Reinforcement Learning: Foundations

draft - do not distribute

Shie Mannor and Yishay Mansour

September 2019

Contents

Chapter 1

Introduction and Overview

Chapter 2

Deterministic Decision Processes

In this chapter we introduce the dynamic system viewpoint of the optimal planning problem. We restrict the discussion here to deterministic (rather than stochastic) systems. We consider two basic settings. The finite-horizon decision problem and its recursive solution via finite-horizon Dynamic Programming. The average cost and it related minimum average weight cycle.

2.1 Discrete Dynamic Systems

We consider a discrete-time dynamic system, of the form:

$$\mathtt{s}_{\mathtt{t}+1} = f_{\mathtt{t}}(\mathtt{s}_{\mathtt{t}}, \mathtt{a}_{\mathtt{t}}), \quad \mathtt{t} = 0, 1, 2, \ldots, \mathtt{T}-1$$

where

- $\mathtt{t}$ is the time index.
- $\mathtt{s}_{\mathtt{t}} \in \mathcal{S}_{\mathtt{t}}$ is the state variable at time $\mathtt{t}$, and $\mathcal{S}_{\mathtt{t}}$ is the set of possible states at time $\mathtt{t}$.
- $\mathtt{a}_{\mathtt{t}} \in \mathcal{A}_{\mathtt{t}}$ is the control variable at time $\mathtt{t}$, and $\mathcal{A}_{\mathtt{t}}$ is the set of possible control actions at time $\mathtt{t}$.
- $f_{\mathtt{t}} : \mathcal{S}_{\mathtt{t}} \times \mathcal{A}_{\mathtt{t}} \to \mathcal{S}_{\mathtt{t}+1}$ is the state transition function, which defines the *state dynamics* at time $\mathtt{t}$.
- $\mathtt{T} > 0$ is the *time horizon* of the system. It can be finite or infinite.

Remark 2.1. *More generally, the set* $\mathcal{A}_t$ *of available actions may depend on the state at time* t*, namely:* $a_t \in \mathcal{A}_t(s_t) \subset \mathcal{A}_t$.

Remark 2.2. *The system is, in general, time-varying. It is called* time invariant *if* $f_t, \mathcal{S}_t, \mathcal{A}_t$ *do not depend on the time* t*. In that case we write*

$$s_{t+1} = f(s_t, a_t), \quad t = 0, 1, 2, \ldots, T-1; \quad s_t \in \mathcal{S},\ a_t \in \mathcal{A}(s_t).$$

Remark 2.3. *The state dynamics may be augmented by an output equation:*

$$o_t = \mathcal{O}_t(s_t, a_t),$$

where o_t *is the system observation, or the output. In most of this book we implicitly assume that* $o_t = s_t$*, namely, the current state* s_t *is fully observed.*

Example 2.1. ***Linear Dynamic Systems***

A well known example of a dynamic system is that of a linear time-invariant system, where:

$$s_{t+1} = As_t + Ba_t$$

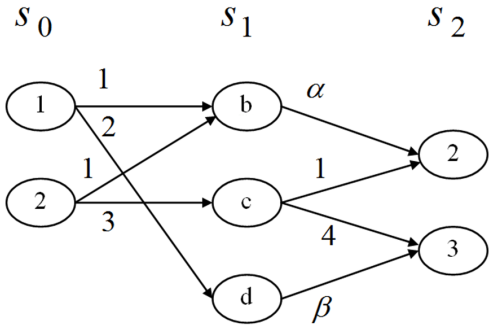
with $s_t \in \mathbb{R}^n$, $a_t \in \mathbb{R}^m$, $A \in \mathbb{R}^{n\times n}$ *and* $B \in \mathbb{R}^{n\times m}$*. Here the state and action spaces are evidently continuous (and not discrete).*

Example 2.2. ***Finite models***

Our emphasis here will be on finite state and action *models. A finite state space contains a finite number of points:* $\mathcal{S}_t = \{1, 2, \ldots, n_t\}$*. Similarly, a finite action space implies a finite number of control actions at each stage:*

$$\mathcal{A}_t(s) = \{1, 2, \ldots, m_t(s)\}, \quad s \in \mathcal{S}_t$$

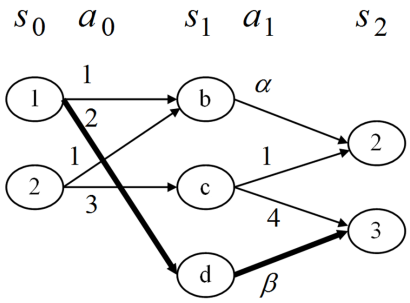
Graphical description: Finite models (over finite time horizons) can be represented by a corresponding decision graph:



Here:

- $\mathtt{T} = 2$, $\mathcal{S}_0 = \{1, 2\}$, $\mathcal{S}_1 = \{b, c.d\}$, $\mathcal{S}_2 = \{2, 3\}$,
- $\mathcal{A}_0(1) = \{1, 2\}$, $\mathcal{A}_0(2) = \{1, 3\}$, $\mathcal{A}_1(b) = \{\alpha\}$, $\mathcal{A}_1(c) = \{1, 4\}$, $\mathcal{A}_1(d) = \{\beta\}$
- $f_0(1, 1) = b$, $f_0(1, 2) = d$, $f_0(2, 1) = b$, $f_0(2, 3) = c$, $f_1(b, \alpha) = 2$, etc.

Definition 2.1. ***Feasible Path***
A feasible path for the specified system is a sequence $(\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{\mathtt{T}-1}, \mathtt{a}_{\mathtt{T}-1}, \mathtt{s}_{\mathtt{T}})$ *of states and actions, such that* $\mathtt{s}_{\mathtt{t}} \in \mathcal{A}_{\mathtt{t}}(\mathtt{s}_{\mathtt{t}})$ *and* $\mathtt{s}_{\mathtt{t}+1} = f_{\mathtt{t}}(\mathtt{s}_{\mathtt{t}}, \mathtt{a}_{\mathtt{t}})$.



A feasible path

2.2 The Finite Horizon Decision Problem

We proceed to define our first and simplest planning problem. For that we need to specify a *performance objective* for our model, and the notion of *control policies*.

2.2.1 Costs and Rewards

The cumulative cost: Let $\mathtt{h}_{\mathtt{T}} = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{a}_{\mathtt{T}-1}, \mathtt{s}_{\mathtt{T}-1}, \mathtt{s}_{\mathtt{T}})$ denote an $\mathtt{T}$-stage feasible path for the system. Each feasible path $\mathtt{h}_{\mathtt{T}}$ is assign some cost $\mathcal{C}_{\mathtt{T}} = \mathcal{C}_{\mathtt{T}}(\mathtt{h}_{\mathtt{T}})$.

The standard definition of the cost $\mathcal{C}_{\mathtt{T}}$ is through the following *cumulative cost functional*:

$$\mathcal{C}_{\mathtt{T}}(\mathtt{h}_{\mathtt{T}}) = \sum_{\mathtt{t}=0}^{\mathtt{T}-1} \mathtt{c}_{\mathtt{t}}(\mathtt{s}_{\mathtt{t}}, \mathtt{a}_{\mathtt{t}}) + c_{\mathtt{T}}(\mathtt{s}_{\mathtt{T}})$$

Here:

- $\mathtt{c}_{\mathtt{t}}(\mathtt{s}_{\mathtt{t}}, \mathtt{a}_{\mathtt{t}})$ is the *instantaneous* cost or *single-stage* cost at stage $\mathtt{t}$, and $\mathtt{c}_{\mathtt{t}}$ is the instantaneous cost function.

- $\mathtt{c}_\mathtt{T}(\mathtt{s}_\mathtt{T})$ is the *terminal* cost, and $\mathtt{c}_\mathtt{T}$ is the terminal cost function.

Note:

- The cost functional defined above is *additive* in time. Other cost functionals are possible, for example the max cost, but additive cost is by far the most common and useful.

- We shall refer to $\mathcal{C}_\mathtt{T}$ as the *cumulative* $\mathtt{T}$*-stage cost*, or just the *cumulative cost*.

Our objective is to *minimize* the cumulative cost $\mathcal{C}_\mathtt{T}$, by a proper choice of actions. We will define that goal more formally in the next section.

Cost versus reward formulation: It is often more natural to consider *maximizing* reward rather than minimizing cost. In that case, we define the cumulative $\mathtt{T}$-stage return function:

$$\mathcal{V}_\mathtt{T}(h_\mathtt{T}) = \sum_{\mathtt{t}=0}^{\mathtt{T}-1} \mathtt{r}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) + \mathtt{r}_\mathtt{T}(\mathtt{s}_\mathtt{T})$$

Here and $\mathtt{r}_\mathtt{t}$ is the instantaneous reward, and $\mathtt{r}_\mathtt{T}$ is the terminal reward. Clearly, minimizing $\mathcal{C}_\mathtt{T}$ is equivalent to maximizing $\mathcal{V}_\mathtt{T}$, if we set:

$$\mathtt{r}_\mathtt{t}(\mathtt{s}, \mathtt{a}) = -\mathtt{c}_\mathtt{t}(\mathtt{s}, \mathtt{a}) \text{ and } \mathtt{r}_\mathtt{T}(\mathtt{s}) = -\mathtt{c}_\mathtt{T}(\mathtt{s}).$$

We denote by $\mathbb{T}$ the set of time steps input setting, i.e., $\mathbb{T} = \{1, \ldots, \mathtt{T}\}$

2.2.2 Optimal Paths

Our first planning problem is the following $\mathtt{T}$*-stage Finite Horizon Problem*:

- For a given initial state $\mathtt{s}_0$, find a feasible path $\mathtt{h}_\mathtt{T} = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{\mathtt{T}-1}, \mathtt{a}_{\mathtt{T}-1}, \mathtt{s}_\mathtt{T})$ that minimizes the cost functional $\mathcal{C}_\mathtt{T}(\mathtt{h}_\mathtt{T})$, over all feasible paths $\mathtt{h}_\mathtt{T}$.

Such a feasible path $\mathtt{h}_\mathtt{T}$ is called an *optimal path* from $\mathtt{s}_0$.

A more general notion than a path is that of a *control policy*, that specifies the action to be taken at each state. Control policies will play an important role in our Dynamic Programming algorithms, and are defined next.

2.2.3 Control Policies

In general we will consider a few classes of control policies. The two basic dimensions in which we will characterize the control policies is their dependence on the history, and their use of randomization.

- A general or **history-dependent** control policy $\pi = (\pi_{\mathtt{t}})_{\mathtt{t}\in\mathbb{T}}$ is a mapping from each possible history $\mathtt{h}_{\mathtt{t}} = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{\mathtt{t}-1}, \mathtt{a}_{\mathtt{t}-1}, \mathtt{s}_{\mathtt{t}})$, $\mathtt{t} \in \mathbb{T}$, to an action $\mathtt{a}_{\mathtt{t}} = \pi_{\mathtt{t}}(\mathtt{h}_{\mathtt{t}}) \in \mathcal{A}_{\mathtt{t}}$. We denote the set of general policies by Π_H.

- A **Markov** control policy π is allowed to depend on the current state and time only: $\mathtt{a}_{\mathtt{t}} = \pi_{\mathtt{t}}(\mathtt{s}_{\mathtt{t}})$. We denote the set of Markov policies by Π_M.

- For stationary models, we may define **stationary** control policies that depend on the current state alone. A stationary policy is defined by a single mapping $\pi : \mathcal{S} \to \mathcal{A}$, so that $\mathtt{a}_{\mathtt{t}} = \pi(\mathtt{s}_{\mathtt{t}})$ for all $\mathtt{t} \in \mathbb{T}$. We denote the set of stationary policies by Π_S.

- Evidently, $\Pi_H \supset \Pi_M \supset \Pi_S$.

Randomized (Stochastic) Control policies

- The control policies defined above specify deterministically the action to be taken at each stage. In some cases we want to allow for a random choice of action.

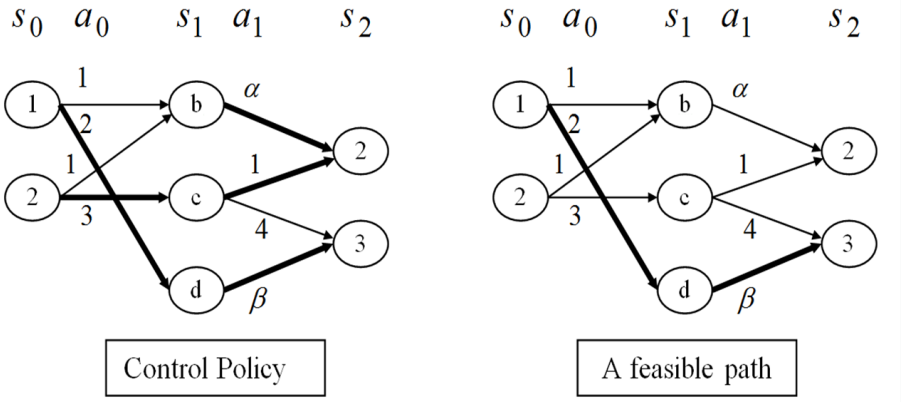
- A general randomized (stochastic) control policy assigns to each possible history $\mathtt{h}_{\mathtt{t}}$ a probability distribution $\pi_{\mathtt{t}}(\cdot|\mathtt{h}_{\mathtt{t}})$ over the action set $\mathcal{A}_{\mathtt{t}}$. That is, $\Pr\{\mathtt{a}_{\mathtt{t}} = \mathtt{a}|\mathtt{h}_{\mathtt{t}}\} = \pi_{\mathtt{t}}(\mathtt{a}|\mathtt{h}_{\mathtt{t}})$. We denote the set of general randomized policies by Π_{HS}.

- Similarly, we can define the set Π_{MS} of Markov randomized (stochastic) control policies, where $\pi_{\mathtt{t}}(\cdot|\mathtt{h}_{\mathtt{t}})$ is replaced by $\pi_{\mathtt{t}}(\cdot|\mathtt{s}_{\mathtt{t}})$, and the set Π_{SS} of stationary randomized (stochastic) control policies, where $\pi_{\mathtt{t}}(\cdot|\mathtt{s}_{\mathtt{t}})$ is replaced by $\pi(\cdot|\mathtt{s}_{\mathtt{t}})$.

- Note that the set Π_{HS} includes all other policy sets as special cases.

- For deterministic control policies, we similarly define $\Pi_{HD} \supset \Pi_{MD} \supset \Pi_{SD}$

Control policies and paths: As mentioned, a control policy specifies an action for each state, whereas a path specifies an action only for states along the path. The definition of a policy, allows us to consider counter-factual events, namely, what would have been the path if we considered a different action. This distinction is illustrated in the following figure.



Induced Path: A control policy π, together with an initial state $\mathtt{s}_0$, specify a feasible path $\mathtt{h}_\mathtt{T} = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{N-1}, \mathtt{a}_{\mathtt{T}-1}, \mathtt{s}_\mathtt{T})$. This path may be computed recursively using $\mathtt{a}_\mathtt{t} = \pi_\mathtt{t}(\mathtt{s}_\mathtt{t})$ and $\mathtt{s}_{\mathtt{t}+1} = f_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$, for $\mathtt{t} = 0, 1, \ldots, \mathtt{T}-1$.

Remark 2.4. *Suppose that for each state $\mathtt{s}_\mathtt{t}$, each action $\mathtt{a}_\mathtt{t} \in \mathcal{A}_\mathtt{t}(\mathtt{s}_\mathtt{t})$ leads to a different state $\mathtt{s}_{\mathtt{t}+1}$ (i.e., at most one edge connects any two states). We can then identify each action $\mathtt{a}_\mathtt{t} \in \mathcal{A}_\mathtt{t}(\mathtt{s}_\mathtt{t})$ with the next state $\mathtt{s}_{\mathtt{t}+1} = f_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$ it induces. In that case a path may be uniquely specified by the state sequence $(\mathtt{s}_0, \mathtt{s}_1, \ldots, \mathtt{s}_\mathtt{T})$.*

2.2.4 Reduction between control policies classes

We first show a reduction from a general history dependent policies to Randomized Markovian policies. The main observation is that the only influence on the cumulative cost is the expected instantaneous cost $\mathbb{E}[\mathtt{c}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})]$. Namely, let

$$\rho^\pi_\mathtt{t}(\mathtt{s}, \mathtt{a}) = \Pr_{\mathtt{h}'_{\mathtt{t}-1}}[\mathtt{a}_\mathtt{t} = \mathtt{a}, \mathtt{s}_\mathtt{t} = s] = \mathbb{E}_{\mathtt{h}'_{\mathtt{t}-1}}[\mathbb{I}[\mathtt{s}_\mathtt{t} = s, \mathtt{a}_\mathtt{t} = a]|\mathtt{h}'_{\mathtt{t}-1}],$$

where $\mathtt{h}'_{\mathtt{t}-1} = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{t-1}, \mathtt{a}_{t-1})$ is the history of the first $\mathtt{t}-1$ time steps generated using π, and the probability and expectation are taken with respect to the

randomness of the policy π. Now we can rewrite the expected cost to go as,

$$\mathbb{E}[\mathcal{C}^{\pi}(\mathtt{s}_0)] = \mathbb{E}[\sum_{\mathtt{t}=1}^{\mathtt{T}-1} \sum_{\mathtt{a}\in\mathcal{A}_\mathtt{t},\mathtt{s}\in\mathcal{S}_\mathtt{t}} \mathtt{c}_\mathtt{t}(\mathtt{s},\mathtt{a})\rho^{\pi}_{\mathtt{t}}(\mathtt{s},\mathtt{a})],$$

where $\mathcal{C}^{\pi}(\mathtt{s}_0)$ is the random variable of the cost when starting at state $\mathtt{s}_0$ and following policy π.

This implies that any two policies π and π' for which $\rho^{\pi}_{\mathtt{t}}(\mathtt{s},\mathtt{a}) = \rho^{\pi'}_{\mathtt{t}}(\mathtt{s},\mathtt{a})$, for any time $\mathtt{t}$, state $\mathtt{s}$ and action $\mathtt{a}$, would have the same expected cumulative cost for any cost function, i.e., $\mathbb{E}[\mathcal{C}^{\pi}(\mathtt{s}_0)] = \mathbb{E}[\mathcal{C}^{\pi'}(\mathtt{s}_0)]$

Theorem 2.1. *For any policy $\pi \in \Pi_{HS}$, there is a policy $\pi' \in \Pi_{MS}$, such that for every state $\mathtt{s}$ and action $\mathtt{a}$ we have, $\rho^{\pi}(\mathtt{s},\mathtt{a}) = \rho^{\pi'}(\mathtt{s},\mathtt{a})$. This will imply that,*

$$\mathbb{E}[\mathcal{C}^{\pi}(\mathtt{s}_0)] = \mathbb{E}[\mathcal{C}^{\pi'}(\mathtt{s}_0)]$$

Proof. Given the policy $\pi \in \Pi_{HS}$, we define $\pi' \in \Pi_{MS}$ as follows. For every state $\mathtt{s} \in \mathcal{S}_\mathtt{t}$ we define

$$\pi'_{\mathtt{t}}(\mathtt{a}|\mathtt{s}) = \Pr_{\mathtt{h}_{\mathtt{t}-1}}[\mathtt{a}_\mathtt{t} = \mathtt{a}|\mathtt{s}_\mathtt{t} = \mathtt{s}] = \frac{\rho^{\pi}_{\mathtt{t}}(\mathtt{s},\mathtt{a})}{\sum_{\mathtt{a}'\in\mathcal{A}_\mathtt{t}} \rho^{\pi}_{\mathtt{t}}(\mathtt{s},\mathtt{a}')}$$

By definition π' is Markovian (depends only on the time $\mathtt{t}$ and the realized state $\mathtt{s}$). By construction $\rho^{\pi}_{\mathtt{t}}(\mathtt{s},\mathtt{a}) = \rho^{\pi'}_{\mathtt{t}}(\mathtt{s},\mathtt{a})$ is identical, which implies that $\mathbb{E}[\mathcal{C}^{\pi}(\mathtt{s}_0)] = \mathbb{E}[\mathcal{C}^{\pi'}(\mathtt{s}_0)]$. □

Next we show that for any stochastic Markovian policy there is a deterministic Markovian policy with at most the same cumulative cost.

Theorem 2.2. *For any policy $\pi \in \Pi_{MS}$, there is a policy $\pi' \in \Pi_{MD}$, such that*

$$\mathbb{E}[\mathcal{C}^{\pi}(\mathtt{s}_0)] \geq \mathbb{E}[\mathcal{C}^{\pi'}(\mathtt{s}_0)]$$

Proof. The proof is by backward induction on the steps. The inductive claim is:
For any policy $\pi \in \Pi_{MS}$ which is deterministic in $[\mathtt{t}+1,\mathtt{T}]$, there is a policy $\pi' \in \Pi_{MS}$ which is deterministic in $[\mathtt{t},\mathtt{T}]$ and $\mathbb{E}[\mathcal{C}^{\pi}(\mathtt{s}_0)] \geq \mathbb{E}[\mathcal{C}^{\pi'}(\mathtt{s}_0)]$.
Clearly, the theorem follows from the case of $\mathtt{t} = 0$.

For the base of the induction we can take $\mathtt{t} = \mathtt{T}$, which holds trivially.

For the inductive step, assume that $\pi \in \Pi_{MS}$ is deterministic in $[\mathtt{t}+1,\mathtt{T}]$.

For every $\mathtt{s}_{\mathtt{t}+1} \in \mathcal{S}_{\mathtt{t}+1}$ define

$$\mathcal{C}_{\mathtt{t}+1}(\mathtt{s}_{\mathtt{t}+1}) = \mathcal{C}(path(\mathtt{s}_{\mathtt{t}+1},\ldots,\mathtt{s}_\mathtt{T})),$$

where $path(\mathtt{s}_{t+1}, \ldots, \mathtt{s}_T)$ is the deterministic path from $\mathtt{s}_{t+1}$ induced by π.

We define π' to be identical to π for all time steps $\mathtt{t}' \neq \mathtt{t}$. We define π'_t for each $\mathtt{s}_t \in \mathcal{S}_t$ as follows:

$$\pi'_t(\mathtt{a}_t, \mathtt{s}_t) = \arg\min_{\mathtt{a} \in \mathcal{A}_t} \mathtt{c}(\mathtt{s}_t, \mathtt{a}) + \mathcal{C}_t(f_t(\mathtt{s}_t, \mathtt{a})). \tag{2.1}$$

Recall that since we have a Deterministic Decision Process $f_t(\mathtt{s}_t, \mathtt{a}_t) \in \mathcal{S}_{t+1}$ is the next state if we take action $\mathtt{a}$ in $\mathtt{s}_t$.

For the analysis, note that π and π' are identical until time $\mathtt{t}$, so they generate exactly the same distribution over paths. At time $\mathtt{t}$, π' is define to minimize the cost to go from $\mathtt{s}_t$, given that we follow π from $\mathtt{t}+1$ to $\mathtt{T}$. Therefore the cost can only decrease. Formally, let $\mathbb{E}^\pi[\cdot]$ be the expectation with respect to policy π.

$$\begin{aligned}
\mathbb{E}^\pi_{\mathtt{s}_t}[\mathcal{C}_t(\mathtt{s}_t)] =& \mathbb{E}^\pi_{\mathtt{s}_t}\mathbb{E}^\pi_{\mathtt{a}_t}[\mathtt{c}(\mathtt{s}_t, \mathtt{a}_t) + \mathcal{C}_{t+1}(f_t(\mathtt{s}_t, \mathtt{a}_t))] \\
\geq& \mathbb{E}^\pi_{\mathtt{s}_t} \min_{\mathtt{a}_t \in \mathcal{A}_t} [\mathtt{c}(\mathtt{s}_t, \mathtt{a}_t) + \mathcal{C}_{t+1}(f_t(\mathtt{s}_t, \mathtt{a}_t))] \\
=& \mathbb{E}^{\pi'}_{\mathtt{s}_t}[\mathcal{C}_t(\mathtt{s}_t)]
\end{aligned}$$

which completes the inductive proof. □

Remark 2.5. *The above proof extends very naturally for the case of a stochastic MDP, which implies that f_t is stochastic. The modification of the proof would simply take the expectation over f_t in Eq. 2.1.*

2.2.5 Optimal Control Policies

Definition 2.2. *A control policy $\pi \in \Pi_{MD}$ is called **optimal** if, for each initial state $\mathtt{s}_0$, it induces an optimal path $\mathtt{h}_T$ from $\mathtt{s}_0$.*

An alternative definition can be given in terms of policies only. For that purpose, let $\mathtt{h}_T(\pi; \mathtt{s}_0)$ denote the path induced by the policy π from $\mathtt{s}_0$. For a given return functional $\mathcal{V}_T(\mathtt{h}_T)$, denote $\mathcal{V}_T(\pi; \mathtt{s}_0) = \mathcal{V}_T(\mathtt{h}_T(\pi; \mathtt{s}_0))$ That is, $\mathcal{V}_T(\pi; \mathtt{s}_0)$ is the cumulative return for the path induced by π from $\mathtt{s}_0$.

Definition 2.3. *A control policy $\pi \in \Pi_{MD}$ is called optimal if, for each initial state $\mathtt{s}_0$, it holds that $\mathcal{V}_T(\pi; \mathtt{s}_0) \geq \mathcal{V}_T(\tilde{\pi}; \mathtt{s}_0)$ for any other policy $\tilde{\pi} \in \Pi_{MD}$.*

Equivalence of the two definitions can be easily established (exercise). An optimal policy is often denoted by π^*.

The standard T-stage finite-horizon planning problem: Find a control policy π for the T-stage Finite Horizon problem that minimizes the cumulative cost (or maximizes the cumulative return) function.

The naive approach to finding an optimal policy: For finite models (i.e., finite state and action spaces), the number of feasible paths (or control policies) is finite. It is therefore possible, in principle, to enumerate all $\mathtt{T}$-stage paths, compute the cumulative return for each one, and choose the one which gives the largest return. Let us evaluate the number of different paths and control policies. Suppose for simplicity that number of states at each stage is the same: $|\mathcal{S}_\mathtt{t}| = n$, and similarly the number of actions at each state is the same: $|\mathcal{A}_\mathtt{t}(x)| = m$ (with $m \leq n$) . The number of feasible $\mathtt{T}$-stage paths for each initial state is seen to be $m^\mathtt{T}$. The number of different policies is $m^{n^\mathtt{T}}$. For example, for a fairly small problem with $\mathtt{T} = n = m = 10$, we obtain 10^{10} paths for each initial state (and 10^{11} overall), and 10^{100} control policies. Clearly it is not computationally feasible to enumerate them all.

Fortunately, Dynamic Programming offers a drastic reduction of the computational complexity for this problem.

2.3 Finite Horizon Dynamic Programming

The Dynamic Programming (DP) algorithm breaks down the $\mathtt{T}$-stage finite-horizon problem into $\mathtt{T}$ sequential single-stage optimization problems. This results in dramatic improvement in computation efficiency.

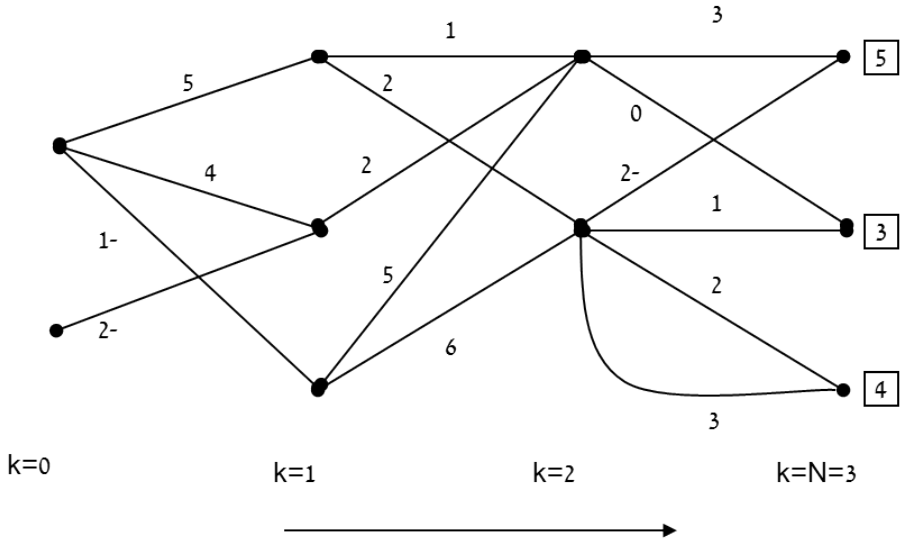
The DP technique for dynamic systems is based on a general observation called Bellman's Principle of Optimality. Essentially it states the following (for deterministic problems):

- **Any sub-path of an optimal path is itself an optimal path between its end point.**

To see why this should hold, consider a sub-path which is not optimal. We can replace it be an optimal sub-path, and improve the return.

Applying this principle recursively from the last stage backward, obtains the (backward) Dynamic Programming algorithm. Let us first illustrate the idea with following example.

Example 2.3. *Shortest path on a decision graph: Suppose we wish to find the shortest path (minimum cost path) from the initial node in* $\mathtt{T}$ *steps.*



The boxed values are the terminal costs at stage $\mathtt{T}$*, the other number are the link costs. Using backward recursion, we may obtain that the minimal path costs from the two initial states are* 7 *and* 3*, as well as the optimal paths and an optimal policy.*

We can now describe the DP algorithm. Recall that we consider the dynamic system

$$\mathtt{s}_{t+1} = f_t(\mathtt{s}_t, \mathtt{a}_t), \quad \mathtt{t} = 0, 1, 2, \ldots, \mathtt{T}-1$$
$$\mathtt{s}_t \in \mathcal{S}_t, \quad \mathtt{a}_t \in \mathcal{A}_t(\mathtt{s}_t)$$

and we wish to maximize the cumulative return:

$$\mathcal{V}_T = \sum_{t=0}^{T-1} \mathtt{r}_t(\mathtt{s}_t, \mathtt{a}_t) + \mathtt{r}_T(\mathtt{s}_T)$$

The DP algorithm computes recursively a set of **value functions** $\mathcal{V}_t : \mathcal{S}_t \to \mathbb{R}$, where $\mathcal{V}_t(\mathtt{s}_t)$ is the value of an optimal sub-path $\mathtt{h}_{t:T} = (\mathtt{s}_t, \mathtt{a}_t, \ldots, \mathtt{s}_T)$ that starts at $\mathtt{s}_t$.

Algorithm 2.1. ***Finite-horizon Dynamic Programming***

1. *Initialize the value function:* $\mathcal{V}_T(\mathtt{s}) = \mathtt{r}_T(\mathtt{s})$*,* $\mathtt{s} \in \mathcal{S}_T$*.*
2. *Backward recursion: For* $\mathtt{t} = \mathtt{T}-1, \ldots, 0$*, compute*

$$\mathcal{V}_t(\mathtt{s}) = \max_{\mathtt{a} \in \mathcal{A}_t} \left\{ \mathtt{r}_t(\mathtt{s}, \mathtt{a}) + \mathcal{V}_{t+1}(f_t(\mathtt{s}, \mathtt{a})) \right\}, \quad \mathtt{s} \in \mathcal{S}_t.$$

3. *Optimal policy: Choose any control policy* $\pi^* = (\pi_{\mathtt{t}}^*)$ *that satisfies:*

$$\pi_{\mathtt{t}}^*(\mathtt{s}) \in \arg\max_{\mathtt{a}\in\mathcal{A}_{\mathtt{t}}} \left\{ \mathtt{r}_{\mathtt{t}}(\mathtt{s},\mathtt{a}) + \mathcal{V}_{\mathtt{t}+1}(f_{\mathtt{t}}(\mathtt{s},\mathtt{a})) \right\}, \quad \mathtt{t} = 0,\ldots,\mathtt{T}-1.$$

Proposition 2.1. *The following holds for finite-horizon dynamic programming:*

1. *The control policy* π^* *computed in Algorithm 2.1 is an optimal control policy for the* T*-stage Finite Horizon problem.*

2. $\mathcal{V}_0(\mathtt{s})$ *is the optimal* T*-stage return from initial state* $\mathtt{s}_0 = \mathtt{s}$*:*

$$\mathcal{V}_0(s) = \max_{\pi} V_0^{\pi}(\mathtt{s}), \qquad \forall \mathtt{s} \in \mathcal{S}_0,$$

where $V_0^{\pi}(\mathtt{s})$ *is the expected return of policy* π *when started at state* s*.*

Proof. We show that the computed policy π^* is optimal and its return from time t is $\mathcal{V}_{\mathtt{t}}$. We will establish the following inductive claim:

For any time t *and any state* s*, the path from* s *define by* π^* *is the maximum return path of length* $\mathtt{T}-\mathtt{t}$*. The value of* $\mathcal{V}_{\mathtt{t}}(\mathtt{s})$ *is the maximum return from* s*.*

The proof is by a backward induction. For the basis of the induction we have: $\mathtt{t} = \mathtt{T}$, and the inductive claim follows from the initialization.

Assume the inductive claim holds for t prove for $\mathtt{t}+1$. For contradiction assume there is a higher return path from s. Let the path generated π^* be $P = (\mathtt{s}, \mathtt{s}_{\mathtt{T}-\mathtt{t}}^*, \ldots, \mathtt{s}_{\mathtt{T}}^*)$. Let $P_1 = (\mathtt{s}, \mathtt{s}_{\mathtt{T}-\mathtt{t}}, \ldots, \mathtt{s}_{\mathtt{T}})$ be an alternative path. Let $P_2 = (\mathtt{s}, \mathtt{s}_{\mathtt{T}-\mathtt{t}}, \mathtt{s}'_{\mathtt{T}-\mathtt{t}+1}, \ldots, \mathtt{s}'_{\mathtt{T}})$ be the path generated by following π^* from $\mathtt{s}_{\mathtt{T}-\mathtt{t}}$. Since P_1 and P_2 are identical except for the last t stages, we can use the inductive hypothesis, which implies that $\mathcal{V}(P_1) \leq \mathcal{V}(P_2)$. From the definition of π^* we have that $\mathcal{V}(P_2) \leq \mathcal{V}(P)$. Hence, $\mathcal{V}(P_1) \leq \mathcal{V}(P_2) \leq \mathcal{V}(P)$, which completes the proof of the inductive hypothesis. □

Let us make the following observations:

1. The algorithm involves visiting each state exactly once, proceeding backward in time. For each time instant (or stage) t, the value function $\mathcal{V}_{\mathtt{t}}(\mathtt{s})$ is computed for all states $\mathtt{s} \in \mathcal{S}_{\mathtt{t}}$ before proceeding to stage $\mathtt{t}-1$.

2. The *backward induction* step of Algorithm 2.1, along with similar equations in the theory of DP, is called **Bellman's equation**.

3. Computational complexity: There is a total of $n\mathtt{T}$ states (excluding the final one), and in each we need m computations. Hence, the number of required calculations is $mn\mathtt{T}$. For the example above with $m = n = \mathtt{T} = 10$, we need $O(10^3)$ calculations.

4. A similar algorithm that proceeds forward in time (from $\mathtt{t} = 0$ to $\mathtt{t} = \mathtt{T}$) can be devised. We note that this will not be possible for stochastic systems (i.e., the stochastic MDP model).

5. The celebrated **Viterbi algorithm** is an important instance of finite-horizon DP. The algorithm essentially finds the most likely sequence of states in a Markov chain ($\mathtt{s}_\mathtt{t}$) that is partially (or noisily) observed. The algorithm was introduced in 1967 for decoding convolution codes over noisy digital communication links. It has found extensive applications in communications, and is a basic computational tool in Hidden Markov Models (HMMs), a popular statistical model that is used extensively in speech recognition and bioinformatics, among other areas.

2.4 Shortest Paths

We can formulate a DDP problems similar to shortest path problems. Given a directed graph $G(V, E)$, there is a set of goal states $\mathcal{S}_G$, and the goal is to reach one of the goal states. Formally, when we reach a goal state we stay there and have a zero cost. For such a DDP the optimal policy would be to compute a shortest path to one of the goal states.

There is a variety of algorithms for computing shortest paths.

1. Bellman-Ford: handles also negative weights, but assumes no negative cycle. Runs in time $O(|V| \cdot |E|)$.

2. Dijkstra: Assumes non-negative weights. Runs in time $O(|V| \log |E| + |E|)$

2.5 Linear Programming for Finite Horizon

In this section we will use linear programming to derive the optimal policy. We will see that both the primal and dual program will play an important part in defining the optimal policy. We will fix an initial state $\mathtt{s}_0$ and compute the optimal policy for it.

We will start with the primal linear program, which will compute the optimal policy. For each time $\mathtt{t}$, state $\mathtt{s}$ and action $\mathtt{a}$ we will have a variable $x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) \in \{0,1\}$ that will indicate by 1 that at time $\mathtt{t}$ we are at state $\mathtt{s}$ and perform action $\mathtt{a}$. For the terminal states $\mathtt{s}$ we will have a variable $x_{\mathtt{T}}(\mathtt{s}) \in \{0,1\}$ that will indicate whether we terminate at state $\mathtt{s}$.

Our main constraint will be a flow constraint, stating that if we reach state $\mathtt{s}$ at time $\mathtt{t}-1$ then we exit it at time $\mathtt{t}$. Formally,

$$\sum_{\mathtt{a}} x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) = \sum_{\mathtt{s}',\mathtt{a}':f_{\mathtt{t}-1}(\mathtt{s}',\mathtt{a}')=\mathtt{s}} x_{\mathtt{t}-1}(\mathtt{s}',\mathtt{a}').$$

and for terminal states simply

$$x_{\mathtt{T}}(\mathtt{s}) = \sum_{\mathtt{s}',\mathtt{a}':f_{\mathtt{T}-1}(\mathtt{s}',\mathtt{a}')=\mathtt{s}} x_{\mathtt{T}-1}(\mathtt{s}',\mathtt{a}')$$

The return, which we would like to maximize, would be

$$\sum_{\mathtt{t},\mathtt{s},\mathtt{a}} \mathtt{r}_{\mathtt{t}}(\mathtt{s},\mathtt{a}) x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) + \sum_{\mathtt{s}} \mathtt{r}_{\mathtt{T}}(\mathtt{s}) x_{\mathtt{T}}(\mathtt{s})$$

To have a linear program we will also need to relax the constraints of $\{0,1\}$ to $[0,1]$

The resulting linear program is the following.

$$\max_{x_{\mathtt{t}}(\mathtt{s},\mathtt{a}),x_{\mathtt{T}}(\mathtt{s})} \sum_{\mathtt{t},\mathtt{s},\mathtt{a}} \mathtt{r}_{\mathtt{t}}(\mathtt{s},\mathtt{a}) x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) + \sum_{\mathtt{s}} \mathtt{r}_{\mathtt{T}}(\mathtt{s}) x_{\mathtt{T}}(\mathtt{s})$$

such that

$$\sum_{\mathtt{a}} x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) = \sum_{\mathtt{s}',\mathtt{a}':f_{\mathtt{t}-1}(\mathtt{s}',\mathtt{a}')=\mathtt{s}} x_{\mathtt{t}-1}(\mathtt{s}',\mathtt{a}'). \qquad \forall \mathtt{s} \in \mathcal{S}_{\mathtt{t}}, \mathtt{t} \in \mathbb{T}$$

$$x_{\mathtt{T}}(\mathtt{s}) = \sum_{\mathtt{s}',\mathtt{a}':f_{\mathtt{T}-1}(\mathtt{s}',\mathtt{a}')=\mathtt{s}} x_{\mathtt{T}-1}(\mathtt{s}',\mathtt{a}') \qquad \forall \mathtt{s} \in \mathcal{S}_{\mathtt{T}}$$

$$x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) \geq 0 \qquad \forall \mathtt{s} \in \mathcal{S}_{\mathtt{t}}, \mathtt{a} \in \mathcal{A}, \mathtt{t} \in \{0,\ldots,\mathtt{T}-1\}$$

$$\sum_{\mathtt{a}} x_0(\mathtt{s}_0,\mathtt{a}) = 1$$

$$x_0(\mathtt{s},\mathtt{a}) = 0, \qquad \forall \mathtt{s} \in \mathcal{S}_0, \mathtt{s} \neq \mathtt{s}_0$$

Note that we (implicitly) have $\sum_{\mathtt{a}\in\mathcal{A},\mathtt{s}\in\mathcal{S}_\mathtt{t}} x_\mathtt{t}(\mathtt{s},\mathtt{a}) = 1$, and we can prove it by induction on $\mathtt{t}$.

Given the primal linear program we can derive the dual linear program.

$$\begin{aligned}
&\min_{z_\mathtt{t}(\mathtt{s})} \ z_0(\mathtt{s}_0) \\
&\text{such that} \\
&\qquad z_\mathtt{T}(\mathtt{s}) = \mathtt{r}_\mathtt{T}(\mathtt{s}) && \forall \mathtt{s}\in\mathcal{S}_\mathtt{t} \\
&\qquad z_\mathtt{t}(\mathtt{s}) \geq \mathtt{r}_\mathtt{t}(\mathtt{s},\mathtt{a}) + z_{\mathtt{t}+1}(f_\mathtt{t}(\mathtt{s},\mathtt{a})), && \forall \mathtt{s}\in\mathcal{S}_\mathtt{t}, \mathtt{a}\in\mathcal{A}, \mathtt{t}\in\mathbb{T}, \\
&\qquad\quad .
\end{aligned}$$

One can identify the dual random variables $z_\mathtt{t}(\mathtt{s})$ with the optimal vale function $\mathcal{V}_\mathtt{t}(\mathtt{s})$. At the optimal solution of the dual linear program one can show that we have

$$z_\mathtt{t}(\mathtt{s}) = \max_\mathtt{a} \left\{ \mathtt{r}_\mathtt{t}(\mathtt{s},\mathtt{a}) + z_{\mathtt{t}+1}(f_\mathtt{t}(\mathtt{s},\mathtt{a})) \right\}, \qquad \forall \mathtt{s}\in\mathcal{S}_\mathtt{t}, \mathtt{t}\in\mathbb{T},$$

which is the familiar Bellman optimality equations.

2.6 Average cost criteria

The average cost criteria considers the limit of the average costs. Formally:

$$\mathcal{C}^\pi_{avg} = \lim_{\mathtt{T}\to\infty} \frac{1}{\mathtt{T}} \sum_{\mathtt{t}=0}^{\mathtt{T}-1} \mathtt{c}_\mathtt{t}(\mathtt{s}_\mathtt{t},\mathtt{a}_\mathtt{t})$$

where the trajectory is generated using π. The aim is to minimize $\mathbb{E}[\mathcal{C}^\pi_{avg}]$. This implies that any finite prefix has no influence of the final average cost, since its influence vanishes as $\mathtt{T}$ goes to infinity.

For a deterministic stationary policy, the policy converges to a simple cycle, and the average cost is the average cost of the edges on the cycle. (Recall, we are considering only DDP.)

Given a directed graph $G(V,E)$, let Ω be the collection of all cycles in $G(V,E)$. For each cycle $\omega = (v_1,\ldots,v_k)$, we define $c(\omega) = \sum_{i=1}^k c(v_i,v_{i+1})$, where (v_i,v_{i+1}) is the i-th edge in the cycle ω. Let $\mu(\omega) = \frac{c(\omega}{k}$. The *min average cost cycle* is

$$\mu^* = \min_{\omega\in\Omega} \mu(\omega)$$

We show that the min average cost cycle is the optimal policy.

Theorem 2.3. *For any Deterministic Decision Process (DPP) the optimal average costs is μ^*, and an optimal policy is π_ω that cycles around a simple cycle of average cost μ^*.*

Proof. Let ω be a cycle of average cost μ^*. Let π_ω be a deterministic stationary policy that first reaches ω and then cycles in ω. Clearly, $\mathcal{C}^{\pi_\omega}_{avg} = \mu^*$.

We show that for any policy π (possibly in Π_{HS}) we have that $\mathbb{E}[\mathcal{C}^{\pi}_{avg}] \geq \mu^*$. For contradiction assume that there is a policy π' that has average cost $\mu^* - \varepsilon$. Consider a sufficiently long run of length $\mathtt{T}$ of π', and fix any realization θ of it. We will show that the cumulative cost $\mathcal{C}(\theta) \geq (\mathtt{T} - n)\mu^*$, which implies that $\mathbb{E}[\mathcal{C}^{\pi}_{avg}] \geq \mu^* - n\mu^*/\mathtt{T}$.

Given θ, consider the first simple cycle ω in θ. The average cost of ω is $\mu(\omega) \geq \mu^*$. Delete ω from θ, reducing the number of edges by $|\omega|$ and the cumulative cost by $\mu(\omega)|\omega|$. We continue the process until there is no remaining cycles, which implies that we have at most $|V| = n$ nodes remaining. Therefore, the costs of ω was at least $(\mathtt{T} - n)\mu^*$. This implies that the average cost of θ is at least $\mathcal{C}(\theta) \geq (1 - \frac{n}{\mathtt{T}})\mu^*$. For $\epsilon > \mu^* n/\mathtt{T}$ we have a contradiction. □

Next we develop an algorithm for computing the minimum average cost cycle, which implies an optimal policy for DDP for average costs. The input is a directed graph $G(V, E)$ with cost $\mathtt{c} : E \to \mathbb{R}$.

We first give a characterization of μ^*. Set a root $r \in V$. Let $F_k(v)$ be paths of length k from r to v. Let $d_k(v) = \min_{p \in F_k(v)} \mathtt{c}(p)$, where if $F_k(v) = \emptyset$ then $d_k(v) = \infty$. The following theorem (due to [R. Karp, A characterization of the minimum cycle mean in digraph, Discrete mathematics, 1978]) gives a characterization of μ^*.

Theorem 2.4.

$$\mu^* = \min_{v \in S} \max_{0 \leq k \leq n-1} \frac{d_n(v) - d_k(v)}{n - k} ,$$

where we define $\infty - \infty$ as ∞.

Proof. We have two cases, $\mu^* = 0$ and $\mu^* > 0$. We assume that the graph has no negative cycle (we can guarantee this by adding a large number M to all the weights).

We start with $\mu^* = 0$. This implies that we have in $G(V, E)$ a cycle of weight zero, but no negative cycle. For the theorem it is sufficient to show that

$$\min_{v \in S} \max_{0 \leq k \leq n-1} \{d_n(v) - d_k(v)\} = 0.$$

For every node $v \in V$ there is a path of length $k \in [0, n-1]$ of cost $d(v)$, the cost of the shortest path from r to v. This implies that

$$\max_{0 \leq k \leq n-1} \{d_n(v) - d_k(v)\} = d_n(v) - d(v) \geq 0$$

We need to show that for some $v \in V$ we have $d_n(v) = d(v)$, which implies that $\min_{v \in S}\{d_n(v) - d(v)\} = 0$.

Consider a cycle ω of cost $\mathcal{C}(\omega) = 0$ (there is one, since $\mu^* = 0$). Let v be a node on the cycle ω. Consider a path P from r to v which then cycles around ω and has length at least n. The path P is a shortest path to v (although not necessarily simple). This implies that any sub-path of P is also a shortest path. Let P' be a sub-path of P of length n and let it end in $u \in V$. Path P' is a shortest path to u, since it is a prefix of a shortest path P. This implies that the cost of P' is $d(u)$. Since P' is of length n, by construction, we have that $d_n(u) = d(u)$. Therefore, $\min_{v \in S}\{d_n(v) - d(v)\} = 0$, which completes the case that $\mu^* = 0$.

For $\mu^* > 0$ we subtract a constant $\Delta = \mu^*$ from al the costs in the graph. This implies that for the new costs we have a zero cycle and no negative cycle. We can now apply the previous case. It only remains to show that the formula changes by exactly $\Delta = \mu^*$.

Formally, for every edge $e \in E$ let $\mathtt{c}'(e) = \mathtt{c}(e) - \Delta$. For any path p we have $\mathcal{C}'(p) = \mathcal{C}(p) - |p|\Delta$, and for any cycle ω we have $\mu'(\omega) = \mu(\omega) - \Delta$. This implies that for $\Delta = \mu^*$ we have a cycle of cost zero and no negative cycles. We now consider the formula,

$$\begin{aligned}
0 = (\mu')^* &= \min_{v \in V} \max_{0 \le k \le n-1} \{\frac{d'_n(v) - d'_k(v)}{n-k}\} \\
&= \min_{v \in V} \max_{0 \le k \le n-1} \{\frac{d_n(v) - n\Delta - d_k(v) + k\Delta}{n-k}\} \\
&= \min_{v \in V} \max_{0 \le k \le n-1} \{\frac{d_n(v) - d_k(v)}{n-k} - \Delta\} \\
&= \min_{v \in V} \max_{0 \le k \le n-1} \{\frac{d_n(v) - d_k(v)}{n-k}\} - \Delta
\end{aligned}$$

Therefore we have

$$\mu^* = \Delta = \min_{v \in V} \max_{0 \le k \le n-1} \{\frac{d_n(v) - d_{\mathtt{t}}(v)}{n-k}\}$$

□

We would like now to recover the minimum average cost cycle. The basic idea is to recover the cycle from the minimizing vertices in the formula, but some care need to be taken. It is true that some minimizing pair (v, k) the path of length n from r to v has a cycle of length $n - k$ which is the suffix of the path. The solution is that for the path p, from r to v of length n, any simple cycle is a minimum average cost

cycle. (See, ["A note of finding minimum mean cycle", Mmamu Chaturvedi and Ross M. McConnell, IPL 2017]).

The running time of computing the minimum average cost cycle is $O(|V| \cdot |E|)$.

2.7 Historical Notes:

- Dynamic Programming was popularized in the 1950's and 1960's by **Richard Bellman** (1920-1984), an American mathematician. Bellman, who coined the term Dynamic Programming, formulated the general theory and proposed numerous applications in control and operations research.

- **Andrew Viterbi** (born 1935) is an American professor of electric engineer, a pioneer in the field of digital communications, and co-founder of Qualcomm Inc. (together with Irwin Jacobs). He has had close ties with the Technion, and has made many contributions to our department.

- **Richard Karp** (born 1935) is an American professor of computer science, known for his contributions to the theory of computing.

Chapter 3

Markov Chains

We start by considering a stochastic dynamics model which does not have any actions, so that we can concentrate on the dynamics.

A Markov chain $\{X_{\mathtt{t}},\ \mathtt{t} = 0, 1, 2, \ldots\}$, with $X_{\mathtt{t}} \in X$, is a discrete-time stochastic process, over a finite or countable state-space X, that satisfies the following Markov property:

$$\mathcal{P}(X_{\mathtt{t}+1} = j | X_{\mathtt{t}} = i, X_{\mathtt{t}-1}, \ldots X_0) = \mathcal{P}(X_{\mathtt{t}+1} = j | X_{\mathtt{t}} = i).$$

We focus on time-homogeneous Markov chains, where

$$\mathcal{P}(X_{\mathtt{t}+1} = j | X_{\mathtt{t}} = i) = \mathcal{P}(X_1 = j | X_0 = i) \stackrel{\Delta}{=} p_{i,j}.$$

The $p_{i,j}$'s are the transition probabilities, which satisfy $p_{i,j} \geq 0$, and for each $i \in X$ we have $\sum_{j \in X} p_{i,j} = 1$, namely, $\{p_{i,j} : j \in X\}$ is a distribution on the next state following state i. The matrix $P = (p_{i,j})$ is the transition matrix. The matrix is row-stochastic (each row sums to 1 and all entries non-negative).

Given the initial distribution p_0 of X_0, namely $\mathcal{P}(X_0 = i) = p_0(i)$, we obtain the finite-dimensional distributions:

$$\mathcal{P}(X_0 = i_0, \ldots, X_{\mathtt{t}} = i_{\mathtt{t}}) = p_0(i_0) p_{i_0,i_1} \cdot \ldots \cdot p_{i_{\mathtt{t}-1},i_{\mathtt{t}}}.$$

Define $p_{i,j}^{(m)} = \mathcal{P}(X_m = j | X_0 = i)$, the m-step transition probabilities. It is easy to verify that $p_{i,j}^{(m)} = [P^m]_{ij}$ (here P^m is the m-th power of the matrix P).

Example 3.1. [1] *Consider the following two state Markov chain, with transition probability P and initial distribution p_0, as follows:*

$$P = \begin{pmatrix} 0.4 & 0.6 \\ 0.2 & 0.8 \end{pmatrix} \qquad p_0 = \begin{pmatrix} 0.5 & 0.5 \end{pmatrix}$$

[1] Simple MC, two states, running example.

Initially, we have both states equally likely. After one step, the distribution of states is $p_1 = p_0 P = (0.3\ ,\ 0.7)$. *After two steps we have* $p_2 = p_1 P = p_0 P^2 = (0.26\ ,\ 0.74)$. *The limit of this sequence would be* $p_\infty = (0.25\ ,\ 0.75)$, *which is called the* steady state distribution*, and would be discussed later.*

State classification:

- State j is *accessible* from i (denoted by $i \to j$) if $p_{i,j}^{(m)} > 0$ for some $m \geq 1$.
  For a finite X we can compute the accessibility property as follows. Construct a directed graph $G(X, E)$ where the vertices are the states X and there is a directed edge (i, j) if $p_{i,j} > 0$. State j is accessible from state i iff there exists a directed path in $G(X, E)$ from i to j.

- States i and j are *communicating* states (or communicate) if $i \to j$ and $j \to i$.
  For a finite X, this implies that in $G(X, E)$ there is both a directed path from i to j and from j to i.

- A *communicating class* (or just class) is a maximal collection of states that communicate.
  For a finite X, this implies that in $G(X, E)$ we have i and j in the same strongly connected component of the graph. (A strongly connected component has a directed path between any pair of vertices.)

- The Markov chain is *irreducible* if all states belong to a single class (i.e., all states communicate with each other).
  For a finite X, this implies that in $G(X, E)$ is strongly connected.

- State i has a *period* $d_i = GCD\{m \geq 1 : p_{i,i}^{(m)} > 0\}$, where GCD is the greatest common divisor. A state is aperiodic if $d_i = 1$.
  State i is periodic with period $d_i \geq 2$ if $p_{i,i}^{(m)} = 0$ for $m \pmod{d_i} \neq 0$ and for any m such that $p_{i,i}^{(m)} > 0$ we have $m \pmod{d_i} = 0$.
  If a state i is aperiodic, then there exists an integer m_0 such that for any $m \geq m_0$ we have $p_{i,i}^{(m)} > 0$.

- Periodicity is a class property: all states in the same class have the same period. Specifically, if some state is *a-periodic*, then all states in the class are a-periodic.

Claim 3.1. *For any two states i and j with periods d_i and d_j, in the same communicating class, we have $d_i = d_j$.*

Proof. For contradiction, assume that $d_j \pmod{d_i} \neq 0$. Since they are in the same communicating class, we have a trajectory from i to j of length $m_{i,j}$ and from j to i of length $m_{j,i}$. This implies that $(m_{i,j} + m_{j,i}) \pmod{d_i} = 0$. Now, there is a trajectory (which is a cycle) of length $m_{j,j}$ from j back to j such that $m_{j,j} \pmod{d_i} \neq 0$ (otherwise d_i divides the period of j). Consider the path from i to itself of length $m_{i,j} + m_{j,j} + m_{j,i}$. We have that $(m_{ij} + m_{jj} + m_{ji}) \pmod{d_i} = m_{jj} \pmod{d_i} \neq 0$. This is a contradiction to the definition of d_i. Therefore, $d_j \pmod{d_i} = 0$ and similarly $d_i \pmod{d_j} = 0$, which implies that $d_i = d_j$. □

The claim shows that periodicity is a class property, and all the states in a class have the same period.

Example 3.2. *Add figures explaining the definitions.*

Recurrence:

- State i is *recurrent* if $\mathcal{P}(X_t = i \text{ for some } t \geq 1 | X_0 = i) = 1$. Otherwise, state i is *transient*.
  We can relate the state property of recurrent and transient to the expected number of returns to a state.

  Claim 3.2. *State i is transient iff $\sum_{m=1}^{\infty} p_{i,i}^{(m)} < \infty$.*

  Proof. Assume that state i is transient. Let $q_i = \mathcal{P}(X_t = i \text{ for some } t \geq 1 | X_0 = i)$. Since state i is transient we have $q_i < 1$. Let Z_i be the number of times the trajectory returns to state i. Note that Z_i is geometrically distributed with parameter q_i, namely $\Pr[Z_i = k] = q_i^k(1 - q_i)$. Therefore the expected number of returns to state i is $1/(1 - q_i)$ and is finite. The expected number of returns to state i is equivalently $\sum_{m=1}^{\infty} p_{i,i}^{(m)}$, and hence if a state is transient we have $\sum_{m=1}^{\infty} p_{i,i}^{(m)} < \infty$.

  For the other direction, assume that $\sum_{m=1}^{\infty} p_{i,i}^{(m)} < \infty$. This implies that there is an m_0 such that $\sum_{m=m_0}^{\infty} p_{i,i}^{(m)} < 1/2$. Consider the probability of returning to i within m_0 stages. This implies that $\mathcal{P}(X_t = i \text{ for some } t \geq m_0 | X_0 = i) < 1/2$. Now consider the probability $q_i' = \mathcal{P}(X_t = i \text{ for some } m_0 \geq t \geq 1 | X_0 = i)$. If $q_i' < 1$, this implies that $\mathcal{P}(X_t = i \text{ for some } t \geq 1 | X_0 = i) < q_i' + (1 - q_i')/2 = (1 + q_i')/2 < 1$, which implies that state i is transient. If $q_i' = 1$, this implies that after at most m_0 stages we are guarantee to return to i, hence the

expected number of return to state i is infinite, i.e., $\sum_{m=1}^{\infty} p_{i,i}^{(m)} = \infty$. This is in contradiction to the assumption that $\sum_{m=1}^{\infty} p_{i,i}^{(m)} < \infty$. □

- Recurrence is a class property.
  To see this consider two states i and j such that j is accessible from i and i is recurrent. Since i is recurrant $\sum_{m=1}^{\infty} p_{i,i}^{(m)} = \infty$. Since j is accessible from i there is a k such that $p_{i,j}^{(k)} > 0$. Since i is recurrent, there exists a k' such that $p_{j,i}^{(k')} > 0$. We can lower bound $\sum_{m=1}^{\infty} p_{j,j}^{(m)}$ by $\sum_{m=1}^{\infty} p_{j,i}^{(k')} p_{i,i}^{(m)} p_{i,j}^{(k)} = \infty$. Therefore we showed that state j is recurrent.

- If states i and j are in the same recurrent (communicating) class, then state j is (eventually) reached from state i with probability 1: $\mathcal{P}(X_{\mathtt{t}} = j \text{ for some } \mathrm{t} \geq 1 | X_0 = i) = 1$.
  This follows from the fact that both states occur infinitely often, with probability 1.

- Let T_i be the *return time* to state i (number of stages required for $(X_{\mathtt{t}})$ starting from state i to the first return to i). If i is a recurrent state, then $T_i < \infty$ w.p. 1.
  Since otherwise, there is a positive probability that we never return to state i, and hence state i is not recurrent.

- State i is *positive recurrent* if $\mathbb{E}(T_i) < \infty$, and null recurrent if $\mathbb{E}(T_i) = \infty$. If the state space X is finite, all recurrent states are positive recurrent.
  This follows since the set of states that are null recurrent cannot have transitions from positive recurrent states and cannot have transition to transient states. If the chain never leaves the set of null recurrent states, then some state would have a return time which is at most the size of the set. If there is a positive probability of leaving the set (and never returning) then the states are transient.

Example 3.3. Random walk *Consider the following Markov chain over the integers. The states are the integers. The initial states is* 0*. At each state i, with probability* $1/2$ *we move to $i+1$ and with probability* $1/2$ *to $i-1$. Namely, $p_{i,i+1} = 1/2$, $p_{i,i-1} = 1/2$, and $p_{i,j} = 0$ for $j \notin \{i-1, i+1\}$. We will show that T_i is finite with probability* 1 *and $\mathbb{E}[T_i] = \infty$. This implies that all the states are null recurrent.*

To compute $\mathbb{E}[T_i]$ consider what happens in one and two steps. Let $Z_{i,i-1}$ be the time to move from i to $i-1$. Note that we have

$$\mathbb{E}[T_i] = 1 + 0.5\mathbb{E}[Z_{i+1,i}] + 0.5\mathbb{E}[Z_{i-1,i}] = 1 + \mathbb{E}[Z_{1,0}],$$

since, due to symmetry, $\mathbb{E}[Z_{i,i+1}] = \mathbb{E}[Z_{i+1,i}] = \mathbb{E}[Z_{1,0}]$.

In two steps we are either back to i*, or at state* $i+2$ *or* $i-2$*. For* $\mathbb{E}[T_i]$ *we have that*

$$\begin{aligned}\mathbb{E}[T_i] =& 2 + \frac{1}{4}\mathbb{E}[Z_{i+2,i}] + \frac{1}{4}\mathbb{E}[Z_{i-2,i}] \\ =& 2 + \frac{1}{4}(\mathbb{E}[Z_{i+2,i+1}] + \mathbb{E}[Z_{i+1,i}]) + \frac{1}{4}(\mathbb{E}[Z_{i-2,i-1}] + \mathbb{E}[Z_{i-1,i}]) \\ =& 2 + E[Z_{1,0}]\end{aligned}$$

This implies that we have

$$\mathbb{E}[T_i] = 1 + \mathbb{E}[Z_{1,0}] = 2 + E[Z_{1,0}]$$

Clearly, there is no finite value for $\mathbb{E}[Z_{1,0}]$ *which will satisfy the equation, which implies* $\mathbb{E}[Z_{1,0}] = \infty$*, and hence* $\mathbb{E}[T_i] = \infty$.

To show that 0 *is a recurrent state, note that the probability that at time* $2k$ *we are at state* 0 *is exactly* $p_{0,0}^{(2k)} = \binom{2k}{k}2^{-2k} \approx \frac{c}{\sqrt{k}}$*, for some* $c > 0$*. This implies that*

$$\sum_{m=1}^{\infty} p_{0,0}^{(m)} \approx \sum_{m=1}^{\infty} \frac{c}{\sqrt{m}} = \infty$$

and therefore state 0 *is recurrent. (By symmetry, this shows that all the states are recurrent).*

Note that this Markov chain has a period of 2*.*

Example 3.4. Random walk with jumps. *Consider the following Markov chain over the integers. The states are the integers. The initial states is* 0*. At each state* i*, with probability* $1/2$ *we move to* $i+1$ *and with probability* $1/2$ *we return to* 0*. Namely,* $p_{i,i+1} = 1/2$*,* $p_{i,0} = 1/2$*, and* $p_{i,j} = 0$ *for* $j \notin \{0, i+1\}$*. We will show that* $\mathbb{E}[T_i] < \infty$ *(which implies that* T_i *is finite with probability* 1*).*

From any state we return to 0 *with probability* $1/2$*, therefore* $\mathbb{E}[T_0] = 2$*. We will show that for state* i *we have* $\mathbb{E}[T_i] \leq 2 + 2 \cdot 2^i$*. We will decompose* T_i *to two parts. The first is the return to* 0*, this part has expectation* 2*. The second is to reach state* i *from state* 0*. Consider an epoch as the time between two visits to* 0*. The probability that an epoch would reach* i *is exactly* 2^{-i}*. The expected time of an epoch is* 2 *(the expected time to return to state* 0*). The expected time to return to state* 0*, given that we did not reach state* i *is less than* 2*. Therefore,* $\mathbb{E}[T_i] \leq 2 + 2 \cdot 2^i$*.*

Note that this Markov chain is aperiodic.

Invariant Distribution: The probability vector $\mu = (\mu_i)$ is an invariant distribution or stationary distribution for the Markov chain if $\mu^\top P = \mu^\top$, namely

$$\mu_j = \sum_i \mu_i p_{i,j} \quad \forall j.$$

Clearly, if $X_\mathtt{t} \sim \mu$ then $X_{\mathtt{t}+1} \sim \mu$. If $X_0 \sim \mu$, then the Markov chain $(X_\mathtt{t})$ is a stationary stochastic process.

Theorem 3.1. *Let $(X_\mathtt{t})$ be an irreducible and a-periodic Markov chain over a finite state space X with transition matrix P. Then there is a unique distribution μ such that $\mu^\top P = \mu^\top > 0$.*

Proof. Assume that x is an eigenvector of P with eigenvalue λ, i.e., we have $Px = \lambda x$. Since P is a stochastic matrix, we have $\|Px\|_\infty \leq \|x\|_\infty$, which implies that $\lambda \leq 1$. Since the matrix P is stochastic, $P\vec{1} = \vec{1}$, which implies that P has a right eigenvalue of 1 and this is the maximal eigenvalue. Since the set of right and left eignevalues is identical, we conclude that there is x such that $x^\top P x^\top$. Our first task is to show that there is such an x with $x \geq 0$.

Since the Markov chain is irreducible and a-periodic, there is an integer m, such that P^m has all the entries strictly positive. Namely, for any $i, j \in X$ we have $p_{i,j}^{(m)} > 0$.

We now show a general property of positive matrices (matrices where are the entries are strictly positive). Let $A = P^m$ be a positive matrix and x an eigenvector of A with eigenvalue 1. First, if x has complex number then $Re(x)$ and $Im(x)$ are an eigenvectors of A of eigenvalue 1 and one of them is non-zero. Therefore we can assume that $x \in \mathbb{R}^d$. We would like to show that there is an $x \geq 0$ such that $x^\top A = x^\top$. If $x \geq 0$ we are done. If $x \leq 0$ we can take $x' = -x$ and we are done. We need to show that x cannot have both positive and negative entries.

For contradiction, assume that we have $x_k > 0$ and $x_{k'} < 0$. This implies that for any weight vector $w > 0$ we have $|x^\top w| < |x|^\top w$, where $|x|$ is point-wise absolute value. Therefore,

$$\sum_j |x_j| = \sum_j |\sum_i x_i P_{i,j}| < \sum_j \sum_i |x_i| P_{i,j} = \sum_i |x_i| \sum_j P_{i,j} = \sum_j |x_j|$$

where the first identity follows since x is an eigenvector. The second since P is strictly positive.The third is a change of order of summation. The last follows since P is a stochastic matrix, so each row sums to 1. Clearly, we reached an contradiction, and therefore x cannot have both positive and negative entries.

We have shown so far that there exists a μ such that $\mu^\top P = \mu^\top$ and $\mu \geq 0$. Since $A = P^m$ is strictly positive, then $\mu^\top = \mu^\top A > 0$.

To show the uniqueness of μ, assume we have x and y such that $x^\top P = x^\top$ and $y^\top P = y^\top$ and $x \neq y$. Recall that we showed that in such a case both $x > 0$ and $y > 0$. Then there is a linear combination $z = ax + by$ such that for some i we have $z_i = 0$. Since $z^\top P = z^\top$, we have showed that z is strictly positive, i.e., $z > 0$, which is a contradiction. Therefore, $x = y$, and hence μ is unique. □

We define the average fraction that a state $j \in X$ occurs, given that we start with an initial state distribution x_0, as follows:

$$\pi_j^{(m)} = \frac{1}{m} \sum_{t=1}^{m} \mathbb{I}(X_t = j)$$

Theorem 3.2. *Let $(X_\mathtt{t})$ be an irreducible and a-periodic Markov chain over a finite state space X with transition matrix P. Let μ be the stationary distribution of P. Then, for any $j \in X$ we have*

$$\mu_j = \lim_{m \to \infty} \mathbb{E}[\pi_j^{(m)}] = \frac{1}{\mathbb{E}[T_j]}$$

Proof. We have that

$$\mathbb{E}[\pi_j^{(m)}] = \mathbb{E}[\frac{1}{m} \sum_{t=1}^{m} \mathbb{I}(X_t = j)] = \frac{1}{m} \sum_{t=1}^{m} \Pr[X_t = j | X_0 = x_0] = \frac{1}{m} \sum_{t=1}^{m} x_0^\top P^t$$

Let $v_1, \ldots, v_n$ be the eigenvectors of P with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$. By Theorem 3.1 that $v_1 = \mu$, the stationary distribution and $\lambda_1 = 1 > \lambda_i$ for $i \geq 2$. Rewrite $x_0 = \sum_i \alpha_i v_i$. Since P^m is a stochastic matrix, $x_0^\top P^m$ is a distribution, and therefore $\lim_{m \to \infty} x_0^\top P^m = \mu$.

We will be interested in the limit $\pi_j = \lim_{m \to \infty} \pi_j^m$, and mainly in the expected value $\mathbb{E}[\pi_j]$. From the above we have that $\mathbb{E}[\pi_j] = \mu_j$.

A different way to express $\mathbb{E}[\pi_j]$ is using a variable time horizon, with a fixed number of occurrences of j. Let $T_{k,j}$ be the time between the k and $k+1$ occurrence of state j. This implies that

$$\lim_{m \to \infty} \frac{1}{m} \sum_{t=1}^{m} \mathbb{I}(X_t = j) = \lim_{n \to \infty} \frac{n}{\sum_{k=1}^{n} T_{k,j}}$$

Note that the $T_{k,j}$ are i.i.d. and equivalent to T_j. By the law of large numbers we have that $\frac{1}{n}\sum_{k=1}^{n} T_{k,j}$ converges to $\mathbb{E}[T_i]$. Therefore,

$$\mathbb{E}[\pi_j] = \frac{1}{\mathbb{E}[T_j]}$$

□

We have established the following general theorem.

Theorem 3.3 (**Recurrence of finite Markov chains**). *Let (X_t) be an irreducible, a-periodic Markov chain over a finite state space X. Then the following properties hold:*

1. *All states are **positive recurrent***
2. *There exists a **unique stationary distribution** μ^*, where $\mu^*(i) = 1/\mathbb{E}[T_i]$.*
3. ***Convergence** to the stationary distribution: $\lim_{t\to\infty} p_{i,j}^{(t)} = \mu_j \quad (\forall j)$*
4. ***Ergodicity**: For any finite f: $\lim_{t\to\infty} \frac{1}{t}\sum_{s=0}^{t-1} f(X_s) = \sum_i \mu_i f(i) \triangleq \pi \cdot f$.*

For countable Markov chains, there are other possibilities.

Theorem 3.4 (**Countable Markov chains**). *Let (X_t) be an irreducible and a-periodic Markov chain over a countable state space X. Then: Either (i) all states are positive recurrent, or (ii) all states are null recurrent, or (iii) all states are transient.*

Proof. Let i is a positive recurrent state, then we will show that all states are positive recurrent. For any state j, since the Markov chain is irreducible, we have for some $m_1, m_2 \geq 0$ that $p_{j,i}^{(m_1)}, p_{i,j}^{(m_2)} > 0$. This implies that the return time to state j is at most $\mathbb{E}[T_j] \leq 1/p_{j,i}^{(m_1)} + \mathbb{E}[T_i] + 1/p_{i,j}^{(m_2)}$, and hence j is positive recurrent.

If there is no positive recurrent state, let i be a null recurrent state, then we will show that all states are null recurrent. For any state j, since the Markov chain is irreducible, we have for some $m_1, m_2 \geq 0$ that $p_{j,i}^{(m_1)}, p_{i,j}^{(m_2)} > 0$. This implies that the probability to return time to state j is at least $\sum_{m=0}^{\infty} p_{j,i}^{(m_1)}\ p_{i,i}^{(m)}\ p_{i,j}^{(m_2)} = \infty$, since we have $\sum_{m=0}^{\infty} p_{i,i}^{(m)} = \infty$, since i is a recurrent state. This implies that j is a recurrent state. Since there are no positive recurrent states, it has to be that j is a null recurrent state.

If there are no positive or null recurrent states, then all states are transient. □

Reversible Markov chains: Suppose there exists a probability vector $\mu = (\mu_i)$ so that

$$\mu_i p_{i,j} = \mu_j p_{j,i}, \qquad i, j \in X. \tag{3.1}$$

It is then easy to verify by direct summation that μ is an invariant distribution for the Markov chain defined by $(p_{i,j})$. This follows since $\sum_i \mu_i p_{i,j} = \sum_i p_{i,j}\mu_j = \mu_j$. The equations (3.1) are called the *detailed balance equations*. A Markov chain that satisfies these equations is called reversible.

Example 3.5 (Discrete-time queue). *Consider a discrete-time queue, with queue length $X_t \in \mathbb{N}_0 = \{0, 1, 2, \dots\}$. At time instant t, A_t new jobs arrive, and then up to S_t jobs can be served, so that*

$$X_{t+1} = (X_t + A_t - S_t)^+.$$

Suppose that (S_t) is a sequence of i.i.d. RVs, and similarly (A_t) is a sequence of i.i.d. RVs, with (S_t), (A_t) and X_0 mutually independent. It may then be seen that $(X_t,\ t \geq 0)$ is a Markov chain. Suppose further that each S_t is a Bernoulli RV with parameter q, namely $P(S_t = 1) = q$, $P(S_t = 0) = 1 - q$. Similarly, let A_t be a Bernoulli RV with parameter p. Then

$$p_{i,j} = \begin{cases} p(1-q) & : \ j = i+1 \\ (1-p)(1-q) + pq & : \ j = i, \ \ i > 0 \\ (1-p)q & : \ j = i-1, \ \ i > 0 \\ (1-p) + pq & : \ j = i = 0 \\ 0 & : \ \text{otherwise} \end{cases}$$

Denote $\lambda = p(1-q)$, $\eta = (1-p)q$, and $\rho = \lambda/\eta$. The detailed balance equations for this case are:

$$\mu_i \lambda = \mu_{i+1}\eta, \qquad i \geq 0$$

These equations have a solution with $\sum_i \mu_i = 1$ if and only if $\rho < 1$. The solution is $\mu_i = \mu_0 \rho^i$, with $\pi_0 = 1 - \rho$. This is therefore the stationary distribution of this queue.

Chapter 4

Markov Decision Processes and Finite Horizon

In Chapter 2 we considered multi-stage decision problems for *deterministic* systems. In many problems of interest, the system dynamics also involves *randomness*, which leads us to stochastic decision problems. In this chapter we introduce the basic model of Markov Decision Processes (MDP), which will be considered in the rest of the book, and discuss the finite horizon return.

4.1 Controlled Markov Chains

A Markov Decision Process consists of two main parts:

1. A controlled dynamic system, with stochastic evolution.
2. A performance objective to be optimized.

In this section we describe the first part, which is modeled as a controlled Markov chain.

Consider a controlled dynamic system, defined over:

- A discrete time axis $\mathbb{T} = \{0, 1, \ldots, \mathtt{T}-1\}$ (finite horizon), or $\mathbb{T} = \{0, 1, 2, \ldots\}$ (infinite horizon). To simplify the discussion we refer below to the infinite horizon case, which can always be "truncated" at $\mathtt{T}$ if needed.
- A finite state space $\mathcal{S}$, where $\mathcal{S}_{\mathtt{t}} \subset \mathcal{S}$ is the set of possible states at time $\mathtt{t} \in \mathbb{T}$.
- A finite action set $\mathcal{A}$, where $\mathcal{A}_{\mathtt{t}}(\mathtt{s}) \subset \mathcal{A}$ is the set of possible actions at time $\mathtt{t} \in \mathbb{T}$ and state $\mathtt{s} \in \mathcal{S}_{\mathtt{t}}$.

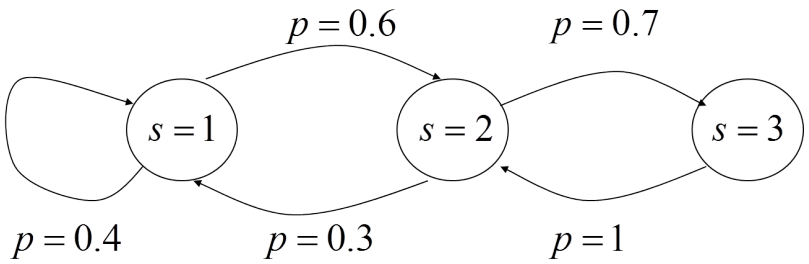


Figure 4.1: Markov chain

State transition probabilities:

- Suppose that at time $\mathtt{t}$ we are in state $\mathtt{s}_\mathtt{t} = \mathtt{s}$, and choose an action $\mathtt{a}_\mathtt{t} = \mathtt{a}$. The next state $\mathtt{s}_{\mathtt{t}+1} = \mathtt{s}'$ is then determined randomly according to a probability distribution $p_\mathtt{t}(\cdot|\mathtt{s}, \mathtt{a})$ on $\mathcal{S}_{\mathtt{t}+1}$. That is,

$$\Pr(\mathtt{s}_{\mathtt{t}+1} = \mathtt{s}'|\mathtt{s}_\mathtt{t} = \mathtt{s}, \mathtt{a}_\mathtt{t} = \mathtt{a}) = p_\mathtt{t}(\mathtt{s}'|\mathtt{s}, \mathtt{a}), \qquad \mathtt{s}' \in \mathcal{S}_{\mathtt{t}+1}$$

- The probability $p_\mathtt{t}(\mathtt{s}'|\mathtt{s}, \mathtt{a})$ is the *transition probability* from state $\mathtt{s}$ to state $\mathtt{s}'$ for a given action $\mathtt{a}$. We naturally require that $p_\mathtt{t}(\mathtt{s}'|\mathtt{s}, \mathtt{a}) \geq 0$, and $\sum_{\mathtt{s}' \in \mathcal{S}_{\mathtt{t}+1}} p_\mathtt{t}(\mathtt{s}'|\mathtt{s}, \mathtt{a}) = 1$ for all $\mathtt{s} \in \mathcal{S}_\mathtt{t}, \mathtt{a} \in \mathcal{A}_\mathtt{t}(\mathtt{s})$.

- Implicit in this definition is the controlled-Markov property:

$$\Pr(\mathtt{s}_{\mathtt{t}+1} = \mathtt{s}'|\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}, \ldots, \mathtt{s}_{0,}\mathtt{a}_0) = \Pr(\mathtt{s}_{\mathtt{t}+1} = \mathtt{s}'|\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$$

- The set of probability distributions

$$P = \{p_\mathtt{t}(\cdot|\mathtt{s}, \mathtt{a}) \quad : \quad \mathtt{s} \in \mathcal{S}_\mathtt{t}, \mathtt{a} \in \mathcal{A}_\mathtt{t}(\mathtt{s}), \mathtt{t} \in \mathbb{T}\}$$

  is called the *transition law* or *transition kernel* of the controlled Markov process.

Stationary Models: The controlled Markov chain is called stationary or time-invariant if the transition probabilities do not depend on the time $\mathtt{t}$. That is:

$$\forall t, \quad p_\mathtt{t}(\mathtt{s}'|\mathtt{s}, \mathtt{a}) \equiv p(\mathtt{s}'|\mathtt{s}, \mathtt{a}), \;\; \mathcal{S}_\mathtt{t} \equiv \mathcal{S}, \;\; \mathcal{A}_\mathtt{t}(\mathtt{s}) \equiv \mathcal{A}(\mathtt{s}).$$

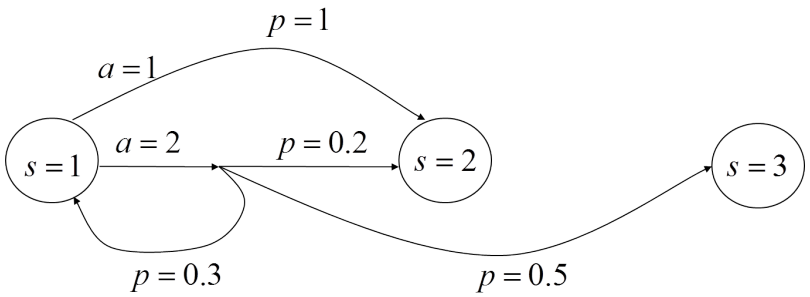


Figure 4.2: Controlled Markov chain

Graphical Notation: The state transition probabilities of a Markov chain are often illustrated via a state transition diagram, such as in Figure 4.1.

A graphical description of a controlled Markov chain is a bit more complicated because of the additional action variable. We obtain the diagram (drawn for state $\mathtt{s} = 1$ only, and for a given time $\mathtt{t}$) in Figure 4.2, reflecting the following transition probabilities:

$$p(\mathtt{s}' = 2|\mathtt{s} = 1, \mathtt{a} = 1) = 1$$
$$p(\mathtt{s}'|\mathtt{s} = 1, \mathtt{a} = 2) = \begin{cases} 0.3 & : \quad \mathtt{s}' = 1 \\ 0.2 & : \quad \mathtt{s}' = 2 \\ 0.5 & : \quad \mathtt{s}' = 3 \end{cases}$$

State-equation notation: The stochastic state dynamics can be equivalently defined in terms of a state equation of the form

$$\mathtt{s}_{\mathtt{t}+1} = f_{\mathtt{t}}(\mathtt{s}_{\mathtt{t}}, \mathtt{a}_{\mathtt{t}}, w_{\mathtt{t}}),$$

where $w_{\mathtt{t}}$ is a random variable (RV). If $(w_{\mathtt{t}})_{t\geq 0}$ is a sequence of independent RVs, and further each $w_{\mathtt{t}}$ is independent of the "past" $(\mathtt{s}_{\mathtt{t}-1}, \mathtt{a}_{\mathtt{t}-1}, \ldots \mathtt{s}_0)$, then $(\mathtt{s}_{\mathtt{t}}, \mathtt{a}_{\mathtt{t}})_{t\geq 0}$ is a controlled Markov process. For example, the state transition law of the last example can be written in this way, using $w_{\mathtt{t}} \in \{4, 5, 6\}$, with $p_w(4) = 0.3,\ p_w(5) = 0.2,\ p_w(6) = 0.5$ and, for $\mathtt{s}_{\mathtt{t}} = 1$:

$$f_{\mathtt{t}}(1, 1, w_{\mathtt{t}}) = 2$$
$$f_{\mathtt{t}}(1, 2, w_{\mathtt{t}}) = w_{\mathtt{t}} - 3$$

This state algebraic equation notation is especially useful for problems with continuous state space, but also for some models with discrete states. Equivalently, we can write

$$f_{\mathtt{t}}(1, 2, w_{\mathtt{t}}) = 1 \cdot \mathbb{I}[w_{\mathtt{t}} = 4] + 2 \cdot \mathbb{I}[w_{\mathtt{t}} = 5] + 3 \cdot \mathbb{I}[w_{\mathtt{t}} = 6],$$

where $\mathbb{I}[\cdot]$ is the indicator function.

Next we recall the definitions of control policies from Chapter 2.

Control Policies

- A general or **history-dependent deterministic** control policy $\pi = (\pi_{\mathtt{t}})_{\mathtt{t}\in\mathbb{T}}$ is a mapping from each possible history $\mathtt{h_t} = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{\mathtt{t}-1}, \mathtt{a}_{\mathtt{t}-1}, \mathtt{s_t})$, and time $\mathtt{t} \in \mathbb{T}$, to an action $\mathtt{a_t} = \pi_{\mathtt{t}}(\mathtt{h_t}) \in \mathcal{A}_{\mathtt{t}}$. We denote the set of general policies by Π_{HD}.

- A **Markov deterministic** control policy π is allowed to depend on the current state and time only: $\mathtt{a_t} = \pi_{\mathtt{t}}(\mathtt{s_t})$. We denote the set of Markov deterministic policies by Π_{MD}.

- For stationary models, we may define **stationary deterministic** control policies that depend on the current state alone. A stationary policy is defined by a single mapping $\pi : \mathcal{S} \to \mathcal{A}$, so that $\mathtt{a_t} = \pi(\mathtt{s_t})$ for all $\mathtt{t} \in \mathbb{T}$. We denote the set of stationary policies by Π_{SD}.

- Evidently, $\Pi_{HD} \supset \Pi_{MD} \supset \Pi_{SD}$.

Randomized Control policies

- The control policies defined above specify deterministically the action to be taken at each stage. In some cases we want to allow for a random choice of action.

- A general randomized control policy assigns to each possible history $\mathtt{h_t}$ a probability distribution $\pi_{\mathtt{t}}(\cdot|\mathtt{h_t})$ over the action set $\mathcal{A}_{\mathtt{t}}$. That is, $\Pr(\mathtt{a_t} = \mathtt{a}|\mathtt{h_t}) = \pi_{\mathtt{t}}(\mathtt{a}|\mathtt{h_t})$. We denote the set of history-dependent stochastic policies by Π_{HS}.

- Similarly, we can define the set Π_{MS} of Markov stochastic control policies, where $\pi_{\mathtt{t}}(\cdot|\mathtt{h_t})$ is replaced by $\pi_{\mathtt{t}}(\cdot|\mathtt{s_t})$, and the set Π_{SS} of stationary stochastic control policies, where $\pi_{\mathtt{t}}(\cdot|\mathtt{s_t})$ is replaced by $\pi(\cdot|\mathtt{s_t})$, namely the policy is independent of the time.

- Note that the set Π_{HS} includes all other policy sets as special cases.

The Induced Stochastic Process Let $p_0 = \{p_0(\mathtt{s}), \mathtt{s} \in \mathcal{S}_0\}$ be a probability distribution for the initial state $\mathtt{s}_0$. (Many time we will assume that the initial state is deterministic and given by $\mathtt{s}_0$.) A control policy $\pi \in \Pi_{HS}$, together with the transition law $P = \{p_\mathtt{t}(\mathtt{s}'|\mathtt{s}, \mathtt{a})\}$ and the initial state distribution $p_0 = (p_0(\mathtt{s}),\ \mathtt{s} \in \mathcal{S}_0)$, induces a probability distribution over any finite state-action sequence $\mathtt{h}_\mathtt{T} = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{\mathtt{T}-1}, \mathtt{a}_{\mathtt{T}-1}, \mathtt{s}_\mathtt{T})$, given by

$$\Pr(\mathtt{h}_\mathtt{T}) = p_0(\mathtt{s}_0) \prod_{t=0}^{\mathtt{T}-1} p_\mathtt{t}(\mathtt{s}_{\mathtt{t}+1}|\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) \pi_\mathtt{t}(\mathtt{a}_\mathtt{t}|\mathtt{h}_\mathtt{t}),$$

where $\mathtt{h}_\mathtt{t} = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{\mathtt{T}-1}, \mathtt{a}_{\mathtt{T}-1}, \mathtt{s}_\mathtt{t})$. To see this, observe the recursive relation:

$$\begin{aligned}\Pr(\mathtt{h}_{\mathtt{t}+1}) &= \Pr(\mathtt{h}_\mathtt{t}, \mathtt{a}_\mathtt{t}, \mathtt{s}_{\mathtt{t}+1}) = \Pr(\mathtt{s}_{\mathtt{t}+1}|\mathtt{h}_\mathtt{t}, \mathtt{a}_\mathtt{t}) \Pr(\mathtt{a}_\mathtt{t}|\mathtt{h}_\mathtt{t}) \Pr(\mathtt{h}_\mathtt{t}) \\ &= p_\mathtt{t}(\mathtt{s}_{\mathtt{t}+1}|\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) \pi_\mathtt{t}(\mathtt{a}_\mathtt{t}|\mathtt{h}_\mathtt{t}) \Pr(\mathtt{h}_\mathtt{t}).\end{aligned}$$

In the last step we used the conditional Markov property of the controlled chain: $\Pr(\mathtt{s}_{\mathtt{t}+1}|\mathtt{h}_\mathtt{t}, \mathtt{a}_\mathtt{t}) = p_\mathtt{t}(\mathtt{s}_{\mathtt{t}+1}|\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$, and the definition of the control policy $\pi_\mathtt{t}$. The required formula follows by recursion.

Therefore, the state-action sequence $\mathtt{h}_\infty = (\mathtt{s}_k, \mathtt{a}_k)_{k \geq 0}$ can now be considered a stochastic process. We denote the probability law of this stochastic process by $\Pr^{\pi, p_0}(\cdot)$. The corresponding expectation operator is denoted by $\mathbb{E}^{\pi, p_0}(\cdot)$. When the initial state $\mathtt{s}_0$ is deterministic (i.e., $p_0(\mathtt{s})$ is concentrated on a single state $\mathtt{s}$), we may simply write $\Pr^{\pi, s}(\cdot)$ or $\Pr^\pi(\cdot|\mathtt{s}_0 = \mathtt{s})$.

Under a Markov control policy, the state sequence $(\mathtt{s}_\mathtt{t})_{\mathtt{t} \geq 0}$ becomes a *Markov chain*, with transition probabilities:

$$\Pr(\mathtt{s}_{\mathtt{t}+1} = \mathtt{s}'|\mathtt{s}_\mathtt{t} = \mathtt{s}) = \sum\nolimits_{\mathtt{a} \in \mathcal{A}_\mathtt{t}} p_\mathtt{t}(\mathtt{s}'|\mathtt{s}, \mathtt{a}) \pi_\mathtt{t}(\mathtt{a}|\mathtt{s}).$$

This follows since:

$$\begin{aligned}\Pr(\mathtt{s}_{\mathtt{t}+1} = \mathtt{s}'|\mathtt{s}_\mathtt{t} = \mathtt{s}) &= \sum\nolimits_{\mathtt{a} \in \mathcal{A}_\mathtt{t}} \Pr(\mathtt{s}_{\mathtt{t}+1} = \mathtt{s}', \mathtt{a}|\mathtt{s}_\mathtt{t} = \mathtt{s}) \\ &= \sum\nolimits_{\mathtt{a} \in \mathcal{A}_\mathtt{t}} \Pr(\mathtt{s}_{\mathtt{t}+1} = \mathtt{s}'|\mathtt{s}_\mathtt{t} = \mathtt{s}, \mathtt{a}) \Pr(\mathtt{a}|\mathtt{s}_\mathtt{t} = \mathtt{s}) \\ &= \sum\nolimits_{\mathtt{a} \in \mathcal{A}_\mathtt{t}} p_\mathtt{t}(\mathtt{s}'|\mathtt{s}, \mathtt{a}) \pi_\mathtt{t}(\mathtt{a}|\mathtt{s})\end{aligned}$$

If the controlled Markov chain is stationary (time-invariant) and the control policy is stationary, then the induced Markov chain is stationary as well.

Remark 4.1. *For most non-learning optimization problems, Markov policies suffice to achieve the optimum.*

Remark 4.2. *Implicit in these definitions of control policies is the assumption that the current state* $\mathtt{s}_\mathtt{t}$ *can be fully observed before the action* $\mathtt{a}_\mathtt{t}$ *is chosen . If this is not the case we need to consider the problem of a Partially Observed MDP (POMDP), which is more involved and will be discussed in Chapter 16.*

4.2 Performance Criteria

4.2.1 Finite Horizon Return

Consider the finite-horizon return, with a fixed time horizon $\mathtt{T}$. As in the deterministic case, we are given a running reward function $\mathtt{r}_\mathtt{t} = \{\mathtt{r}_\mathtt{t}(\mathtt{s},\mathtt{a}) : \mathtt{s} \in \mathcal{S}_\mathtt{t}, \mathtt{a} \in \mathcal{A}_\mathtt{t}\}$ for $0 \leq \mathtt{t} \leq \mathtt{T}-1$, and a terminal reward function $\mathtt{r}_\mathtt{T} = \{\mathtt{r}_\mathtt{T}(\mathtt{s}) : \mathtt{s} \in \mathcal{S}_\mathtt{T}\}$. The obtained reward is $R_\mathtt{t} = \mathtt{r}_\mathtt{t}(\mathtt{s}_\mathtt{t},\mathtt{a}_\mathtt{t})$ at times $\mathtt{t} \leq \mathtt{T}-1$, and $R_\mathtt{T} = \mathtt{r}_\mathtt{T}(\mathtt{s}_\mathtt{T})$ at the last stage. (Note that $\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}$ and $\mathtt{s}_\mathtt{T}$ are random variables that depend both on the policy π and the stochastic transitions.) Our general goal is to maximize the cumulative return:

$$\sum_{t=0}^{\mathtt{T}} R_\mathtt{t} = \sum_{t=0}^{\mathtt{T}-1} \mathtt{r}_\mathtt{t}(\mathtt{s}_\mathtt{t},\mathtt{a}_\mathtt{t}) + \mathtt{r}_\mathtt{T}(\mathtt{s}_\mathtt{T}).$$

However, since the system is stochastic, the cumulative return will generally be a random variable, and we need to specify in which sense to maximize it. A natural first option is to consider the expected value of the return. That is, define:

$$\mathcal{V}^\pi_\mathtt{T}(\mathtt{s}) = \mathbb{E}^\pi(\sum_{t=0}^{\mathtt{T}} R_\mathtt{t} | \mathtt{s}_0 = \mathtt{s}) \equiv \mathbb{E}^{\pi,s}(\sum_{t=0}^{\mathtt{T}} R_\mathtt{t}).$$

Here π is the control policy as defined above, and $\mathtt{s}$ denotes the initial state. Hence, $\mathcal{V}^\pi_\mathtt{T}(\mathtt{s})$ is the expected cumulative return under the control policy π. Our goal is to find an optimal control policy that maximizes $\mathcal{V}^\pi_\mathtt{T}(\mathtt{s})$.

Remarks:

1. Dependence on the next state: In some problems, the obtained reward may depend on the next state as well: $R_\mathtt{t} = \tilde{\mathtt{r}}_\mathtt{t}(\mathtt{s}_\mathtt{t},\mathtt{a}_\mathtt{t},\mathtt{s}_{\mathtt{t}+1})$. For control purposes, when we only consider the expected value of the reward, we can reduce this reward function to the usual one by defining

$$\mathtt{r}_\mathtt{t}(\mathtt{s},\mathtt{a}) \stackrel{\Delta}{=} \mathbb{E}(R_\mathtt{t} | \mathtt{s}_\mathtt{t} = \mathtt{s}, \mathtt{a}_\mathtt{t} = \mathtt{a}) \equiv \sum\nolimits_{\mathtt{s}' \in S} p(\mathtt{s}' | \mathtt{s},\mathtt{a}) \tilde{r}_\mathtt{t}(s,a,\mathtt{s}')$$

2. Random rewards: The reward $R_{\mathtt{t}}$ may also be random, namely a random variable whose distribution depends on $(\mathtt{s}_{\mathtt{t}}, \mathtt{a}_{\mathtt{t}})$. This can also be reduced to our standard model for planning purposes by looking at the expected value of $R_{\mathtt{t}}$, namely
$$\mathtt{r}_{\mathtt{t}}(\mathtt{s}, \mathtt{a}) = \mathbb{E}(R_{\mathtt{t}} | \mathtt{s}_{\mathtt{t}} = \mathtt{s}, \mathtt{a}_{\mathtt{t}} = \mathtt{a}).$$

3. Risk-sensitive criteria: The expected cumulative return is by far the most common goal for planning. However, it is not the only one possible. For example, one may consider the following risk-sensitive return function:
$$\mathcal{V}^{\pi}_{\mathtt{T},\lambda}(\mathtt{s}) = \frac{1}{\lambda} \log \mathbb{E}^{\pi,s}(\exp(\lambda \sum_{t=0}^{\mathtt{T}} R_{\mathtt{t}})).$$
For $\lambda > 0$, the exponent gives higher weight to high rewards, and the opposite for $\lambda < 0$.

In the case that the rewards are stochastic, but have a discrete support, we can construct an equivalent MDP in which all the rewards are deterministic and has the same distribution of rewards. This implies that the important challenge is the stochastic state transition function, and the rewards can be assumed to be deterministic. Formally, given a trajectory we define a *rewards trajectory* as the sub-trajectory that includes only the rewards.

Theorem 4.1. *Given an MDP $M(\mathcal{S}, \mathcal{A}, p, \mathtt{r}, \mathtt{s}_0)$, where the rewards are stochastic, with support $\mathcal{K} = \{1, \ldots, k\}$, there is an MDP $M'(\mathcal{S} \times \mathcal{K}, \mathcal{A}, p', \mathtt{r}', \mathtt{s}'_0)$, such that and a mapping of policies π of M to π' policies of M', such that: running π in M for horizon $\mathtt{T}$ generates reward trajectory $R = (R_0, \ldots, R_{\mathtt{T}})$ and running π' in M' for horizon $\mathtt{T}+1$ generates reward trajectory $R = (R_1, \ldots, R_{\mathtt{T}+1})$, then the distribution of R and R' are identical.*

Proof. The basic idea is to encode the rewards in the states of M' which are $\mathcal{S} \times \mathcal{K} = \mathcal{S}'$. For each $(\mathtt{s}, i) \in \mathcal{S}'$ and action $\mathtt{a} \in \mathcal{A}$ we have $p'_t((\mathtt{s}', j)|(\mathtt{s}, i), \mathtt{a}) = p_t(\mathtt{s}'|\mathtt{s}, \mathtt{a}) \Pr[R_t(\mathtt{s}, \mathtt{a}) = j]$, and $p'_{\mathtt{T}}((\mathtt{s}', j)|(\mathtt{s}, i)) = \mathbb{I}(\mathtt{s}' = \mathtt{s}) \Pr[R_{\mathtt{T}}(\mathtt{s}) = j]$. The reward is $\mathtt{r}'_t((\mathtt{s}, i), \mathtt{a}) = i$. The initial state is $\mathtt{s}'_0 = (\mathtt{s}_0, 0)$.

For any policy $\pi(\mathtt{a}|\mathtt{s})$ in M we have a policy π' is M' where $\pi'(\mathtt{a}|(\mathtt{s}, i)) = \pi(\mathtt{a}|\mathtt{s})$.

We map trajectories of M to trajectories of M' which have identical probabilities. A trajectory $(\mathtt{s}_0, \mathtt{a}_0, R_0, \mathtt{s}_1, \mathtt{a}_1, R_1, \mathtt{s}_2 \ldots, R_{\mathtt{T}})$ is map to $((\mathtt{s}_0, 0), \mathtt{a}_0, 0, (\mathtt{s}_1, R_0), \mathtt{a}_1, R_0, (\mathtt{s}_2, R_1) \ldots, R_{\mathtt{T}+1})$. Let R and R' be the respective reward trajectories. Clearly, the two trajectory have identical probabilities. This implies that the rewards trajectories R and R' have are identical probabilities (up to a shift of one in the index). □

Theorem 4.1 requires the number of rewards be bounded, and guarantees that the reward distribution be identical. In the case that the rewards are continuous, we can have a similar guarantee for linear return functions.

Theorem 4.2. *Given an MDP $M(\mathcal{S}, \mathcal{A}, p, \mathtt{r}, \mathtt{s}_0)$, where the rewards are stochastic, with support $[0, 1]$, there is an MDP $M'(\mathcal{S}, \mathcal{A}, p, \mathtt{r}', \mathtt{s}_0)$, where the rewards are stochastic, with support $\{0, 1\}$, such that for any policy $\pi \in \Pi_{MS}$ the distribution of the expected rewards trajectory is identical.*

Proof. We simply change the reward of $(\mathtt{s}, \mathtt{a})$ to be $\{0, 1\}$ by changing them to be a Bernoulli random variables with a parameter $\mathtt{r}_t(\mathtt{s}, \mathtt{a})$, i.e., $\Pr[\mathtt{R}_\mathtt{t}(\mathtt{s}, \mathtt{a}) = 1] = \mathtt{r}_\mathtt{t}(\mathtt{s}, \mathtt{a})$ and $\Pr[\mathtt{R}_\mathtt{t}(\mathtt{s}, \mathtt{a}) = 0] = 1 - \mathtt{r}_\mathtt{t}(\mathtt{s}, \mathtt{a})$. Clearly, the expected value of the rewards is identical. Further, since $\pi \in \Pi_{MS}$, it depends only of $\mathtt{s}$ and $\mathtt{t}$, which implies that the behavior (states and actions) will be identical in M and M'. □

We have also established the following corollary.

Corollary 4.1. *Given an MDP $M(\mathcal{S}, \mathcal{A}, p, \mathtt{r}, \mathtt{s}_0)$, where the rewards are stochastic, with support $[0, 1]$, there is an MDP $M'(\mathcal{S} \times \{0, 1\}, \mathcal{A}, p', \mathtt{r}', \mathtt{s}'_0)$, and a mapping of policies $\pi \in \Pi_{MS}$ of M to $\pi' \in \Pi_{MD}$ policies of M', such that $\mathcal{V}_\mathtt{T}^{\pi, M}(\mathtt{s}_0) = \mathcal{V}_{\mathtt{T}+1}^{\pi', M'}(\mathtt{s}'_0)$*

4.2.2 Infinite Horizon Problems

We next consider planning problems that extend to an infinite time horizon, $\mathtt{t} = 0, 1, 2, \ldots$. Such planning problems arise when the system in question is expected to operate for a long time, or a large number of steps, possibly with no specific "closing" time. Infinite horizon problems are most often defined for stationary problems. In that case, they enjoy the important advantage that optimal policies can be found among the class of stationary policies. We will restrict attention here to stationary models. As before, we have the running reward function $\mathtt{r}(\mathtt{s}, \mathtt{a})$, which extends to all $\mathtt{t} \geq 0$. The expected reward obtained at stage $\mathtt{t}$ is $\mathbb{E}[R_\mathtt{t}] = \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$.

Discounted return: The most common performance criterion for infinite horizon problems is the expected discounted return:

$$\mathcal{V}_\gamma^\pi(\mathtt{s}) = \mathbb{E}^\pi(\sum_{t=0}^{\infty} \gamma^t \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) | \mathtt{s}_0 = \mathtt{s}) \equiv \mathbb{E}^{\pi, s}(\sum_{t=0}^{\infty} \gamma^t \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}))\ ,$$

where $0 < \gamma < 1$ is the discount factor. Mathematically, the discount factor ensures convergence of the sum (whenever the reward sequence is bounded). This make the problem "well behaved", and relatively easy to analyze. The discounted return is discussed in Chapter 5.

Average return: Here we are interested to maximize the long-term average return. The most common definition of the long-term average return is

$$\mathcal{V}^{\pi}_{av}(\mathtt{s}) = \liminf_{\mathtt{T}\to\infty} \mathbb{E}^{\pi,s}(\frac{1}{\mathtt{T}}\sum_{t=0}^{\mathtt{T}-1} \mathtt{r}(\mathtt{s_t}, \mathtt{a_t}))$$

The theory of average-return planning problems is more involved, and relies to a larger extent on the theory of Markov chains (see Chapter 3). The average return is discussed in Chapter 7.

4.2.3 Stochastic Shortest-Path Problems

In an important class of planning problems, the time horizon is not set beforehand, but rather the problem continues until a certain event occurs. This event can be defined as reaching some goal state. Let $\mathcal{S}_G \subset \mathcal{S}$ define the set of *goal states*. Define

$$\tau = \inf\{t \geq 0 : \mathtt{s_t} \in \mathcal{S}_G\}$$

as the first time in which a goal state is reached. The total expected return for this problem is defined as:

$$\mathcal{V}^{\pi}_{ssp}(\mathtt{s}) = \mathbb{E}^{\pi,\mathtt{s}}(\sum_{\mathtt{t}=0}^{\tau-1} \mathtt{r}(\mathtt{s_t}, \mathtt{a_t}) + \mathtt{r}_G(\mathtt{s}_\tau))$$

Here $\mathtt{r}_G(\mathtt{s})$, $\mathtt{s} \in \mathcal{S}_G$ specified the reward at goal states. Note that the length of the run τ is a random variable.

Stochastic shortest path includes, naturally, the finite horizon case. This can be shown by creating a leveled MDP where at each time step we move to the next level and terminate at level $\mathtt{T}$. Specifically, we define a new state space $\mathcal{S}' = \mathcal{S} \times \mathbb{T}$, transition function $p((\mathtt{s}', i+1)|\mathtt{s}, \mathtt{a}) = p(\mathtt{s}'|\mathtt{s}, \mathtt{a})$, and goal states $\mathcal{S}_G = \{(\mathtt{s}, \mathtt{T}) : \mathtt{s} \in \mathcal{S}\}$.

Stochastic shortest path includes also the discounted infinite horizon. To see that, add a new goal state, and from each state with probability $1-\gamma$ jump to the goal state and terminate. The expected return of a policy would be the same in both models. Specifically, we add a state $\mathtt{s}_G$, such that $p(\mathtt{s}_G|\mathtt{s}, \mathtt{a}) = 1-\gamma$, for any state $\mathtt{s} \in \mathcal{S}$ and action $\mathtt{a} \in \mathcal{A}$ and $p(\mathtt{s}'|\mathtt{s}, \mathtt{a}) = \gamma p(\mathtt{s}'|\mathtt{s}, \mathtt{a})$. The probability that we do not terminate by time $\mathtt{t}$ is exactly $\gamma^{\mathtt{t}}$. Therefore, the expected return is $\sum_{t=1}^{\infty} \gamma^t \mathtt{r}(\mathtt{s}_t, \mathtt{a}_t)$ which is identical to the discounted return.

This class of problems provides a natural extension of the standard shortest-path problem to stochastic settings. Some conditions on the system dynamics and reward

function must be imposed for the problem to be well posed (e.g., that a goal state may be reached with probability one). Such problems are known as stochastic shortest path problems, or also episodic MDP planning problems. See more in Chapter 6.

4.3 Sufficiency of Markov Policies

In all the performance criteria defined above, the criterion is composed of sums of terms of the form $\mathbb{E}(\mathtt{r}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}))$. It follows that if two control policies induce the same marginal probability distributions $q_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$ over the state-action pairs $(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$ for all $\mathtt{t} \geq 0$, they will have the same performance for any linear return function.

Using this observation, the next claim implies that it is enough to consider the set of (stochastic) Markov policies in the above planning problems.

Proposition 4.1. *Let $\pi \in \Pi_{HS}$ be a general (history-dependent, stochastic) control policy. Let*

$$p_\mathtt{t}^{\pi,\mathtt{s}_0}(\mathtt{s}, \mathtt{a}) = P^{\pi,\mathtt{s}_0}(\mathtt{s}_\mathtt{t} = \mathtt{s}, \mathtt{a}_\mathtt{t} = \mathtt{a}), \qquad (\mathtt{s}, \mathtt{a}) \in \mathcal{S}_\mathtt{t} \times \mathcal{A}_\mathtt{t}$$

Denote the marginal distributions induced by $q_t(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$ on the state-action pairs $(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$, for all $\mathtt{t} \geq 0$. Then there exists a stochastic Markov policy $\tilde{\pi} \in \Pi_{MS}$ that induces the same marginal probabilities (for all initial states $\mathtt{s}_0$).

In Chapter 2 we showed for Deterministic Decision Process for the finite horizon that there is an optimal deterministic policy. The proof that every stochastic history dependent strategy has an equivalent stochastic Markovian policy (Theorem 2.1) showed how to generate the same state-action distribution, and applies to other setting as well. The proof that every stochastic Markovian policy has a deterministic Markovian policy (Theorem 2.2) depended on the finite horizon, but it is easy to extend it to any linear return function as well.

4.4 Finite-Horizon Dynamic Programming

Recall that we consider the expected total reward criterion, which we denote as

$$\mathcal{V}^\pi(\mathtt{s}_0) = \mathbb{E}^{\pi,\mathtt{s}_0}\left(\sum\nolimits_{\mathtt{t}=0}^{\mathtt{T}-1} \mathtt{r}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) + \mathtt{r}_\mathtt{T}(\mathtt{s}_\mathtt{T})\right) ,$$

where π is the control policy used, and $\mathtt{s}_0$ is a given initial state. We wish to maximize the expected return $\mathcal{V}^\pi(\mathtt{s}_0)$ over all control policies, and find an optimal policy π^*

that achieves the maximal expected return $\mathcal{V}^*(\mathtt{s}_0)$ for all initial states $\mathtt{s}_0$. Thus,

$$\mathcal{V}^*_{\mathtt{T}}(\mathtt{s}_0) \stackrel{\Delta}{=} \mathcal{V}^{\pi *}_{\mathtt{T}}(\mathtt{s}_0) = \max_{\pi \in \Pi_{HS}} \mathcal{V}^{\pi}_{\mathtt{T}}(\mathtt{s}_0)$$

4.4.1 The Principle of Optimality

The celebrated principle of optimality (stated by Bellman) applies to a large class of multi-stage optimization problems, and is at the heart of Dynamic Programming. As a general principle, it states that:

The tail of an optimal policy is optimal for the "tail" problem.

This principle is not an actual claim, but rather a guiding principle that can be applied in different ways to each problem. For example, considering our finite-horizon problem, let $\pi^* = (\pi_0, \ldots, \pi_{\mathtt{T}-1})$ denote an optimal Markov policy. Take any state $\mathtt{s}_\mathtt{t} = \mathtt{s}'$ which has a positive probability to be reached under π^*, namely $P^{\pi^*,\mathtt{s}_0}(\mathtt{s}_\mathtt{t} = \mathtt{s}') > 0$. Then the tail policy $\pi^*_{t:T} = (\pi_\mathtt{t}, \ldots, \pi_{\mathtt{T}-1})$ is optimal for the "tail" criterion $\mathcal{V}^{\pi}_{t:T}(\mathtt{s}') = \mathbb{E}^{\pi}\left(\sum_{k=t}^{\mathtt{T}} R_k | \mathtt{s}_\mathtt{t} = \mathtt{s}'\right)$.

Note that the reverse is not true. The prefix of the optimal policy is not optimal for the "prefix" problem. When we plan for a long horizon, we might start with non-greedy actions, so we can improve our return in later time steps. Specifically, the first action taken does not have to be the optimal action for horizon $\mathtt{T} = 1$, for which the greedy action is optimal.

4.4.2 Dynamic Programming for Policy Evaluation

As a "warmup", let us evaluate the reward of a given policy. Let $\pi = (\pi_0, \ldots, \pi_{\mathtt{T}-1})$ be a given Markov policy. Define the following reward-to-go function, or value function:

$$V^{\pi}_k(\mathtt{s}) = \mathbb{E}^{\pi}\left(\sum\nolimits_{\mathtt{t}=k}^{\mathtt{T}} R_\mathtt{t} | \mathtt{s}_k = \mathtt{s}\right)$$

Observe that $V^{\pi}_0(\mathtt{s}_0) = \mathcal{V}^{\pi}(\mathtt{s}_0)$.

Lemma 4.1 (**Value Iteration**). *$V^{\pi}_k(\mathtt{s})$ may be computed by the backward recursion:*

$$V^{\pi}_k(\mathtt{s}) = \left\{ \mathtt{r}_k(\mathtt{s},\mathtt{a}) + \sum\nolimits_{\mathtt{s}' \in \mathcal{S}_{k+1}} p_k(\mathtt{s}'|\mathtt{s},\mathtt{a})\, V^{\pi}_{k+1}(\mathtt{s}') \right\}_{a=\pi_k(\mathtt{s})}, \quad \mathtt{s} \in \mathcal{S}_k$$

for $k = \mathtt{T}-1, \ldots, 0$, starting with $V^{\pi}_{\mathtt{T}}(\mathtt{s}) = \mathtt{r}_\mathtt{T}(\mathtt{s})$.

Proof. Observe that:

$$\begin{aligned}
V_k^\pi(\mathtt{s}) &= \mathbb{E}^\pi\left(R_k + \sum\nolimits_{\mathtt{t}=k+1}^{\mathtt{T}} R_\mathtt{t} |\ \mathtt{s}_k = \mathtt{s}, \mathtt{a}_k = \pi_k(\mathtt{s})\right) \\
&= \mathbb{E}^\pi\left(\mathbb{E}^\pi\left(R_k + \sum\nolimits_{\mathtt{t}=k+1}^{\mathtt{T}} R_\mathtt{t} |\ \mathtt{s}_k = \mathtt{s}, \mathtt{a}_k = \pi_k(\mathtt{s}), \mathtt{s}_{k+1}\right) | \mathtt{s}_k = \mathtt{s}, \mathtt{a}_k = \pi_k(\mathtt{s})\right) \\
&= \mathbb{E}^\pi\left(\mathtt{r}_k(\mathtt{s}_k, \mathtt{a}_k) + V_{k+1}^\pi(\mathtt{s}_{k+1}) | \mathtt{s}_k = \mathtt{s}, \mathtt{a}_k = \pi_k(\mathtt{s})\right) \\
&= \mathtt{r}_k(s, \pi_k(\mathtt{s})) + \sum\nolimits_{\mathtt{s}' \in \mathcal{S}_{k+1}} p_k(\mathtt{s}'|s, \pi_k(\mathtt{s}))\ V_{k+1}^\pi(\mathtt{s}')
\end{aligned}$$

The first identity is simply writing the value function explicitly, starting at state $\mathtt{s}$ at time k and using action $\mathtt{a} = \pi_k(\mathtt{s})$. We split the sum to R_k, immediate reward, and the sum of other rewards. The second identity uses the law of total probability, we are conditioning on state $\mathtt{s}_{k+1}$, and taking the expectation over it. The third identity observes that the expected value of the sum is actually the value function at $\mathtt{s}_{k+1}$. The last identity writes the expectation over $\mathtt{s}_{k+1}$ explicitly. □

Remarks:

- Note that $\sum_{\mathtt{s}' \in \mathcal{S}_{k+1}} p_k(\mathtt{s}'|\mathtt{s}, \mathtt{a})\ V_{k+1}^\pi(\mathtt{s}') = \mathbb{E}^\pi(V_{k+1}^\pi(\mathtt{s}_{k+1})|\mathtt{s}_k = \mathtt{s}, \mathtt{a}_k = \mathtt{a})$.
- For the more general reward function $\tilde{\mathtt{r}}_\mathtt{t}(\mathtt{s}, \mathtt{a}, \mathtt{s}')$, the recursion takes the form

$$V_k^\pi(\mathtt{s}) = \left\{\sum\nolimits_{\mathtt{s}' \in \mathcal{S}_{k+1}} p_k(\mathtt{s}'|\mathtt{s}, \mathtt{a})[\tilde{\mathtt{r}}_k(\mathtt{s}, \mathtt{a}, \mathtt{s}') + V_{k+1}^\pi(\mathtt{s}')]\right\}_{\mathtt{a}=\pi_k(\mathtt{s})}.$$

  A similar observation applies to the Dynamic Programming equations in the next section.

4.4.3 Dynamic Programming for Policy Optimization

We next define the optimal value function at each time $k \geq 0$:

$$V_k^*(\mathtt{s}) = \max_{\pi^k} \mathbb{E}^{\pi^k}\left(\sum\nolimits_{\mathtt{t}=k}^{\mathtt{T}} R_\mathtt{t} | \mathtt{s}_k = \mathtt{s}\right), \quad \mathtt{s} \in \mathcal{S}_k\ ,$$

where the maximum is taken over "tail" policies $\pi^k = (\pi_k, \ldots, \pi_{\mathtt{T}-1})$ that start from time k. Note that π^k is allowed to be a general policy, i.e., history-dependent and stochastic. Obviously, $V_0(\mathtt{s}_0) = \mathcal{V}^*(\mathtt{s}_0)$.

Theorem 4.3 (**Finite-horizon Dynamic Programming**). *The following holds:*

1. *Backward recursion: Set* $V_{\mathtt{T}}(\mathtt{s}) = \mathtt{r}_{\mathtt{T}}(\mathtt{s})$ *for* $\mathtt{s} \in \mathcal{S}_{\mathtt{T}}$.
   For $k = \mathtt{T}-1, \ldots, 0$, $V_k(\mathtt{s})$ *compute using the following recursion:*

$$V_k(\mathtt{s}) = \max_{\mathtt{a}\in\mathcal{A}_k} \left\{ \mathtt{r}_k(\mathtt{s},\mathtt{a}) + \sum\nolimits_{\mathtt{s}'\in\mathcal{S}_{k+1}} p_k(\mathtt{s}'|\mathtt{s},\mathtt{a})\, V_{k+1}(\mathtt{s}') \right\}, \quad \mathtt{s} \in \mathcal{S}_k.$$

   We have that $V_k(s) = V_k^*(s)$.

2. *Optimal policy: Any Markov policy* π^* *that satisfies, for* $\mathtt{t} = 0, \ldots, \mathtt{T}-1$,

$$\pi_{\mathtt{t}}^*(\mathtt{s}) \in \arg\max_{\mathtt{a}\in\mathcal{A}_{\mathtt{t}}} \left\{ \mathtt{r}_{\mathtt{t}}(\mathtt{s},\mathtt{a}) + \sum\nolimits_{\mathtt{s}'\in\mathcal{S}_{\mathtt{t}+1}} p_{\mathtt{t}}(\mathtt{s}'|\mathtt{s},\mathtt{a})\, V_{\mathtt{t}+1}(\mathtt{s}') \right\}, \quad \mathtt{s} \in \mathcal{S}_{\mathtt{t}},$$

   is an optimal control policy. Furthermore, π^* *maximizes* $\mathcal{V}^{\pi}(\mathtt{s}_0)$ *simultaneously for every initial state* $\mathtt{s}_0 \in \mathcal{S}_0$.

Note that Theorem 4.3 specifies an optimal control policy which is a deterministic Markov policy.

Proof. **Part (i):**

We use induction to show that the stated backward recursion indeed yields the optimal value function $V_{\mathtt{t}}^*$. The idea is simple, but some care is needed with the notation since we consider general policies, and not just Markov policies.

For the base of the induction we start with $\mathtt{t} = \mathtt{T}$. The equality $V_{\mathtt{T}}(\mathtt{s}) = \mathtt{r}_{\mathtt{T}}(\mathtt{s})$ follows directly from the definition of $V_{\mathtt{T}}$. Clearly this is also the optimal value function $V_{\mathtt{T}}^*$.

We proceed by backward induction. Suppose that $V_{k+1}(\mathtt{s})$ is the optimal value function for time $k+1$, i.e., $V_{k+1}(\mathtt{s}) = V_{k+1}^*$. We need to show that $V_k(\mathtt{s}) = V_k^*(\mathtt{s})$ and we do it by showing that $V_k^*(\mathtt{s}) = W_k(\mathtt{s})$, where

$$W_k(\mathtt{s}) \stackrel{\Delta}{=} \max_{\mathtt{a}\in\mathcal{A}_k} \left\{ \mathtt{r}_k(\mathtt{s},\mathtt{a}) + \sum\nolimits_{\mathtt{s}'\in\mathcal{S}_{k+1}} p_k(\mathtt{s}'|\mathtt{s},\mathtt{a})\, V_{k+1}(\mathtt{s}') \right\}.$$

We will first establish that $V_k^*(\mathtt{s}) \geq W_k(\mathtt{s})$, and then that $V_k^*(\mathtt{s}) \leq W_k(\mathtt{s})$.

(a) We first show that $V_k^*(\mathtt{s}) \geq W_k(\mathtt{s})$. For that purpose, it is enough to find a policy π^k so that $V_k^{\pi^k}(\mathtt{s}) = W_k(\mathtt{s})$, since $V_k^*(s) \geq V_k^{\pi}(s)$ for any strategy π.

Fix $\mathtt{s} \in \mathcal{S}_k$, and define π^k as follows: Choose $\mathtt{a}_k = \bar{\mathtt{a}}$, where

$$\bar{\mathtt{a}} \in \arg\max_{\mathtt{a}\in\mathcal{A}_k} \left\{ \mathtt{r}_k(\mathtt{s},\mathtt{a}) + \sum\nolimits_{\mathtt{s}'\in\mathcal{S}_{k+1}} p_k(\mathtt{s}'|\mathtt{s},\mathtt{a})\, V_{k+1}(\mathtt{s}') \right\},$$

and then, after observing $\mathtt{s}_{k+1} = \mathtt{s}'$, proceed with the optimal tail policy $\pi^{k+1}(\mathtt{s}')$ that obtains $V_{k+1}^{\pi^{k+1}(\mathtt{s}')}(\mathtt{s}') = V_{k+1}(\mathtt{s}')$. Proceeding similarly to the proof of Lemma 4.1 (value iteration for a fixed policy), we obtain:

$$V_k^{\pi^k}(\mathtt{s}) = \mathtt{r}_k(\mathtt{s}, \bar{\mathtt{a}}) + \sum\nolimits_{\mathtt{s}' \in \mathcal{S}_{k+1}} p(\mathtt{s}'|\mathtt{s}, \bar{\mathtt{a}})\, V_{k+1}^{\pi^{k+1}(\mathtt{s}')}(\mathtt{s}') \tag{4.1}$$

$$= \mathtt{r}_k(\mathtt{s}, \bar{\mathtt{a}}) + \sum\nolimits_{\mathtt{s}' \in \mathcal{S}_{k+1}} p(\mathtt{s}'|\mathtt{s}, \bar{\mathtt{a}})\, V_{k+1}(\mathtt{s}') = W_k(\mathtt{s}), \tag{4.2}$$

as was required.

(b) To establish $V_k^*(\mathtt{s}) \leq W_k(\mathtt{s})$, it is enough to show that $V_k^{\pi^k}(\mathtt{s}) \leq W_k(\mathtt{s})$ for any (general, randomized) "tail" policy π^k.

Fix $\mathtt{s} \in \mathcal{S}_k$. Consider then some tail policy $\pi^k = (\pi_k, \dots \pi_{\mathtt{T}-1})$. Note that this means that $\mathtt{a}_\mathtt{t} \sim \pi_\mathtt{t}(\mathtt{a}|\mathtt{h}_{k:t})$, where $\mathtt{h}_{k:t} = (\mathtt{s}_k, \mathtt{a}_k, \mathtt{s}_{k+1}, \mathtt{a}_{k+1}, \dots, \mathtt{s}_\mathtt{t})$. For each state-action pair $\mathtt{s} \in \mathcal{S}_k$ and $\mathtt{a} \in \mathcal{A}_k$, let $(\pi^k|\mathtt{s}, \mathtt{a})$ denote the tail policy π^{k+1} from time $k+1$ onwards which is obtained from π^k given that $\mathtt{s}_k = \mathtt{s},\ \mathtt{a}_k = \mathtt{a}$. As before, by value iteration for a fixed policy,

$$V_k^{\pi^k}(\mathtt{s}) = \sum\nolimits_{\mathtt{a} \in \mathcal{A}_k} \pi_k(\mathtt{a}|\mathtt{s}) \left\{ \mathtt{r}_k(\mathtt{s}, \mathtt{a}) + \sum\nolimits_{\mathtt{s}' \in \mathcal{S}_{k+1}} p_k(\mathtt{s}'|\mathtt{s}, \mathtt{a})\, V_{k+1}^{(\pi^k|\mathtt{s},\mathtt{a})}(\mathtt{s}') \right\}.$$

But since V_{k+1} is optimal,

$$\begin{aligned} V_k^{\pi^k}(\mathtt{s}) &\leq \sum\nolimits_{\mathtt{a} \in \mathcal{A}_k} \pi_k(\mathtt{a}|\mathtt{s}) \left\{ \mathtt{r}_k(\mathtt{s}, \mathtt{a}) + \sum\nolimits_{\mathtt{s}' \in \mathcal{S}_{k+1}} p_k(\mathtt{s}'|\mathtt{s}, \mathtt{a})\, V_{k+1}(\mathtt{s}') \right\} \\ &\leq \max_{\mathtt{a} \in \mathcal{A}_k} \left\{ \mathtt{r}_k(\mathtt{s}, \mathtt{a}) + \sum\nolimits_{\mathtt{s}' \in S_{k+1}} p_k(\mathtt{s}'|\mathtt{s}, \mathtt{a})\, V_{k+1}(\mathtt{s}') \right\} = W_k(\mathtt{s}), \end{aligned}$$

which is the required inequality in (b).

Part (ii) The main point is to show that the it is sufficient that the optimal policy would be Markov (rather than history dependent) and deterministic (rather than stochastic).

We will only sketch the proof. Let π^* be the (Markov) policy defined in part 2 of Theorem 4.3. Our goal is to show that the value function of π^* coincides with that of the optima policy, which we showed is equal to V_k that we computed. Once we show that, we prove that π^* is optimal.

Consider the value iteration (Lemma 4.1). The updates for $V_k^{\pi^*}$ in the value iteration, given the action selection of π^*, are identical to those of V_k. This implies that $V_k^{\pi^*} = V_k$ (formally, by induction of k). Since V_k is the optimal value function, it implies that π^* is the optimal policy. $\square$

4.4.4 The Q function

Let

$$Q_k^*(\mathtt{s},\mathtt{a}) \stackrel{\Delta}{=} \mathtt{r}_k(\mathtt{s},\mathtt{a}) + \sum\nolimits_{\mathtt{s}' \in S_k} p_k(\mathtt{s}'|\mathtt{s},\mathtt{a})\, V_k^*(\mathtt{s}').$$

This is known as the optimal state-action value function, or simply as the *Q-function*. $Q_k^*(\mathtt{s},\mathtt{a})$ is the expected return from stage k onward, if we choose $\mathtt{a}_k = \mathtt{a}$ and then proceed optimally.

Theorem 4.3 can now be succinctly expressed as

$$V_k^*(\mathtt{s}) = \max_{\mathtt{a} \in \mathcal{A}_k} Q_k^*(\mathtt{s},\mathtt{a}),$$

and

$$\pi_k^*(\mathtt{s}) \in \arg\max_{\mathtt{a} \in \mathcal{A}_k} Q_k^*(\mathtt{s},\mathtt{a}).$$

The Q function provides the basis for the Q-learning algorithm, which is one of the basic Reinforcement Learning algorithms, and would be discussed in Chapter 9.

4.5 Linear Program

In this section we will extend the linear programming given in Section 2.5 from Deterministic Decision Processes to Markov Decision Processes. The derivation is almost identical and the difference is that replacing the deterministic next state $f_{\mathtt{t}}(\mathtt{s},\mathtt{a})$ by an expectation over $\mathtt{s}'$, such that each $\mathtt{s}'$ has a probability $p_{\mathtt{t}}(\mathtt{s}'|\mathtt{s},\mathtt{a})$. For completeness we do it in a detailed way.

We will see that both the primal and dual program will play an important part in defining the optimal policy. We will fix an initial state $\mathtt{s}_0$ and compute the optimal policy for it.

We will start with the primal linear program, which will compute the optimal policy. For each time $\mathtt{t}$, state $\mathtt{s}$ and action $\mathtt{a}$ we will have a variable $x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) \in [0,1]$ that will indicate by probability that at time $\mathtt{t}$ we are at state $\mathtt{s}$ and perform action $\mathtt{a}$. For the terminal states $\mathtt{s}$ we will have a variable $x_{\mathtt{T}}(\mathtt{s}) \in [0,1]$ that will indicate whether the probability that we terminate at state $\mathtt{s}$.

Our main constraint will be a flow constraint, stating that the probability mass that leaves a state $\mathtt{s}$ at time $\mathtt{t}$ is bounded by the probability mass of reaching state $\mathtt{s}$ at time $\mathtt{t}-1$. Formally,

$$\sum_{\mathtt{a}} x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) \leq \sum_{\mathtt{s}',\mathtt{a}'} x_{\mathtt{t}-1}(\mathtt{s}',\mathtt{a}') p_{\mathtt{t}-1}(\mathtt{s}|\mathtt{s}'\mathtt{a}').$$

and for terminal states simply

$$x_{\mathtt{T}}(\mathtt{s}) \leq \sum_{\mathtt{s}',\mathtt{a}'} x_{\mathtt{T}-1}(\mathtt{s}',\mathtt{a}')p_{\mathtt{T}-1}(\mathtt{s}|\mathtt{s}',\mathtt{a}')$$

The return, which we would like to maximize, would be

$$\sum_{\mathtt{t},\mathtt{s},\mathtt{a}} \mathtt{r}_{\mathtt{t}}(\mathtt{s},\mathtt{a})x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) + \sum_{\mathtt{s}} \mathtt{r}_{\mathtt{T}}(\mathtt{s})x_{\mathtt{T}}(\mathtt{s})$$

The resulting linear program is the following.

$$\max_{x_{\mathtt{t}}(\mathtt{s},\mathtt{a}),x_{\mathtt{T}}(\mathtt{s})} \quad \sum_{\mathtt{t},\mathtt{s},\mathtt{a}} \mathtt{r}_{\mathtt{t}}(\mathtt{s},\mathtt{a})x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) + \sum_{\mathtt{s}} \mathtt{r}_{\mathtt{T}}(\mathtt{s})x_{\mathtt{T}}(\mathtt{s})$$

such that

$$\sum_{\mathtt{a}} x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) \leq \sum_{\mathtt{s}',\mathtt{a}'} x_{\mathtt{t}-1}(\mathtt{s}',\mathtt{a}')p_{\mathtt{t}-1}(\mathtt{s}|\mathtt{s}'\mathtt{a}'). \qquad \forall \mathtt{s} \in \mathcal{S}_{\mathtt{t}}, \mathtt{t} \in \mathbb{T}$$

$$x_{\mathtt{T}}(\mathtt{s}) \leq \sum_{\mathtt{s}',\mathtt{a}'} x_{\mathtt{T}-1}(\mathtt{s}',\mathtt{a}')p_{\mathtt{T}-1}(\mathtt{s}|\mathtt{s}'\mathtt{a}') \qquad \forall \mathtt{s} \in \mathcal{S}_{\mathtt{T}}$$

$$x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) \geq 0 \qquad \forall \mathtt{s} \in \mathcal{S}_{\mathtt{t}}, \mathtt{a} \in \mathcal{A}, \mathtt{t} \in \{0,\ldots,\mathtt{T}-1\}$$

$$\sum_{\mathtt{a}} x_0(\mathtt{s}_0,\mathtt{a}) = 1$$

$$x_0(\mathtt{s},\mathtt{a}) = 0, \qquad \forall \mathtt{s} \in \mathcal{S}_0, \mathtt{s} \neq \mathtt{s}_0$$

At first sight it is surprising that we do not impose that $x_{\mathtt{t}}(\mathtt{s},\mathtt{a}) \leq 1$, however this is implicit in the program. Let $\Phi(\mathtt{t}) = \sum_{\mathtt{s},\mathtt{a}} x_{\mathtt{t}}(\mathtt{s},\mathtt{a})$. From the initial conditions we have that $\Phi(0) = 1$. When we sum over the state the flow condition (first inequality) we have that $\Phi(\mathtt{t}) \leq \Phi(\mathtt{t}-1)$. This implies that $\Phi(\mathtt{t}) \leq 1$.

Given the primal linear program we can derive the dual linear program.

$$\min_{z_{\mathtt{t}}(\mathtt{s})} \; z_0(\mathtt{s}_0)$$

such that

$$z_{\mathtt{T}}(\mathtt{s}) = \mathtt{r}_{\mathtt{T}}(\mathtt{s}) \qquad \forall \mathtt{s} \in \mathcal{S}_{\mathtt{t}}$$

$$z_{\mathtt{t}}(\mathtt{s}) \geq \mathtt{r}_{\mathtt{t}}(\mathtt{s},\mathtt{a}) + \sum_{\mathtt{s}'} z_{\mathtt{t}+1}(\mathtt{s}')p_{\mathtt{t}}(\mathtt{s}'|\mathtt{s},\mathtt{a}), \qquad \forall \mathtt{s} \in \mathcal{S}_{\mathtt{t}}, \mathtt{a} \in \mathcal{A}, \mathtt{t} \in \mathbb{T},$$

.

One can identify the dual random variables $z_{\mathtt{t}}(\mathtt{s})$ with the optimal vale function $\mathcal{V}_{\mathtt{t}}(\mathtt{s})$. At the optimal solution of the dual linear program one can show that we have

$$z_{\mathtt{t}}(\mathtt{s}) = \max_{\mathtt{a}} \left\{ \mathtt{r}_{\mathtt{t}}(\mathtt{s},\mathtt{a}) + \sum_{\mathtt{s}'} z_{\mathtt{t}+1}(\mathtt{s}') p_{\mathtt{t}}(\mathtt{s}'|\mathtt{s},\mathtt{a}) \right\}, \qquad \forall \mathtt{s} \in \mathcal{S}_{\mathtt{t}}, \mathtt{t} \in \mathbb{T},$$

which are the familiar Bellman optimality equations.

4.6 Summary

- The optimal value function can be computed by backward recursion. This recursive equation is known as the *dynamic programming equation*, *optimality equation*, or *Bellman's Equation*.
- Computation of the value function in this way is known as the *finite-horizon value iteration* algorithm.
- The value function is computed for all states at each stage.
- An optimal policy is easily derived from the optimal value.
- The optimization in each stage is performed in the action space. The total number of minimization operations needed is $\mathtt{T} \times |\mathcal{S}|$ - each over $|\mathcal{A}|$ choices. This replaces "brute force" optimization in policy space, with tremendous computational savings as the number of Markov policies is $|\mathcal{A}|^{\mathtt{T}\times|\mathcal{S}|}$.

Chapter 5

MDPs with Discounted Infinite Return

This chapter covers the basic theory and main solution methods for stationary MDPs over an infinite horizon, with the discounted return criterion. In this case, we will show that stationary policies are optimal.

The discounted return problem is the most "well behaved" among all infinite horizon problems (such as average return and stochastic shortest path), and the theory of it is relatively simple, both in the planning and the learning contexts. For that reason, as well as its usefulness, we will consider here the discounted problem and its solution in some detail.

5.1 Problem Statement

We consider a stationary (time-invariant) MDP, with a finite state space $\mathcal{S}$, finite action set $\mathcal{A}$, and transition kernel $P = (P(\mathtt{s}'|\mathtt{s},\mathtt{a}))$ over the infinite time horizon $\mathbb{T} = \{0, 1, 2, \ldots\}$.

Our goal is to maximize the expected discounted return, which is defined for each control policy π and initial state $\mathtt{s}_0 = s$ as follows:

$$\begin{aligned}\mathcal{V}_\gamma^\pi(\mathtt{s}) &= \mathbb{E}^\pi(\sum_{\mathtt{t}=0}^{\infty} \gamma^{\mathtt{t}} \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) | \mathtt{s}_0 = \mathtt{s}) \\ &\equiv \mathbb{E}^{\pi,\mathtt{s}}(\sum_{\mathtt{t}=0}^{\infty} \gamma^{\mathtt{t}} \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}))\end{aligned}$$

where $\mathbb{E}^{\pi,\mathtt{s}}$ uses the distribution induced by policy π starting at state $\mathtt{s}$. Here,

- $\mathtt{r} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the (running, or instantaneous) expected reward function, i.e., $\mathtt{r}(\mathtt{s}, \mathtt{a}) = \mathbb{E}[R|\mathtt{s}, \mathtt{a}]$.
- $\gamma \in (0, 1)$ is the discount factor.

We observe that $\gamma < 1$ ensures convergence of the infinite sum (since the rewards $\mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$ are uniformly bounded). With $\gamma = 1$ we obtain the total return criterion, which is harder to handle due to possible divergence of the sum.

Let $\mathcal{V}^*_\gamma(\mathtt{s})$ denote the maximal expected value of the discounted return, over all (possibly history dependent and randomized) control policies, i.e.,

$$\mathcal{V}^*_\gamma(\mathtt{s}) = \sup_{\pi \in \Pi_{HS}} \mathcal{V}^\pi_\gamma(\mathtt{s}).$$

Our goal is to find an optimal control policy π^* that attains that maximum (for all initial states), and compute the numeric value of the optimal return $\mathcal{V}^*_\gamma(\mathtt{s})$. As we shall see, for this problem there always exists an optimal policy which is a (deterministic) stationary policy.

Note: As usual, the discounted performance criterion can be defined in terms of cost:

$$\mathcal{C}^\pi_\gamma(\mathtt{s}) = \mathbb{E}^{\pi,\mathtt{s}}(\sum_{\mathtt{t}=0}^{\infty} \gamma^\mathtt{t} \mathtt{c}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})) \ ,$$

where $\mathtt{c}(\mathtt{s}, \mathtt{a})$ is the running cost function. Our goal is then to minimize the discounted cost $\mathcal{C}^\pi_\gamma(\mathtt{s})$.

5.2 The Fixed-Policy Value Function

We start the analysis by defining and computing the value function for a fixed stationary policy. This intermediate step is required for later analysis of our optimization problem, and also serves as a gentle introduction to the value iteration approach.

For a stationary policy $\pi : \mathcal{S} \to \mathcal{A}$, we define the value function $\mathcal{V}^\pi(\mathtt{s}),\ \mathtt{s} \in \mathcal{S}$ simply as the corresponding discounted return:

$$\mathcal{V}^\pi(\mathtt{s}) \stackrel{\Delta}{=} \mathbb{E}^{\pi,\mathtt{s}} \left(\sum_{\mathtt{t}=0}^{\infty} \gamma^\mathtt{t} \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) \right) = \mathcal{V}^\pi_\gamma(\mathtt{s}), \quad \mathtt{s} \in S$$

Lemma 5.1. *For $\pi \in \Pi_{SD}$, the value function $\mathcal{V}^\pi$ satisfies the following set of $|\mathcal{S}|$ linear equations:*

$$\mathcal{V}^\pi(\mathtt{s}) = \mathtt{r}(\mathtt{s}, \pi(\mathtt{s})) + \gamma \sum\nolimits_{\mathtt{s}' \in \mathcal{S}} p(\mathtt{s}'|\mathtt{s}, \pi(\mathtt{s})) \mathcal{V}^\pi(\mathtt{s}') \ , \quad \mathtt{s} \in \mathcal{S}. \tag{5.1}$$

Proof. We first note that

$$\begin{aligned}\mathcal{V}^{\pi}(\mathtt{s}) &\triangleq \mathbb{E}^{\pi}(\sum_{\mathtt{t}=0}^{\infty}\gamma^{\mathtt{t}}\mathtt{r}(\mathtt{s}_{\mathtt{t}},\mathtt{a}_{\mathtt{t}})|\mathtt{s}_0=\mathtt{s})\\ &= \mathbb{E}^{\pi}(\sum_{\mathtt{t}=1}^{\infty}\gamma^{t-1}\mathtt{r}(\mathtt{s}_{\mathtt{t}},\mathtt{a}_{\mathtt{t}})|\mathtt{s}_1=\mathtt{s}),\end{aligned}$$

since both the model and the policy are stationary. Now,

$$\begin{aligned}\mathcal{V}^{\pi}(\mathtt{s}) &= \mathtt{r}(\mathtt{s},\pi(\mathtt{s})) + \mathbb{E}^{\pi}(\sum_{\mathtt{t}=1}^{\infty}\gamma^{\mathtt{t}}\mathtt{r}(\mathtt{s}_{\mathtt{t}},\pi(\mathtt{s}_{\mathtt{t}}))|\mathtt{s}_0=\mathtt{s})\\ &= \mathtt{r}(\mathtt{s},\pi(\mathtt{s})) + \mathbb{E}^{\pi}\left[\mathbb{E}^{\pi}\left(\sum_{\mathtt{t}=1}^{\infty}\gamma^{\mathtt{t}}\mathtt{r}(\mathtt{s}_{\mathtt{t}},\pi(\mathtt{s}_{\mathtt{t}}))|\mathtt{s}_0=\mathtt{s},\mathtt{s}_1=\mathtt{s}'\right)\middle|\,\mathtt{s}_0=s\right]\\ &= \mathtt{r}(\mathtt{s},\pi(\mathtt{s})) + \sum_{\mathtt{s}'\in\mathcal{S}}p(\mathtt{s}'|\mathtt{s},\pi(\mathtt{s}))\mathbb{E}^{\pi}(\sum_{\mathtt{t}=1}^{\infty}\gamma^{\mathtt{t}}\mathtt{r}(\mathtt{s}_{\mathtt{t}},\pi(\mathtt{s}_{\mathtt{t}}))|\mathtt{s}_1=\mathtt{s}')\\ &= \mathtt{r}(\mathtt{s},\pi(\mathtt{s})) + \gamma\sum_{\mathtt{s}'\in\mathcal{S}}p(\mathtt{s}'|\mathtt{s},\pi(\mathtt{s}))\mathbb{E}^{\pi}(\sum_{\mathtt{t}=1}^{\infty}\gamma^{t-1}\mathtt{r}(\mathtt{s}_{\mathtt{t}},\mathtt{a}_{\mathtt{t}})|\mathtt{s}_1=\mathtt{s}')\\ &= \mathtt{r}(\mathtt{s},\pi(\mathtt{s})) + \gamma\sum_{\mathtt{s}'\in\mathcal{S}}p(\mathtt{s}'|\mathtt{s},\pi(\mathtt{s}))\mathcal{V}^{\pi}(\mathtt{s}').\end{aligned}$$

The first equality is by the definition of the value function. The second equality follows from the law of total expectation, conditioning $\mathtt{s}_1 = \mathtt{s}'$ and taking the expectation over it. By definition $\mathtt{a}_{\mathtt{t}} = \pi(\mathtt{s}_{\mathtt{t}})$. The third equality follows similarly to the finite-horizon case (Lemma 4.1, in Chapter 2.1). The fourth is simple algebra, taking a one multiple of the discount factor γ outside. The last by the observation in the beginning of the proof. □

We can write the linear equations in (5.1) in vector form as follows. Define the column vector $r^{\pi} = (r^{\pi}(\mathtt{s}))_{\mathtt{s}\in\mathcal{S}}$ with components $r^{\pi}(\mathtt{s}) = \mathtt{r}(\mathtt{s},\pi(\mathtt{s}))$, and the transition matrix P^{π} with components $P^{\pi}(\mathtt{s}'|\mathtt{s}) = p(\mathtt{s}'|\mathtt{s},\pi(\mathtt{s}))$. Finally, let $\mathcal{V}^{\pi}$ denote a column vector with components $\mathcal{V}^{\pi}(\mathtt{s})$. Then (5.1) is equivalent to the linear equation set

$$\mathcal{V}^{\pi} = r^{\pi} + \gamma P^{\pi}\mathcal{V}^{\pi} \tag{5.2}$$

Lemma 5.2. *The set of linear equations* (5.1) *or* (5.2)*, with* $\mathcal{V}^{\pi}$ *as variables, has a unique solution* $\mathcal{V}^{\pi}$*, which is given by*

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

Proof. We only need to show that the square matrix $I - \gamma P^\pi$ is non-singular. Let (λ_i) denote the eigenvalues of the matrix P^π. Since P^π is a stochastic matrix (row sums are 1), then $|\lambda_i| \leq 1$. Now, the eignevalues of $I - \gamma P^\pi$ are $(1 - \gamma\lambda_i)$, and satisfy $|1 - \gamma\lambda_i| \geq 1 - \gamma > 0$. □

Combining Lemma 5.1 and Lemma 5.2, we obtain

Proposition 5.1. *Let $\pi \in \Pi_{SD}$. The value function $\mathcal{V}^\pi = [\mathcal{V}^\pi(\mathtt{s})]$ is the unique solution of equation (5.2), given by*

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

Proposition 5.1 provides a closed-form formula for computing $\mathcal{V}^\pi$. However, for large systems, computing the inverse $(I - \gamma P^\pi)^{-1}$ may be computationally expensive. In that case, the following value iteration algorithm provides an alternative, iterative method for computing $\mathcal{V}^\pi$.

Algorithm 5.1. ***Fixed-policy value iteration***

1. *Let $\mathcal{V}_0 = (\mathcal{V}_0(\mathtt{s}))_{\mathtt{s}\in\mathcal{S}}$ be arbitrary.*
2. *For $n = 0, 1, 2, \ldots$, set*

$$\mathcal{V}_{n+1}(\mathtt{s}) = \mathtt{r}(\mathtt{s}, \pi(\mathtt{s})) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s}, \pi(\mathtt{s}))\mathcal{V}_n(\mathtt{s}') \,, \qquad \mathtt{s} \in \mathcal{S}$$

or, equivalently,

$$\mathcal{V}_{n+1} = r^\pi + \gamma P^\pi \mathcal{V}_n.$$

Proposition 5.2 (**Convergence of fixed-policy value iteration**). *We have $\mathcal{V}_n \to \mathcal{V}^\pi$ component-wise, that is,*

$$\lim_{n\to\infty} \mathcal{V}_n(\mathtt{s}) = \mathcal{V}^\pi(\mathtt{s}), \qquad \mathtt{s} \in \mathcal{S}.$$

Proof. Note first that

$$\begin{aligned}\mathcal{V}_1(\mathtt{s}) &= \mathtt{r}(\mathtt{s}, \pi(\mathtt{s})) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s}, \pi(\mathtt{s}))\mathcal{V}_0(\mathtt{s}') \\ &= \mathbb{E}^\pi(\mathtt{r}(\mathtt{s}_0, a_0) + \gamma\mathcal{V}_0(\mathtt{s}_1)|\mathtt{s}_0 = \mathtt{s}).\end{aligned}$$

Continuing similarly, we obtain that

$$\mathcal{V}_n(\mathtt{s}) = \mathbb{E}^\pi(\sum_{\mathtt{t}=0}^{n-1} \gamma^{\mathtt{t}} \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) + \gamma^n \mathcal{V}_0(s_n)|\mathtt{s}_0 = \mathtt{s}).$$

Note that $\mathcal{V}_n(\mathtt{s})$ is the n-stage discounted return, with terminal reward $\mathtt{r}_n(\mathtt{s}_n) = \mathcal{V}_0(\mathtt{s}_n)$. Comparing with the definition of $\mathcal{V}^\pi$, we can see that

$$\mathcal{V}^\pi(\mathtt{s}) - \mathcal{V}_n(\mathtt{s}) = \mathbb{E}^\pi(\sum_{\mathtt{t}=n}^{\infty} \gamma^{\mathtt{t}} \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) - \gamma^n \mathcal{V}_0(\mathtt{s}_n) | \mathtt{s}_0 = \mathtt{s}).$$

Denoting $\mathtt{R}_{\max} = \max_{\mathtt{s},\mathtt{a}} |\mathtt{r}(\mathtt{s},\mathtt{a})|$, $\bar{\mathcal{V}}_0 = \max_s |\mathcal{V}_0(\mathtt{s})|$ we obtain

$$|\mathcal{V}^\pi(\mathtt{s}) - \mathcal{V}_n(\mathtt{s})| \leq \gamma^n (\frac{\mathtt{R}_{\max}}{1-\gamma} + \bar{\mathcal{V}}_0)$$

which converges to 0 since $\gamma < 1$. □

Comments:

- The proof provides an explicit bound on $|\mathcal{V}^\pi(\mathtt{s}) - \mathcal{V}_n(\mathtt{s})|$. It may be seen that the convergence is exponential, with rate $O(\gamma^n)$.
- Using vector notation, it may be seen that

$$\mathcal{V}_n = r^\pi + P^\pi r^\pi + \ldots + (P^\pi)^{n-1} r^\pi + (P^\pi)^n \mathcal{V}_0 = \sum_{\mathtt{t}=0}^{n-1} (P^\pi)^{\mathtt{t}} r^\pi + (P^\pi)^n \mathcal{V}_0.$$

  Similarly, $\mathcal{V}^\pi = \sum_{\mathtt{t}=0}^{\infty} (P^\pi)^{\mathtt{t}} r^\pi$.

In summary:

- Proposition 5.1 allows to compute $\mathcal{V}^\pi$ by solving a set of $|\mathcal{S}|$ linear equations.
- Proposition 5.2 computes $\mathcal{V}^\pi$ by an infinite recursion, that converges exponentially fast.

5.3 Overview: The Main DP Algorithms

We now return to the optimal planning problem defined in Section 5.1. Recall that $\mathcal{V}^*_\gamma(\mathtt{s}) = \sup_{\pi \in \Pi_{HS}} \mathcal{V}^\pi_\gamma(\mathtt{s})$ is the optimal discounted return. We further denote

$$\mathcal{V}^*(\mathtt{s}) \stackrel{\Delta}{=} \mathcal{V}^*_\gamma(\mathtt{s}), \quad \mathtt{s} \in \mathcal{S},$$

and refer to $\mathcal{V}^*$ as the optimal value function. Depending on the context, we consider $\mathcal{V}^*$ either as a function $\mathcal{V}^* : \mathcal{S} \to \mathbb{R}$, or as a column vector $\mathcal{V}^* = [\mathcal{V}(\mathtt{s})]_{\mathtt{s}\in\mathcal{S}}$.

The following optimality equation provides an explicit characterization of the value function, and shows that an optimal stationary policy can easily be computed if the value function is known. (See the proof in Section 5.5.)

Theorem 5.1 (**Bellman's Optimality Equation**). *The following statements hold:*

1. *$\mathcal{V}^*$ is the unique solution of the following set of (nonlinear) equations:*

$$\mathcal{V}(\mathtt{s}) = \max_{\mathtt{a}\in\mathcal{A}} \left\{ \mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\mathtt{a})\mathcal{V}(\mathtt{s}') \right\}, \quad \mathtt{s} \in \mathcal{S}. \tag{5.3}$$

2. *Any stationary policy π^* that satisfies*

$$\pi^*(\mathtt{s}) \in \arg\max_{\mathtt{a}\in\mathcal{A}} \left\{ \mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\mathtt{a})\mathcal{V}(\mathtt{s}') \right\},$$

is an optimal policy (for any initial state $\mathtt{s}_0 \in \mathcal{S}$).

The optimality equation (5.3) is non-linear, and generally requires iterative algorithms for its solution. The main iterative algorithms are **value iteration** and **policy iteration**. In the following we provide the algorithms and the basic claims. Later in this chapter we formally prove the results regarding value iteration (Section 5.6) and policy iteration (Section 5.7).

Algorithm 5.2. ***Value Iteration (VI)***

1. *Let $\mathcal{V}_0 = (\mathcal{V}_0(\mathtt{s}))_{\mathtt{s}\in\mathcal{S}}$ be arbitrary.*
2. *For $n = 0, 1, 2, \ldots$, set*

$$\mathcal{V}_{n+1}(\mathtt{s}) = \max_{\mathtt{a}\in\mathcal{A}} \left\{ \mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\mathtt{a})\mathcal{V}_n(\mathtt{s}') \right\}, \quad \forall \mathtt{s} \in \mathcal{S}$$

Theorem 5.2 (**Convergence of value iteration**). *We have $\lim_{n\to\infty}\mathcal{V}_n = \mathcal{V}^*$ (component-wise). The rate of convergence is exponential, at rate $O(\gamma^n)$.*

Proof. Using our previous results on value iteration for the finite-horizon problem, namely the proof of Proposition 5.2, it follows that

$$\mathcal{V}_n(\mathtt{s}) = \max_{\pi} \mathbb{E}^{\pi,\mathtt{s}}\Big(\sum_{\mathtt{t}=0}^{n-1} \gamma^{\mathtt{t}} R_t + \gamma^n \mathcal{V}_0(\mathtt{s}_n)\Big).$$

Comparing to the optimal value function

$$\mathcal{V}^*(\mathtt{s}) = \max_{\pi} \mathbb{E}^{\pi,\mathtt{s}}(\sum_{\mathtt{t}=0}^{\infty} \gamma^{\mathtt{t}} R_t),$$

it may be seen that that

$$|\mathcal{V}_n(\mathtt{s}) - \mathcal{V}^*(\mathtt{s})| \leq \gamma^n (\frac{\mathtt{R}_{\max}}{1-\gamma} + ||\mathcal{V}_0||_\infty).$$

As $\gamma < 1$, this implies that $\mathcal{V}_n$ converges to $\mathcal{V}^*_\gamma$ exponentially fast. □

The value iteration algorithm iterates over the value functions, with asymptotic convergence. The policy iteration algorithm iterates over stationary policies, with each new policy better than the previous one. This algorithm converges to the optimal policy in a finite number of steps.

Algorithm 5.3. ***Policy Iteration (PI)***

1. *Initialization: choose some stationary policy* π_0.
2. *For* $k = 0, 1, \ldots$:
   (a) *Policy evaluation: compute* $\mathcal{V}^{\pi_k}$.
   (For example, use the explicit formula $\mathcal{V}^{\pi_k} = (I - \gamma P^{\pi_k})^{-1} r^{\pi_k}$.*)*
   (b) *Policy Improvement: Compute* π_{k+1}*, a greedy policy with respect to* $\mathcal{V}^{\pi_k}$*:*

$$\pi_{k+1}(\mathtt{s}) \in \arg\max_{\mathtt{a}\in\mathcal{A}} \left\{ \mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\mathtt{a}) \mathcal{V}^{\pi_k}(\mathtt{s}') \right\}, \quad \forall \mathtt{s} \in \mathcal{S}.$$

   (c) *Stop if* $\pi_{k+1} = \pi_k$ *(or if* $\mathcal{V}^{\pi_k}$ *satisfied the optimality equation), else continue.*

Theorem 5.3 (**Convergence of policy iteration**)**.** *The following statements hold:*

1. *Each policy* π_{k+1} *is improving over the previous one* π_k*, in the sense that* $\mathcal{V}^{\pi_{k+1}} \geq \mathcal{V}^{\pi_k}$ *(component-wise).*
2. $\mathcal{V}^{\pi_{k+1}} = \mathcal{V}^{\pi_k}$ *if and only if* π_k *is an optimal policy.*
3. *Consequently, since the number of stationary policies is finite,* π_k *converges to the optimal policy after a finite number of steps.*

An additional solution method for DP planning relies on a Linear Programming formulation of the problem. We will provide additional details later in this Chapter, in Section A.

5.4 Contraction Operators

The basic proof methods of the DP results mentioned above rely on the concept of a *contraction operator*. We provide here the relevant mathematical background, and illustrate the contraction properties of some basic Dynamic Programming operators.

5.4.1 The contraction property

Recall that a norm $||\cdot||$ over $\mathbb{R}^n$ is a real-valued function $\|\cdot\| : \mathbb{R}^d \to \mathbb{R}$ such that, for any pair of vectors $x, y \in \mathbb{R}^d$ and scalar $a \in \mathbb{R}$,

1. $||ax|| = |a| \cdot ||x||$,
2. $||x + y|| \leq ||x|| + ||y||$,
3. $||x|| = 0$ only if $x = 0$.

Common examples are the p-norm $||x||_p = (\sum_{i=1}^{d} |x_i|^p)^{1/p}$ for $p \geq 1$, and in particular the Euclidean norm ($p = 2$). Here we will mostly use the max-norm:

$$||x||_\infty = \max_{1 \leq i \leq d} |x_i|.$$

Let $T : \mathbb{R}^d \to \mathbb{R}^d$ be a vector-valued function over $\mathbb{R}^d$ ($d \geq 1$). We equip $\mathbb{R}^d$ with some norm $||\cdot||$, and refer to T as an *operator* over $\mathbb{R}^d$. Thus, $T(v) \in \mathbb{R}^d$ for any $v \in \mathbb{R}^d$. We also denote $T^n(v) = T(T^{n-1}(v))$ for $n \geq 2$. For example, $T^2(v) = T(T(v))$.

Definition 5.1. *The operator T is called a contraction operator if there exists $\beta \in (0, 1)$ (the contraction coefficient) such that*

$$||T(v_1) - T(v_2)|| \leq \beta ||v_1 - v_2||,$$

for all $v_1, v_2 \in \mathbb{R}^d$

5.4.2 The Banach Fixed Point Theorem

The following celebrated result applies to contraction operators. While we quote the result for $\mathbb{R}^d$, we note that it applies in much greater generality to any Banach space (a complete normed space), or even to any complete metric space, with essentially the same proof.

Theorem 5.4 (**Banach's fixed point theorem**). *Let $T : \mathbb{R}^d \to \mathbb{R}^d$ be a contraction operator. Then*

1. *The equation $T(v) = v$ has a unique solution $\mathcal{V}^* \in \mathbb{R}^d$.*
2. *For any $v_0 \in \mathbb{R}^d$, $\lim_{n\to\infty} T^n(v_0) = \mathcal{V}^*$. In fact, $||T^n(v_0) - \mathcal{V}^*|| \leq O(\beta^n)$, where β is the contraction coefficient.*

Proof. Fix any v_0 and define $v_{n+1} = T(v_n)$. We will show that: (1) there exists a limit to the sequence, and (2) the limit is a fixed point of T.

Existence of a limit v^* of the sequence v_n
We show that the sequence of v_n is a cauchy sequence. We consider two elements v_n and v_{m+n} and bound the distance between them.

$$\begin{aligned}
\|v_{n+m} - v_n\| &= \|\sum_{k=0}^{m-1} v_{n+k+1} - v_{n+k}\| \\
&\leq \sum_{k=0}^{m-1} \|v_{n+k+1} - v_{n+k}\| \quad (\textit{according to the triangle inequality}) \\
&= \sum_{k=0}^{m-1} \|T^{n+k}v_1 - T^{n+k}v_0\| \\
&\leq \sum_{k=0}^{m-1} \beta^{n+k}\|v_1 - v_0\| \quad (\textit{contraction } n+k \textit{ times}) \\
&= \frac{\beta^n(1-\beta^m)}{1-\beta}\|v_1 - v_0\|
\end{aligned}$$

Since the coefficient decreases as n increases, for any $\epsilon > 0$ there exists $N > 0$ such that for all $n, m \geq N$, we have $\|\vec{v}_{n+m} - \vec{v}_n\| < \epsilon$. This implies that the sequence is a Cauchy sequence, and hence the sequence v_n has a limit. Let us call this limit v^*. Next we show that v^* is a fixed point of the operator T.

v^* is a fixed point
We need to show that $T(v^*) = v^*$, or equivalently $\|T(v^*) - v^*\| = 0$.

$$\begin{aligned}
0 &\leq \|T(v^*) - v^*\| \\
&\leq \|T(v^*) - v_n\| + \|v_n - v^*\| \ (\textit{according to the triangle inequality}) \\
&= \|T(v^*) - T(v_{n-1})\| + \|v_n - v^*\| \\
&\leq \beta\|\underbrace{v^* - v_{n-1}}_{\to 0}\| + \|\underbrace{v^n - v^*}_{\to 0}\|
\end{aligned}$$

Since v^* is the limit of v_n, i.e., $\lim_{n\to\infty} \|\vec{v}_n - \vec{v^*}\| = 0$ hence

$$\|T\vec{v^*} - \vec{v^*}\| = 0.$$

Thus, v^* is a fixed point of the operator T.

Uniqueness of $\vec{v^*}$
Assume that $T(v_1) = v_1$, and $T(v_2) = v_2$, and $v_1 \neq v_2$. Then

$$\|v_1 - v_2\| = \|T(v_1) - T(v_2)\| \leq \beta\|v_1 - v_2\|$$

Hence, this is in contradiction to $\beta < 1$. Therefore, v^* is unique. □

5.4.3 The Dynamic Programming Operators

We next define the basic Dynamic Programming operators, and show that they are in fact contraction operators.

For a fixed stationary policy $\pi : \mathcal{S} \to \mathcal{A}$, define the fixed policy DP operator $T^\pi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ as follows: For any $V = (V(\mathtt{s})) \in \mathbb{R}^{|\mathcal{S}|}$,

$$(T^\pi(V))(\mathtt{s}) = \mathtt{r}(\mathtt{s}, \pi(\mathtt{s})) + \gamma \sum\nolimits_{\mathtt{s}' \in \mathcal{S}} p(\mathtt{s}'|\mathtt{s}, \pi(\mathtt{s}))V(\mathtt{s}'), \quad \mathtt{s} \in \mathcal{S}$$

In our column-vector notation, this is equivalent to $T^\pi(V) = r^\pi + \gamma P^\pi V$.

Similarly, define the discounted-return **Dynamic Programming Operator** $T^* : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ as follows: For any $V = (V(\mathtt{s})) \in R^{|\mathcal{S}|}$,

$$(T^*(V))(\mathtt{s}) = \max_{\mathtt{a}\in\mathcal{A}} \left\{ \mathtt{r}(\mathtt{s}, \mathtt{a}) + \gamma \sum\nolimits_{\mathtt{s}' \in \mathcal{S}} p(\mathtt{s}'|\mathtt{s}, \mathtt{a})V(\mathtt{s}') \right\}, \quad \mathtt{s} \in \mathcal{S}$$

We note that T^π is a linear operator, while T^* is generally non-linear due to the maximum operation.

Let $||V||_\infty \stackrel{\Delta}{=} \max_{\mathtt{s}\in\mathcal{S}}|V(\mathtt{s})|$ denote the max-norm of V. Recall that $0 < \gamma < 1$.

Theorem 5.5 (**Contraction property**). *The following statements hold:*

1. *T^π is a γ-contraction operator with respect to the max-norm, namely $||T^\pi(\mathcal{V}_1) - T^\pi(\mathcal{V}_2)||_\infty \leq \gamma||\mathcal{V}_1 - \mathcal{V}_2||_\infty$ for all $\mathcal{V}_1, \mathcal{V}_2 \in \mathbb{R}^{|\mathcal{S}|}$.*

2. *Similarly, T^* is an γ-contraction operator with respect to the max-norm.*

Proof. 1. Fix $\mathcal{V}_1, \mathcal{V}_2$. For every state $\mathtt{s}$,

$$\begin{aligned}|(T^\pi(\mathcal{V}_1))(\mathtt{s}) - (T^\pi(\mathcal{V}_2))(\mathtt{s})| &= \left|\gamma \sum_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\pi(\mathtt{s}))[\mathcal{V}_1(\mathtt{s}') - \mathcal{V}_2(\mathtt{s}')]\right| \\ &\leq \gamma \sum_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\pi(\mathtt{s}))\,|\mathcal{V}_1(\mathtt{s}') - \mathcal{V}_2(\mathtt{s}')| \\ &\leq \gamma \sum_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\pi(\mathtt{s}))\,\|\mathcal{V}_1 - \mathcal{V}_2\|_\infty = \gamma\,\|\mathcal{V}_1 - \mathcal{V}_2\|_\infty\ .\end{aligned}$$

Since this holds for every $\mathtt{s} \in \mathcal{S}$ the required inequality follows.

2. The proof here is more intricate due to the maximum operation. As before, we need to show that $|T^*(\mathcal{V}_1)(\mathtt{s}) - T^*(\mathcal{V}_2)(\mathtt{s})| \leq \gamma\|\mathcal{V}_1 - \mathcal{V}_2\|_\infty$. Fixing the state $\mathtt{s}$, we consider separately the positive and negative parts of the absolute value:

(a) Showing $T^*(\mathcal{V}_1)(\mathtt{s}) - T^*(\mathcal{V}_2)(\mathtt{s}) \leq \gamma\|\mathcal{V}_1 - \mathcal{V}_2\|_\infty$: Let $\bar{\mathtt{a}}$ denote an action that attains the maximum in $T^*(\mathcal{V}_1)(\mathtt{s})$, namely

$$\bar{\mathtt{a}} \in \arg\max_{\mathtt{a}\in\mathcal{A}} \left\{\mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\mathtt{a})\mathcal{V}_1(\mathtt{s}')\right\}.$$

Then

$$\begin{aligned}T^*(\mathcal{V}_1)(\mathtt{s}) &= \mathtt{r}(\mathtt{s},\bar{\mathtt{a}}) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\bar{\mathtt{a}})\mathcal{V}_1(\mathtt{s}') \\ T^*(\mathcal{V}_2)(\mathtt{s}) &\geq \mathtt{r}(\mathtt{s},\bar{\mathtt{a}}) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\bar{\mathtt{a}})\mathcal{V}_2(\mathtt{s}')\end{aligned}$$

Since the same action $\bar{\mathtt{a}}$ appears in both expressions, we can now continue to show the inequality (a) similarly to 1. Namely,

$$\begin{aligned}(T^\pi(\mathcal{V}_1))(\mathtt{s}) - (T^\pi(\mathcal{V}_2))(\mathtt{s}) &= \gamma \sum_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\pi(\mathtt{s}))\,(\mathcal{V}_1(\mathtt{s}') - \mathcal{V}_2(\mathtt{s}')) \\ &\leq \gamma \sum_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\pi(\mathtt{s}))\,\|\mathcal{V}_1 - \mathcal{V}_2\|_\infty = \gamma\,\|\mathcal{V}_1 - \mathcal{V}_2\|_\infty\ .\end{aligned}$$

(b) Showing $T^*(\mathcal{V}_2)(\mathtt{s}) - T^*(\mathcal{V}_1)(\mathtt{s}) \leq \gamma\|\mathcal{V}_1 - \mathcal{V}_2\|_\infty$. By (a) we have

$$T^*(\mathcal{V}_2)(\mathtt{s}) - T^*(\mathcal{V}_1)(\mathtt{s}) \leq \gamma\|\mathcal{V}_2 - \mathcal{V}_1\|_\infty = \gamma\|\mathcal{V}_1 - \mathcal{V}_2\|_\infty.$$

The inequalities (a) and (b) together imply that $|T^*(\mathcal{V}_1)(\mathtt{s}) - T^*(\mathcal{V}_2)(\mathtt{s})| \leq \gamma\|\mathcal{V}_1 - \mathcal{V}_2\|_\infty$. Since this holds for any state $\mathtt{s}$, it follows that $||T^*(\mathcal{V}_1) - T^*(\mathcal{V}_2)||_\infty \leq \gamma\|\mathcal{V}_1 - \mathcal{V}_2\|_\infty$.

□

5.5 Proof of Bellman's Optimality Equation

We prove in this section Theorem 5.1, which is restated here:

Theorem (Bellman's Optimality Equation)**.** *The following statements hold:*

1. $\mathcal{V}^*$ *is the unique solution of the following set of (nonlinear) equations:*

$$\mathcal{V}(\mathtt{s}) = \max_{\mathtt{a}\in\mathcal{A}} \left\{ \mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\mathtt{a})\mathcal{V}(\mathtt{s}') \right\}, \quad \mathtt{s}\in\mathcal{S}. \tag{5.4}$$

2. *Any stationary policy* π^* *that satisfies*

$$\pi^*(\mathtt{s}) \in \arg\max_{\mathtt{a}\in\mathcal{A}} \left\{ \mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\mathtt{a})\mathcal{V}(\mathtt{s}') \right\},$$

is an optimal policy (for any initial state $\mathtt{s}_0 \in \mathcal{S}$*).*

We observe that the Optimality equation in part 1 is equivalent to $V = T^*(V)$ where T^* is the optimal DP operator from the previous section, which was shown to be a contraction operator with coefficient γ. The proof also uses the value iteration property of Theorem 5.2.

Proof of Theorem 5.1: We prove each part.

1. As T^* is a contraction operator, existence and uniqueness of the solution to $V = T^*(V)$ follows from the Banach fixed point theorem (Theorem 5.4). Let $\hat{V}$ denote that solution. It also follows by that theorem (Theorem 5.4) that $(T^*)^n(\mathcal{V}_0) \to \hat{V}$ for any $\mathcal{V}_0$. By Theorem 5.2 we have that $(T^*)^n(\mathcal{V}_0) \to \mathcal{V}^*$, hence $\hat{V} = \mathcal{V}^*$, so that $\mathcal{V}^*$ is indeed the unique solution of $V = T^*(V)$.

2. By definition of π^* we have

$$T^{\pi^*}(\mathcal{V}^*) = T^*(\mathcal{V}^*) = \mathcal{V}^*,$$

where the last equality follows from part 1. Thus the optimal value function satisfied the equation $T^{\pi^*}(\mathcal{V}^*) = \mathcal{V}^*$. But we already know (from Prop. 5.2) that $\mathcal{V}^{\pi^*}$ is the unique solution of that equation, hence $\mathcal{V}^{\pi^*} = \mathcal{V}^*$. This implies that π^* achieves the optimal value (for any initial state), and is therefore an optimal policy as stated.

□

5.6 Value Iteration (VI)

The value iteration algorithm allows to compute the optimal value function $\mathcal{V}^*$ iteratively to any required accuracy. The Value Iteration algorithm (Algorithm 5.2) can be stated as follows:

1. Start with any initial value function $\mathcal{V}_0 = (\mathcal{V}_0(\mathtt{s}))$.

2. Compute recursively, for $n = 0, 1, 2, \ldots$,

$$\mathcal{V}_{n+1}(\mathtt{s}) = \max_{\mathtt{a}\in\mathcal{A}} \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\mathtt{a})[\mathtt{r}(\mathtt{s},\mathtt{a},\mathtt{s}') + \gamma\mathcal{V}_n(\mathtt{s}')], \quad \forall \mathtt{s} \in \mathcal{S}.$$

3. Apply a stopping rule to obtain a required accuracy (see below).

In terms of the DP operator T^*, value iteration is simply stated as:

$$\mathcal{V}_{n+1} = T^*(\mathcal{V}_n), \quad n \geq 0.$$

Note that the number of operations for each iteration is $O(|\mathcal{A}| \cdot |\mathcal{S}|^2)$. Theorem 5.2 states that $\mathcal{V}_n \to \mathcal{V}^*$, exponentially fast.

5.6.1 Error bounds and stopping rules:

While we showed an exponential convergence rate, it is important to have a criteria that would depend only on the observed quantities.

Lemma 5.3. *If* $\|\mathcal{V}_{n+1} - \mathcal{V}_n\|_\infty < \epsilon \cdot \frac{1-\gamma}{2\gamma}$ *then* $\|\mathcal{V}_{n+1} - \mathcal{V}^*\|_\infty < \frac{\varepsilon}{2}$ *and* $|\mathcal{V}^{\pi_{n+1}} - \mathcal{V}^*| \leq \varepsilon$, *where* π_{n+1} *is the greedy policy w.r.t.* $\mathcal{V}_{n+1}$.

Proof. Assume that $\|\mathcal{V}_{n+1} - \mathcal{V}_n\| < \varepsilon \cdot \frac{1-\gamma}{2\gamma}$, and we show that $\|\mathcal{V}^{\pi_{n+1}} - \mathcal{V}^*\| < \varepsilon$, which would make the policy π_{n+1} ε-optimal. We bound the difference between $\mathcal{V}^{\pi_{n+1}}$ and $\mathcal{V}^*$. (All the norms are max-norm.) We consider the following:

$$\|\mathcal{V}^{\pi_{n+1}} - \mathcal{V}^*\| \leq \|\mathcal{V}^{\pi_{n+1}} - \mathcal{V}_{n+1}\| + \|\mathcal{V}_{n+1} - \mathcal{V}^*\| \tag{5.5}$$

We now bound each part of the sum separately:

$$\begin{aligned} \|\mathcal{V}^{\pi_{n+1}} - \mathcal{V}_{n+1}\| &= \|T^{\pi_{n+1}}(\mathcal{V}^{\pi_{n+1}}) - \mathcal{V}_{n+1}\| \qquad (\textit{because } \mathcal{V}^{\pi_{n+1}} \textit{ is the fixed point of } T^{\pi_{n+1}}) \\ &\leq \|T^{\pi_{n+1}}(\mathcal{V}^{\pi_{n+1}}) - T^*(\mathcal{V}_{n+1})\| + \|T^*(\mathcal{V}_{n+1}) - \mathcal{V}_{n+1}\| \end{aligned}$$

Since π_{n+1} is maximal over the actions using $\mathcal{V}_{n+1}$, it implies that $T^{\pi_{n+1}}(\mathcal{V}_{n+1}) = T^*(\mathcal{V}_{n+1})$ and we conclude that:

$$\begin{aligned}\|\mathcal{V}^{\pi_{n+1}} - \mathcal{V}_{n+1}\| &\leq \|T^{\pi_{n+1}}(\mathcal{V}^{\pi_{n+1}}) - T^{\pi_{n+1}}(\mathcal{V}_{n+1})\| + \|T^*(\mathcal{V}_{n+1}) - T^*(\mathcal{V}_n)\| \\ &\leq \gamma\|\mathcal{V}^{\pi_{n+1}} - \mathcal{V}_{n+1}\| + \gamma\|\mathcal{V}_{n+1} - \mathcal{V}_n\|\end{aligned}$$

Rearranging, this implies that,

$$\|\mathcal{V}^{\pi_{n+1}} - \mathcal{V}_{n+1}\| \leq \frac{\gamma}{1-\gamma}\|\mathcal{V}_{n+1} - \mathcal{V}_n\| < \frac{\gamma}{1-\gamma} \cdot \epsilon \cdot \frac{1-\gamma}{2\gamma} = \frac{\epsilon}{2}$$

For the second part of the sum we derive similarly that:

$$\|\mathcal{V}_{n+1} - \mathcal{V}^*\| \leq \frac{\gamma}{1-\gamma}\|\mathcal{V}_{n+1} - \mathcal{V}_n\| < \frac{\gamma}{1-\gamma} \cdot \epsilon \cdot \frac{1-\gamma}{2\gamma} = \frac{\epsilon}{2}$$

Returning to inequality (5.5), it follows:

$$\|\mathcal{V}^{\pi_{n+1}} - \mathcal{V}^*\| \leq \frac{2\gamma}{1-\gamma}\|\mathcal{V}_{n+1} - \mathcal{V}_n\| < \epsilon$$

Therefore the selected policy π_{n+1} is ϵ-optimal. □

5.7 Policy Iteration (PI)

The policy iteration algorithm, introduced by Howard (1960), computes an optimal policy π^* in a finite number of steps. This number is typically small (on the same order as $|\mathcal{S}|$).

The basic principle behind Policy Iteration is Policy Improvement. Let π be a stationary policy, and let $\mathcal{V}^\pi$ denote its value function. A stationary policy $\bar{\pi}$ is called π- improving if it is a greedy policy with respect to $\mathcal{V}^\pi$, namely

$$\bar{\pi}(\mathtt{s}) \in \arg\max_{\mathtt{a}\in\mathcal{A}} \left\{ \mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma \sum\nolimits_{\mathtt{s}'\in\mathcal{S}} p(\mathtt{s}'|\mathtt{s},\mathtt{a})\mathcal{V}^\pi(\mathtt{s}') \right\}, \quad \mathtt{s} \in \mathcal{S}.$$

Lemma 5.4 (Policy Improvement)**.** *Let π be a stationary policy and $\bar{\pi}$ be a π-improving policy. We have $\mathcal{V}^{\bar{\pi}} \geq \mathcal{V}^\pi$ (component-wise), and $\mathcal{V}^{\bar{\pi}} = \mathcal{V}^\pi$ if and only if π is an optimal policy.*

Proof. Observe first that

$$\mathcal{V}^\pi = T^\pi(\mathcal{V}^\pi) \leq T^*(\mathcal{V}^\pi) = T^{\bar{\pi}}(\mathcal{V}^\pi)$$

The first equality follows since $\mathcal{V}^\pi$ is the value function for the policy π, the inequality follows because of the maximization in the definition of T^*, and the last equality by definition of the improving policy $\bar{\pi}$.

It is easily seen that T^π is a monotone operator (for any policy π), namely $\mathcal{V}_1 \leq \mathcal{V}_2$ implies $T^\pi(\mathcal{V}_1) \leq T^\pi(\mathcal{V}_2)$. Applying $T^{\bar{\pi}}$ repeatedly to both sides of the above inequality $\mathcal{V}^\pi \leq T^{\bar{\pi}}(\mathcal{V}^\pi)$ therefore gives

$$\mathcal{V}^\pi \leq T^{\bar{\pi}}(\mathcal{V}^\pi) \leq (T^{\bar{\pi}})^2(\mathcal{V}^\pi) \leq \cdots \leq \lim_{n\to\infty} (T^{\bar{\pi}})^n(\mathcal{V}^\pi) = \mathcal{V}^{\bar{\pi}},$$

where the last equality follows by Theorem 5.2 This establishes the first claim.

We now show that π is optimal if and only if $\mathcal{V}^{\bar{\pi}} = \mathcal{V}^\pi$. We showed that $\mathcal{V}^{\bar{\pi}} \geq \mathcal{V}^\pi$. If $\mathcal{V}^{\bar{\pi}} > \mathcal{V}^\pi$ then clearly π is not optimal. Assume that $\mathcal{V}^{\bar{\pi}} = \mathcal{V}^\pi$. We have the following identities:

$$\mathcal{V}^\pi = \mathcal{V}^{\bar{\pi}} = T^{\bar{\pi}}(\mathcal{V}^{\bar{\pi}}) = T^{\bar{\pi}}(\mathcal{V}^\pi) = T^*(\mathcal{V}^{\bar{\pi}}) = T^*(\mathcal{V}^\pi)$$

where the first equality is by our assumption. The second equality follows since $\mathcal{V}^{\bar{\pi}}$ is the fixed point of its operator $T^{\bar{\pi}}$. The third follows since we assume that $\mathcal{V}^{\bar{\pi}} = \mathcal{V}^\pi$. The last equality follows since $T^{\bar{\pi}}$ and T^* are identical on $\mathcal{V}^\pi$.

We have established that: $\mathcal{V}^\pi = T^*(\mathcal{V}^\pi)$, and hence $\mathcal{V}^\pi$ and π is a fixed point of T^* and therefore, by Theorem 5.1, policy π is optimal. □

The policy iteration algorithm performs successive rounds of policy improvement, where each policy π_{k+1} improves the previous one π_k. Since the number of stationary policies is bounded, so is the number of strict improvements, and the algorithm must terminate with an optimal policy after a finite number of iterations.

In terms of computational complexity, Policy Iteration requires $O(|\mathcal{A}| \cdot |\mathcal{S}|^2 + |\mathcal{S}|^3)$ operations per iteration, while Value Iteration requires $O(|\mathcal{A}| \cdot |\mathcal{S}|^2)$ per iteration. However, in many cases the Policy Iteration has a smaller number of iteration than Value Iteration, as we show in the next section. Another consideration is that the number of iterations of Value Iteration increases as the discount factor γ approaches 1, while the number of policies (which upper bound the number of iterations of Policy Iteration) is independent from γ.

5.8 A Comparison between VI and PI Algorithms

In this section we will compare the convergence rate of the VI and PI algorithms. We show that, assuming that the two algorithms begin with the same approximated value, the PI algorithm converges in less iterations.

Theorem 5.6. *Let $\{VI_n\}$ be the sequence of values created by the VI algorithm (where $VI_{n+1} = T^*(VI_n)$) and let $\{PI_n\}$ be the sequence of values created by PI algorithm, i.e., $PI_n = \mathcal{V}^{\pi_n}$. If $VI_0 = PI_0$, then for all n we have $VI_n \leq PI_n \leq \mathcal{V}^*$.*

Proof. The proof is by induction on n.
Induction Basis: By construction $VI_0 = PI_0$. Since $PI_0 = \mathcal{V}^{\pi_0}$, it is clearly bounded by $\mathcal{V}^*$.
Induction Step: Assume that $VI_n \leq PI_n$. For VI_{n+1} we have,

$$VI_{n+1} = T^*(VI_n) = T^{\pi'}(VI_n),$$

where π' is the greedy policy w.r.t. VI_n, i.e.,

$$\pi'(\mathtt{s}) \in \arg\max_{a \in A}\{\mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma \sum_{\mathtt{s}' \in \mathcal{S}} p(\mathtt{s}'|\mathtt{s},\mathtt{a}) VI_n(\mathtt{s}')\}.$$

Since $VI_n \leq PI_n$, and $T^{\pi'}$ is monotonic it follows that:

$$T^{\pi'}(VI_n) \;\; \leq \;\; T^{\pi'}(PI_n)$$

Since T^* is upper bounding any T^{π}:

$$T^{\pi'}(PI_n) \;\; \leq \;\; T^*(PI_n)$$

The policy determined by PI algorithm in iteration $n+1$ is π_{n+1} and we have:

$$T^*(PI_n) = T^{\pi_{n+1}}(PI_n)$$

From the definition of π_{n+1}, we have

$$T^{\pi_{n+1}}(PI_n) \;\; \leq \;\; \mathcal{V}^{\pi_{n+1}} = PI_{n+1}$$

Therefore, $VI_{n+1} \leq PI_{n+1}$. Since $PI_{n+1} = \mathcal{V}^{\pi_{n+1}}$, it implies that $PI_{n+1} \leq \mathcal{V}^*$. □

5.9 Some Variants on Value Iteration and Policy Iteration

5.9.1 Value Iteration - Gauss Seidel Iteration

In the standard value iteration: $\mathcal{V}_{n+1} = T^*(\mathcal{V}_n)$, the vector $\mathcal{V}_n$ is held fixed while all entries of $\mathcal{V}_{n+1}$ are updated. An alternative is to update each element $\mathcal{V}_n(\mathtt{s})$ of that vector as to $\mathcal{V}_{n+1}(\mathtt{s})$ as soon as the latter is computed, and continue the calculation with the new value. This procedure is guaranteed to be "as good" as the standard one, in some sense, and often speeds up convergence.

5.9.2 Asynchronous Value Iteration

Here, in each iteration $\mathcal{V}_n \Rightarrow \mathcal{V}_{n+1}$, only a subset of the entries of $\mathcal{V}_n$ (namely, a subset of all states) is updated. It can be shown that if each state is updated infinitely often, then $\mathcal{V}_n \to \mathcal{V}^*$. Asynchronous update can be used to focus the computational effort on "important" parts of a large-state space.

5.9.3 Modified (a.k.a. Generalized or Optimistic) Policy Iteration

This scheme combines policy improvement steps with value iteration for policy evaluation. This way the requirement for exact policy evaluation (computing $\mathcal{V}^{\pi_k} = (I - \gamma P^{\pi_k})^{-1} r^{\pi_k}$) is avoided.

The procedure starts with some initial value vector $\mathcal{V}_0$, and iterates as follows:

- Greedy policy computation:

  Compute $\pi_k \in \arg\max_\pi T^\pi(\mathcal{V}_k)$, a greedy policy with respect to $\mathcal{V}_k$.

- Partial value iteration:

  Perform m_k steps of value iteration, $\mathcal{V}_{k+1} = (T_\gamma^{\pi_k})^{m_k}(\mathcal{V}_k)$.

This algorithm guarantees convergence of $\mathcal{V}_k$ to $\mathcal{V}^*$.

5.10 Linear Program

In this section we will use linear programming to derive the optimal policy for discounted return. We will extend the linear programming given in Section 4.5 from Finite Horizon return to discounted return, but the derivation is similar in spirit. We will see that both the primal and dual program will play an important part in defining the optimal policy. We will fix an initial state $\mathtt{s}_0$ and compute the optimal policy for it.

We will start with the primal linear program, which will compute the optimal policy. For each state $\mathtt{s}$ and action $\mathtt{a}$ we will have a variable $x(\mathtt{s}, \mathtt{a})$ that will indicate the discounted fraction of time we are at state $\mathtt{s}$ and perform action $\mathtt{a}$.

To better understand what we mean by the "discounted fraction of time" consider a fixed policy π and a trajectory $(\mathtt{s}_0, \ldots)$ generated by π. Define $X^\pi(\mathtt{s}, \mathtt{a}) = \sum_\mathtt{t} \mathbb{I}(\mathtt{s}_\mathtt{t} = \mathtt{s}, \mathtt{a}_\mathtt{t} = \mathtt{a})$, which is a random variable. We are interested in $x^\pi(\mathtt{s}, \mathtt{a}) = \mathbb{E}[X^\pi(\mathtt{s}, \mathtt{a})]$ which is the expected discounted fraction of time

policy π is in state $\mathtt{s}$ and performs action $\mathtt{a}$. The goal of our linear program is to compute x^{π^*} for the optimal policy π^*.

Our main constraint will be a flow constraint, stating that the discounted fraction of time we reach state $\mathtt{s}$ upper bounds the discounted fraction of time we exit it, times the discounted factor. Formally, for $\mathtt{s} \in \mathcal{S}$,

$$\mathbb{I}(\mathtt{s} = \mathtt{s}_0) + \sum_{\mathtt{a}} x(\mathtt{s}, \mathtt{a}) \leq \gamma \sum_{\mathtt{s}', \mathtt{a}'} x(\mathtt{s}', \mathtt{a}') p(\mathtt{s} | \mathtt{s}' \mathtt{a}').$$

Note that if we sum the inequalities over all states, we have

$$1 + \sum_{\mathtt{s}, \mathtt{a}} x(\mathtt{s}, \mathtt{a}) \leq \gamma \sum_{\mathtt{s}', \mathtt{a}'} x(\mathtt{s}', \mathtt{a}') \sum_{\mathtt{s}} p(\mathtt{s} | \mathtt{s}' \mathtt{a}') = \gamma \sum_{\mathtt{s}', \mathtt{a}'} x(\mathtt{s}', \mathtt{a}').$$

which implies that $\sum_{\mathtt{s},\mathtt{a}} x(\mathtt{s}, \mathtt{a}) \leq 1/(1-\gamma)$, as we should expect. Namely, in each time we are in some state, therefore the sum over states should be $\sum_{\mathtt{t}} \gamma^{\mathtt{t}} = 1/(1-\gamma)$.

The discounted return, which we would like to maximize, is $\mathbb{E}[\sum_{\mathtt{t}} \gamma^{\mathtt{t}} \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})]$. We can regroup the sum by state and action and have $\sum_{\mathtt{s},\mathtt{a}} \mathbb{E}[\sum_{\mathtt{t}} \gamma^{\mathtt{t}} \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) \mathbb{I}(\mathtt{s}_\mathtt{t} = \mathtt{s}, \mathtt{a}_\mathtt{t} = \mathtt{a})]$, which is equivalent to $\sum_{\mathtt{s},\mathtt{a}} \mathtt{r}(\mathtt{s}, \mathtt{a}) \mathbb{E}[\sum_{\mathtt{t}} \gamma^{\mathtt{t}} \mathbb{I}(\mathtt{s}_\mathtt{t} = \mathtt{s}, \mathtt{a}_\mathtt{t} = \mathtt{a})]$. Now our variable are $x(\mathtt{s}, \mathtt{a}) = \mathbb{E}[\sum_{\mathtt{t}} \gamma^{\mathtt{t}} \mathbb{I}(\mathtt{s}_\mathtt{t} = \mathtt{s}, \mathtt{a}_\mathtt{t} = \mathtt{a})]$, and the expected return would be

$$\sum_{\mathtt{s},\mathtt{a}} \mathtt{r}(\mathtt{s}, \mathtt{a}) x(\mathtt{s}, \mathtt{a})$$

The resulting linear program is the following.

$$\begin{aligned}
\max_{x(\mathtt{s},\mathtt{a})} \quad & \sum_{\mathtt{s},\mathtt{a}} \mathtt{r}(\mathtt{s}, \mathtt{a}) x(\mathtt{s}, \mathtt{a}) \\
& \text{such that} \\
& \mathbb{I}(\mathtt{s} = \mathtt{s}_0) + \sum_{\mathtt{a}} x(\mathtt{s}, \mathtt{a}) \leq \gamma \sum_{\mathtt{s}', \mathtt{a}'} x(\mathtt{s}', \mathtt{a}') p(\mathtt{s} | \mathtt{s}' \mathtt{a}') && \forall \mathtt{s} \in \mathcal{S}, \mathtt{a} \in \mathcal{A}, \\
& x(\mathtt{s}, \mathtt{a}) \geq 0 && \forall \mathtt{s} \in \mathcal{S}, \mathtt{a} \in \mathcal{A}, \\
& \sum_{\mathtt{a}} x_0(\mathtt{s}_0, \mathtt{a}) = 1 && \forall \mathtt{a} \in \mathcal{A}, \\
& x_0(\mathtt{s}, \mathtt{a}) = 0, && \forall \mathtt{s} \in \mathcal{S}, \mathtt{s} \neq \mathtt{s}_0
\end{aligned}$$

Given the primal linear program we can derive the dual linear program.

$$\min_{z(\mathtt{s})} \; z_0(\mathtt{s}_0)$$

such that

$$z(\mathtt{s}) \geq \mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma \sum_{\mathtt{s}'} z(\mathtt{s}')p(\mathtt{s}'|\mathtt{s},\mathtt{a}), \qquad \forall \mathtt{s} \in \mathcal{S}, \mathtt{a} \in \mathcal{A},$$

.

One can identify the dual random variables $z(\mathtt{s})$ with the optimal vale function $\mathcal{V}(\mathtt{s})$. At the optimal solution of the dual linear program one can show that we have

$$z(\mathtt{s}) = \max_{\mathtt{a}} \left\{ \mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma \sum_{\mathtt{s}'} z(\mathtt{s}')p_{\mathtt{t}}(\mathtt{s}'|\mathtt{s},\mathtt{a}) \right\}, \qquad \forall \mathtt{s} \in \mathcal{S},$$

which are the familiar Bellman optimality equations.

Chapter 6

Stochastic shortest paths

Chapter 7

Average cost

Chapter 8

Model based learning: Tabular case

Until now we looked at planning problems, where we are given a complete model of the MDP, and the goal is to either evaluate a given policy or compute the optimal policy. In this chapter we will start looking at learning problems, where we need to learn from interaction. This chapter will concentrate on *model based* learning, where the main goal is to learn an accurate model and use it. In the following chapters we will look at *model free* learning, where we learn a policy without recovering the actual underlying model.

8.1 Effective horizon of discounted return

Before we start looking at the learning setting, we will show an "reduction" from discounted return to finite horizon return. The main issue will be to show that the discounted return has an *effective horizon* such that rewards beyond it have a negligible effect on the discounted return.

Theorem 8.1. *Given a discount factor γ, the discounted return in the first $\mathtt{T} = \frac{1}{1-\gamma} \log \frac{\mathtt{R}_{\max}}{\varepsilon(1-\gamma)}$ time steps, is within ε of the total discounted return.*

Proof. Recall that the rewards are $\mathtt{r}_\mathtt{t} \in [0, \mathtt{R}_{\max}]$. Fix an infinite sequence of rewards $(\mathtt{r}_0, \ldots, \mathtt{r}_\mathtt{t}, \ldots)$. We would like to consider the following difference

$$\Delta = \sum_{\mathtt{t}=1}^{\infty} \mathtt{r}_\mathtt{t} \gamma^\mathtt{t} - \sum_{\mathtt{t}=0}^{\mathtt{T}-1} \mathtt{r}_\mathtt{t} \gamma^\mathtt{t} = \sum_{\mathtt{t}=\mathtt{T}}^{\infty} \mathtt{r}_\mathtt{t} \gamma^\mathtt{t} \leq \frac{\gamma^\mathtt{T}}{1-\gamma} \mathtt{R}_{\max}$$

We want this difference Δ to be bounded by ε, hence

$$\frac{\gamma^{\mathtt{T}}}{1-\gamma}\mathtt{R}_{\max} \leq \varepsilon \ .$$

This implies that

$$\mathtt{T}\log(1/\gamma) \geq \log \frac{\mathtt{R}_{\max}}{\varepsilon(1-\gamma)} \ .$$

We can bound $\log(1/\gamma) = \log(1 + \frac{1-\gamma}{\gamma}) \leq \frac{1-\gamma}{\gamma}$. Since $\gamma < 1$, and the theorem follows. □

8.2 Off-Policy Model-Based Learning

In the off-policy setting we would have access to previous executed trajectories, and we would like to use them to learn. Naturally, we will have to make some assumption about the trajectories. Intuitively, we will need to assume that they are sufficiently exploratory.

We will decompose the trajectories to quadruples, which are composed:

$$(\mathtt{s}, \mathtt{a}, \mathtt{r}, \mathtt{s}')$$

where $\mathtt{r}$ is sampled from $\mathtt{R}(\mathtt{s}, \mathtt{a})$ and $\mathtt{s}'$ is sampled from $p(\cdot|\mathtt{s}, \mathtt{a})$.

Our goal is to output an MDP $(\mathcal{S}, \mathcal{A}, \widehat{\mathtt{r}}, \widehat{p})$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $\widehat{\mathtt{r}}(\mathtt{s}, \mathtt{a})$ is the approximate expected reward of $\mathtt{R}(\mathtt{s}, \mathtt{a}) \in [0, \mathtt{R}_{\max}]$, and $\widehat{p}(\mathtt{s}'|\mathtt{s}, \mathtt{a})$ is the approximate probability of reaching state $\mathtt{s}'$ when we are in state $\mathtt{s}$ and doing action $\mathtt{a}$.

8.2.1 Mean estimation

We start with a basic mean estimation problem, which appeared in many settings including supervised learning. Suppose we are given access to a random variable $\mathtt{R} \in [0, 1]$ and would like to approximate its mean $\mu = \mathbb{E}[\mathtt{R}]$. We observe m samples of $\mathtt{R}$, which are $R_1, \ldots, R_m$, and compute their observed mean $\widehat{\mu} = \frac{1}{m}\sum_{i=1}^{m} R_i$.

By the law of large numbers we know that when m goes to infinity we have that $\widehat{\mu}$ converges to μ. We would like to have concrete finite convergence bounds, mainly to derive the value of m as a function of the desired accuracy ε. For this we use concentration bounds (known as Chernoff-Hoffding bounds). The bounds have both an additive form and a multiplicative form, given as follows:

Lemma 8.1 (Chernoff-Hoffding)**.** *Let $R_1, \ldots, R_m$ be m i..i.d. samples of a random variable $\mathtt{R} \in [0,1]$. Let $\mu = \mathbb{E}[\mathtt{R}]$ and $\widehat{\mu} = \frac{1}{m}\sum_{i=1}^{m} R_i$. For any $\varepsilon \in (0,1)$ we have,*

$$\Pr[|\mu - \widehat{\mu}| \geq \varepsilon] \leq 2e^{-2\varepsilon^2 m}$$

In addition,

$$\Pr[\widehat{\mu} \leq (1-\varepsilon)\mu] \leq e^{-\varepsilon^2 m/2} \qquad \text{and} \qquad \Pr[\widehat{\mu} \geq (1+\varepsilon)\mu] \leq e^{-\varepsilon^2 m/3}$$

We will refer to the first bound as *additive* and the second set of bound as *multiplicative*.

Using the additive bound of Lemma 8.1, we have

Corollary 8.1. *Let $R_1, \ldots, R_m$ be m i..i.d. samples of a random variable $\mathtt{R} \in [0,1]$. Let $\mu = \mathbb{E}[\mathtt{R}]$ and $\widehat{\mu} = \frac{1}{m}\sum_{i=1}^{m} R_i$. Fix $\varepsilon, \delta > 0$. Then, for $m \geq \frac{1}{2\varepsilon^2}\log(2/\delta)$, with probability $1-\delta$, we have that $|\mu - \widehat{\mu}| \leq \varepsilon$.*

We can now use the above concentration bound in order to estimate the expected rewards. For each state-action $(\mathtt{s},\mathtt{a})$ let $\widehat{\mathtt{r}}(\mathtt{s},\mathtt{a}) = \frac{1}{m}\sum_{i=1}^{m} \mathtt{R}_i(\mathtt{s},\mathtt{a})$ be the average of m samples. We can show the following:

Claim 8.1. *If we have $m \geq \frac{\mathtt{R}_{\max}}{2\varepsilon^2}\log\frac{2|\mathcal{S}|\,|\mathcal{A}|}{\delta}$ samples for each state action $(\mathtt{s},\mathtt{a})$, then with probability $1-\delta$ we have for every $(\mathtt{s},\mathtt{a})$ that $|\mathtt{r}(\mathtt{s},\mathtt{a}) - \widehat{\mathtt{r}}(\mathtt{s},\mathtt{a})| \leq \varepsilon$.*

Proof. First, we will need to scale the random variables to $[0,1]$, which will be achieved by dividing by $\mathtt{R}_{\max}$. Then, by the Chernoff-Hoffding bound (Corollary 8.1), using $\varepsilon = \frac{\varepsilon}{\mathtt{R}_{\max}}$ and $\delta' = \frac{\delta}{|\mathcal{S}|\,|\mathcal{A}|}$, we have that for each $(\mathtt{s},\mathtt{a})$ we have that with probability $1 - \frac{\delta}{|\mathcal{S}|\,|\mathcal{A}|}$ that $|\frac{\mathtt{r}(\mathtt{s},\mathtt{a})}{\mathtt{R}_{\max}} - \frac{\widehat{\mathtt{r}}(\mathtt{s},\mathtt{a})}{\mathtt{R}_{\max}}| \leq \frac{\varepsilon}{\mathtt{R}_{\max}}$.

We bound the probability over all state-action pairs using a union bound,

$$\begin{aligned}\Pr\left[\exists(\mathtt{s},\mathtt{a}) : |\frac{\mathtt{r}(\mathtt{s},\mathtt{a})}{\mathtt{R}_{\max}} - \frac{\widehat{\mathtt{r}}(\mathtt{s},\mathtt{a})}{\mathtt{R}_{\max}}|| > \frac{\varepsilon}{\mathtt{R}_{\max}}\right] &\leq \sum_{(\mathtt{s},\mathtt{a})} \Pr\left[|\frac{\mathtt{r}(\mathtt{s},\mathtt{a})}{\mathtt{R}_{\max}} - \frac{\widehat{\mathtt{r}}(\mathtt{s},\mathtt{a})}{\mathtt{R}_{\max}}|| > \frac{\varepsilon}{\mathtt{R}_{\max}}\right] \\ &\leq \sum_{(\mathtt{s},\mathtt{a})} \frac{\delta}{|\mathcal{S}|\,|\mathcal{A}|} = \delta\end{aligned}$$

Therefore, we have that with probability $1-\delta$ for every $(\mathtt{s},\mathtt{a})$ simultaneously we have $|\mathtt{r}(\mathtt{s},\mathtt{a}) - \widehat{\mathtt{r}}(\mathtt{s},\mathtt{a})| \leq \varepsilon$. □

8.2.2 Influence of reward estimation errors

We would like to quantify the influence of having inaccurate estimates of the rewards. We will look both at the finite horizon return and the discounted return. We start with the case of finite horizon.

Influence of reward estimation errors: Finite horizon

Fix a policy $\pi \in MD$. We want to compare the return using $\mathtt{r}_\mathtt{t}(\mathtt{s},\mathtt{a})$ versus $\widehat{\mathtt{r}}(\mathtt{s},\mathtt{a})$ and $\mathtt{r}_\mathtt{T}(\mathtt{s})$ versus $\widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s})$. We will assume that for every $(\mathtt{s},\mathtt{a})$ and $\mathtt{t}$ we have $|\mathtt{r}_\mathtt{t}(\mathtt{s},\mathtt{a}) - \widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s},\mathtt{a})| \leq \varepsilon$ and $|\mathtt{r}_\mathtt{T}(\mathtt{s}) - \widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s})| \leq \varepsilon$. We will show that the difference in return is bounded by $\varepsilon(\mathtt{T}+1)$, where $\mathtt{T}$ is the finite horizon.

Define the expected return of π with the true expectations

$$\mathcal{V}^\pi_\mathtt{T}(\mathtt{s}_0) = \mathbb{E}^{\pi,\mathtt{s}_0}[\sum_{\mathtt{t}=0}^{\mathtt{T}} \mathtt{r}_\mathtt{t}(\mathtt{s}_\mathtt{t},\mathtt{a}_\mathtt{t}) + \mathtt{r}_\mathtt{T}(\mathtt{s}_\mathtt{T})].$$

and with the estimated expectations

$$\widehat{\mathcal{V}}^\pi_\mathtt{T}(\mathtt{s}_0) = \mathbb{E}^{\pi,\mathtt{s}_0}[\sum_{\mathtt{t}=0}^{\mathtt{T}} \widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s}_\mathtt{t},\mathtt{a}_\mathtt{t}) + \widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s}_\mathtt{T})].$$

We are interested in bounding the difference between the two

$$error(\pi) = |\mathcal{V}^\pi_\mathtt{T}(\mathtt{s}_0) - \widehat{\mathcal{V}}^\pi_\mathtt{T}(\mathtt{s}_0)|.$$

Note that in both cases we use the true transition probability. For a given trajectory $\sigma = (\mathtt{s}_0,\mathtt{a}_0,\ldots,\mathtt{s}_{\mathtt{T}-1},\mathtt{a}_{\mathtt{T}-1},\mathtt{s}_\mathtt{T})$ we define

$$error(\pi,\sigma) = \left(\sum_{\mathtt{t}=0}^{\mathtt{T}} \mathtt{r}_\mathtt{t}(\mathtt{s}_\mathtt{t},\mathtt{a}_\mathtt{t}) + \mathtt{r}_\mathtt{T}(\mathtt{s}_\mathtt{T})\right) - \left(\sum_{\mathtt{t}=0}^{\mathtt{T}} \widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s}_\mathtt{t},\mathtt{a}_\mathtt{t}) + \widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s}_\mathtt{T})\right)$$

taking the expectation over trajectories we have

$$error(\pi) = |\mathbb{E}^{\pi,\mathtt{s}_0}[error(\pi,\sigma)]|$$

Lemma 8.2. *Assume that for every* $(\mathtt{s},\mathtt{a})$ *and* $\mathtt{t}$ *we have* $|\mathtt{r}_\mathtt{t}(\mathtt{s},\mathtt{a}) - \widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s},\mathtt{a})| \leq \varepsilon$ *and for every* $\mathtt{s}$ *we have* $|\mathtt{r}_\mathtt{T}(\mathtt{s}) - \widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s})| \leq \varepsilon$. *Then, for any policy* $\pi \in MD$ *we have* $error(\pi) \leq \varepsilon(\mathtt{T}+1)$.

Proof. The probability of each trajectory $\sigma = (\mathtt{s}_0,\mathtt{a}_0,\ldots,\mathtt{s}_{\mathtt{T}-1},\mathtt{a}_{\mathtt{T}-1},\mathtt{s}_\mathtt{T})$ is the same under $\mathtt{r}_\mathtt{t}(\mathtt{s},\mathtt{a})$ and $\widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s},\mathtt{a})$, since $\pi \in MD$ implies that π depends only on the time

$\mathtt{t}$ and state $\mathtt{s}_\mathtt{t}$. For each trajectory $\sigma = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{\mathtt{T}-1}, \mathtt{a}_{\mathtt{T}-1}, \mathtt{s}_\mathtt{T})$, we have,

$$\begin{aligned}
error(\pi, \sigma) &= \left| \sum_{\mathtt{t}=0}^{\mathtt{T}} (\mathtt{r}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) + \mathtt{r}_\mathtt{T}(\mathtt{s}_\mathtt{T})) - \sum_{\mathtt{t}=0}^{\mathtt{T}} (\widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) + \widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s}_\mathtt{T})) \right| \\
&= \left| \sum_{\mathtt{t}=0}^{\mathtt{T}} (\mathtt{r}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) - \widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})) + (\mathtt{r}_\mathtt{T}(\mathtt{s}_\mathtt{T}) - \widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s}_\mathtt{T})) \right| \\
&\leq \sum_{\mathtt{t}=0}^{\mathtt{T}} |\mathtt{r}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) - \widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})| + |\mathtt{r}_\mathtt{T}(\mathtt{s}_\mathtt{T}) - \widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s}_\mathtt{T})| \\
&\leq \varepsilon \mathtt{T} + \varepsilon
\end{aligned}$$

The theorem follows since $error(\pi) = |E^{\pi, \mathtt{s}_0}[error(\pi, \sigma)]| \leq \varepsilon(\mathtt{T}+1)$, and the bound hold for every trajectory σ. □

Computing approximate optimal policy: finite horizon

We can now describe how to compute a near optimal policy for the finite horizon case. We start with the sample requirement. We need a sample of size $m \geq \frac{1}{2\varepsilon^2} \log \frac{2|\mathcal{S}|\,|\mathcal{A}|\,\mathtt{T}}{\delta}$ for each $\mathtt{R}_\mathtt{t}(\mathtt{s}, \mathtt{a})$ and $\mathtt{R}_\mathtt{T}(\mathtt{s})$. Given the sample, we compute $\widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s}, \mathtt{a})$ and $\widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s})$. By Claim 8.1, with probability $1-\delta$ we have $|\mathtt{r}_\mathtt{t}(\mathtt{s}, \mathtt{a}) - \widehat{\mathtt{r}}_(\mathtt{s}, \mathtt{a})| \leq \varepsilon$ and $|\mathtt{r}_\mathtt{T}(\mathtt{s}) - \widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s})| \leq \varepsilon$. Now we can compute the optimal policy $\widehat{\pi}^*$ for the estimated rewards $\widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s}, \mathtt{a})$ and $\widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s})$. The main goal is to show that $\widehat{\pi}^*$ is a near optimal policy.

Theorem 8.2. *Assume that for every* $(\mathtt{s}, \mathtt{a})$ *and* $\mathtt{t}$ *we have* $|\mathtt{r}_\mathtt{t}(\mathtt{s}, \mathtt{a}) - \widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s}, \mathtt{a})| \leq \varepsilon$ *and for every* $\mathtt{s}$ *we have* $|\mathtt{r}_\mathtt{T}(\mathtt{s}) - \widehat{\mathtt{r}}_\mathtt{T}(\mathtt{s})| \leq \varepsilon$. *Then,*

$$\mathcal{V}_\mathtt{T}^{\pi^*}(\mathtt{s}_0) - \mathcal{V}_\mathtt{T}^{\widehat{\pi}^*} \leq 2\varepsilon(\mathtt{T}+1)$$

Proof. By Lemma 8.2 we have that $error(\pi) \leq \varepsilon(\mathtt{T}+1)$. This implies that,

$$\mathcal{V}_\mathtt{T}^{\pi^*}(\mathtt{s}_0) - \widehat{\mathcal{V}}_\mathtt{T}^{\pi^*}(\mathtt{s}_0) \leq error(\pi^*) \leq \varepsilon(\mathtt{T}+1)$$

and

$$\widehat{\mathcal{V}}_\mathtt{T}^{\widehat{\pi}^*}(\mathtt{s}_0) - \mathcal{V}_\mathtt{T}^{\widehat{\pi}^*}(\mathtt{s}_0) \leq error(\widehat{\pi}^*) \leq \varepsilon(\mathtt{T}+1)$$

Since $\widehat{\pi}^*$ is optimal for $\widehat{\mathtt{r}}_\mathtt{t}$ we have

$$\widehat{\mathcal{V}}_\mathtt{T}^{\pi^*}(\mathtt{s}_0) \leq \widehat{\mathcal{V}}_\mathtt{T}^{\widehat{\pi}^*}(\mathtt{s}_0)$$

The theorem follows by adding the three inequalities. □

Influence of reward estimation errors: discounted return

Fix a policy $\pi \in SD$. Again, define the expected return of π with the true rewards

$$\mathcal{V}_\gamma^\pi(\mathtt{s}_0) = \mathbb{E}^{\pi,\mathtt{s}_0}[\sum_{\mathtt{t}=0}^{\infty} \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})\gamma^\mathtt{t}]$$

and with the estimated rewards

$$\widehat{\mathcal{V}}_\gamma^\pi(\mathtt{s}_0) = \mathbb{E}^{\pi,\mathtt{s}_0}[\sum_{\mathtt{t}=0}^{\infty} \widehat{\mathtt{r}}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})\gamma^\mathtt{t}]$$

We are interested in bounding the difference between the two

$$\mathit{error}(\pi) = |\mathcal{V}_\gamma^\pi(\mathtt{s}_0) - \widehat{\mathcal{V}}_\gamma^\pi(\mathtt{s}_0)|$$

For a given trajectory $\sigma = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{\mathtt{T}-1}, \mathtt{a}_{\mathtt{T}-1}, \mathtt{s}_\mathtt{T})$ we define

$$\mathit{error}(\pi, \sigma) = \sum_{\mathtt{t}=0}^{\infty} \gamma^\mathtt{t} \mathtt{r}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) - \sum_{\mathtt{t}=0}^{\infty} \gamma^\mathtt{t} \hat{\mathtt{r}}_\mathtt{t}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$$

Lemma 8.3. *Assume that for every* $(\mathtt{s}, \mathtt{a})$ *we have* $|\mathtt{r}_\mathtt{t}(\mathtt{s}, \mathtt{a}) - \widehat{\mathtt{r}}_\mathtt{t}(\mathtt{s}, \mathtt{a})| \leq \varepsilon$. *Then, for any policy* $\pi \in SD$ *we have* $\mathit{error}(\pi) \leq \frac{\varepsilon}{1-\gamma}$.

Proof. The probability of each trajectory $\sigma = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{\mathtt{T}-1}, \mathtt{a}_{\mathtt{T}-1}, \mathtt{s}_\mathtt{T})$ is the same under $\mathtt{r}(\mathtt{s}, \mathtt{a})$ and $\widehat{\mathtt{r}}(\mathtt{s}, \mathtt{a})$, since $\pi \in SD$. For each trajectory $\sigma = (\mathtt{s}_0, \mathtt{a}_0, \ldots, \mathtt{s}_{\mathtt{T}-1}, \mathtt{a}_{\mathtt{T}-1}, \mathtt{s}_\mathtt{T})$, we have,

$$\begin{aligned}
\mathit{error}(\pi, \sigma) &= \left| \sum_{\mathtt{t}=0}^{\infty} \mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})\gamma^\mathtt{t} - \sum_{\mathtt{t}=0}^{\infty} \widehat{\mathtt{r}}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})\gamma^\mathtt{t} \right| \\
&= \left| \sum_{\mathtt{t}=0}^{\infty} \left(\mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) - \widehat{\mathtt{r}}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})\right) \gamma^\mathtt{t} \right| \\
&\leq \sum_{\mathtt{t}=0}^{\infty} |\mathtt{r}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) - \widehat{\mathtt{r}}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})| \, \gamma^\mathtt{t} \\
&\leq \frac{\varepsilon}{1-\gamma}
\end{aligned}$$

The lemma follows since $\mathit{error}(\pi) = |E^{\pi,\mathtt{s}_0}[\mathit{error}(\pi, \sigma)]| \leq \frac{\varepsilon}{1-\gamma}$. □

Computing approximate optimal policy: discounted return

We can now describe how to compute a near optimal policy for the discounted return. We need a sample of size $m \geq \frac{\mathtt{R}_{\max}}{2\varepsilon^2} \log \frac{2|\mathcal{S}|\,|\mathcal{A}|}{\delta}$ for each $\mathtt{R}(\mathtt{s},\mathtt{a})$. Given the sample, we compute $\widehat{\mathtt{r}}(\mathtt{s},\mathtt{a})$. As we saw, with probability $1-\delta$ we have for every $(\mathtt{s},\mathtt{a})$ that $|\mathtt{r}(\mathtt{s},\mathtt{a}) - \widehat{\mathtt{r}}(\mathtt{s},\mathtt{a})| \leq \varepsilon$. Now we can compute the policy $\widehat{\pi}^*$ for the estimated rewards $\widehat{\mathtt{r}}_{\mathtt{t}}(\mathtt{s},\mathtt{a})$. Again, the main goal is to show that $\widehat{\pi}^*$ is a near optimal policy.

Theorem 8.3. *Assume that for every* $(\mathtt{s},\mathtt{a})$ *we have* $|\mathtt{r}(\mathtt{s},\mathtt{a}) - \widehat{\mathtt{r}}(\mathtt{s},\mathtt{a})| \leq \varepsilon$. *Then,*

$$\mathcal{V}_\gamma^{\pi^*}(\mathtt{s}_0) - \mathcal{V}_\gamma^{\widehat{\pi}^*} \leq \frac{2\varepsilon}{1-\gamma}$$

Proof. From the definition of $error(\pi)$ and since for any $\pi \in SD$ we showed that $error(\pi) \leq \frac{\varepsilon}{1-\gamma}$, we have,

$$\mathcal{V}_\gamma^{\pi^*}(\mathtt{s}_0) - \widehat{\mathcal{V}}_\gamma^{\pi^*}(\mathtt{s}_0) \leq error(\pi^*) \leq \frac{\varepsilon}{1-\gamma}$$

and

$$\widehat{\mathcal{V}}_\gamma^{\widehat{\pi}^*}(\mathtt{s}_0) - \mathcal{V}_\gamma^{\widehat{\pi}^*}(\mathtt{s}_0) \leq error(\widehat{\pi}^*) \leq \frac{\varepsilon}{1-\gamma}$$

Since $\widehat{\pi}^*$ is optimal for $\widehat{\mathtt{r}}$ we have

$$\widehat{\mathcal{V}}_\gamma^{\pi^*}(\mathtt{s}_0) \leq \widehat{\mathcal{V}}_\gamma^{\widehat{\pi}^*}(\mathtt{s}_0)$$

The claim follows by adding the three inequalities. □

8.2.3 Estimating the transition probabilities

We now estimate the transition probabilities. Again, we will look at the observed model. Namely, for a given state-action $(\mathtt{s},\mathtt{a})$, we consider the transitions $(\mathtt{s},\mathtt{a},\mathtt{s}'_i)$, for $1 \leq i \leq m$. We define the observed transition distribution,

$$\widehat{p}(\mathtt{s}'|\mathtt{s},\mathtt{a}) = \frac{|\{i : \mathtt{s}'_i = \mathtt{s}'\}|}{m}$$

Our main goal would be to evaluate the observed model as a function of the sample size m.

We start by considering two Markov chains which differ by at most α for each transition. Namely, we have Markov chains M_1 and M_2, where $M_1[i,j] = \Pr[i \to j] = p_1(j|i)$ and $M_2[i,j] = \Pr[i \to j] = p_2(j|i)$. We assume that for every i,j we have

$|M_1[i,j] - M_2[i,j]| \leq \alpha$. We would like to relate α to the statistical distance between the state distributions generated by the two Markov chains. Clearly if $\alpha \approx 0$ then the probabilities will be almost identical, but we like to get a quantitative bound on the difference, which will allow us to derive the sample size m.

We start with a general well-known observation about distributions.

Theorem 8.4. *Let q_1 and q_2 be two distributions over $\mathcal{S}$. Let $f : \mathcal{S} \to [0, F_{max}]$. Then,*

$$|\mathbb{E}_{s\sim q_1}[f(\mathtt{s})] - \mathbb{E}_{s\sim q_2}[f(\mathtt{s})]| \leq F_{max}\|q_1 - q_2\|_1$$

where $\|q_1 - q_2\|_1 = \sum_{\mathtt{s}\in\mathcal{S}} |q_1(\mathtt{s}) - q_2(\mathtt{s})|$.

Proof.

$$\begin{aligned}|\mathbb{E}_{s\sim q_1}[f(\mathtt{s})] - \mathbb{E}_{s\sim q_2}[f(\mathtt{s})]| &= |\sum_{\mathtt{s}\in\mathcal{S}} f(\mathtt{s})q_1(\mathtt{s}) - \sum_{\mathtt{s}\in\mathcal{S}} f(\mathtt{s})q_2(\mathtt{s})| \\ &\leq \sum_{\mathtt{s}\in\mathcal{S}} f(\mathtt{s})|q_1(\mathtt{s}) - q_2(\mathtt{s})| \\ &\leq F_{max}\|q_1 - q_2\|_1\end{aligned}$$

□

Therefore, we would like to bound the L_1-norm between the state distributions generated by M_1 and M_2.

Theorem 8.5. *Let $q_1^{\mathtt{t}}$ and $q_2^{\mathtt{t}}$ be the distribution over states after trajectories of length $\mathtt{t}$ of M_1 and M_2, respectively. Then,*

$$\|q_1^{\mathtt{t}} - q_2^{\mathtt{t}}\|_1 \leq \alpha|\mathcal{S}|\mathtt{t}$$

Proof. Let p_0 be the distribution of the start state. Then $q_1^{\mathtt{t}} = x_0 M_1^{\mathtt{t}}$ and $q_2^{\mathtt{t}} = p_0^\top M_2^{\mathtt{t}}$. The proof is by induction on $\mathtt{t}$. Clearly, for $\mathtt{t} = 0$ we have $q_1^0 = q_2^0 = p_0$.

We start with a few basic facts about matrix norms. Let $\|M\|_{\infty,1} = \max_i \sum_j |M[i,j]|$. Then

$$\|zM\|_1 = \sum_j |\sum_i z[i]M[i,j]| \leq \sum_{i,j} |z[i]| \cdot |M[i,j]| \leq \sum_i |z[i]| \cdot \|M\|_\infty \leq \|z\|_1\|M\|_{\infty,1} \tag{8.1}$$

This implies the following two simple facts. First, let q be a distribution, i.e., $\|q\|_1 = 1$, and M a matrix with all the entries at most α, i.e., $|M[i,j]| \leq \alpha$ which implies $\|M\|_{\infty,1} \leq \alpha|\mathcal{S}|$. Then,

$$\|qM\|_1 \leq \|q\|_1\|M\|_{\infty,1} \leq \alpha|\mathcal{S}| \tag{8.2}$$

Second, let M be a row-stochastic matrix, implies that $\|M\|_{\infty,1} = 1$. Then,

$$\|zM\|_1 \leq \|z\|_1 \|M\|_{\infty,1} \leq \|z\|_1 \tag{8.3}$$

For the induction step, let $z^{\mathtt{t}} = q_1^{\mathtt{t}} - q_2^{\mathtt{t}}$. We have,

$$\begin{aligned}
\|q_1^{\mathtt{t}} - q_2^{\mathtt{t}}\|_1 &= \|p_0^\top M_1^{\mathtt{t}} - p_0^\top M_2^{\mathtt{t}}\|_1 \\
&= \|q_1^{\mathtt{t}-1} M_1 - (q_1^{\mathtt{t}-1} - z^{\mathtt{t}-1}) M_2\|_1 \\
&\leq \|q_1 (M_1 - M_2)\|_1 + \|z^{\mathtt{t}-1} M_2\|_1 \\
&\leq \alpha|\mathcal{S}| + \alpha|\mathcal{S}|(\mathtt{t}-1) = \alpha|\mathcal{S}|\mathtt{t}
\end{aligned}$$

where in the last inequality, for the first term we used the first fact and for the second term we used the second fact with the inductive claim. □

Approximate model and simulation lemma

We define an *α-approximate model* as follows. A model $\widehat{M}$ is an α-approximate model of M if for every state-action $(\mathtt{s}, \mathtt{a})$ we have: (1) $|\widehat{\mathtt{r}}(\mathtt{s}, \mathtt{a}) - \mathtt{r}(\mathtt{s}, \mathtt{a})| \leq \alpha$ and (2) for every $\mathtt{s}'$ we have $|\widehat{p}(\mathtt{s}'|\mathtt{s}, \mathtt{a}) - p(\mathtt{s}'|\mathtt{s}, \mathtt{a})| \leq \alpha$.

The following simulation lemma, for the finite horizon case, guarantees that approximate models have similar return.

Lemma 8.4. *Assume that model $\widehat{M}$ is an α-approximate model of M. For the finite horizon return, for any policy $\pi \in MD$, we have*

$$|\mathcal{V}_{\mathtt{T}}^{\pi}(\mathtt{s}_0; M) - \mathcal{V}_{\mathtt{T}}^{\pi}(\mathtt{s}_0; \widehat{M})| \leq \varepsilon$$

for $\alpha \leq \frac{\varepsilon}{\mathtt{R}_{\max}|\mathcal{S}|\mathtt{T}^2}$

Proof. By Theorem 8.5 the distance between the state distributions of M and $\widehat{M}$ at time $\mathtt{t}$ is bounded by $\alpha|\mathcal{S}|\mathtt{t}$. Since the maximum reward is $\mathtt{R}_{\max}$, by Theorem 8.4 the difference is bounded by $\sum_{\mathtt{t}=0}^{\mathtt{T}} \alpha|\mathcal{S}|\mathtt{t}\mathtt{R}_{\max} \leq \alpha|\mathcal{S}|\mathtt{T}^2\mathtt{R}_{\max}$. For $\alpha \leq \frac{\varepsilon}{\mathtt{R}_{\max}|\mathcal{S}|\mathtt{T}^2}$ it implies that the difference is at most ε. □

We now switch to the simulation lemma, for the discounted return case, which also guarantees that approximate models have similar return.

Lemma 8.5. *Assume that model $\widehat{M}$ is an α-approximate model of M. For the discounted return, for any policy $\pi \in MD$, we have*

$$|\mathcal{V}_{\gamma}^{\pi}(\mathtt{s}_0; M) - \mathcal{V}_{\gamma}^{\pi}(\mathtt{s}_0; \widehat{M})| \leq \varepsilon$$

for $\alpha \leq \frac{0.5\varepsilon(1-\gamma)^2}{\mathtt{R}_{\max}|\mathcal{S}|\log^2(\mathtt{R}_{\max}/(\varepsilon(1-\gamma)))}$. $c > 0$.

Proof. We want to move from the discounted setting to the finite horizon setting. By Theorem 8.1, a horizon $\mathtt{T} = \frac{1}{1-\gamma}\log\frac{\mathtt{R}_{\max}}{\varepsilon(1-\gamma)/2}$, guarantees that the error due to truncating at horizon $\mathtt{T}$ is at most $\varepsilon/2$. By Lemma 8.2, for a horizon $\mathtt{T}$ and $\alpha = \frac{\varepsilon/2}{\mathtt{R}_{\max}|\mathcal{S}|\mathtt{T}^2}$ we get and error at most $\varepsilon/2$. By substituting the bound for $\mathtt{T}$ in the expression for α we derive the theorem. □

putting it all together

We want with high probability $(1-\delta)$ to have accuracy α. To get accuracy α we need $m = O(\frac{1}{\alpha^2}\log(|\mathcal{S}|^2|\mathcal{A}|/\delta))$ samples for each state-action pair $(\mathtt{s},\mathtt{a})$. Plugging in the value of α we have, for the finite horizon,

$$m = O(\frac{\mathtt{R}_{\max}^2}{\varepsilon^2}|\mathcal{S}|^2\mathtt{T}^4\log(|\mathcal{S}|\ |\mathcal{A}|/\delta))$$

and for the discounted return

$$m = O(\frac{\mathtt{R}_{\max}^2}{\varepsilon^2}|\mathcal{S}|^2\frac{1}{(1-\gamma)^4}\log(|\mathcal{S}|\ |\mathcal{A}|/\delta)\log^2\frac{\mathtt{R}_{\max}}{\varepsilon(1-\gamma)})$$

Assume we have a sample of m for each $(\mathtt{s},\mathtt{a})$. Then with probability $1-\delta$ we have an α-approximate model $\widehat{M}$. We find an optimal policy $\widehat{\pi}^*$ for $\widehat{M}$. This implies that $\widehat{\pi}^*$ is a 2ε-optimal policy. Namely,

$$|\mathcal{V}^*(\mathtt{s}_0) - \mathcal{V}^{\widehat{\pi}^*}(\mathtt{s}_0)| \leq 2\varepsilon$$

We can now look on the dependency of the sampling bounds on the parameters.

1. The error scales like $\frac{\mathtt{R}_{\max}^2}{\varepsilon^2}$ which looks like the right bound, even for estimation of random variables expectations.

2. The dependency on the horizon is necessary, although it is probably not optimal. In [2] a sample bound of $O(\frac{|\mathcal{S}|^2\mathcal{A}|\mathtt{T}^2\mathtt{R}_{\max}^2}{\varepsilon^2}\log\frac{1}{\delta})$ is given.

3. The dependency on the number on the number of states $|\mathcal{S}|$, is due to the fact that we like a very high approximation of the next state distribution. Simply we need to approximate $|\mathcal{S}|^2$ entries, so for this the bound is reasonable. However, we will show that for approximation the optimal policy we can do better.

8.2.4 Improved sample bound: using approximate value iteration

We would like to exhibit a better sample complexity, for the very interesting case of deriving an approximately optimal policy. The construction and proof would use the Value Iteration algorithm. Recall, that the Value Iteration algorithm works as follows. Initially, we set the values arbitrarily,

$$V_0 = \{V_0(\mathtt{s})\}_{\mathtt{s}\in\mathcal{S}}$$

In iteration n we compute

$$\begin{aligned} V_{n+1}(\mathtt{s}) &= \max_{\mathtt{a}\in\mathcal{A}}\{\mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma\sum_{\mathtt{s}'\in S} p(\mathtt{s}'|\mathtt{s},\mathtt{a})V_n(\mathtt{s}')\} \\ &= \max_{\mathtt{a}\in\mathcal{A}}\{\mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma E_{\mathtt{s}'\sim p(\cdot|\mathtt{s},\mathtt{a})}[V_n(\mathtt{s}')]\} \end{aligned}$$

We showed that $\lim_{n\to\infty} V_n = V^*$, and that the convergence rate is $O(\frac{\gamma^n}{1-\gamma}\mathtt{R}_{\max})$. This implies that if we run for N, where $N = \frac{1}{1-\gamma}\log\frac{\mathtt{R}_{\max}}{\varepsilon(1-\gamma)}$, we have an error of at most ε.

We would like to approximate the Value Iteration algorithm using a sample. Namely, for each $(\mathtt{s},\mathtt{a})$ we have a sample of size m, i.e., $\{(\mathtt{s},\mathtt{a},\mathtt{s}'_i)\}_{i\in[1,m]}$ The Value Iteration using the sample would be,

$$\widehat{V}_{n+1}(\mathtt{s}) = \max_{\mathtt{a}\in\mathcal{A}}\{\mathtt{r}(\mathtt{s},\mathtt{a}) + \gamma\frac{1}{m}\sum_{i=1}^{m}\widehat{V}_n(\mathtt{s}'_i)\}$$

The intuition is that if we have a large enough sample, it will approximate the Value Iteration. We set m such that, with probability $-\delta$, for every $(\mathtt{s},\mathtt{a})$ and any iteration $n\in[1,N]$ we have:

$$|\mathbb{E}[\widehat{V}_n(\mathtt{s}')] - \frac{1}{m}\sum_{i=1}^{m}\widehat{V}_n(\mathtt{s}'_i)| \le \varepsilon$$

and also

$$|\widehat{\mathtt{r}}(\mathtt{s},\mathtt{a}) - \mathtt{r}(\mathtt{s},\mathtt{a})| \le \varepsilon$$

We can have this for $m = O(\frac{V_{max}^2}{\varepsilon^2}\log(1/\delta))$ where V_{max} bound the maximum value. I.e., for finite horizon $V_{max} = \mathtt{T}\,\mathtt{R}_{\max}$ and for discounted return $V_{max} = \frac{\mathtt{R}_{\max}}{1-\gamma}$.

Assume that we have, for every $\mathtt{s}\in\mathcal{S}$,

$$|\widehat{V}_n(\mathtt{s}) - V_n(\mathtt{s})| \le \lambda$$

Then

$$|\widehat{V}_{n+1}(\mathtt{s}) - V_{n+1}(\mathtt{s})| \leq \varepsilon + \gamma\lambda \leq \lambda$$

where the last inequality holds for $\lambda \geq \frac{\varepsilon}{1-\gamma}$.

Therefore, if we sample $m = O(\frac{1}{\varepsilon^2} \log \frac{N|\mathcal{S}|\,|\mathcal{A}|}{\delta})$, we have that with probability $1-\delta$ for every $(\mathtt{s}, \mathtt{a})$ the approximation is at most ε. This implies that the approximate Value Iteration has error at most $\lambda = \frac{\varepsilon}{1-\gamma}$.

The main result is that we can run Value Iteration algorithm for N iterations and approximate well the optimal value function and policy. The implicit drawback is that we are approximating only the optimal policy, and cannot evaluate an arbitrary policy.

8.3 On-Policy Learning

In the off-policy setting we where given some trajectories and used them to learn and model, and using it an approximately optimal policy. Essentially, we assumed that the trajectories are exploratory enough, in the sense that each $(\mathtt{s}, \mathtt{a})$ has a sufficient number of samples. In the on-line setting it is the responsibility of the learner to perform the exploration. This would be the main challenge in this section

8.3.1 Learning Deterministic Decision Process

Recall that a Deterministic Decision Process is modeled by a directed graph, where the states are the vertices, and each action is associate with an edge.

We first define the observed model. Given an observation set $\{(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}, \mathtt{r}_\mathtt{t}, \mathtt{s}_{\mathtt{t}+1})\}$, we define an observed model $\widehat{M}$, where $\widehat{f}(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) = \mathtt{s}_{\mathtt{t}+1}$ and $\widehat{\mathtt{r}}(\mathtt{s}, \mathtt{a}) = \mathtt{r}_\mathtt{t}$. For $(\mathtt{s}, \mathtt{a})$ which do not appear in the observation set, we define $\widehat{f}(\mathtt{s}, \mathtt{a}) = \mathtt{s}$ and $\widehat{\mathtt{r}}(\mathtt{s}, \mathtt{a}) = \mathtt{R}_{\max}$.

First we claim that for any $\pi \in SD$ the completion can only increase the value.

$$\widehat{V}^\pi(\mathtt{s}; \widehat{M}) \geq V^\pi(\mathtt{s}; M)$$

The increase holds for any trajectory, and note that once π reaches $(\mathtt{s}, \mathtt{a})$ that was not observed, its reward will be $\mathtt{R}_{\max}$ forever. (This is since $\pi \in SD$.)

We can now present the on-policy learning algorithm.

1. At time $\mathtt{T}$ let $\{(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}, \mathtt{r}_\mathtt{t}, \mathtt{s}_{\mathtt{t}+1}) : 0 \leq \mathtt{t} \leq \mathtt{T}-1\}$ be the previous observations.

2. Build the observed model $\widehat{M}_\mathtt{T}$.

3. Compute $\widehat{\pi}^*_{\mathtt{T}}$, the optimal policy for $\widehat{M}_{\mathtt{T}}$.

4. Use $\mathtt{a}_{\mathtt{T}} = \widehat{\pi}^*_{\mathtt{T}}(\mathtt{s}_{\mathtt{T}})$.

5. Observe the reward $\mathtt{r}_{\mathtt{T}}$ and the next state $\mathtt{s}_{\mathtt{T}+1}$.

We first show that the model $\widehat{M}_{\mathtt{T}}$ cannot change too often.

Lemma 8.6. *The observed model $\widehat{M}_{\mathtt{T}}$ can change at most $|\mathcal{S}|\ |\mathcal{A}|$ times.*

Proof. Each time we change the observed model, we add one $(\mathtt{s}, \mathtt{a})$ to be known. Since there are $|\mathcal{S}|\ |\mathcal{A}|$ such pairs, this bounds the number of changes. □

Next we show that we either make progress in the next $|\mathcal{S}|$ steps or we never make any more changes.

Lemma 8.7. *Either we change $\widehat{M}_{\mathtt{T}}$ in some time $\mathtt{t} \in [\mathtt{T}, \mathtt{T} + |\mathcal{S}|]$ or we never change $\widehat{M}_{\mathtt{T}}$.*

Proof. The model is deterministic. If we do not make any update in the next $|\mathcal{S}|$ steps, the policy $\widehat{\pi}^*_{\mathtt{T}} \in MD$ will close a cycle and continue on this cycle forever. Hence, the model will never change. □

We can now state the convergence of the algorithm to the optimal policy.

Theorem 8.6. *After $\mathtt{T} = |\mathcal{S}|^2|\mathcal{A}|$ time steps $\widehat{\pi}^*_{\mathtt{T}}$ is optimal.*

Proof. We showed that the number of changes is at most $|\mathcal{S}|\ |\mathcal{A}|$, and the time between changes is at most $|\mathcal{S}|$. This implies that after time $\mathtt{T} = |\mathcal{S}|^2|\mathcal{A}|$ we never change.

The return of $\widehat{\pi}^*_{\mathtt{T}}$ after time $\mathtt{T}$ is identical in $\widehat{M}_{\mathtt{T}}$ and M, since all the edges it traverses are known. Let $\widehat{V}^*$ be its return. Assume that the policy π^* has a strictly higher return in M, say $V^* > \widehat{V}^*$. This implies that the return of π^* is at least $V^* > \widehat{V}^*$ in $\widehat{M}_{\mathtt{T}}$, since the rewards in $\widehat{M}_{\mathtt{T}}$ are always at least those in M. This contradicts the fact that $\widehat{\pi}^*_{\mathtt{T}}$ is optimal for $\widehat{M}_{\mathtt{T}}$. □

8.3.2 On-policy learning MDP

Similar to the DDP, we will use the principle of *Optimism in face of uncertainty*. Namely, we substitute the unknown quantities by the maximum possible values.

Similar to DDP, we will partition the state-action pairs $(\mathtt{s}, \mathtt{a})$ `known` and `unknown`. The main difference is that in a DDP it is sufficient to have a single sample to move

$(\mathtt{s},\mathtt{a})$ from `unknown` to `known`. In a general MDP we need have a larger sample to move $(\mathtt{s},\mathtt{a})$ from `unknown` to `known`. Otherwise, the algorithm would be very similar.

We describe algorithm `R-MAX`, which performs on-policy learning of MDPs.

Initialization: Initially we set for each $(\mathtt{s},\mathtt{a})$ a next state distribution which always return to $\mathtt{s}$, i.e., $p(\mathtt{s}|\mathtt{s},\mathtt{a}) = 1$ and $p(\mathtt{s}'|\mathtt{s},\mathtt{a}) = 0$ for $\mathtt{s}' \neq s$. We set the reward to be maximal, i.e., $\mathtt{r}(\mathtt{s},\mathtt{a}) = \mathtt{R}_{\max}$. We mark $(\mathtt{s},\mathtt{a})$ to be `unknown`.

Execution: At time $\mathtt{t}$. (1) Build a model $\widehat{M}_{\mathtt{t}}$, explained later. (2) Compute $\widehat{\pi}^*_{\mathtt{t}}$ the optimal policy for $\widehat{M}_{\mathtt{t}}$, and (3) Execute $\mathtt{a}_{\mathtt{t}} = \widehat{\pi}^*_{\mathtt{t}}(\mathtt{s}_{\mathtt{t}})$ and observe $\mathtt{r}_{\mathtt{t}}$ and $\mathtt{s}_{\mathtt{t}+1}$.

Building a model: At time $\mathtt{t}$, if the number of samples of $(\mathtt{s},\mathtt{a})$ is *exactly* m for the *first time*, then: modify $p(\cdot|\mathtt{s},\mathtt{a})$ to the observed transition distribution, $\mathtt{r}(\mathtt{s},\mathtt{a})$ to the average observed reward for $(\mathtt{s},\mathtt{a})$ and mark $(\mathtt{s},\mathtt{a})$ as `known`. Note that we update each $(\mathtt{s},\mathtt{a})$ only once, when it moves from `unknown` to `known`.

Here is the basic intuition for algorithm `R-MAX`. We consider the discounted return. Fix a large enough horizon $\mathtt{T}$, at least the effective horizon. Assume we run $\widehat{\pi}^*_{\mathtt{t}}$ for $\mathtt{T}$ time steps. Either, we explore a state-action $(\mathtt{s},\mathtt{a})$ which is `unknown`, in this case we make progress on the exploration. Also this can happen at most $m|\mathcal{S}|\ |\mathcal{A}|$ times. Alternatively, we did not reach any state-action $(\mathtt{s},\mathtt{a})$ which is `unknown`, in which case we are optimal on the observed model, and near optimal on the true model.

For the analysis consider the event W, which is that event that we visit a state-action $(\mathtt{s},\mathtt{a})$ which is `unknown` during the block of $\mathtt{T}$ time steps. For the return of $\widehat{\pi}^*_{\mathtt{t}}$ we have,

$$V^{\widehat{\pi}^*_{\mathtt{t}}} \geq V^* - \Pr[W]\frac{\mathtt{R}_{\max}}{1-\gamma} - \lambda$$

where λ is the approximation error, and we can bound it by $\varepsilon/2$ by setting m large enough.

We consider two cases, depending on the probability of W. First, we consider the case that the probability of W is small. If $\Pr[W] \leq \frac{\varepsilon(1-\gamma)/2}{\mathtt{R}_{\max}}$, then $V^{\widehat{\pi}^*_{\mathtt{t}}} \geq V^* - \varepsilon/2 - \varepsilon/2$, since we assume that $\lambda \leq \varepsilon/2$.

Second, we consider the case that the probability of W is large. If $\Pr[W] > \frac{\varepsilon(1-\gamma)/2}{\mathtt{R}_{\max}}$. Then, there is a good probability to visit an `unknown` $(\mathtt{s},\mathtt{a})$, but this can happen at most $m|\mathcal{S}|\ |\mathcal{A}|$. Therefore, the expected number of such blocks is at most $m|\mathcal{S}|\ |\mathcal{A}|\frac{\mathtt{R}_{\max}}{\varepsilon(1-\gamma)/2}$.

This implies that only in

$$m|\mathcal{S}|\ |\mathcal{A}|\frac{\mathtt{R}_{\max}}{\varepsilon(1-\gamma)/2}$$

blocks, the algorithm `R-MAX` will be more than ε sub-optimal, i.e., have an expected return less than $V^* - \varepsilon$.

8.4 Bibliography Remarks

The first polynomial time model based learning algorithm is E^3 of [4]. While we did not outline the E^3 algorithm, we did describe the effective horizon and the simulation lemma from there.

The improved sampling bounds for the optimal policy using approximate Value Iteration is following [3].

The `R-MAX` algorithm is due to [1].

Analysis of related models, especially the `PAC-MDP` model appears in [6, 5].

Bibliography

[1] Ronen I. Brafman and Moshe Tennenholtz. R-MAX - A general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2002.

[2] Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2818–2826, 2015.

[3] Michael J. Kearns and Satinder P. Singh. Finite-sample convergence rates for q-learning and indirect algorithms. In *Advances in Neural Information Processing Systems 11, [NIPS Conference, Denver, Colorado, USA, November 30 - December 5, 1998]*, pages 996–1002, 1998.

[4] Michael J. Kearns and Satinder P. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

[5] Lihong Li. Sample complexity bounds of exploration. In Marco Wiering and Martijn van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, pages 175–204. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[6] Alexander L. Strehl, Lihong Li, and Michael L. Littman. Reinforcement learning in finite mdps: PAC analysis. *Journal of Machine Learning Research*, 10:2413–2444, 2009.

Chapter 9

Model free learning: Tabular case

Chapter 10

Tabular learning: off-policy

Chapter 11

Large state space: value function approximation

Chapter 12

Large state space: value function approximation: Policy Gradient Methods

Chapter 13

Large state space: tree based optimization

Chapter 14

Deep RL

Chapter 15

Multi-Arm bandits

Chapter 16

POMDP

Appendix A

Linear Programming Solutions: Background

A Linear Program (LP) is an optimization problem that involves minimizing (or maximizing) a linear objective function subject to linear constraints. A standard form of a LP is

$$\text{minimize } b^\top x, \quad \text{subject to } Ax \geq c, x \geq 0. \tag{A.1}$$

where $x = (x_1, x_2, \ldots, x_n)^\top$ is a vector of real variables arranged as a column vector. The set of constraints is linear and defines a *convex polytope* in $\mathbb{R}^n$, namely a closed and convex set U that is the intersection of a finite number of half-spaces. U has a finite number of vertices, which are points that cannot be generated as a convex combination of other points in U. If U is bounded, it equals the convex combination of its vertices. It can be seen that an optimal solution (if finite) will be in one of these vertices.

The LP problem has been extensively studied, and many efficient solvers exist. In 1947, Danzig introduced the Simplex algorithm, which essentially moves greedily along neighboring vertices. In the 1980's effective algorithms (interior point and others) were introduced which had polynomial time guarantees.

Duality: The *dual* of the LP in (A.1) is defined as the following LP:

$$\text{maximize } c^\top y, \quad \text{subject to } A^\top y \leq b, y \geq 0. \tag{A.2}$$

The two dual LPs have the same optimal value, and (in many cases) the solution of one can be obtained from that of the other. The common optimal value can be understood by the following computation:

$$\min_{x\geq 0, Ax\geq c} b^\top x = \min_{x\geq 0}\max_{y\geq 0}\left\{b^\top x + y^\top(c-Ax)\right\}$$
$$= \max_{y\geq 0}\min_{x\geq 0}\left\{c^\top y + x^\top(b-Ay)\right\} = \max_{y\geq 0, Ay\leq b} c^\top x.$$

where the second equality follows by the min-max theorem.

Note: For an LP of the form:

$$\text{minimize } b^\top x, \quad \text{subject to } Ax \geq c,$$

the dual is

$$\text{maximize } c^\top y, \quad \text{subject to } A^\top y = b, y \geq 0.$$