} at state T, i.e., $\mathcal{A}'_{\mathsf{T}} = \{a_{\mathsf{T}}\}$, and the rest of actions in M' will be the same as in M. The transition and reward functions of M' are given by:

$$P'(j|s,a) = \begin{cases} \gamma P(j|s,a) & j \neq \mathsf{T}, s \neq \mathsf{T} \\ 1-\gamma & j = \mathsf{T}, s \neq \mathsf{T} \\ 1 & j = \mathsf{T}, s = \mathsf{T}, a = a_{\mathsf{T}}. \end{cases}$$

$$R'(s,a,j) = \begin{cases} R(s,a,j) & j \neq \mathsf{T} \\ 0 & j = \mathsf{T}, \text{ or } s = \mathsf{T}, a = a_{\mathsf{T}}. \end{cases}$$

The following result relates $V^{\pi}_{M,\nu}$ to V^{π}_{M}.

Proposition 1 ([1, Proposition 5.3.1]) *Suppose the bounded reward assumption holds and $\nu \sim$ Geo(γ). Then, for any policy π, $V^{\pi}_{\nu,M}(s) = V^{\pi}_{M'}(s)$ for all $s \in \mathcal{S}$.*

Proof. Since $\nu \sim \text{Geo}(\gamma)$, we have

$$V^{\pi}_{M,\nu}(s) = \mathbb{E}^{\pi}_{s}\Big[\mathbb{E}_{\nu}\Big[\sum_{t=1}^{\nu} r(s_t, a_t)\Big]\Big] = \mathbb{E}^{\pi}_{s}\Big[\sum_{n=1}^{\infty}\sum_{t=1}^{n} r(s_t, a_t)(1-\gamma)\gamma^{n-1}\Big]\,.$$

In view of the bounded reward assumption and $\gamma < 1$, the series converges. Hence, by changing the order of summations, we have

$$V^{\pi}_{M,\nu}(s) = \mathbb{E}^{\pi}_{s}\Big[\sum_{t=1}^{\infty} r(s_t, a_t)\sum_{n=t}^{\infty}(1-\gamma)\gamma^{n-1}\Big] = \mathbb{E}^{\pi}_{s}\Big[\sum_{t=1}^{\infty} r(s_t, a_t)\gamma^{t-1}\Big] = V^{\pi}_{M'}(s)\,,$$

where we used the identity $\sum_{n=t}^{\infty}\gamma^{n-1} = \gamma^{t-1}/(1-\gamma)$. □

5 Bellman's Equation and Operators

The following proposition states that the value function of any stationary policy satisfies the Bellman equation:

Proposition 2 *Let $\pi \in \Pi^{\text{SR}}$. For all $s \in \mathcal{S}$,*

$$V^\pi(s) = \mathbb{E}_{a\sim\pi(s)}[r(s,a)] + \gamma \mathbb{E}_{a\sim\pi(s)}\Big[\sum_{x\in\mathcal{S}} P(x|s,a)V^\pi(x)\Big].$$

Equivalently, $V^\pi = r^\pi + \gamma P^\pi V^\pi$.

For a deterministic policy $\pi \in \Pi^{\text{SD}}$, we have:

$$V^\pi(s) = r(s,\pi(s)) + \gamma \sum_{x\in\mathcal{S}} P(x|s,\pi(s))V^\pi(s)$$

for all $s \in \mathcal{S}$.

Proof. Let $\pi \in \Pi^{\text{SR}}$ and $s \in \mathcal{S}$. We have

$$\begin{aligned}
V^\pi(s) &= \mathbb{E}\Big[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t,a_t)\Big| s_1 = s, a_t \sim \pi(s_t), \forall t\Big] \\
&= \mathbb{E}_{a\sim\pi(s)}[r(s,a)] + \mathbb{E}\Big[\sum_{t=2}^{\infty} \gamma^{t-1} r(s_t,a_t)\Big| s_1 = s, a_t \sim \pi(s_t), \forall t\Big] \\
&= \mathbb{E}_{a\sim\pi(s)}[r(s,a)] + \gamma \sum_{x\in\mathcal{S}} \mathbb{P}(s_2 = x|s_1 = s, a_1 \sim \pi(s_1))\mathbb{E}\Big[\sum_{t=2}^{\infty} \gamma^{t-2} r(s_t,a_t)\Big| s_2 = x, a_t \sim \pi(s_t), \forall t\Big] \\
&= \mathbb{E}_{a\sim\pi(s)}[r(s,a)] + \gamma \sum_{x\in\mathcal{S}} \mathbb{P}(s_2 = x|s_1 = s, a_1 \sim \pi(s_1))V^\pi(x) \\
&= \mathbb{E}_{a\sim\pi(s)}[r(s,a)] + \gamma \mathbb{E}_{a\sim\pi(s)}\Big[\sum_{x\in\mathcal{S}} P(x|s,a)V^\pi(x)\Big].
\end{aligned}$$

□

Let $\pi \in \Pi^{\text{SR}}$. The **Bellman operator** associated to π is a mapping $\mathcal{T}^\pi : \mathbb{R}^S \to \mathbb{R}^S$ defined as follows: For any function $f : \mathcal{S} \to \mathbb{R}^S$,

$$\mathcal{T}^\pi f := r^\pi + \gamma P^\pi f\,,$$

where $P^\pi f$ can be interpreted as multiplying the matrix P^π by the vector f: $[P^\pi f]_s = \sum_{x\in\mathcal{S}} P^\pi_{s,x} f(x)$.

Intuitively, $\mathcal{T}^\pi$ is the value of π for the same one-stage problem. The above relation indicates that $\mathcal{T}^\pi$ is applied to (or *operates on*) the bounded functions defined on $\mathcal{S}$ and returns another bounded function defined on $\mathcal{S}$. Using this notation, we can rewrite the relation $V^\pi = r^\pi + \gamma P^\pi V^\pi$ concisely as:

$$V^\pi = \mathcal{T}^\pi V^\pi.$$

In other words, V^π is the *unique* fixed-point of the operator $\mathcal{T}^\pi$.

We have, for all $s \in \mathcal{S}$,

$$Q^\pi(s,\pi(s)) = V^\pi(s)$$

The notion of the Bellman operator can be extended to action value functions (Q-functions). For any function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^S$,

$$\mathcal{T}^\pi f(s,a) = \mathbb{E}_{a\sim\pi(s)}[r(s,a)] + \gamma \mathbb{E}_{a\sim\pi(s)}\Big[\sum_{y} P(y|s,a)f(s,a)\Big], \quad (s,a) \in \mathcal{S}\times\mathcal{A}.$$

Hence, we have $Q^\pi = \mathcal{T}^\pi Q^\pi$. In other words, Q^π is the fixed point of the operator $\mathcal{T}^\pi$ (for Q-function).

We have (see, e.g., [1, Theorem 6.1.1]):

Theorem 1 *For any stationary policy π and $\gamma \in [0,1)$, V^π is the unique solution (in the space of functions with bounded norms) of the fixed-point equation $f = \mathcal{T}^\pi f$. Furthermore,*

$$V^\pi = (I - \gamma P^\pi)^{-1} r^\pi\,.$$

Proof. Rewriting the Bellman equation for π yields $(I - \gamma P^\pi)V^\pi = r^\pi$. To show that $I - \gamma P^\pi$ is invertible, we show that the spectral radius σ of γP^π is less than 1, i.e., $\sigma(\gamma P^\pi) < 1$. We have $\sigma(P^\pi) \le \|P^\pi\|_\infty = 1$. Hence, $\sigma(\gamma P^\pi) \le \|\gamma P^\pi\|_\infty = \gamma < 1$. Hence, $(I - \gamma P^\pi)^{-1}$ exists and thus, $V^\pi = (I - \gamma P^\pi)^{-1} r^\pi$. □

5.1 Bellman's Optimality Equation and Operators

Definition 1 *An operator (or mapping) $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}^n$ is called a* **contraction mapping** *if there exists $\kappa \in [0, 1)$ such that for all $v, v' \in \mathbb{R}^n$,*

$$\|\mathcal{L}v - \mathcal{L}v'\| \leq \kappa \|v - v'\| .$$

κ is called the modulus of $\mathcal{L}$. Alternatively, $\mathcal{L}$ is called a κ-contraction mapping.[4]

We have the following fixed-point theorem for contraction mappings:

Theorem 2 (Banach Fixed-Point Theorem) *Suppose $\mathcal{L}$ is a contraction mapping. Then*

(i) there exists a unique $v^\star \in \mathbb{R}^n$ such that $\mathcal{L}v^\star = v^\star$;

(ii) for any $v_0 \in \mathbb{R}^n$, the sequence $(v_n)_{n\geq 0}$ with $v_{n+1} = \mathcal{L}v_n = \mathcal{L}^{n+1}v_0$ for $n \geq 0$ converges to $v^\star$.

We have:

Lemma 1 *The Bellman operators $\mathcal{T}^\pi$ and $\mathcal{T}$ are γ-contraction mappings w.r.t. the L_∞-norm. In other words, for any $v, v' \in \mathbb{R}^n$,*

$$\|\mathcal{T}^\pi v - \mathcal{T}^\pi v'\|_\infty \leq \gamma \|v - v'\|_\infty ,$$
$$\|\mathcal{T} v - \mathcal{T} v'\|_\infty \leq \gamma \|v - v'\|_\infty .$$

Proof. For the second statement, we have:

$$\begin{aligned}
&\|\mathcal{T}v - \mathcal{T}v'\|_\infty \\
&\quad = \max_s \Big| \max_{a\in\mathcal{A}_s} \Big(r(s,a) + \gamma \sum_j P(j|s,a)v(j) \Big) - \max_{a\in\mathcal{A}_s} \Big(r(s,a) + \gamma \sum_j P(j|s,a)v'(j) \Big) \Big| \\
&\quad \leq \max_s \max_{a\in\mathcal{A}_s} \Big| \gamma \sum_j P(j|s,a)(v(j) - v'(j)) \Big| \\
&\quad \leq \gamma \max_s \max_{a\in\mathcal{A}_s} \max_j |v(j) - v'(j)| \sum_j P(j|s,a) = \gamma \|v - v'\|_\infty,
\end{aligned}$$

where the first inequality uses the inequality $|\max_x f(x) - \max_x g(x)| \leq \max_x |f(x) - g(x)|$, valid for real-valued functions f and g. The proof of the first statement follows a similar argument. □

We now provide two theorems that are fundamental results in the theoretical foundations of discounted MDPs:

Theorem 3 *Suppose the state-space $\mathcal{S}$ is finite. Then there exists a stationary deterministic policy which is optimal.*

Theorem 3 implies that when seeking an optimal policy in a discounted MDP with a finite state-space, we can restrict our attention to those in Π^{SD}. This further implies that for finite $\mathcal{S}$:

$$\sup_{\pi\in\Pi^{\text{HR}}} V^\pi = \sup_{\pi\in\Pi^{\text{SD}}} V^\pi = \max_{\pi\in\Pi^{\text{SD}}} V^\pi$$

Theorem 4 *A stationary deterministic policy π is optimal if and only if*

$$\mathcal{T}^\pi V^\star = \mathcal{T} V^\star$$

Equivalently, π is optimal if and only if it attains the maximum in the Bellman optimality equations: For all $s \in \mathcal{S}$,

$$\pi(s) \in \operatorname*{argmax}_{a\in\mathcal{A}_s} \Big(r(s,a) + \gamma \sum_x P(x|s,a)V^\star(x) \Big) .$$

[4] In the case of $\kappa = 1$, the mapping is said to be **non-expansive**.

Example 1 (Continuation). *Consider Example 1 and suppose we are interested in finding a processing strategy that minimizes the total (discounted) cost. The Bellman's equation takes the form:*

$$V(i) = \min\{K + \gamma(1-\alpha)V(0) + \gamma\alpha V(1), cs + \gamma(1-\alpha)V(i) + \gamma pV(i+1)\}, \;\; i = 0, 1, \ldots, n-1,$$
$$V(n) = K + \gamma(1-p)V(0) + \gamma pV(1), \;\; i = n.$$

We show below through induction that the optimal cost $V(i)$ is monotonically non-decreasing in i. Hence, if processing a batch of m orders is optimal, that is,

$$K + \lambda(1-p)V(0) + \lambda V(1) \leq cm + \lambda(1-p)V(m) + \lambda pV(m+1),$$

then processing a batch of $m+1$ orders is also optimal. Therefore, the optimal policy is a threshold policy, which decides to process the orders if their number exceeds some threshold integer $m^\star$ which satisfies:

$$K + \lambda(1-p)V(0) + \lambda V(1) \leq cm^\star + \lambda(1-p)V(m^\star) + \lambda pV(m^\star + 1).$$

Suppose that $V_k(i+1) \geq V_k(i)$ for all i. We will show that $V_{k+1}(i+1) \geq V_{k+1}(i)$ for all i. Consider first the case $i+1 < n$. Then by induction hypothesis, we have that

$$c(i+1) + \lambda(1-p)V_k(i+1) + \lambda pV_k(i+2) \geq ci + \lambda(1-p)V_k(i) + \lambda pV_k(i+1).$$

Define for any scalar γ, $F_k(\gamma) = \min\{K + \lambda(1-p)V_k(0) + \lambda pV_k(1), \gamma\}$. Since $F_k(\gamma)$ is monotonically increasing in γ, from the above equations we have that

$$\begin{aligned} V_{k+1}(i+1) &= F_k\Big(c(i+1) + \lambda(1-p)V_k(i+1) + \lambda pV_k(i+2)\Big) \\ &\geq F_k\Big(ci + \lambda(1-p)V_k(i) + \lambda pV_k(i+1)\Big) = V_{k+1}(i). \end{aligned}$$

Finally consider the case $i+1 = n$. It follows that

$$\begin{aligned} V_{k+1}(n) &= K + \lambda(1-p)V_k(0) + \lambda pV_k(1) \\ &\geq F_k\Big(ci + \lambda(1-p)V_k(i) + \lambda pV_k(i+1)\Big) = V_{k+1}(n-1) \end{aligned}$$

and hence the induction is complete.

6 Algorithms for Finding Optimal Policies

In this section, we present algorithms for solving discounted MDPs.

6.1 Value Iteration

The Value Iteration algorithm is perhaps the most well-known method for solving discounted MDPs. Value Iteration, often abbreviated as `VI`, This algorithm has been known with other names such as *successive approximation* and *backward induction*.

`VI` has been around since the early days of MDPs, and so far several variants of it have been developed. The most basic variant of `VI` is an iterative algorithm, which outputs an ε-optimal stationary policy within a finite number of iterations, where $\varepsilon > 0$ is an input parameter of the algorithm. Precisely speaking, `VI` proceeds as follows. It takes as input a parameter $\varepsilon > 0$ and a vector V_0, which serves an initial approximate of $V^\star$. — V_0 can be chosen to be $\mathbf{0}$. At each iterate n, `VI` maintains an approximate V_n of the optimal value function $V^\star$, which is updated as follows:

$$V_{n+1} = \mathcal{T}V_n$$

where $\mathcal{T}$ denotes the optimal Bellman operator. This is repeated until $\|V_{n+1} - V_n\| < \frac{\varepsilon(1-\gamma)}{2\gamma}$. In other words, the algorithm keeps refining the values until V_{n+1} is close (in norm) to V_n. Finally, it outputs the

Algorithm 1 Value Iteration (VI)

Input: ε

Initialize: Select $V_0 \in \mathbb{R}^S$, $V_1 = r_{\max}/(1-\gamma)\mathbf{1}$, and set $n = 0$.

while $\|V_{n+1} - V_n\| \geq \frac{\varepsilon(1-\gamma)}{2\gamma}$ **do**

Update, for each $s \in \mathcal{S}$,

$$V_{n+1}(s) = \max_{a \in \mathcal{A}_s} \Big(r(s,a) + \gamma \sum_{x \in \mathcal{S}} P(x|s,a) V_n(x) \Big)$$

Increment n.

end while

Output:

$$\pi^{\text{VI}}(s) = \arg\max_{a \in \mathcal{A}_s} \Big(r(s,a) + \gamma \sum_{x \in \mathcal{S}} P(x|s,a) V_n(x) \Big), \quad s \in \mathcal{S}$$

greedy policy w.r.t. $r(s,a) + \gamma \sum_{x \in \mathcal{S}} P(x|s,a) V_{n+1}(x)$ as the output policy. The pseudocode of VI is presented in Algorithm 1.

The following theorem summarizes the convergence guarantees of VI. In particular, it establishes that VI is a *globally convergent* method for finding an ε-optimal policy.

Theorem 5 ([1, Theorem 6.3.1]) *Let $(V_n)_{n \geq 0}$ be a sequence of value functions generated by VI with some $\varepsilon > 0$ starting from an arbitrary initial point $V_0 \in \mathbb{R}^S$. Then,*

(i) V_n converges to $V^\star$ in norm;

(ii) the algorithm stops after finitely many iterations;

(iii) π^{VI} is ε-optimal;

(iv) when convergence criterion is satisfied, $\|V_{n+1} - V^\star\| < \varepsilon/2$.

Proof. Parts (i) and (ii) are direct consequences of the Banach fixed point theorem (Theorem 2). To prove (iii), assume that n is such that the stopping criterion holds. Then,

$$\|V^{\pi^{\text{VI}}} - V^\star\| = \|V^{\pi^{\text{VI}}} - V_{n+1} + V_{n+1} - V^\star\| \leq \|V^{\pi^{\text{VI}}} - V_{n+1}\| + \|V_{n+1} - V^\star\|$$

By construction, $V^{\pi^{\text{VI}}}$ is a fixed point of $\mathcal{T}^{\pi^{\text{VI}}}$: $V^{\pi^{\text{VI}}} = \mathcal{T}^{\pi^{\text{VI}}} V^{\pi^{\text{VI}}}$. Moreover, $\mathcal{T}^{\pi^{\text{VI}}} V_{n+1} = \mathcal{T} V_{n+1}$.

Hence,

$$\begin{aligned}
\|V^{\pi^{\text{VI}}} - V_{n+1}\| &= \|\mathcal{T}^{\pi^{\text{VI}}} V^{\pi^{\text{VI}}} - V_{n+1}\| \\
&\leq \|\mathcal{T}^{\pi^{\text{VI}}} V^{\pi^{\text{VI}}} - \mathcal{T} V_{n+1}\| + \|\mathcal{T} V_{n+1} - V_{n+1}\| \\
&= \|\mathcal{T}^{\pi^{\text{VI}}} V^{\pi^{\text{VI}}} - \mathcal{T}^{\pi^{\text{VI}}} V_{n+1}\| + \|\mathcal{T} V_{n+1} - \mathcal{T} V_n\| \\
&\leq \gamma \|V^{\pi^{\text{VI}}} - V_{n+1}\| + \gamma \|V_{n+1} - V_n\|,
\end{aligned}$$

where the last inequality follows from Lemma 1. Rearranging the right-hand side yields

$$\|V^{\pi^{\text{VI}}} - V_{n+1}\| \leq \frac{\gamma}{1-\gamma} \|V_{n+1} - V_n\|.$$

We also have

$$\|V_{n+1} - V^\star\| \leq \frac{\gamma}{1-\gamma} \|V_{n+1} - V_n\|.$$

Hence, when the stopping criterion holds, then $\|V_{n+1} - V^\star\| < \varepsilon/2$, thus proving (iv). Further,

$$\|V^{\pi^{\text{VI}}} - V^\star\| \leq \frac{2\gamma}{1-\gamma} \|V_{n+1} - V_n\| < \varepsilon,$$

so that $V^{\pi^{\text{VI}}}(s) > V^\star(s) - \varepsilon$ for all $s \in \mathcal{S}$, and thus proving (iii). □

Remark 4 *Theorem 5 establishes that* `VI` *returns a ε-optimal policy within a finite number of iterations. When ε is small enough*[5]*, then the output policy is optimal. However,* `VI` *has no ability to determine whether the output policy is optimal or not.*

Computational Complexity. It is easy to verify that each iteration of `VI` has involves $O(S^2A)$ arithmetic calculations. The iteration complexity of `VI` depends on both ε and γ. The large the γ, the more iteration until the algorithm finds an ε-optimal policy.

6.2 Policy Iteration

The Policy Iteration algorithm is another popular classic method for solving discounted MDPs. Similar to `VI`, policy iteration has been around since the early days of MDPs, and so far several variants of it have been developed. We discuss the most basic variant of policy iteration (referred to as `PI`here), which is also known as *Howard's Policy Iteration* and was presented by Howard in 60s [4]. Another well-known variant of policy iteration is *Modified Policy Iteration* presented by Puterman and Shin in [5]. `PI` outputs an *optimal policy*, within a finite number of iterations. `PI` is an iterative algorithm, where in each iterate (until convergence) an refinement of the current approximation of the optimal policy $\pi^\star$ is performed. Specifically, each iterate n comprises two steps: The first step is *policy evaluation*, where the value V_n of the current policy π_n is computed. This is followed by a *policy improvement* step where a policy maximizing $r^\pi + \gamma P^\pi V_n$ is computed. The pseudocode of `PI` is presented in Algorithm 2.

Algorithm 2 Policy Iteration (`PI`)

Initialize: Select an aribitrary policy π_0, and set $n = 0$.
while $\pi_{n+1} \neq \pi_n$ **do**
Policy Evaluation: Find V_n, the value of π_n by solving

$$(I - \gamma P^{\pi_n})V_n = r^{\pi_n}$$

Policy Improvement: Choose π_{n+1} such that

$$\pi_{n+1}(s) \in \arg\max_{a \in \mathcal{A}_s} \Big(r(s,a) + \gamma \sum_{x \in \mathcal{S}} P(x|s,a) V_n(x) \Big)$$

and if possible, set $\pi_{n+1} = \pi_n$.
Increment n.
end while
Output: $\pi^\star = \pi_n$

A key property of `PI` is that the values of successive stationary policies generated by `PI` are non-decreasing. In other words, for any n, $V_{n+1} \geq V_n$. The following theorem states that `PI` converges within a finite number of iterations and returns an optimal policy.

Theorem 6 ([1, Theorem 6.4.2]) *Suppose M has a finite state-action space. Then,* `PI` *terminates after finitely many iterations and outputs an optimal policy of M.*

Proof. First we show that for any iteration n, $V_{n+1} \geq V_n$. To show this, note that the policy improvement step can be rewritten as

$$\pi_{n+1} \in \operatorname*{argmax}_{\pi \in \Pi} \left(r^\pi + \gamma P^\pi V_n \right).$$

Hence,

$$r^{\pi_{n+1}} + \gamma P^{\pi_{n+1}} V_n \geq r^{\pi_n} + \gamma P^{\pi_n} V_n = V_n \,,$$

where the last step follows from the policy evaluation step. Hence, $r^{\pi_{n+1}} \geq (I - \gamma P^{\pi_{n+1}})V_n$, so that

$$V_{n+1} = (I - \gamma P^{\pi_{n+1}})^{-1} r^{\pi_{n+1}} \geq (I - \gamma P^{\pi_{n+1}})^{-1}(I - \gamma P^{\pi_{n+1}})V_n = V_n \,.$$

[5]It suffices that ε be smaller than $\min_{s \in \mathcal{S}}(V^\star(s) - \max_{\pi \neq \pi^\star} V^\pi(s))$.

Now, in view of $V_{n+1} \geq V_n$ for all n, and since there are only a finite number of deterministic stationary policies (since the state-action space is assumed finite), under the stopping criterion of `PI`, it must terminate after a finite number of iterations. It then find a policy $\pi_{n+1} = \pi_n$ such that

$$V_n = r^{\pi_{n+1}} + \gamma P^{\pi_{n+1}} V_n = \max_{\pi \in \Pi} \Big(r^{\pi} + \gamma P^{\pi} V_n \Big).$$

Thus, V_n solves the optimality equation (hence, $V_n = V^\star$), and since $V_n = V^{\pi_n}$, $\pi_n = \pi^\star$. □

Iteration Complexity. A trivial upper bound on the iteration complexity is A^S, which is the total number of stationary deterministic policies. This upper bound is improved to $O\big(\frac{A^S}{S}\big)$ by Mansour and Singh [6], but still exponential in the size S of state-space (and not far from enumerating all possible stationary policies). In practice, `PI` converges within, at most, a few tens of iterations. We also have the following result on the iteration complexity of `PI`:

Theorem 7 ([7]) *`PI` converges in at most $O\big(\frac{SA}{1-\gamma} \log \frac{1}{1-\gamma}\big)$ iterations.*

Computational Complexity. Each iteration in `PI` involves solving a linear system with S equations and S unknowns. Hence, per iteration complexity of `PI` is $O(S^2A + S^3)$.

References

[1] Martin L Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 2005.

[2] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

[3] Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

[4] Ronald A Howard. Dynamic programming and Markov processes. 1960.

[5] Martin L Puterman and Moon Chirl Shin. Modified policy iteration algorithms for discounted Markov decision problems. *Management Science*, 24(11):1127–1137, 1978.

[6] Yishay Mansour and Satinder Singh. On the complexity of Policy Iteration. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 401–408, 1999.

[7] Bruno Scherrer. Improved and generalized upper bounds on the complexity of Policy Iteration. *Advances in Neural Information Processing Systems*, 26, 2013.