# Reinforcement Learning: Theory and Algorithms

Alekh Agarwal    Nan Jiang    Sham M. Kakade    Wen Sun

September 14, 2021

<span style="color:red">WORKING DRAFT:</span>

We will be frequently updating the book this fall, 2021. Please email
bookrltheory@gmail.com with any typos or errors you find.
We appreciate it!

# Contents

# 2  Strategic Exploration     55

# Notation

The reader might find it helpful to refer back to this notation section.

- We slightly abuse notation and let $[K]$ denote the set $\{0, 1, 2, \ldots K - 1\}$ for an integer $K$.

- For a vector $v$, we let $(v)^2$, $\sqrt{v}$, and $|v|$ be the component-wise square, square root, and absolute value operations.

- Inequalities between vectors are elementwise, e.g. for vectors $v, v'$, we say $v \leq v'$, if the inequality holds elementwise.

- For a vector $v$, we refer to the $j$-th component of this vector by either $v(j)$ or $[v]_j$

- Denote the variance of any real valued $f$ under a distribution $\mathcal{D}$ as:

$$\text{Var}_{\mathcal{D}}(f) := E_{x \sim \mathcal{D}}[f(x)^2] - (E_{x \sim \mathcal{D}}[f(x)])^2$$

- We overload notation where, for a distribution $\mu$ over $\mathcal{S}$, we write:

$$V^\pi(\mu) = \mathbb{E}_{s \sim \mu}\left[V^\pi(s)\right].$$

- It is helpful to overload notation and let $P$ also refer to a matrix of size $(\mathcal{S} \cdot \mathcal{A}) \times \mathcal{S}$ where the entry $P_{(s,a),s'}$ is equal to $P(s'|s, a)$. We also will define $P^\pi$ to be the transition matrix on state-action pairs induced by a deterministic policy $\pi$. In particular, $P^\pi_{(s,a),(s',a')} = P(s'|s, a)$ if $a' = \pi(s')$ and $P^\pi_{(s,a),(s',a')} = 0$ if $a' \neq \pi(s')$. With this notation,

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

- For a vector $Q \in \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|}$, denote the greedy policy and value as:

$$\pi_Q(s) \quad := \quad \text{argmax}_{a \in \mathcal{A}} Q(s, a)$$
$$V_Q(s) \quad := \quad \max_{a \in \mathcal{A}} Q(s, a)..$$

- For a vector $Q \in \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|}$, the *Bellman optimality operator* $\mathcal{T} : \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|} \to \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|}$ is defined as:

$$\mathcal{T}Q := r + PV_Q. \tag{0.1}$$

# Part 1

**Fundamentals**

# Chapter 1

# Markov Decision Processes

## 1.1 Discounted (Infinite-Horizon) Markov Decision Processes

In reinforcement learning, the interactions between the agent and the environment are often described by an infinite-horizon, discounted Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu)$, specified by:

- A state space $\mathcal{S}$, which may be finite or infinite. For mathematical convenience, we will assume that $\mathcal{S}$ is finite or countable infinite.

- An action space $\mathcal{A}$, which also may be discrete or infinite. For mathematical convenience, we will assume that $\mathcal{A}$ is finite.

- A transition function $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ is the space of probability distributions over $\mathcal{S}$ (i.e., the probability simplex). $P(s'|s, a)$ is the probability of transitioning into state $s'$ upon taking action $a$ in state $s$. We use $P_{s,a}$ to denote the vector $P(\cdot \mid s, a)$.

- A reward function $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$. $r(s, a)$ is the immediate reward associated with taking action $a$ in state $s$. More generally, the $r(s, a)$ could be a random variable (where the distribution depends on $s, a$). While we largely focus on the case where $r(s, a)$ is deterministic, the extension to methods with stochastic rewards are often straightforward.

- A discount factor $\gamma \in [0, 1)$, which defines a horizon for the problem.

- An initial state distribution $\mu \in \Delta(\mathcal{S})$, which specifies how the initial state $s_0$ is generated.

In many cases, we will assume that the initial state is fixed at $s_0$, i.e. $\mu$ is a distribution supported only on $s_0$.

### 1.1.1 The objective, policies, and values

**Policies.** In a given MDP $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu)$, the agent interacts with the environment according to the following protocol: the agent starts at some state $s_0 \sim \mu$; at each time step $t = 0, 1, 2, \ldots$, the agent takes an action $a_t \in \mathcal{A}$, obtains the immediate reward $r_t = r(s_t, a_t)$, and observes the next state $s_{t+1}$ sampled according to $s_{t+1} \sim P(\cdot|s_t, a_t)$. The interaction record at time $t$,

$$\tau_t = (s_0, a_0, r_0, s_1, \ldots, s_t, a_t, r_t),$$

is called a *trajectory*, which includes the observed state at time $t$.

In the most general setting, a policy specifies a decision-making strategy in which the agent chooses actions adaptively based on the history of observations; precisely, a policy is a (possibly randomized) mapping from a trajectory to an action, i.e. $\pi : \mathcal{H} \to \Delta(\mathcal{A})$ where $\mathcal{H}$ is the set of all possible trajectories (of all lengths) and $\Delta(\mathcal{A})$ is the space of probability distributions over $\mathcal{A}$. A *stationary* policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ specifies a decision-making strategy in which the agent chooses actions based only on the current state, i.e. $a_t \sim \pi(\cdot|s_t)$. A deterministic, stationary policy is of the form $\pi : \mathcal{S} \to \mathcal{A}$.

**Values.** We now define values for (general) policies. For a fixed policy and a starting state $s_0 = s$, we define the value function $V_M^\pi : \mathcal{S} \to \mathbb{R}$ as the discounted sum of future rewards

$$V_M^\pi(s) = \mathbb{E}\Big[ \sum_{t=0}^\infty \gamma^t r(s_t, a_t) \,\big|\, \pi, s_0 = s \Big].$$

where expectation is with respect to the randomness of the trajectory, that is, the randomness in state transitions and the stochasticity of $\pi$. Here, since $r(s, a)$ is bounded between 0 and 1, we have $0 \le V_M^\pi(s) \le 1/(1 - \gamma)$.

Similarly, the action-value (or Q-value) function $Q_M^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as

$$Q_M^\pi(s, a) = \mathbb{E}\Big[ \sum_{t=0}^\infty \gamma^t r(s_t, a_t) \,\big|\, \pi, s_0 = s, a_0 = a \Big].$$

and $Q_M^\pi(s, a)$ is also bounded by $1/(1 - \gamma)$.

**Goal.** Given a state $s$, the goal of the agent is to find a policy $\pi$ that maximizes the value, i.e. the optimization problem the agent seeks to solve is:

$$\max_\pi V_M^\pi(s) \tag{0.1}$$

where the max is over all (possibly non-stationary and randomized) policies. As we shall see, there exists a deterministic and stationary policy which is simultaneously optimal for all starting states $s$.

We drop the dependence on $M$ and write $V^\pi$ when it is clear from context.

**Example 1.1** (Navigation). Navigation is perhaps the simplest to see example of RL. The state of the agent is their current location. The four actions might be moving 1 step along each of east, west, north or south. The transitions in the simplest setting are deterministic. Taking the north action moves the agent one step north of their location, assuming that the size of a step is standardized. The agent might have a goal state $g$ they are trying to reach, and the reward is 0 until the agent reaches the goal, and 1 upon reaching the goal state. Since the discount factor $\gamma < 1$, there is incentive to reach the goal state earlier in the trajectory. As a result, the optimal behavior in this setting corresponds to finding the shortest path from the initial to the goal state, and the value function of a state, given a policy is $\gamma^d$, where $d$ is the number of steps required by the policy to reach the goal state.

**Example 1.2** (Conversational agent). This is another fairly natural RL problem. The state of an agent can be the current transcript of the conversation so far, along with any additional information about the world, such as the context for the conversation, characteristics of the other agents or humans in the conversation etc. Actions depend on the domain. In the most basic form, we can think of it as the next statement to make in the conversation. Sometimes, conversational agents are designed for task completion, such as travel assistant or tech support or a virtual office receptionist. In these cases, there might be a predefined set of *slots* which the agent needs to fill before they can find a good solution. For instance, in the travel agent case, these might correspond to the dates, source, destination and mode of travel. The actions might correspond to natural language queries to fill these slots.

In task completion settings, reward is naturally defined as a binary outcome on whether the task was completed or not, such as whether the travel was successfully booked or not. Depending on the domain, we could further refine it based on the quality or the price of the travel package found. In more generic conversational settings, the ultimate reward is whether the conversation was satisfactory to the other agents or humans, or not.

**Example 1.3** (Strategic games). This is a popular category of RL applications, where RL has been successful in achieving human level performance in Backgammon, Go, Chess, and various forms of Poker. The usual setting consists of the state being the current game board, actions being the potential next moves and reward being the eventual win/loss outcome or a more detailed score when it is defined in the game. Technically, these are multi-agent RL settings, and, yet, the algorithms used are often non-multi-agent RL algorithms.

### 1.1.2 Bellman consistency equations for stationary policies

Stationary policies satisfy the following consistency conditions:

**Lemma 1.4.** *Suppose that $\pi$ is a stationary policy. Then $V^\pi$ and $Q^\pi$ satisfy the following* Bellman consistency equations*: for all $s \in \mathcal{S}, a \in \mathcal{A}$,*

$$V^\pi(s) = Q^\pi(s, \pi(s)).$$
$$Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[V^\pi(s')\right].$$

We leave the proof as an exercise to the reader.

It is helpful to view $V^\pi$ as vector of length $|\mathcal{S}|$ and $Q^\pi$ and $r$ as vectors of length $|\mathcal{S}| \cdot |\mathcal{A}|$. We overload notation and let $P$ also refer to a matrix of size $(|\mathcal{S}| \cdot |\mathcal{A}|) \times |\mathcal{S}|$ where the entry $P_{(s,a),s'}$ is equal to $P(s'|s,a)$.

We also will define $P^\pi$ to be the transition matrix on state-action pairs induced by a stationary policy $\pi$, specifically:

$$P^\pi_{(s,a),(s',a')} := P(s'|s,a)\pi(a'|s').$$

In particular, for deterministic policies we have:

$$P^\pi_{(s,a),(s',a')} := \begin{cases} P(s'|s,a) & \text{if } a' = \pi(s') \\ 0 & \text{if } a' \neq \pi(s') \end{cases}$$

With this notation, it is straightforward to verify:

$$Q^\pi = r + \gamma P V^\pi$$
$$Q^\pi = r + \gamma P^\pi Q^\pi.$$

**Corollary 1.5.** *We have that:*

$$Q^\pi = (I - \gamma P^\pi)^{-1} r \tag{0.2}$$

*where $I$ is the identity matrix.*

**Proof:** To see that the $I - \gamma P^\pi$ is invertible, observe that for any non-zero vector $x \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$,

$$
\begin{aligned}
\|(I - \gamma P^\pi)x\|_\infty &= \|x - \gamma P^\pi x\|_\infty \\
&\geq \|x\|_\infty - \gamma \|P^\pi x\|_\infty & \text{(triangle inequality for norms)} \\
&\geq \|x\|_\infty - \gamma \|x\|_\infty & \text{(each element of } P^\pi x \text{ is an average of } x) \\
&= (1 - \gamma)\|x\|_\infty > 0 & (\gamma < 1, x \neq 0)
\end{aligned}
$$

which implies $I - \gamma P^\pi$ is full rank. ∎

The following is also a helpful lemma:

**Lemma 1.6.** *We have that:*

$$[(1-\gamma)(I-\gamma P^{\pi})^{-1}]_{(s,a),(s',a')} = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathbb{P}^{\pi}(s_t = s', a_t = a'|s_0 = s, a_0 = a)$$

*so we can view the $(s, a)$-th row of this matrix as an induced distribution over states and actions when following $\pi$ after starting with $s_0 = s$ and $a_0 = a$.*

We leave the proof as an exercise to the reader.

### 1.1.3 Bellman optimality equations

A remarkable and convenient property of MDPs is that there exists a stationary and deterministic policy that simultaneously maximizes $V^{\pi}(s)$ for all $s \in \mathcal{S}$. This is formalized in the following theorem:

**Theorem 1.7.** *Let $\Pi$ be the set of all non-stationary and randomized policies. Define:*

$$V^{\star}(s) := \sup_{\pi \in \Pi} V^{\pi}(s)$$
$$Q^{\star}(s, a) := \sup_{\pi \in \Pi} Q^{\pi}(s, a).$$

*which is finite since $V^{\pi}(s)$ and $Q^{\pi}(s, a)$ are bounded between $0$ and $1/(1-\gamma)$.*

*There exists a stationary and deterministic policy $\pi$ such that for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$,*

$$V^{\pi}(s) = V^{\star}(s)$$
$$Q^{\pi}(s, a) = Q^{\star}(s, a).$$

*We refer to such a $\pi$ as an optimal policy.*

**Proof:** First, let us show that conditioned on $(s_0, a_0, r_0, s_1) = (s, a, r, s')$, the maximum future discounted value, from time 1 onwards, is not a function of $s, a, r$. Specifically,

$$\sup_{\pi \in \Pi} \mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^t r(s_t, a_t) \mid \pi, (s_0, a_0, r_0, s_1) = (s, a, r, s')\Big] = \gamma V^{\star}(s')$$

For any policy $\pi$, define an "offset" policy $\pi_{(s,a,r)}$, which is the policy that chooses actions on a trajectory $\tau$ according to the same distribution that $\pi$ chooses actions on the trajectory $(s, a, r, \tau)$. For example, $\pi_{(s,a,r)}(a_0 = a'|s_0 = s')$ is equal to the probability $\pi(a_1 = a'|(s_0, a_0, r_0, s_1) = (s, a, r, s'))$. By the Markov property, we have that:

$$\mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^t r(s_t, a_t) \mid \pi, (s_0, a_0, r_0, s_1) = (s, a, r, s')\Big] = \gamma\mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t) \mid \pi_{(s,a,r)}, s_0 = s'\Big] = \gamma V^{\pi_{(s,a,r)}}(s').$$

Hence, due to that $V^{\pi}(s')$ is not a function of $(s, a, r)$, we have

$$\sup_{\pi \in \Pi}\mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^t r(s_t, a_t) \mid \pi, (s_0, a_0, r_0, s_1) = (s, a, r, s')\Big] = \gamma \cdot \sup_{\pi \in \Pi} V^{\pi_{(s,a,r)}}(s') = \gamma \cdot \sup_{\pi \in \Pi} V^{\pi}(s') = \gamma V^{\star}(s'),$$

thus proving the claim.

We now show the deterministic and stationary policy $\pi(s) = \operatorname{argmax}_{a \in \mathcal{A}} \sup_{\pi' \in \Pi} Q^{\pi'}(s,a)$ satisfies $V^\pi(s) = \sup_{\pi' \in \Pi} V^{\pi'}(s)$. For this, we have that:

$$
\begin{aligned}
V^\star(s_0) &= \sup_{\pi \in \Pi} \mathbb{E}\Big[r(s_0, a_0) + \sum_{t=1}^\infty \gamma^t r(s_t, a_t)\Big] \\
&= \sup_{\pi \in \Pi} \mathbb{E}\Big[r(s_0, a_0) + \mathbb{E}\Big[\sum_{t=1}^\infty \gamma^t r(s_t, a_t) \mid \pi, (s_0, a_0, r_0, s_1)\Big]\Big] \\
&\leq \sup_{\pi \in \Pi} \mathbb{E}\Big[r(s_0, a_0) + \sup_{\pi' \in \Pi} \mathbb{E}\Big[\sum_{t=1}^\infty \gamma^t r(s_t, a_t) \mid \pi', (s_0, a_0, r_0, s_1)\Big]\Big] \\
&= \sup_{\pi \in \Pi} \mathbb{E}\Big[r(s_0, a_0) + \gamma V^\star(s_1)\Big] \\
&= \mathbb{E}\Big[r(s_0, a_0) + \gamma V^\star(s_1) \mid \pi\Big].
\end{aligned}
$$

where the second equality is by the tower property of conditional expectations, and the last equality follows from the definition of $\pi$. Now, by recursion,

$$
V^\star(s_0) \leq \mathbb{E}\Big[r(s_0, a_0) + \gamma V^\star(s_1) \mid \pi\Big] \leq \mathbb{E}\Big[r(s_0, a_0) + \gamma r(s_1, a_1) + \gamma^2 V^\star(s_2) \mid \pi\Big] \leq \ldots \leq V^\pi(s_0).
$$

Since $V^\pi(s) \leq \sup_{\pi' \in \Pi} V^{\pi'}(s) = V^\star(s)$, we have that $V^\pi = V^\star$, which completes the proof of the first claim.

For the same policy $\pi$, an analogous argument can be used prove the second claim. ∎

This shows that we may restrict ourselves to using stationary and deterministic policies without any loss in performance. The following theorem, also due to [Bellman, 1956], gives a precise characterization of the optimal value function.

Let us say that a vector $Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ satisfies the *Bellman optimality equations* if:

$$
Q(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}\Big[\max_{a' \in \mathcal{A}} Q(s', a')\Big].
$$

**Theorem 1.8** (Bellman Optimality Equations). *For any $Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, we have that $Q = Q^\star$ if and only if $Q$ satisfies the Bellman optimality equations. Furthermore, the deterministic policy $\pi(s) \in Q^\star(s,a)$ is an optimal policy (where ties are broken in some arbitrary and deterministic manner).*

Before we prove this claim, we will provide a few definitions. Let $\pi_Q$ denote the greedy policy with respect to a vector $Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, i.e

$$
\pi_Q(s) := \operatorname{argmax}_{a \in \mathcal{A}} Q(s,a).
$$

where ties are broken in some arbitrary (and deterministic) manner. With this notation, by the above theorem, the optimal policy $\pi^\star$ is given by:

$$
\pi^\star = \pi_{Q^\star}.
$$

Let us also use the following notation to turn a vector $Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ into a vector of length $|\mathcal{S}|$.

$$
V_Q(s) := \max_{a \in \mathcal{A}} Q(s,a).
$$

The *Bellman optimality operator* $\mathcal{T}_M : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is defined as:

$$
\mathcal{T}Q := r + \gamma P V_Q. \tag{0.3}
$$

9

This allows us to rewrite the Bellman optimality equation in the concise form:

$$Q = \mathcal{T}Q,$$

and, so, the previous theorem states that $Q = Q^\star$ if and only if $Q$ is a fixed point of the operator $\mathcal{T}$.

**Proof:** We first show sufficiency, i.e. that $Q^\star$ (the state-action value of an optimal policy) satisfies $Q^\star = \mathcal{T}Q^\star$. Let $\pi^\star$ be an optimal stationary and deterministic policy, which exists by Theorem 1.7. First let us show that $V^\star(s) = \max_a Q^\star(s,a)$. We have that $V^\star(s) = V^{\pi^\star}(s) = Q^{\pi^\star}(s, \pi^\star(a)) = Q^\star(s, \pi^\star(a))$, by Lemma 1.4 and Theorem 1.7. Also,

$$\max_a Q^\star(s,a) \geq Q^\star(s, \pi^\star(a)) = V^\star(s) \geq \max_a \max_\pi Q^\pi(s,a) \geq \max_a Q^{\pi^\star}(s,a) = \max_a Q^\star(s,a),$$

which proves the claim. Now for all actions $a \in \mathcal{A}$, we have:

$$\begin{aligned}
Q^\star(s,a) &= \max_\pi Q^\pi(s,a) = r(s,a) + \gamma \max_\pi \mathbb{E}_{s' \sim P(\cdot|s,a)}[V^\pi(s')] \\
&\overset{(a)}{=} r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V^\star(s')] \\
&= r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[\max_{a'} Q^\star(s',a')].
\end{aligned}$$

Here the equality $(a)$ follows from Theorem 1.7. This proves sufficiency.

For the converse, suppose $Q = \mathcal{T}Q$ for some $Q$. We now show that $Q = Q^\star$. Let $\pi = \pi_Q$. That $Q = \mathcal{T}Q$ implies that $Q = r + \gamma P^{\pi_Q}Q$, and so:

$$Q = (I - \gamma P^{\pi_Q})^{-1}r = Q^\pi$$

using Equation 0.2 in the last step. In other words, $Q$ is the action value of the policy $\pi_Q$. Now observe for any other deterministic and stationary policy $\pi'$:

$$\begin{aligned}
Q - Q^{\pi'} &= Q^\pi - Q^{\pi'} \\
&= Q^\pi - (I - \gamma P^{\pi'})^{-1}r \\
&= (I - \gamma P^{\pi'})^{-1}((I - \gamma P^{\pi'}) - (I - \gamma P^\pi))Q^\pi \\
&= \gamma(I - \gamma P^{\pi'})^{-1}(P^\pi - P^{\pi'})Q^\pi.
\end{aligned}$$

The proof is completed by noting that $(P^\pi - P^{\pi'})Q^\pi \geq 0$. To see this, recall that $(1 - \gamma)(I - \gamma P^{\pi'})^{-1}$ is a matrix with all positive entries (see Lemma 1.6), and now we can observe that:

$$[(P^\pi - P^{\pi'})Q^\pi]_{s,a} = \mathbb{E}_{s' \sim P(\cdot|s,a)}[Q^\pi(s', \pi(s')) - Q^\pi(s', \pi'(s'))] \geq 0$$

where the last step uses that $\pi = \pi_Q$. Thus we have that $Q \geq Q^{\pi'}$ for all deterministic and stationary $\pi'$ which shows $Q = Q^\star$, using Theorem 1.7. This completes the proof. ∎

## 1.2 Finite-Horizon Markov Decision Processes

In some cases, it is natural to work with finite-horizon (and time-dependent) Markov decision Processes (see the discussion below). Here, a finite horizon, time-dependent Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, \{P\}_h, \{r\}_h, H, \mu)$ is specified as follows:

- A state space $\mathcal{S}$, which may be finite or infinite.

- An action space $\mathcal{A}$, which also may be discrete or infinite.

- A time-dependent transition function $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ is the space of probability distributions over $\mathcal{S}$ (i.e., the probability simplex). $P_h(s'|s, a)$ is the probability of transitioning into state $s'$ upon taking action $a$ in state $s$ at time step $h$. Note that time-dependent setting generalizes the stationary setting where all steps share the same transition.

- A time-dependent reward function $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$. $r_h(s, a)$ is the immediate reward associated with taking action $a$ in state $s$ at time step $h$.

- A integer $H$ which defines the horizon of the problem.

- An initial state distribution $\mu \in \Delta(\mathcal{S})$, which species how the initial state $s_0$ is generated.

Here, for a policy $\pi$, a state $s$, and $h \in \{0, \ldots H - 1\}$, we define the value function $V_h^\pi : \mathcal{S} \to \mathbb{R}$ as

$$V_h^\pi(s) = \mathbb{E}\Big[ \sum_{t=h}^{H-1} r_h(s_t, a_t) \mid \pi, s_h = s \Big],$$

where again the expectation is with respect to the randomness of the trajectory, that is, the randomness in state transitions and the stochasticity of $\pi$. Similarly, the state-action value (or Q-value) function $Q_h^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as

$$Q_h^\pi(s, a) = \mathbb{E}\Big[ \sum_{t=h}^{H-1} r_h(s_t, a_t) \mid \pi, s_h = s, a_h = a \Big].$$

We also use the notation $V^\pi(s) = V_0^\pi(s)$.

Again, given a state $s$, the goal of the agent is to find a policy $\pi$ that maximizes the value, i.e. the optimization problem the agent seeks to solve is:

$$\max_\pi V^\pi(s) \tag{0.4}$$

where recall that $V^\pi(s) = V_0^\pi(s)$.

**Theorem 1.9.** *(Bellman optimality equations) Define*

$$Q_h^\star(s, a) = \sup_{\pi \in \Pi} Q_h^\pi(s, a)$$

*where the* sup *is over all non-stationary and randomized policies. Suppose that $Q_H = 0$. We have that $Q_h = Q_h^\star$ for all $h \in [H]$ if and only if for all $h \in [H]$,*

$$Q_h(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)} \left[ \max_{a' \in \mathcal{A}} Q_{h+1}(s', a') \right].$$

*Furthermore, $\pi(s, h) = \mathrm{argmax}_{a \in \mathcal{A}} Q_h^\star(s, a)$ is an optimal policy.*

We leave the proof as an exercise to the reader.

**Discussion: Stationary MDPs vs Time-Dependent MDPs**  For the purposes of this book, it is natural for us to study both of these models, where we typically assume stationary dynamics in the infinite horizon setting and time-dependent dynamics in the finite-horizon setting. From a theoretical perspective, the finite horizon, *time-dependent* setting is often more amenable to analysis, where optimal statistical rates often require simpler arguments. However, we should note that from a practical perspective, time-dependent MDPs are rarely utilized because they lead to policies and value functions that are $O(H)$ larger (to store in memory) than those in the stationary setting. In practice, we often incorporate temporal information directly into the definition of the state, which leads to more compact value functions and policies (when coupled with function approximation methods, which attempt to represent both the values and policies in a more compact form).

| | Value Iteration | Policy Iteration | LP-Algorithms |
|---|---|---|---|
| Poly? | $|\mathcal{S}|^2|\mathcal{A}|\frac{L(P,r,\gamma)\log\frac{1}{1-\gamma}}{1-\gamma}$ | $(|\mathcal{S}|^3+|\mathcal{S}|^2|\mathcal{A}|)\frac{L(P,r,\gamma)\log\frac{1}{1-\gamma}}{1-\gamma}$ | $|\mathcal{S}|^3|\mathcal{A}|L(P,r,\gamma)$ |
| Strongly Poly? | ✗ | $(|\mathcal{S}|^3+|\mathcal{S}|^2|\mathcal{A}|)\cdot\min\left\{\frac{|\mathcal{A}|^{|\mathcal{S}|}}{|\mathcal{S}|},\frac{|\mathcal{S}|^2|\mathcal{A}|\log\frac{|\mathcal{S}|^2}{1-\gamma}}{1-\gamma}\right\}$ | $|\mathcal{S}|^4|\mathcal{A}|^4\log\frac{|\mathcal{S}|}{1-\gamma}$ |

Table 0.1: Computational complexities of various approaches (we drop universal constants). Polynomial time algorithms depend on the bit complexity, $L(P,r,\gamma)$, while strongly polynomial algorithms do not. Note that only for a fixed value of $\gamma$ are value and policy iteration polynomial time algorithms; otherwise, they are not polynomial time algorithms. Similarly, only for a fixed value of $\gamma$ is policy iteration a strongly polynomial time algorithm. In contrast, the LP-approach leads to both polynomial time and strongly polynomial time algorithms; for the latter, the approach is an interior point algorithm. See text for further discussion, and Section 1.6 for references. Here, $|\mathcal{S}|^2|\mathcal{A}|$ is the assumed runtime per iteration of value iteration, and $|\mathcal{S}|^3+|\mathcal{S}|^2|\mathcal{A}|$ is the assumed runtime per iteration of policy iteration (note that for this complexity we would directly update the values $V$ rather than $Q$ values, as described in the text); these runtimes are consistent with assuming cubic complexity for linear system solving.

## 1.3   Computational Complexity

This section will be concerned with computing an optimal policy, when the MDP $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ is known; this can be thought of as the solving the *planning* problem. While much of this book is concerned with statistical limits, understanding the computational limits can be informative. We will consider algorithms which give both exact and approximately optimal policies. In particular, we will be interested in polynomial time (and strongly polynomial time) algorithms.

Suppose that $(P,r,\gamma)$ in our MDP $M$ is specified with rational entries. Let $L(P,r,\gamma)$ denote the total bit-size required to specify $M$, and assume that basic arithmetic operations $+,-,\times,\div$ take unit time. Here, we may hope for an algorithm which (exactly) returns an optimal policy whose runtime is polynomial in $L(P,r,\gamma)$ and the number of states and actions.

More generally, it may also be helpful to understand which algorithms are *strongly* polynomial. Here, we do not want to explicitly restrict $(P,r,\gamma)$ to be specified by rationals. An algorithm is said to be strongly polynomial if it returns an optimal policy with runtime that is polynomial in only the number of states and actions (with no dependence on $L(P,r,\gamma)$).

The first two subsections will cover classical *iterative* algorithms that compute $Q^\star$, and then we cover the linear programming approach.

### 1.3.1   Value Iteration

Perhaps the simplest algorithm for discounted MDPs is to iteratively apply the fixed point mapping: starting at some $Q$, we iteratively apply $\mathcal{T}$:

$$Q \leftarrow \mathcal{T}Q\,,$$

This is algorithm is referred to as *Q-value iteration*.

**Lemma 1.10.** *(contraction) For any two vectors* $Q, Q' \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$,

$$\|\mathcal{T}Q - \mathcal{T}Q'\|_\infty \leq \gamma\|Q - Q'\|_\infty$$

**Proof:** First, let us show that for all $s \in \mathcal{S}$, $|V_Q(s) - V_{Q'}(s)| \leq \max_{a \in \mathcal{A}} |Q(s,a) - Q'(s,a)|$. Assume $V_Q(s) > V_{Q'}(s)$ (the other direction is symmetric), and let $a$ be the greedy action for $Q$ at $s$. Then

$$|V_Q(s) - V_{Q'}(s)| = Q(s,a) - \max_{a' \in \mathcal{A}} Q'(s,a') \leq Q(s,a) - Q'(s,a) \leq \max_{a \in \mathcal{A}} |Q(s,a) - Q'(s,a)|.$$

Using this,

$$
\begin{aligned}
\|\mathcal{T}Q - \mathcal{T}Q'\|_\infty &= \gamma \|PV_Q - PV_{Q'}\|_\infty \\
&= \gamma \|P(V_Q - V_{Q'})\|_\infty \\
&\leq \gamma \|V_Q - V_{Q'}\|_\infty \\
&= \gamma \max_s |V_Q(s) - V_{Q'}(s)| \\
&\leq \gamma \max_s \max_a |Q(s,a) - Q'(s,a)| \\
&= \gamma \|Q - Q'\|_\infty
\end{aligned}
$$

where the first inequality uses that each element of $P(V_Q - V_{Q'})$ is a convex average of $V_Q - V_{Q'}$ and the second inequality uses our claim above. ∎

The following result bounds the sub-optimality of the greedy policy itself, based on the error in $Q$-value function.

**Lemma 1.11.** *(Q-Error Amplification) For any vector $Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$,*

$$V^{\pi_Q} \geq V^\star - \frac{2\|Q - Q^\star\|_\infty}{1 - \gamma} \mathbb{1},$$

*where $\mathbb{1}$ denotes the vector of all ones.*

**Proof:** Fix state $s$ and let $a = \pi_Q(s)$. We have:

$$
\begin{aligned}
V^\star(s) - V^{\pi_Q}(s) &= Q^\star(s, \pi^\star(s)) - Q^{\pi_Q}(s,a) \\
&= Q^\star(s, \pi^\star(s)) - Q^\star(s,a) + Q^\star(s,a) - Q^{\pi_Q}(s,a) \\
&= Q^\star(s, \pi^\star(s)) - Q^\star(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V^\star(s') - V^{\pi_Q}(s')] \\
&\leq Q^\star(s, \pi^\star(s)) - Q(s, \pi^\star(s)) + Q(s,a) - Q^\star(s,a) \\
&\quad + \gamma \mathbb{E}_{s' \sim P(s,a)}[V^\star(s') - V^{\pi_Q}(s')] \\
&\leq 2\|Q - Q^\star\|_\infty + \gamma \|V^\star - V^{\pi_Q}\|_\infty.
\end{aligned}
$$

where the first inequality uses $Q(s, \pi^\star(s)) \leq Q(s, \pi_Q(s)) = Q(s,a)$ due to the definition of $\pi_Q$. ∎

**Theorem 1.12.** *(Q-value iteration convergence). Set $Q^{(0)} = 0$. For $k = 0, 1, \ldots$, suppose:*

$$Q^{(k+1)} = \mathcal{T}Q^{(k)}$$

*Let $\pi^{(k)} = \pi_{Q^{(k)}}$. For $k \geq \frac{\log \frac{2}{(1-\gamma)^2 \epsilon}}{1 - \gamma}$,*

$$V^{\pi^{(k)}} \geq V^\star - \epsilon \mathbb{1}.$$

**Proof:** Since $\|Q^\star\|_\infty \leq 1/(1 - \gamma)$, $Q^{(k)} = \mathcal{T}^k Q^{(0)}$ and $Q^\star = \mathcal{T}Q^\star$, Lemma 1.10 gives

$$\|Q^{(k)} - Q^\star\|_\infty = \|\mathcal{T}^k Q^{(0)} - \mathcal{T}^k Q^\star\|_\infty \leq \gamma^k \|Q^{(0)} - Q^\star\|_\infty = (1 - (1-\gamma))^k \|Q^\star\|_\infty \leq \frac{\exp(-(1-\gamma)k)}{1 - \gamma}.$$

The proof is completed with our choice of $\gamma$ and using Lemma 1.11. ∎

**Iteration complexity for an exact solution.** With regards to computing an exact optimal policy, when the gap between the current objective value and the optimal objective value is smaller than $2^{-L(P,r,\gamma)}$, then the greedy policy will be optimal. This leads to claimed complexity in Table 0.1. Value iteration is not strongly polynomial algorithm due to that, in finite time, it may never return the optimal policy.

### 1.3.2 Policy Iteration

The policy iteration algorithm, for discounted MDPs, starts from an arbitrary policy $\pi_0$, and repeats the following iterative procedure: for $k = 0, 1, 2, \ldots$

1. *Policy evaluation.* Compute $Q^{\pi_k}$

2. *Policy improvement.* Update the policy:

$$\pi_{k+1} = \pi_{Q^{\pi_k}}$$

In each iteration, we compute the Q-value function of $\pi_k$, using the analytical form given in Equation 0.2, and update the policy to be greedy with respect to this new $Q$-value. The first step is often called *policy evaluation*, and the second step is often called *policy improvement*.

**Lemma 1.13.** *We have that:*

1. $Q^{\pi_{k+1}} \geq \mathcal{T} Q^{\pi_k} \geq Q^{\pi_k}$

2. $\|Q^{\pi_{k+1}} - Q^\star\|_\infty \leq \gamma \|Q^{\pi_k} - Q^\star\|_\infty$

**Proof:** First let us show that $\mathcal{T} Q^{\pi_k} \geq Q^{\pi_k}$. Note that the policies produced in policy iteration are always deterministic, so $V^{\pi_k}(s) = Q^{\pi_k}(s, \pi_k(s))$ for all iterations $k$ and states $s$. Hence,

$$\mathcal{T} Q^{\pi_k}(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[\max_{a'} Q^{\pi_k}(s', a')]$$
$$\geq r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[Q^{\pi_k}(s', \pi_k(s'))] = Q^{\pi_k}(s, a).$$

Now let us prove that $Q^{\pi_{k+1}} \geq \mathcal{T} Q^{\pi_k}$. First, let use see that $Q^{\pi_{k+1}} \geq Q^{\pi_k}$:

$$Q^{\pi_k} = r + \gamma P^{\pi_k} Q^{\pi_k} \leq r + \gamma P^{\pi_{k+1}} Q^{\pi_k} \leq \sum_{t=0}^{\infty} \gamma^t (P^{\pi_{k+1}})^t r = Q^{\pi_{k+1}}.$$

where we have used that $\pi_{k+1}$ is the greedy policy in the first inequality and recursion in the second inequality. Using this,

$$Q^{\pi_{k+1}}(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[Q^{\pi_{k+1}}(s', \pi_{k+1}(s'))]$$
$$\geq r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[Q^{\pi_k}(s', \pi_{k+1}(s'))]$$
$$= r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[\max_{a'} Q^{\pi_k}(s', a')] = \mathcal{T} Q^{\pi_k}(s, a)$$

which completes the proof of the first claim.

For the second claim,

$$\|Q^\star - Q^{\pi_{k+1}}\|_\infty \leq \|Q^\star - \mathcal{T} Q^{\pi_k}\|_\infty = \|\mathcal{T} Q^\star - \mathcal{T} Q^{\pi_k}\|_\infty \leq \gamma \|Q^\star - Q^{\pi_k}\|_\infty$$

where we have used that $Q^\star \geq Q^{\pi_{k+1}} \geq Q^{\pi_k}$ in second step and the contraction property of $\mathcal{T}(\cdot)$ (see Lemma 1.10 in the last step. ∎

With this lemma, a convergence rate for the policy iteration algorithm immediately follows.

**Theorem 1.14.** *(Policy iteration convergence). Let $\pi_0$ be any initial policy. For $k \geq \frac{\log \frac{1}{(1-\gamma)\epsilon}}{1-\gamma}$, the $k$-th policy in policy iteration has the following performance bound:*

$$Q^{\pi^{(k)}} \geq Q^\star - \epsilon \mathbb{1}.$$

**Iteration complexity for an exact solution.** With regards to computing an exact optimal policy, it clear from the previous results that policy iteration is no worse than value iteration. However, with regards to obtaining an exact solution MDP that is independent of the bit complexity, $L(P, r, \gamma)$, improvements are possible (and where we assume basic arithmetic operations on real numbers are order one cost). Naively, the number of iterations of policy iterations is bounded by the number of policies, namely $|\mathcal{A}|^{|\mathcal{S}|}$; here, a small improvement is possible, where the number of iterations of policy iteration can be bounded by $\frac{|\mathcal{A}|^{|\mathcal{S}|}}{|\mathcal{S}|}$. Remarkably, for a fixed value of $\gamma$, policy iteration can be show to be a strongly polynomial time algorithm, where policy iteration finds an exact policy in at most $\frac{|\mathcal{S}|^2 |\mathcal{A}| \log \frac{|\mathcal{S}|^2}{1-\gamma}}{1-\gamma}$ iterations. See Table 0.1 for a summary, and Section 1.6 for references.

### 1.3.3 Value Iteration for Finite Horizon MDPs

Let us now specify the value iteration algorithm for finite-horizon MDPs. For the finize-horizon setting, it turns out that the analogues of value iteration and policy iteration lead to identical algorithms. The value iteration algorithm is specified as follows:

1. Set $Q_{H-1}(s, a) = r_{H-1}(s, a)$.

2. For $h = H - 2, \ldots 0$, set:

$$Q_h(s, a) = r_h(s, a) + \gamma \mathbb{E}_{s' \sim P_h(\cdot|s,a)} \left[ \max_{a' \in \mathcal{A}} Q_{h+1}(s', a') \right].$$

By Theorem 1.9, it follows that $Q_h(s, a) = Q_h^\star(s, a)$ and that $\pi(s, h) = \text{argmax}_{a \in \mathcal{A}} Q_h^\star(s, a)$ is an optimal policy.

### 1.3.4 The Linear Programming Approach

It is helpful to understand an alternative approach to finding an optimal policy for a known MDP. With regards to computation, consider the setting where our MDP $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu)$ is known and $P$, $r$, and $\gamma$ are all specified by rational numbers. Here, from a computational perspective, the previous iterative algorithms are, strictly speaking, not polynomial time algorithms, due to that they depend polynomially on $1/(1 - \gamma)$, which is not polynomial in the description length of the MDP . In particular, note that any rational value of $1 - \gamma$ may be specified with only $O(\log \frac{1}{1-\gamma})$ bits of precision. In this context, we may hope for a fully polynomial time algorithm, when given knowledge of the MDP, which would have a computation time which would depend polynomially on the description length of the MDP $M$, when the parameters are specified as rational numbers. We now see that the LP approach provides a polynomial time algorithm.

**The Primal LP and A Polynomial Time Algorithm**

Consider the following optimization problem with variables $V \in \mathbb{R}^{|\mathcal{S}|}$:

$$\min \quad \sum_s \mu(s)V(s)$$

$$\text{subject to} \quad V(s) \geq r(s,a) + \gamma \sum_{s'} P(s'|s,a)V(s') \quad \forall a \in \mathcal{A}, \, s \in \mathcal{S}$$

Provided that $\mu$ has full support, then the optimal value function $V^\star(s)$ is the unique solution to this linear program. With regards to computation time, linear programming approaches only depend on the description length of the coefficients in the program, due to that this determines the computational complexity of basic additions and multiplications. Thus, this approach will only depend on the bit length description of the MDP, when the MDP is specified by rational numbers.

**Computational complexity for an exact solution.** Table 0.1 shows the runtime complexity for the LP approach, where we assume a standard runtime for solving a linear program. The strongly polynomial algorithm is an interior point algorithm. See Section 1.6 for references.

**Policy iteration and the simplex algorithm.** It turns out that the policy iteration algorithm is actually the simplex method with block pivot. While the simplex method, in general, is not a strongly polynomial time algorithm, the policy iteration algorithm is a strongly polynomial time algorithm, provided we keep the discount factor fixed. See [Ye, 2011].

**The Dual LP and the State-Action Polytope**

For a fixed (possibly stochastic) policy $\pi$, let us define let us define a visitation measure over states *and* actions induced by following $\pi$ after starting at $s_0$. Precisely, define this distribution, $d_{s_0}^\pi$, as follows:

$$d_{s_0}^\pi(s,a) := (1-\gamma) \sum_{t=0}^\infty \gamma^t \Pr^\pi(s_t = s, a_t = a|s_0) \tag{0.5}$$

where $\Pr^\pi(s_t = s, a_t = a|s_0)$ is the probability that $s_t = s$ and $a_t = a$, after starting at state $s_0$ and following $\pi$ thereafter. It is straightforward to verify that $d_{s_0}^\pi$ is a distribution over $\mathcal{S} \times \mathcal{A}$. We also overload notation and write:

$$d_\mu^\pi(s,a) = \mathbb{E}_{s_0 \sim \mu} \left[ d_{s_0}^\pi(s,a) \right].$$

for a distribution $\mu$ over $\mathcal{S}$. Recall Lemma 1.6 provides a way to easily compute $d_\mu^\pi(s,a)$ through an appropriate vector-matrix multiplication.

It is straightforward to verify that $d_\mu^\pi$ satisfies, for all states $s \in \mathcal{S}$:

$$\sum_a d_\mu^\pi(s,a) = (1-\gamma)\mu(s) + \gamma \sum_{s',a'} P(s|s',a')d_\mu^\pi(s',a').$$

Let us define the state-action polytope as follows:

$$\mathcal{K}_\mu := \{d| \, d \geq 0 \text{ and } \sum_a d(s,a) = (1-\gamma)\mu(s) + \gamma \sum_{s',a'} P(s|s',a')d(s',a')\}$$

We now see that this set precisely characterizes all state-action visitation distributions.

16

**Proposition 1.15.** *We have that $\mathcal{K}_\mu$ is equal to the set of all feasible state-action distributions, i.e. $d \in \mathcal{K}_\mu$ if and only if there exists a stationary (and possibly randomized) policy $\pi$ such that $d_\mu^\pi = d$.*

With respect the variables $d \in \mathbb{R}^{|\mathcal{S}| \cdot |\mathcal{A}|}$, the dual LP formulation is as follows:

$$\max \qquad \frac{1}{1-\gamma} \sum_{s,a} d_\mu(s,a) r(s,a)$$
$$\text{subject to} \qquad d \in \mathcal{K}_\mu$$

Note that $\mathcal{K}_\mu$ is itself a polytope, and one can verify that this is indeed the dual of the aforementioned LP. This approach provides an alternative approach to finding an optimal solution.

If $d^\star$ is the solution to this LP, and provided that $\mu$ has full support, then we have that:

$$\pi^\star(a|s) = \frac{d^\star(s,a)}{\sum_{a'} d^\star(s,a')},$$

is an optimal policy. An alternative optimal policy is $\mathrm{argmax}_a\, d^\star(s,a)$ (and these policies are identical if the optimal policy is unique).

## 1.4 Sample Complexity and Sampling Models

Much of reinforcement learning is concerned with finding a near optimal policy (or obtaining near optimal reward) in settings where the MDPs is not known to the learner. We will study these questions in a few different models of how the agent obtains information about the unknown underlying MDP. In each of these settings, we are interested understanding the number of samples required to find a near optimal policy, i.e. the *sample complexity*. Ultimately, we interested in obtaining results which are applicable to cases where number of states and actions is large (or, possibly, countably or uncountably infinite). This is many ways analogous to the supervised learning question of generalization, though, as we shall see, this question if fundamentally more challenging in the reinforcement learning setting.

**The Episodic Setting.** In the episodic setting, in every episode, the learner acts for some finite number of steps, starting from a fixed starting state $s_0 \sim \mu$, the learner observes the trajectory, and the state resets to $s_0 \sim \mu$. This episodic model of feedback is applicable to both the finte-horizon and infinite horizon settings.

- (Finite Horizon MDPs) Here, each episode lasts for $H$-steps, and then the state is reset to $s_0 \sim \mu$.

- (Infinite Horizon MDPs) Even for infinite horizon MDPs it is natural to work in an episodic model for learning, where each episode terminates after a finite number of steps. Here, it is often natural to assume either the agent can terminate the episode at will or that the episode will terminate at each step with probability $1 - \gamma$. After termination, we again assume that the state is reset to $s_0 \sim \mu$. Note that, if each step in an episode is terminated with probability $1 - \gamma$, then the observed cumulative reward in an episode of a policy provides an unbiased estimate of the infinite-horizon, discounted value of that policy.

In this setting, we are often interested in either the number of episodes it takes to find a near optimal policy, which is a *PAC* (probably, approximately correct) guarantee, or we are interested in a *regret* guarantee (which we will study in Chapter 7). Both of these questions are with regards to statistical complexity (i.e. the sample complexity) of learning.

The episodic setting is challenging in that the agent has to engage in some exploration in order to gain information at the relevant state. As we shall see in Chapter 7, this exploration must be strategic, in the sense that simply behaving

17

randomly will not lead to information being gathered quickly enough. It is often helpful to study the statistical complexity of learning in a more abstract sampling model, a generative model, which allows to avoid having to directly address this exploration issue. Furthermore, this sampling model is natural in its own right.

**The generative model setting.** A *generative model* provides takes as input a state action pair $(s, a)$ and returns a sample $s' \sim P(\cdot|s, a)$ and the reward $r(s, a)$ (or a sample of the reward if the rewards are stochastic).

**The offline RL setting.** The *offline RL setting* is where the agent has access to an offline dataset, say generated under some policy (or a collection of policies). In the simplest of these settings, we may assume our dataset is of the form $\{(s, a, s', r)\}$ where $r$ is the reward (corresponding to $r(s, a)$ if the reward is deterministic) and $s' \sim P(\cdot|s, a)$. Furthermore, for simplicity, it can be helpful to assume that the $s, a$ pairs in this dataset were sampled i.i.d. from some fixed distribution $\nu$ over $\mathcal{S} \times \mathcal{A}$.

## 1.5   Bonus: Advantages and The Performance Difference Lemma

Throughout, we will overload notation where, for a distribution $\mu$ over $\mathcal{S}$, we write:

$$V^\pi(\mu) = \mathbb{E}_{s \sim \mu} [V^\pi(s)].$$

The *advantage* $A^\pi(s, a)$ of a policy $\pi$ is defined as

$$A^\pi(s, a) := Q^\pi(s, a) - V^\pi(s).$$

Note that:

$$A^*(s, a) := A^{\pi^*}(s, a) \leq 0$$

for all state-action pairs.

Analogous to the state-action visitation distribution (see Equation 0.5), we can define a visitation measure over just the states. When clear from context, we will overload notation and also denote this distribution by $d_{s_0}^\pi$, where:

$$d_{s_0}^\pi(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathrm{Pr}^\pi(s_t = s|s_0). \tag{0.6}$$

Here, $\mathrm{Pr}^\pi(s_t = s|s_0)$ is the state visitation probability, under $\pi$ starting at state $s_0$. Again, we write:

$$d_\mu^\pi(s) = \mathbb{E}_{s_0 \sim \mu} [d_{s_0}^\pi(s)].$$

for a distribution $\mu$ over $\mathcal{S}$.

The following lemma is helpful in the analysis of RL algorithms.

**Lemma 1.16.** *(The performance difference lemma) For all policies $\pi, \pi'$ and distributions $\mu$ over $\mathcal{S}$,*

$$V^\pi(\mu) - V^{\pi'}(\mu) = \frac{1}{1 - \gamma} \mathbb{E}_{s' \sim d_\mu^\pi} \mathbb{E}_{a' \sim \pi(\cdot|s')} \left[ A^{\pi'}(s', a') \right].$$

**Proof:** Let $\mathrm{Pr}^\pi(\tau|s_0 = s)$ denote the probability of observing a trajectory $\tau$ when starting in state $s$ and following the policy $\pi$. By definition of $d_{s_0}^{\pi_\theta}$, observe that for any function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$,

$$\mathbb{E}_{\tau \sim \mathrm{Pr}^\pi} \left[ \sum_{t=0}^{\infty} \gamma^t f(s_t, a_t) \right] = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d_{s_0}^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ f(s, a) \right]. \tag{0.7}$$

18

Using a telescoping argument, we have:

$$
\begin{aligned}
V^{\pi}(s) - V^{\pi'}(s) &= \mathbb{E}_{\tau \sim \mathrm{Pr}^{\pi}(\tau | s_0 = s)}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] - V^{\pi'}(s)\\
&= \mathbb{E}_{\tau \sim \mathrm{Pr}^{\pi}(\tau | s_0 = s)}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + V^{\pi'}(s_t) - V^{\pi'}(s_t)\right)\right] - V^{\pi'}(s)\\
&\overset{(a)}{=} \mathbb{E}_{\tau \sim \mathrm{Pr}^{\pi}(\tau | s_0 = s)}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \gamma V^{\pi'}(s_{t+1}) - V^{\pi'}(s_t)\right)\right]\\
&\overset{(b)}{=} \mathbb{E}_{\tau \sim \mathrm{Pr}^{\pi}(\tau | s_0 = s)}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \gamma \mathbb{E}[V^{\pi'}(s_{t+1}) | s_t, a_t] - V^{\pi'}(s_t)\right)\right]\\
&\overset{(c)}{=} \mathbb{E}_{\tau \sim \mathrm{Pr}^{\pi}(\tau | s_0 = s)}\left[\sum_{t=0}^{\infty} \gamma^t \left(Q^{\pi'}(s_t, a_t) - V^{\pi'}(s_t)\right)\right]\\
&= \mathbb{E}_{\tau \sim \mathrm{Pr}^{\pi}(\tau | s_0 = s)}\left[\sum_{t=0}^{\infty} \gamma^t A^{\pi'}(s_t, a_t)\right]\\
&= \frac{1}{1-\gamma}\mathbb{E}_{s' \sim d_s^{\pi}} \mathbb{E}_{a \sim \pi(\cdot | s)} A^{\pi'}(s', a),
\end{aligned}
$$

where $(a)$ rearranges terms in the summation via telescoping; $(b)$ uses the tower property of conditional expectations; $(c)$ follows by definition; and the final equality follows from Equation 0.7. ∎

## 1.6 Bibliographic Remarks and Further Reading

We refer the reader to [Puterman, 1994] for a more detailed treatment of dynamic programming and MDPs. [Puterman, 1994] also contains a thorough treatment of the dual LP, along with a proof of Lemma 1.15

With regards to the computational complexity of policy iteration, [Ye, 2011] showed that policy iteration is a strongly polynomial time algorithm for a fixed discount rate [1]. Also, see [Ye, 2011] for a good summary of the computational complexities of various approaches. [Mansour and Singh, 1999] showed that the number of iterations of policy iteration can be bounded by $\frac{|\mathcal{A}|^{|\mathcal{S}|}}{|\mathcal{S}|}$.

With regards to a strongly polynomial algorithm, the CIPA algorithm [Ye, 2005] is an interior point algorithm with the claimed runtime in Table 0.1.

Lemma 1.11 is due to Singh and Yee [1994].

The performance difference lemma is due to [Kakade and Langford, 2002, Kakade, 2003], though the lemma was implicit in the analysis of a number of prior works.

---

[1]The stated strongly polynomial runtime in Table 0.1 for policy iteration differs from that in [Ye, 2011] due to we assume that the runtime per iteration of policy iteration is $|\mathcal{S}|^3 + |\mathcal{S}|^2|\mathcal{A}|$.

# Chapter 2

# Sample Complexity with a Generative Model

This chapter begins our study of the sample complexity, where we focus on the (minmax) number of transitions we need to observe in order to accurately estimate $Q^\star$ or in order to find a near optimal policy. We assume that we have access to a generative model (as defined in Section 1.4) and that the reward function is deterministic (the latter is often a mild assumption, due to that much of the difficulty in RL is due to the uncertainty in the transition model $P$).

This chapter follows the results due to [Azar et al., 2013], along with some improved rates due to [Agarwal et al., 2020c]. One of the key observations in this chapter is that we can find a near optimal policy using a number of observed transitions that is *sublinear* in the model size, i.e. use a number of samples that is smaller than $O(|\mathcal{S}|^2|\mathcal{A}|)$. In other words, we do not need to learn an accurate model of the world in order to learn to act near optimally.

**Notation.** We define $\widehat{M}$ to be the empirical MDP that is identical to the original $M$, except that it uses $\widehat{P}$ instead of $P$ for the transition model. When clear from context, we drop the subscript on $M$ on the values, action values (and one-step variances and variances which we define later). We let $\widehat{V}^\pi$, $\widehat{Q}^\pi$, $\widehat{Q}^\star$, and $\widehat{\pi}^\star$ denote the value function, state-action value function, optimal state-action value, and optimal policy in $\widehat{M}$, respectively.

## 2.1   Warmup: a naive model-based approach

A central question in this chapter is: *Do we require an accurate model of the world in order to find a near optimal policy?* Recall that a generative model takes as input a state action pair $(s, a)$ and returns a sample $s' \sim P(\cdot|s, a)$ and the reward $r(s, a)$ (or a sample of the reward if the rewards are stochastic). Let us consider the most naive approach to learning (when we have access to a generative model): suppose we call our simulator $N$ times at each state action pair. Let $\widehat{P}$ be our empirical model, defined as follows:

$$\widehat{P}(s'|s, a) = \frac{\text{count}(s', s, a)}{N}$$

where $\text{count}(s', s, a)$ is the number of times the state-action pair $(s, a)$ transitions to state $s'$. As the $N$ is the number of calls for each state action pair, the total number of calls to our generative model is $|\mathcal{S}||\mathcal{A}|N$. As before, we can view $\widehat{P}$ as a matrix of size $|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|$.

Note that since $P$ has a $|\mathcal{S}|^2|\mathcal{A}|$ parameters, we would expect that observing $O(|\mathcal{S}|^2|\mathcal{A}|)$ transitions is sufficient to provide us with an accurate model. The following proposition shows that this is the case.

**Proposition 2.1.** *There exists an absolute constant $c$ such that the following holds. Suppose $\epsilon \in \left(0, \frac{1}{1-\gamma}\right)$ and that we obtain*

$$\# \text{ samples from generative model } = |\mathcal{S}||\mathcal{A}|N \geq \frac{\gamma}{(1-\gamma)^4} \frac{|\mathcal{S}|^2|\mathcal{A}| \log(c|\mathcal{S}||\mathcal{A}|/\delta)}{\epsilon^2}$$

*where we uniformly sample every state action pair. Then, with probability greater than $1 - \delta$, we have:*

- *(Model accuracy) The transition model has error bounded as:*

$$\max_{s,a} \|P(\cdot|s,a) - \widehat{P}(\cdot|s,a)\|_1 \leq (1-\gamma)^2 \epsilon.$$

- *(Uniform value accuracy) For all policies $\pi$,*

$$\|Q^\pi - \widehat{Q}^\pi\|_\infty \leq \epsilon$$

- *(Near optimal planning) Suppose that $\widehat{\pi}$ is the optimal policy in $\widehat{M}$. We have that:*

$$\|\widehat{Q}^\star - Q^\star\|_\infty \leq \epsilon, \quad \text{and} \quad \|Q^{\widehat{\pi}} - Q^\star\|_\infty \leq 2\epsilon.$$

Before we provide the proof, the following lemmas will be helpful throughout:

**Lemma 2.2.** *(Simulation Lemma) For all $\pi$ we have that:*

$$Q^\pi - \widehat{Q}^\pi = \gamma(I - \gamma\widehat{P}^\pi)^{-1}(P - \widehat{P})V^\pi$$

**Proof:** Using our matrix equality for $Q^\pi$ (see Equation 0.2), we have:

$$
\begin{aligned}
Q^\pi - \widehat{Q}^\pi &= (I - \gamma P^\pi)^{-1}r - (I - \gamma\widehat{P}^\pi)^{-1}r \\
&= (I - \gamma\widehat{P}^\pi)^{-1}((I - \gamma\widehat{P}^\pi) - (I - \gamma P^\pi))Q^\pi \\
&= \gamma(I - \gamma\widehat{P}^\pi)^{-1}(P^\pi - \widehat{P}^\pi)Q^\pi \\
&= \gamma(I - \gamma\widehat{P}^\pi)^{-1}(P - \widehat{P})V^\pi
\end{aligned}
$$

which proves the claim. ∎

**Lemma 2.3.** *For any policy $\pi$, MDP $M$ and vector $v \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$, we have $\left\|(I - \gamma P^\pi)^{-1}v\right\|_\infty \leq \|v\|_\infty / (1-\gamma)$.*

**Proof:** Note that $v = (I - \gamma P^\pi)(I - \gamma P^\pi)^{-1}v = (I - \gamma P^\pi)w$, where $w = (I - \gamma P^\pi)^{-1}v$. By triangle inequality, we have

$$\|v\|_\infty = \|(I - \gamma P^\pi)w\|_\infty \geq \|w\|_\infty - \gamma\|P^\pi w\|_\infty \geq \|w\|_\infty - \gamma\|w\|_\infty,$$

where the final inequality follows since $P^\pi w$ is an average of the elements of $w$ by the definition of $P^\pi$ so that $\|P^\pi w\|_\infty \leq \|w\|_\infty$. Rearranging terms completes the proof. ∎

Now we are ready to complete the proof of our proposition.

**Proof:** Using the concentration of a distribution in the $\ell_1$ norm (Lemma A.8), we have that for a fixed $s, a$ that, with probability greater than $1 - \delta$, we have:

$$\|P(\cdot|s,a) - \widehat{P}(\cdot|s,a)\|_1 \leq c\sqrt{\frac{|\mathcal{S}| \log(1/\delta)}{m}}$$

where $m$ is the number of samples used to estimate $\widehat{P}(\cdot|s, a)$. The first claim now follows by the union bound (and redefining $\delta$ and $c$ appropriately).

For the second claim, we have that:

$$\|Q^\pi - \widehat{Q}^\pi\|_\infty = \|\gamma(I - \gamma\widehat{P}^\pi)^{-1}(P - \widehat{P})V^\pi\|_\infty \leq \frac{\gamma}{1-\gamma}\|(P - \widehat{P})V^\pi\|_\infty$$

$$\leq \frac{\gamma}{1-\gamma}\left(\max_{s,a}\|P(\cdot|s,a) - \widehat{P}(\cdot|s,a)\|_1\right)\|V^\pi\|_\infty \leq \frac{\gamma}{(1-\gamma)^2}\max_{s,a}\|P(\cdot|s,a) - \widehat{P}(\cdot|s,a)\|_1$$

where the penultimate step uses Holder's inequality. The second claim now follows.

For the final claim, first observe that $|\sup_x f(x) - \sup_x g(x)| \leq \sup_x |f(x) - g(x)|$, where $f$ and $g$ are real valued functions. This implies:

$$|\widehat{Q}^\star(s,a) - Q^\star(s,a)| = |\sup_\pi \widehat{Q}^\pi(s,a) - \sup_\pi Q^\pi(s,a)| \leq \sup_\pi |\widehat{Q}^\pi(s,a) - Q^\pi(s,a)| \leq \epsilon$$

which proves the first inequality. The second inequality is left as an exercise to the reader. ∎

## 2.2 Sublinear Sample Complexity

In the previous approach, we are able to accurately estimate the value of *every* policy in the unknown MDP $M$. However, with regards to planning, we only need an accurate estimate $\widehat{Q}^\star$ of $Q^\star$, which we may hope would require less samples. Let us now see that the model based approach can be refined to obtain minmax optimal sample complexity, which we will see is sublinear in the model size.

We will state our results in terms of $N$, and recall that $N$ is the # of calls to the generative models per state-action pair, so that:

$$\# \text{ samples from generative model } = |\mathcal{S}||\mathcal{A}|N.$$

Let us start with a crude bound on the optimal action-values, which provides a sublinear rate. In the next section, we will improve upon this to obtain the minmax optimal rate.

**Proposition 2.4.** *(Crude Value Bounds) Let $\delta \geq 0$. With probability greater than $1 - \delta$,*

$$\|Q^\star - \widehat{Q}^\star\|_\infty \leq \Delta_{\delta,N}$$
$$\|Q^\star - \widehat{Q}^{\pi^\star}\|_\infty \leq \Delta_{\delta,N},$$

*where:*

$$\Delta_{\delta,N} := \frac{\gamma}{(1-\gamma)^2}\sqrt{\frac{2\log(2|\mathcal{S}||\mathcal{A}|/\delta)}{N}}$$

Note that the first inequality above shows a sublinear rate on estimating the value function. Ultimately, we are interested in the value $V^{\widehat{\pi}^\star}$ when we execute $\widehat{\pi}^\star$, not just an estimate $\widehat{Q}^\star$ of $Q^\star$. Here, by Lemma 1.11, we lose an additional horizon factor and have:

$$\|Q^\star - \widehat{Q}^{\widehat{\pi}^\star}\|_\infty \leq \frac{1}{1-\gamma}\Delta_{\delta,N}.$$

As we see in Theorem 2.6, this is improvable.

Before we provide the proof, the following lemma will be helpful throughout.

**Lemma 2.5.** *(Component-wise Bounds) We have that:*

$$Q^\star - \widehat{Q}^\star \leq \gamma(I - \gamma\widehat{P}^{\pi^\star})^{-1}(P - \widehat{P})V^\star$$
$$Q^\star - \widehat{Q}^\star \geq \gamma(I - \gamma\widehat{P}^{\widehat{\pi}^\star})^{-1}(P - \widehat{P})V^\star$$

**Proof:** For the first claim, the optimality of $\pi^\star$ in $M$ implies:

$$Q^\star - \widehat{Q}^\star = Q^{\pi^\star} - \widehat{Q}^{\widehat{\pi}^\star} \leq Q^{\pi^\star} - \widehat{Q}^{\pi^\star} = \gamma(I - \gamma\widehat{P}^{\pi^\star})^{-1}(P - \widehat{P})V^\star,$$

where we have used Lemma 2.2 in the final step. This proves the first claim.

For the second claim,

$$
\begin{aligned}
Q^\star - \widehat{Q}^\star &= Q^{\pi^\star} - \widehat{Q}^{\widehat{\pi}^\star} \\
&= (1-\gamma)\left((I - \gamma P^{\pi^\star})^{-1}r - (I - \gamma\widehat{P}^{\widehat{\pi}^\star})^{-1}r\right) \\
&= (I - \gamma\widehat{P}^{\pi^\star})^{-1}((I - \gamma\widehat{P}^{\widehat{\pi}^\star}) - (I - \gamma P^{\pi^\star}))Q^\star \\
&= \gamma(I - \gamma\widehat{P}^{\pi^\star})^{-1}(P^{\pi^\star} - \widehat{P}^{\widehat{\pi}^\star})Q^\star \\
&\geq \gamma(I - \gamma\widehat{P}^{\pi^\star})^{-1}(P^{\pi^\star} - \widehat{P}^{\pi^\star})Q^\star \\
&= \gamma(I - \gamma\widehat{P}^{\pi^\star})^{-1}(P - \widehat{P})V^\star,
\end{aligned}
$$

where the inequality follows from $\widehat{P}^{\widehat{\pi}^\star}Q^\star \leq \widehat{P}^{\pi^\star}Q^\star$, due to the optimality of $\pi^\star$. This proves the second claim. ∎

**Proof:** Following from the simulation lemma (Lemma 2.2) and Lemma 2.3, we have:

$$\|Q^\star - \widehat{Q}^{\pi^\star}\|_\infty \leq \frac{\gamma}{1-\gamma}\|(P - \widehat{P})V^\star\|_\infty.$$

Also, the previous lemma, implies that:

$$\|Q^\star - \widehat{Q}^\star\|_\infty \leq \frac{\gamma}{1-\gamma}\|(P - \widehat{P})V^\star\|_\infty$$

By applying Hoeffding's inequality and the union bound,

$$\|(P - \widehat{P})V^\star\|_\infty = \max_{s,a}|\mathbb{E}_{s'\sim P(\cdot|s,a)}[V^\star(s')] - \mathbb{E}_{s'\sim\widehat{P}(\cdot|s,a)}[V^\star(s')]| \leq \frac{1}{1-\gamma}\sqrt{\frac{2\log(2|\mathcal{S}||\mathcal{A}|/\delta)}{N}}$$

which holds with probability greater than $1 - \delta$. This completes the proof. ∎

## 2.3 Minmax Optimal Sample Complexity (and the Model Based Approach)

We now see that the model based approach is minmax optimal, for both the discounted case and the finite horizon setting.

### 2.3.1 The Discounted Case

**Upper bounds.** The following theorem refines our crude bound on $\widehat{Q}^\star$.

**Theorem 2.6.** *For $\delta \geq 0$ and for an appropriately chosen absolute constant c, we have that:*

- *(Value estimation) With probability greater than $1 - \delta$,*

$$\|Q^\star - \widehat{Q}^\star\|_\infty \leq \gamma \sqrt{\frac{c}{(1-\gamma)^3} \frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}} + \frac{c\gamma}{(1-\gamma)^3} \frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N} \,.$$

- *(Sub-optimality) If $N \geq \frac{1}{(1-\gamma)^2}$, then with probability greater than $1 - \delta$,*

$$\|Q^\star - Q^{\widehat{\pi}^\star}\|_\infty \leq \gamma \sqrt{\frac{c}{(1-\gamma)^3} \frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}}.$$

This immediately provides the following corollary.

**Corollary 2.7.** *Provided that $\epsilon \leq 1$ and that*

$$\text{\# samples from generative model } = |\mathcal{S}||\mathcal{A}|N \geq \frac{c|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3} \frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{\epsilon^2},$$

*then with probability greater than $1 - \delta$,*
$$\|Q^\star - \widehat{Q}^\star\|_\infty \leq \epsilon.$$
*Furthermore, provided that $\epsilon \leq \sqrt{\frac{1}{1-\gamma}}$ and that*

$$\text{\# samples from generative model } = |\mathcal{S}||\mathcal{A}|N \geq \frac{c|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3} \frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{\epsilon^2},$$

*then with probability greater than $1 - \delta$,*
$$\|Q^\star - Q^{\widehat{\pi}^\star}\|_\infty \leq \epsilon.$$

We only prove the first claim in Theorem 2.6 on the estimation accuracy. With regards to the sub-optimality, note that Theorem 1.11 already implies a sub-optimality gap, though with an amplification of the estimation error by $2/(1 - \gamma)$. The argument for the improvement provided in the second claim is more involved (See Section 2.6 for further discussion).

**Lower Bounds.** Let us say that an estimation algorithm $\mathcal{A}$, which is a map from samples to an estimate $\widehat{Q}^\star$, is $(\epsilon, \delta)$-good on MDP $M$ if $\|Q^\star - \widehat{Q}^\star\|_\infty \leq \epsilon$ holds with probability greater than $1 - \delta$.

**Theorem 2.8.** *There exists $\epsilon_0, \delta_0, c$ and a set of MDPs $\mathcal{M}$ such that for $\epsilon \in (0, \epsilon_0)$ and $\delta \in (0, \delta_0)$ if algorithm $\mathcal{A}$ is $(\epsilon, \delta)$-good on all $M \in \mathcal{M}$, then $\mathcal{A}$ must use a number of samples that is lower bounded as follows*

$$\text{\# samples from generative model } \geq \frac{c}{(1-\gamma)^3} \frac{|\mathcal{S}||\mathcal{A}| \log(c|\mathcal{S}||\mathcal{A}|/\delta)}{\epsilon^2} \,.$$

In other words, this theorem shows that the model based approach minmax optimal.

## 2.3.2 Finite Horizon Setting

Recall the setting of finite horizon MDPs defined in Section 1.2. Again, we can consider the most naive approach to learning (when we have access to a generative model): suppose we call our simulator $N$ times for every $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, i.e. we obtain $N$ i.i.d. samples where $s' \sim P_h(\cdot|s, a)$, for every $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$. Note that the total number of observed transitions is $H|\mathcal{S}||\mathcal{A}|N$.

**Upper bounds.** The following theorem provides an upper bound on the model based approach.

**Theorem 2.9.** *For $\delta \geq 0$ and with probability greater than $1 - \delta$, we have that:*

- *(Value estimation)*

$$\|Q_0^\star - \widehat{Q}_0^\star\|_\infty \leq cH\sqrt{\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}} + cH\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N} \,,$$

- *(Sub-optimality)*

$$\|Q_0^\star - Q_0^{\widehat{\pi}^\star}\|_\infty \leq cH\sqrt{\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}} + cH\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N} \,,$$

*where $c$ is an absolute constant.*

Note that the above bound requires $N$ to be $O(H^2)$ in order to achieve an $\epsilon$-optimal policy, while in the discounted case, we require $N$ to be $O(1/(1 - \gamma)^3)$ for the same guarantee. While this may seem like an improvement by a horizon factor, recall that for the finite horizon case, $N$ corresponds to observing $O(H)$ more transitions than in the discounted case.

**Lower Bounds.** In the minmax sense of Theorem 2.8, the previous upper bound provided by the model based approach for the finite horizon setting achieves the minmax optimal sample complexity.

## 2.4 Analysis

We now prove (the first claim in) Theorem 2.6.

### 2.4.1 Variance Lemmas

The key to the sharper analysis is to more sharply characterize the variance in our estimates.

Denote the variance of any real valued $f$ under a distribution $\mathcal{D}$ as:

$$\mathrm{Var}_\mathcal{D}(f) := E_{x\sim\mathcal{D}}[f(x)^2] - (E_{x\sim\mathcal{D}}[f(x)])^2$$

Slightly abusing the notation, for $V \in R^{|\mathcal{S}|}$, we define the vector $\mathrm{Var}_P(V) \in R^{|\mathcal{S}||\mathcal{A}|}$ as:

$$\mathrm{Var}_P(V)(s,a) := \mathrm{Var}_{P(\cdot|s,a)}(V)$$

Equivalently,

$$\mathrm{Var}_P(V) = P(V)^2 - (PV)^2 \,.$$

Now we characterize a relevant deviation in terms of the its variance.

**Lemma 2.10.** *Let $\delta > 0$. With probability greater than $1 - \delta$,*

$$|(P - \widehat{P})V^\star| \leq \sqrt{\frac{2\log(2|\mathcal{S}||\mathcal{A}|/\delta)}{N}}\sqrt{\mathrm{Var}_P(V^\star)} + \frac{1}{1 - \gamma}\frac{2\log(2|\mathcal{S}||\mathcal{A}|/\delta)}{3N}\mathbb{1} \,.$$

26

**Proof:** The claims follows from Bernstein's inequality along with a union bound over all state-action pairs. ∎

The key ideas in the proof are in how we bound $\|(I - \gamma \widehat{P}^{\pi^\star})^{-1}\sqrt{\mathrm{Var}_P(V^\star)}\|_\infty$ and $\|(I - \gamma \widehat{P}^{\widehat{\pi}^\star})^{-1}\sqrt{\mathrm{Var}_P(V^\star)}\|_\infty$.

It is helpful to define $\Sigma_M^\pi$ as the variance of the discounted reward, i.e.

$$\Sigma_M^\pi(s, a) := \mathbb{E}\left[\left(\sum_{t=0}^\infty \gamma^t r(s_t, a_t) - Q_M^\pi(s, a)\right)^2 \middle| s_0 = s, a_0 = a\right]$$

where the expectation is induced under the trajectories induced by $\pi$ in $M$. It is straightforward to verify that $\|\Sigma_M^\pi\|_\infty \leq \gamma^2/(1-\gamma)^2$.

The following lemma shows that $\Sigma_M^\pi$ satisfies a Bellman consistency condition.

**Lemma 2.11.** *(Bellman consistency of $\Sigma$) For any MDP $M$,*

$$\Sigma_M^\pi = \gamma^2 \mathrm{Var}_P(V_M^\pi) + \gamma^2 P^\pi \Sigma_M^\pi \tag{0.1}$$

*where $P$ is the transition model in MDP $M$.*

The proof is left as an exercise to the reader.

**Lemma 2.12.** *(Weighted Sum of Deviations) For any policy $\pi$ and MDP $M$,*

$$\left\|(I - \gamma P^\pi)^{-1}\sqrt{\mathrm{Var}_P(V_M^\pi)}\right\|_\infty \leq \sqrt{\frac{2}{(1-\gamma)^3}},$$

*where $P$ is the transition model of $M$.*

**Proof:** Note that $(1 - \gamma)(I - \gamma P^\pi)^{-1}$ is matrix whose rows are a probability distribution. For a positive vector $v$ and a distribution $\nu$ (where $\nu$ is vector of the same dimension of $v$), Jensen's inequality implies that $\nu \cdot \sqrt{v} \leq \sqrt{\nu \cdot v}$. This implies:

$$
\begin{aligned}
\|(I - \gamma P^\pi)^{-1}\sqrt{v}\|_\infty &= \frac{1}{1-\gamma}\|(1-\gamma)(I - \gamma P^\pi)^{-1}\sqrt{v}\|_\infty \\
&\leq \sqrt{\left\|\frac{1}{1-\gamma}(I - \gamma P^\pi)^{-1}v\right\|_\infty} \\
&\leq \sqrt{\left\|\frac{2}{1-\gamma}(I - \gamma^2 P^\pi)^{-1}v\right\|_\infty}.
\end{aligned}
$$

where we have used that $\|(I - \gamma P^\pi)^{-1}v\|_\infty \leq 2\|(I - \gamma^2 P^\pi)^{-1}v\|_\infty$ (which we will prove shortly). The proof is completed as follows: by Equation 0.1, $\Sigma_M^\pi = \gamma^2(I - \gamma^2 P^\pi)^{-1}\mathrm{Var}_P(V_M^\pi)$, so taking $v = \mathrm{Var}_P(V_M^\pi)$ and using that $\|\Sigma_M^\pi\|_\infty \leq \gamma^2/(1-\gamma)^2$ completes the proof.

Finally, to see that $\|(I - \gamma P^\pi)^{-1}v\|_\infty \leq 2\|(I - \gamma^2 P^\pi)^{-1}v\|_\infty$, observe:

$$
\begin{aligned}
\|(I - \gamma P^\pi)^{-1}v\|_\infty &= \|(I - \gamma P^\pi)^{-1}(I - \gamma^2 P^\pi)(I - \gamma^2 P^\pi)^{-1}v\|_\infty \\
&= \|(I - \gamma P^\pi)^{-1}\left((1-\gamma)I + \gamma(I - \gamma P^\pi)\right)(I - \gamma^2 P^\pi)^{-1}v\|_\infty \\
&= \|\left((1-\gamma)(I - \gamma P^\pi)^{-1} + \gamma I\right)(I - \gamma^2 P^\pi)^{-1}v\|_\infty \\
&\leq (1-\gamma)\|(I - \gamma P^\pi)^{-1}(I - \gamma^2 P^\pi)^{-1}v\|_\infty + \gamma\|(I - \gamma^2 P^\pi)^{-1}v\|_\infty \\
&\leq \frac{1-\gamma}{1-\gamma}\|(I - \gamma^2 P^\pi)^{-1}v\|_\infty + \gamma\|(I - \gamma^2 P^\pi)^{-1}v\|_\infty \\
&\leq 2\|(I - \gamma^2 P^\pi)^{-1}v\|_\infty
\end{aligned}
$$

which proves the claim. ∎

### 2.4.2 Completing the proof

**Lemma 2.13.** *Let* $\delta \geq 0$. *With probability greater than* $1 - \delta$, *we have:*

$$\mathrm{Var}_P(V^\star) \leq 2\mathrm{Var}_{\widehat{P}}(\widehat{V}^{\pi^\star}) + \Delta'_{\delta,N}\mathbb{1}$$
$$\mathrm{Var}_P(V^\star) \leq 2\mathrm{Var}_{\widehat{P}}(\widehat{V}^\star) + \Delta'_{\delta,N}\mathbb{1}$$

*where*

$$\Delta'_{\delta,N} := \frac{1}{(1-\gamma)^2}\sqrt{\frac{18\log(6|\mathcal{S}||\mathcal{A}|/\delta)}{N}} + \frac{1}{(1-\gamma)^4}\frac{4\log(6|\mathcal{S}||\mathcal{A}|/\delta)}{N}.$$

**Proof:** By definition,

$$\begin{aligned}
\mathrm{Var}_P(V^\star) &= \mathrm{Var}_P(V^\star) - \mathrm{Var}_{\widehat{P}}(V^\star) + \mathrm{Var}_{\widehat{P}}(V^\star) \\
&= P(V^\star)^2 - (PV^\star)^2 - \widehat{P}(V^\star)^2 + (\widehat{P}V^\star)^2 + \mathrm{Var}_{\widehat{P}}(V^\star) \\
&= (P - \widehat{P})(V^\star)^2 - \left((PV^\star)^2 - (\widehat{P}V^\star)^2\right) + \mathrm{Var}_{\widehat{P}}(V^\star)
\end{aligned}$$

Now we bound each of these terms with Hoeffding's inequality and the union bound. For the first term, with probability greater than $1 - \delta$,

$$\|(P - \widehat{P})(V^\star)^2\|_\infty \leq \frac{1}{(1-\gamma)^2}\sqrt{\frac{2\log(2|\mathcal{S}||\mathcal{A}|/\delta)}{N}}.$$

For the second term, again with probability greater than $1 - \delta$,

$$\|(PV^\star)^2 - (\widehat{P}V^\star)^2\|_\infty \leq \|PV^\star + \widehat{P}V^\star\|_\infty\|PV^\star - \widehat{P}V^\star\|_\infty$$
$$\leq \frac{2}{1-\gamma}\|(P - \widehat{P})V^\star\|_\infty \leq \frac{2}{(1-\gamma)^2}\sqrt{\frac{2\log(2|\mathcal{S}||\mathcal{A}|/\delta)}{N}}.$$

where we have used that $(\cdot)^2$ is a component-wise operation in the second step. For the last term:

$$\begin{aligned}
\mathrm{Var}_{\widehat{P}}(V^\star) &= \mathrm{Var}_{\widehat{P}}(V^\star - \widehat{V}^{\pi^\star} + \widehat{V}^{\pi^\star}) \\
&\leq 2\mathrm{Var}_{\widehat{P}}(V^\star - \widehat{V}^{\pi^\star}) + 2\mathrm{Var}_{\widehat{P}}(\widehat{V}^{\pi^\star}) \\
&\leq 2\|V^\star - \widehat{V}^{\pi^\star}\|_\infty^2 + 2\mathrm{Var}_{\widehat{P}}(\widehat{V}^{\pi^\star}) \\
&= 2\Delta_{\delta,N}^2 + 2\mathrm{Var}_{\widehat{P}}(\widehat{V}^{\pi^\star}).
\end{aligned}$$

where $\Delta_{\delta,N}$ is defined in Proposition 2.4. To obtain a cumulative probability of error less than $\delta$, we replace $\delta$ in the above claims with $\delta/3$. Combining these bounds completes the proof of the first claim. The argument in the above display also implies that $\mathrm{Var}_{\widehat{P}}(V^\star) \leq 2\Delta_{\delta,N}^2 + 2\mathrm{Var}_{\widehat{P}}(\widehat{V}^\star)$ which proves the second claim. ∎

Using Lemma 2.10 and 2.13, we have the following corollary.

**Corollary 2.14.** *Let* $\delta \geq 0$. *With probability greater than* $1 - \delta$, *we have:*

$$|(P - \widehat{P})V^\star| \leq c\sqrt{\frac{\mathrm{Var}_{\widehat{P}}(\widehat{V}^{\pi^\star})\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}} + \Delta''_{\delta,N}\mathbb{1}$$

$$|(P - \widehat{P})V^\star| \leq c\sqrt{\frac{\mathrm{Var}_{\widehat{P}}(\widehat{V}^\star)\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}} + \Delta''_{\delta,N}\mathbb{1},$$

*where*

$$\Delta''_{\delta,N} := c\frac{1}{1-\gamma}\left(\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}\right)^{3/4} + \frac{c}{(1-\gamma)^2}\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N},$$

*and where c is an absolute constant.*

28

**Proof:**(of Theorem 2.6) The proof consists of bounding the terms in Lemma 2.5. We have:

$$
\begin{aligned}
Q^\star - \widehat{Q}^\star \;\leq\;& \gamma\|(I - \gamma\widehat{P}^{\pi^\star})^{-1}(P - \widehat{P})V^\star\|_\infty \\
\leq\;& c\gamma\sqrt{\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}}\|(I - \gamma\widehat{P}^{\pi^\star})^{-1}\sqrt{\mathrm{Var}_{\widehat{P}}(\widehat{V}^{\pi^\star})}\|_\infty + \frac{c\gamma}{(1-\gamma)^2}\left(\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}\right)^{3/4} \\
& + \frac{c\gamma}{(1-\gamma)^3}\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N} \\
\leq\;& \gamma\sqrt{\frac{2}{(1-\gamma)^3}}\sqrt{\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}} + \frac{c\gamma}{(1-\gamma)^2}\left(\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}\right)^{3/4} + \frac{c\gamma}{(1-\gamma)^3}\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N} \\
\leq\;& 3\gamma\sqrt{\frac{1}{(1-\gamma)^3}}c\sqrt{\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}} + 2\frac{c\gamma}{(1-\gamma)^3}\frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}\,,
\end{aligned}
$$

where the first step uses Corollary 2.14; the second uses Lemma 2.12; and the last step uses that $2ab \leq a^2 + b^2$ (and choosing $a, b$ appropriately). The proof of the lower bound is analogous. Taking a different absolute constant completes the proof. ∎

## 2.5   Scalings and Effective Horizon Dependencies

It will be helpful to more intuitively understand why $1/(1 - \gamma)^3$ is the effective horizon dependency one might hope to expect, from a dimensional analysis viewpoint. Due to that $Q^\star$ is a quantity that is as large as $1/(1 - \gamma)$, to account for this scaling, it is natural to look at obtaining relative accuracy.

In particular, if

$$
N \geq \frac{c}{1 - \gamma}\frac{|\mathcal{S}||\mathcal{A}|\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{\epsilon^2},
$$

then with probability greater than $1 - \delta$, then

$$
\|Q^\star - Q^{\widehat{\pi}^\star}\|_\infty \leq \frac{\epsilon}{1 - \gamma}, \text{ and } \|Q^\star - \widehat{Q}^\star\|_\infty \leq \frac{\epsilon}{1 - \gamma}.
$$

(provided that $\epsilon \leq \sqrt{1 - \gamma}$ using Theorem 2.6). In other words, if we had normalized the value functions [1], then for additive accuracy (on our normalized value functions) our sample size would scale linearly with the effective horizon.

## 2.6   Bibliographic Remarks and Further Readings

The notion of a generative model was first introduced in [Kearns and Singh, 1999], which made the argument that, up to horizon factors and logarithmic factors, both model based methods and model free methods are comparable. [Kakade, 2003] gave an improved version of this rate (analogous to the crude bounds seen here).

The first claim in Theorem 2.6 is due to [Azar et al., 2013], and the proof in this section largely follows this work. Improvements are possible with regards to bounding the quality of $\widehat{\pi}^\star$; here, Theorem 2.6 shows that the model based approach is near optimal even for policy itself; showing that the quality of $\widehat{\pi}^\star$ does suffer any amplification factor of $1/(1-\gamma)$. [Sidford et al., 2018] provides the first proof of this improvement using a variance reduction algorithm with value iteration. The second claim in Theorem 2.6 is due to [Agarwal et al., 2020c], which shows that the naive model based approach is sufficient. The lower bound in Theorem 2.8 is due to [Azar et al., 2013].

---

[1]Rescaling the value functions by multiplying by $(1-\gamma)$, i.e. $Q^\pi \leftarrow (1-\gamma)Q^\pi$, would keep the values bounded between 0 and 1. Throughout,

We also remark that we may hope for the sub-optimality bounds (on the value of the argmax policy) to hold up to for "large" $\epsilon$, i.e. up to $\epsilon \leq 1/(1-\gamma)$ (see the second claim in Theorem 2.6). Here, the work in [Li et al., 2020] shows this limit is achievable, albeit with a slightly different algorithm where they introduce perturbations. It is currently an open question if the naive model based approach also achieves the non-asymptotic statistical limit.

This chapter also provided results, without proof, on the optimal sample complexity in the finite horizon setting (see Section 2.3.2). The proof of this claim would also follow from the line of reasoning in [Azar et al., 2013], with the added simplification that the sub-optimality analysis is simpler in the finite-horizon setting with time-dependent transition matrices (e.g. see [Yin et al., 2021]).

---

this book it is helpful to understand sample size with regards to normalized quantities.

# Chapter 3

# Linear Bellman Completeness

Up to now we have focussed on "tabular" MDPs, where the sample complexity (and the computational complexity) scaled polynomially in the size of the state and action spaces. Ultimately, we seek to obtain methods which are applicable to cases where number of states and actions is large (or, possibly, countably or uncountably infinite).

In this chapter, we will consider one of the simplest conditions, where such a result is possible. In particular, we will consider a setting where we have a set of features which satisfy the *linear Bellman completion* condition, and we show that, with access to a generative model, there is a simple algorithm that learns a near optimal policy with polynomial sample complexity in the dimension of these features.

The linear Bellman completeness concept and first analysis was due to [Munos, 2005]. This chapter studies this concept in the setting of finite horizon MDPs, where we have access to a generative model.

## 3.1   The Linear Bellman Completeness Condition

We will work with given a feature mapping $\phi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^d$, where we assume the following property is true: given any linear function $f(s, a) := \theta^\top \phi(s, a)$, we have that the Bellman operator applied to $f(s, a)$ also returns a linear function with respect to $\phi$. Precisely, we have:

**Definition 3.1** (Linear Bellman Completeness). We say the features $\phi$ satisfy the *linear Bellman completeness* property if for all $\theta \in \mathbb{R}^d$ and $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, there exists $w \in \mathbb{R}^d$ such that:

$$w^\top \phi(s, a) = r(s, a) + \mathbb{E}_{s' \sim P_h(s,a)} \max_{a'} \theta^\top \phi(s', a').$$

As $w$ depends on $\theta$, we use the notation $\mathcal{T}_h : \mathbb{R}^d \mapsto \mathbb{R}^d$ to represent such a $w$, i.e., $w := \mathcal{T}_h(\theta)$ in the above equation.

Note that the above implies that $r(s, a)$ is in the span of $\phi$ (to see this, take $\theta = 0$). Furthermore, it also implies that $Q_h^\star(s, a)$ is linear in $\phi$, i.e., there exists $\theta_h^\star$ such that $Q_h^\star(s, a) = (\theta_h^\star)^\top \phi(s, a)$. We can easily see that tabular MDP is captured by the linear Bellman completeness with $\phi(s, a) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ being a one-hot encoding vector with zeros everywhere except one at the entry corresponding to state-action $(s, a)$. In Chapters 8 and 9, we will provide precise examples of models which satisfy the linear Bellman completeness condition.

We focus on the generative model setting, i.e., where we can input any $(s, a)$ pair to obtain a next state sample $s' \sim P(s, a)$ and reward $r(s, a)$. Our goal here is to design an algorithm that finds a policy $\widehat{\pi}$ such that $V^{\widehat{\pi}} \geq V^\star - \epsilon$, with number of samples scaling polynomially with respect to all relevant parameters $d, H, 1/\epsilon$.

## 3.2 The LSVI Algorithm

We first present the least square value iteration (LSVI) algorithm (Alg. 1) here. The algorithm takes $H$ many datasets $\mathcal{D}_0, \ldots, \mathcal{D}_{H-1}$ as inputs, where each $\mathcal{D}_h = \{s, a, r, s'\}$.

---

**Algorithm 1** Least Square Value Iteration

---

1: **Input**: $\mathcal{D}_0, \ldots, \mathcal{D}_{H-1}$
2: Set $V_H(s) = 0$ for all $s \in \mathcal{S}$
3: **for** $h = H - 1 \to 0$ **do**
4:     Solve least squares

$$\theta_h = \arg\min_{\theta} \sum_{s,a,r,s' \in \mathcal{D}_h} \left( \theta^\top \phi(s, a) - r - V_{h+1}(s') \right)^2 \tag{0.1}$$

5:     Set $V_h(s) = \max_{a \in \mathcal{A}} \theta_h^\top \phi(s, a), \forall s$
6: **end for**
7: **Return:** $\{\widehat{\pi}_h(s)\}_{h=0}^{H-1}$ where $\widehat{\pi}_h(s) := \arg\max_a \theta_h^\top \phi(s, a), \forall s, h$.

---

For each time step, LSVI estimate $Q_h^\star(s, a)$ via $\theta_h^\top \phi(s, a)$ with $\theta_h$ computed from a least square problem.

To make sure the algorithm succeeds in terms of finding a near optimal policy, we need to design the datasets $\mathcal{D}_0, \ldots, \mathcal{D}_{H-1}$. Intuitively, we may want to make sure that in each $\mathcal{D}_h$, we have $\sum_{s,a \in \mathcal{D}_h} \phi(s, a)\phi(s, a)^\top$ being full rank, so that least square has closed-form solution, and $\theta_h$ has a good generalization bound. In the next section, we will leverage the generative model setting and the D-optimal design to construct such datasets.

## 3.3 LSVI with D-Optimal Design

In this section, we start by introducing the D-optimal design and its properties. We then use the D-optimal design to construct a dataset and give a generalization bound for the solution to the least squares problem on the constructed dataset. Finally, we present the sample complexity bound for LSVI and its analysis.

We will use the notation $\|x\|_M^2 = x^\top M x$ for a matrix $M$ and vector $x$ of appropriate dimensions.

### 3.3.1 D-Optimal Design

We now specify a sampling distribution which, roughly speaking, ensures good coverage (in a spectral sense) over our feature set.

**Theorem 3.2.** *Suppose $\mathcal{X} \subset \mathbb{R}^d$ is a compact set (and full dimensional). There exists a distribution $\rho$ on $\mathcal{X}$ such that:*

- *$\rho$ is supported on at most $d(d + 1)/2$ points (all of which are in $\mathcal{X}$).*

- *Define*

$$\Sigma = \mathbb{E}_{x \sim \rho} x x^\top.$$

*We have that for all $x \in \mathcal{X}$,*

$$\|x\|_{\Sigma^{-1}}^2 \leq d.$$

*The distribution $\rho$ is referred to as the D-optimal design.*

With these two properties, one can show that for any $x \in \mathcal{X}$, it can be written as a linear combination of the points on the support of $\rho$, i.e., $x = \sum_{x_i \in \text{support}(\rho)} \alpha_i \rho(x_i) x_i$, where $\|\alpha\|_2^2 \leq d$.

The following specifies an explicit construction: $\rho$ is the following maximizer:

$$\rho \in \text{argmax}_{\nu \in \Delta(\mathcal{X})} \ln \det \left( \mathbb{E}_{x \sim \nu} x x^\top \right). \tag{0.2}$$

(and there exists an optimizer which is supported on at most $d(d+1)/2$ points). The design $\rho$ has a geometric interpretation: the centered ellipsoid $\mathcal{E} = \{v : \|v\|_{\Sigma^{-1}}^2 \leq d\}$ is the unique minimum volume centered ellipsoid containing $\mathcal{X}$ We do not provide the proof here (See Section 3.5).

In our setting, we will utilize the $D$-optimal design on the set $\Phi := \{\phi(s,a) : s, a \in \mathcal{S} \times \mathcal{A}\}$.

### 3.3.2 Performace Guarantees

Now we explain how we can construct a dataset $\mathcal{D}_h$ for $h = 0, \ldots, H-1$ using the D-optimal design $\rho$ and the generative model:

- At $h$, for each $s, a \in \text{support}(\rho)$, we sample $n_{s,a} := \lceil \rho(s,a) N \rceil$ many next states independently from $P_h(\cdot | s, a)$; combine all $(s, a, r(s,a), s')$ and denote the whole dataset as $\mathcal{D}_h$

- Repeat for all $h \in [0, \ldots, H-1]$ to construct $\mathcal{D}_0, \ldots, \mathcal{D}_{H-1}$.

The key property of the dataset $\mathcal{D}_h$ is that the empirical covariance matrix $\sum_{s,a \in \mathcal{D}_h} \phi(s,a)\phi(s,a)^\top$ is full rank. Denote $\Lambda_h = \sum_{s,a \in \mathcal{D}_h} \phi(s,a)\phi(s,a)^\top$, we can verify that

$$\Lambda_h = \sum_{s,a \in \mathcal{D}} \phi(s,a)\phi(s,a)^\top \succcurlyeq N\Sigma \tag{0.3}$$

where the inequality comes from the fact that in $\mathcal{D}_h$, for each $s, a \in \text{support}(\rho)$, we have $\lceil \rho(s,a) N \rceil$ many copies of it. The D-optimal design ensures that $\Lambda$ covers all $\phi(s,a)$ in the sense that $\max_{s,a} \phi(s,a)^\top \Lambda_h^{-1} \phi(s,a)^\top \leq d$ due to the property of the D-optimal design. This is the key property that ensures the solution of the least squares on $\mathcal{D}_h$ is accurate globally.

Now we state the main sample complexity bound theorem for LSVI.

**Theorem 3.3** (Sample Complexity of LSVI). *Fix $\delta \in (0,1)$ and $\epsilon \in (0,1)$. Set parameter $N := \left\lceil \frac{64H^6 d^2 \ln(1/\delta)}{\epsilon^2} \right\rceil$. With probability at least $1 - \delta$, Algorithm 1 outputs $\widehat{\pi}$ such that:*

$$\mathbb{E}_{s \sim \mu} V_0^\star(s) - \mathbb{E}_{s \sim \mu} V_0^{\widehat{\pi}}(s) \leq \epsilon,$$

*with total number of samples $H \left( d^2 + \frac{64H^6 d^2 \ln(1/\delta)}{\epsilon^2} \right)$.*

### 3.3.3 Analysis

To prove the theorem, we first show that if we had $\{\widehat{Q}_h\}_h$ such that the Bellman residual $\|\widehat{Q}_h - \mathcal{T}_h \widehat{Q}_{h+1}\|_\infty$ is small, then the performance of the greedy policies with respect to $\widehat{Q}_h$ is already near optimal. Note that this result is general and has nothing to do with the LSVI algorithm.

**Lemma 3.4.** *Assume that for all $h$, we have $\left\| \widehat{Q}_h - \mathcal{T}_{h+1} \widehat{Q}_{h+1} \right\|_\infty \leq \epsilon$. Then we have:*

33

1. *Accuracy of $\widehat{Q}_h$: for all $h$:* $\left\|\widehat{Q}_h - Q_h^\star\right\|_\infty \leq (H-h)\epsilon$,

2. *Policy performance: for $\widehat{\pi}_h(s) := \arg\max_a \widehat{Q}_h(s,a)$, we have $\left|V^{\widehat{\pi}} - V^\star\right| \leq 2H^2\epsilon$.*

**Proof:** The proof of this lemma is elementary, which is analog to what we have proved in Value Iteration in the discounted setting. For completeness, we provide a proof here.

Starting from $Q_H(s,a) = 0, \forall s,a$, we have $\mathcal{T}_{H-1}Q_H = r$ by definition. The condition implies that $\|\widehat{Q}_{H-1} - r\|_\infty \leq \epsilon$, which means that $\|\widehat{Q}_{H-1} - Q_{H-1}^\star\|_\infty \leq \epsilon$. This proves the base case.

Our inductive hypothesis is that $\|\widehat{Q}_{h+1} - Q_{h+1}^\star\|_\infty \leq (H-h-1)\epsilon$. Note that:

$$|\widehat{Q}_h(s,a) - Q_h^\star(s,a)| \leq |\widehat{Q}_h(s,a) - \mathcal{T}_h\widehat{Q}_{h+1}(s,a)| + |\mathcal{T}_h\widehat{Q}_{h+1}(s,a) - Q_h^\star(s,a)|$$
$$\leq \epsilon + |\mathcal{T}_h\widehat{Q}_{h+1}(s,a) - \mathcal{T}_hQ_{h+1}^\star(s,a)| \leq \epsilon + (H-h-1)\epsilon = (H-h)\epsilon,$$

which concludes the proof for the first claim.

For the second claim, Starting from time step $H-1$, for any $s$, we have:

$$
\begin{aligned}
V_{H-1}^{\widehat{\pi}}(s) - V_{H-1}^\star(s) &= Q_{H-1}^{\widehat{\pi}}(s, \widehat{\pi}_{H-1}(s)) - Q_{H-1}^\star(s, \pi^\star(s)) \\
&= Q_{H-1}^{\widehat{\pi}}(s, \widehat{\pi}_{H-1}(s)) - Q_{H-1}^\star(s, \widehat{\pi}_{H-1}(s)) + Q_{H-1}^\star(s, \widehat{\pi}_{H-1}(s)) - Q_{H-1}^\star(s, \pi^\star(s)) \\
&= Q_{H-1}^\star(s, \widehat{\pi}_{H-1}(s)) - Q_{H-1}^\star(s, \pi^\star(s)) \\
&\geq Q_{H-1}^\star(s, \widehat{\pi}_{H-1}(s)) - \widehat{Q}_{H-1}(s, \widehat{\pi}_{H-1}(s)) + \widehat{Q}_{H-1}(s, \pi^\star(s)) - Q_{H-1}^\star(s, \pi^\star(s)) \\
&\geq 2\epsilon,
\end{aligned}
$$

where the third equality uses the fact that $Q_{H-1}^{\widehat{\pi}}(s,a) = Q_{H-1}^\star(s,a) = r(s,a)$, the first inequality uses the definition of $\widehat{\pi}_{H-1}$ and $\widehat{Q}_{H-1}(s, \widehat{\pi}_{H-1}(s)) \geq \widehat{Q}_{H-1}(s,a), \forall a$. This concludes the base case. For time step $h+1$, our inductive hypothesis is defined as $V_{h+1}^{\widehat{\pi}}(s) - V_{h+1}^\star(s) \geq -2(H-h-1)H\epsilon$. For time step $h$, we have:

$$
\begin{aligned}
V_h^{\widehat{\pi}}(s) - V_h^\star(s) &= Q_h^{\widehat{\pi}}(s, \widehat{\pi}_h(s)) - Q_h^\star(s, \pi^\star(s)) \\
&= Q_h^{\widehat{\pi}}(s, \widehat{\pi}_h(s)) - Q_h^\star(s, \widehat{\pi}_h(s)) + Q_h^\star(s, \widehat{\pi}_h(s)) - Q_h^\star(s, \pi^\star(s)) \\
&= \mathbb{E}_{s' \sim P_h(s, \widehat{\pi}_h(s))}\left[V_{h+1}^{\widehat{\pi}}(s') - V_{h+1}^\star(s')\right] + Q_h^\star(s, \widehat{\pi}_h(s)) - Q_h^\star(s, \pi^\star(s)) \\
&\geq -2(H-h-1)\epsilon + Q_h^\star(s, \widehat{\pi}_h(s)) - \widehat{Q}_h(s, \widehat{\pi}_h(s)) + \widehat{Q}_h(s, \pi_h^\star(s)) - Q_h^\star(s, \pi^\star(s)) \\
&\geq -2(H-h-1)H\epsilon - 2(H-h)\epsilon \geq -2(H-h)H\epsilon.
\end{aligned}
$$

This concludes that for any $s$, at time step $h=0$, we have $V_0^{\widehat{\pi}}(s) - V_0^\star(s) \geq -2H^2\epsilon$. ∎

To conclude the proof, what we left to show is that the $\widehat{Q}_h$ that is returned by LSVI satisfies the condition in the above lemma. Note that $\widehat{Q}_h(s,a) = \widehat{\theta}_h^\top \phi(s,a)$, and $\mathcal{T}_h\widehat{Q}_{h+1}$ is indeed the Bayes optimal solution of the least squares in Eq. 0.1, and via the Bellman completion assumption, we have $\mathcal{T}_h\widehat{Q}_{h+1}(s,a) = \mathcal{T}_h(\widehat{\theta}_{h+1})^\top \phi(s,a), \forall s,a$. With the constructed $\mathcal{D}_h$ based on the D-optimal design, Linear regression immediately gives us a generalization bound on $\|\widehat{Q}_h - \mathcal{T}_h\widehat{Q}_{h+1}\|_\infty$. The following theorem formalize the above statement.

**Lemma 3.5.** *Fix $\delta \in (0,1)$ and $\epsilon \in (0,1)$, set $N = \left\lceil \frac{16H^2d^2\ln(H/\delta)}{\epsilon^2} \right\rceil$. With probability at least $1-\delta$, for all $h$, we have:*

$$\forall h: \quad \left\|\widehat{Q}_h - \mathcal{T}_h\widehat{Q}_{h+1}\right\|_\infty \leq \epsilon.$$

**Proof:** The proof uses a standard result for ordinary least squares (see Theorem A.10 in the Appendix). Consider a time step $h$. In time step $h$, we perform linear regression from $\phi(s, a)$ to $r(s, a) + \max_{a'} \hat{\theta}_{h+1}^\top \phi(s', a')$. Note that the Bayes optimal here is

$$\mathbb{E}_{s' \sim P_h(s,a)} \left[ r(s, a) + \max_{a'} \hat{\theta}_{h+1}^\top \phi(s', a') \right] = \mathcal{T}_h(\hat{\theta}_{h+1})^\top \phi(s, a).$$

Also we can show that the noise $\epsilon_{s,a} := \left( r(s, a) + \max_{a'} \widehat{Q}_{h+1}(s', a') - \mathcal{T}_h \widehat{Q}_h(s, a) \right)$ is bounded by $2H$, and also note that the noises are independent, and has mean zero: $\mathbb{E}_{s' \sim P_h(s,a)} \left[ r(s, a) + \max_{a'} \widehat{Q}_{h+1}(s', a') - \mathcal{T}_h \widehat{Q}_h(s, a) \right] = 0$. Using Theorem A.10, we have that with probability at least $1 - \delta$,

$$\left\| \hat{\theta}_h - \mathcal{T}_h(\hat{\theta}_{h+1}) \right\|_{\Lambda_h}^2 \leq 4H^2 \left( d + \sqrt{d \ln(1/\delta)} + 2\ln(1/\delta) \right).$$

Now for any $s, a$, we must have:

$$\left| (\hat{\theta}_h - \mathcal{T}_h(\hat{\theta}_{h+1}))^\top \phi(s, a) \right|^2 \leq \left\| \hat{\theta}_h - \mathcal{T}_h(\hat{\theta}_{h+1}) \right\|_{\Lambda_h}^2 \|\phi(s, a)\|_{\Lambda_h^{-1}}^2 \leq \left\| \hat{\theta}_h - \mathcal{T}_h(\hat{\theta}_{h+1}) \right\|_{\Lambda_h}^2 \frac{d}{N}$$

$$\leq 4H^2 \frac{d^2 + d^{1.5}\sqrt{\ln(1/\delta)} + 2d\ln(1/\delta)}{N} \leq \frac{16H^2 d^2 \ln(1/\delta)}{N},$$

which implies that:

$$\left\| \widehat{Q}_h - \mathcal{T}_h \widehat{Q}_{h+1} \right\|_\infty = \max_{s,a} \left| (\hat{\theta}_h - \mathcal{T}_h(\hat{\theta}_{h+1}))^\top \phi(s, a) \right| \leq \frac{4Hd\sqrt{\ln(1/\delta)}}{\sqrt{N}}.$$

Set the right hand side to be $\epsilon$, we have $N = \frac{16H^2 d^2 \ln(1/\delta)}{\epsilon^2}$.

Finally, apply union bound over all $h = 0, \ldots, H - 1$, we conclude the proof. ∎

Now we are ready to conclude the proof for the main theorem.

**Proof:**[Proof of Theorem 3.3] Using Lemma 3.5, with $N = \left\lceil \frac{64H^6 d^2 \ln(H/\delta)}{\epsilon^2} \right\rceil$, we have that $\|\widehat{Q}_h - \mathcal{T}_h \widehat{Q}_{h+1}\|_\infty \leq \epsilon/(2H^2)$ for all $h$, with probability at least $1 - \delta$. Now apply Lemma 3.4, we immediately have that $|V^{\hat{\pi}} - V^\star| \leq \epsilon$.

Regarding sample complexity, we have:

$$H \sum_{s,a} \lceil \rho(s, a)N \rceil \leq H \sum_{s,a \in \text{support}(\rho)} (1 + \rho(s, a)N) \leq H \left( d^2 + \frac{64H^6 d^2 \ln(1/\delta)}{\epsilon^2} \right),$$

where the second inequality use the support size upper bound of the $D$-optimal design $\rho$ (see Lemma 3.2). This completes the proof. ∎

## 3.4 How Strong is Bellman Completion as a Modeling?

To be added.

## 3.5 Bibliographic Remarks and Further Readings

The idea of Bellman completion under general function class was introduced in [Munos, 2005] under the setting of batch RL. For the episodic online learning setting, Zanette et al. [2020] provided a statistically efficient algorithm

under the linear Bellman completion condition, and [Jin et al., 2021] proposes a statistically efficient algorithms under the Bellman completion condition with general function approximation.

We refer readers to to [Lattimore and Szepesvári, 2020] for a proof of the D-optimal design; the idea directly follows from John's theorem (e.g. see [Ball, 1997, Todd, 2016]).

# Chapter 4

# Fitted Dynamic Programming Methods

Let us again consider the question of learning in large MDPs, when the underlying MDPs is unknown. In the previous chapter, we relied on the linear Bellman completeness assumption to provide strong guarantees with only polynomial dependence on the dimension of the feature space and the horizon $H$. This chapter considers the approach of using *function approximation* methods with iterative dynamic programming approaches.

In particular, we now consider approaches which rely on (average-case) supervised learning methods (namely regression), where we use regression to approximate the target functions in both the value iteration and policy iteration algorithms. We refer to these algorithms as *fitted Q-iteration* (FQI) and *fitted policy iteration* (FPI). The FQI algorithm can be implemented in an offline RL sampling model while the FPI algorithm requires the sampling access to an episodic model (see Chapter 1 for review of these sampling models).

This chapter focuses on obtaining of *average case function approximation error* bounds, provided we have a somewhat stringent condition on how the underlying MDP behaves, quantified by the *concentrability coefficient*. This notion was introduced in [Munos, 2003, 2005]. While the notion is somewhat stringent, we will see that it is not avoidable without further assumptions. The next chapter more explicitly considers lower bounds, while Chapters 13 and 14 seek to relax the concentrability notion.

## 4.1   Fitted $Q$-Iteration (FQI) and Offline RL

We consider infinite horizon discounted MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \gamma, P, r, \mu\}$ where $\mu$ is the initial state distribution. We assume reward is bounded, i.e., $\sup_{s,a} r(s,a) \in [0,1]$. For notation simplicity, we denote $V_{\max} := 1/(1-\gamma)$.

Given any $f : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, we denote the Bellman operator $\mathcal{T}f : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ as follows. For all $s, a \in \mathcal{S} \times \mathcal{A}$,

$$\mathcal{T}f(s,a) := r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \max_{a' \in \mathcal{A}} f(s', a').$$

We assume that we have a distribution $\nu \in \Delta(\mathcal{S} \times \mathcal{A})$. We collect a dataset $\mathcal{D} := \{(s_i, a_i, r_i, s_i')\}_{i=1}^n$ where $s_i, a_i \sim \nu, r_i = r(s_i, a_i), s_i' \sim P(\cdot|s_i, a_i)$. Given $\mathcal{D}$, our goal is to output a near optimal policy for the MDP, that is we would like our algorithm to produce a policy $\hat{\pi}$ such that, with probability at least $1 - \delta$, $V(\hat{\pi}) \geq V^\star - \epsilon$, for some $(\epsilon, \delta)$ pair. As usual, the number of samples $n$ will depend on the accuracy parameters $(\epsilon, \delta)$ and we would like $n$ to scale favorably with these. Given any distribution $\nu \in \mathcal{S} \times \mathcal{A}$, and any function $f : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, we write $\|f\|_{2,\nu}^2 := \mathbb{E}_{s,a \sim \nu} f^2(s,a)$

Denote a function class $\mathcal{F} = \{f : \mathcal{S} \times \mathcal{A} \mapsto [0, V_{\max}]\}$.

We require the data distribution $\nu$ is exploratory enough.

**Assumption 4.1** (Concentrability). There exists a constant $C$ such that for any policy $\pi$ (including non-stationary policies), we have:

$$\forall \pi, x, a : \frac{d^\pi(s, a)}{\nu(s, a)} \leq C.$$

Note that concentrability does not require that the state space is finite, but it does place some constraints on the system dynamics. Note that the above assumption requires that $\nu$ to cover all possible policies's state-action distribution, even including non-stationary policies.

In additional to the above two assumptions, we need an assumption on the representational condition of class $\mathcal{F}$.

**Assumption 4.2** (Inherent Bellman Error). We assume the following error bound holds:

$$\epsilon_{approx,\nu} := \max_{f' \in \mathcal{F}} \min_{f \in \mathcal{F}} \|f' - \mathcal{T}f\|_{2,\nu}^2.$$

We refer to $\epsilon_{approx,\nu}$ as the *inherent Bellman error* with respect to the distribution $\nu$.

### 4.1.1 The FQI Algorithm

Fitted Q Iteration (FQI) simply performs the following iteration. Start with some $f_0 \in \mathcal{F}$, FQI iterates:

$$\text{FQI:} \quad f_t \in \operatorname{argmin}_{f \in \mathcal{F}} \sum_{i=1}^{n} \left( f(s'_i, a_i) - r_i - \gamma \max_{a' \in \mathcal{A}} f_{t-1}(s_i, a_i) \right)^2. \tag{0.1}$$

After $k$ many iterations, we output a policy $\pi^k(s) := \operatorname{argmax}_a f_k(s, a), \forall s$.

To get an intuition why this approach can work, let us assume the inherent Bellman error $\epsilon_{approx} = 0$ which is saying that for any $f \in \mathcal{F}$, we have $\mathcal{T}f \in \mathcal{F}$ (i.e., Bellman completion). Note that the Bayes optimal solution is $\mathcal{T}f_{t-1}$. Due to the Bellman completion assumption, the Bayes optimal solution $\mathcal{T}f_{t-1} \in \mathcal{F}$. Thus, we should expect that $f_t$ is close to the Bayes optimal $\mathcal{T}f_{t-1}$ under the distribution $\nu$, i.e., with a uniform convergence argument, for the generalization bound, we should expect that:

$$\mathbb{E}_{s,a \sim \nu} \left( f_t(s, a) - \mathcal{T}f_{t-1}(s, a) \right)^2 \approx \sqrt{1/n}.$$

Thus in high level, $f_t \approx \mathcal{T}f_{t-1}$ as our data distribution $\nu$ is exploratory, and we know that based on value iteration, $\mathcal{T}f_{k-1}$ is a better approximation of $Q^\star$ than $f_k$, i.e., $\|\mathcal{T}f_{t-1} - Q^\star\|_\infty \leq \gamma \|f_{t-1} - Q^\star\|_\infty$, we can expect FQI to converge to the optimal solution when $n \to \infty, t \to \infty$. We formalize the above intuition below.

### 4.1.2 Performance Guarantees of FQI

We first state the performance guarantee of FQI.

**Theorem 4.3** (FQI guarantee). *Fix $K \in \mathbb{N}^+$. Fitted Q Iteration guarantees that with probability $1 - \delta$,*

$$V^\star - V^{\pi^K} \leq \frac{1}{(1-\gamma)^3} \left( \sqrt{\frac{144 C V_{\max}^2 \ln(|\mathcal{F}|^2 K/\delta)}{n}} + \sqrt{8C\epsilon_{approx}} \right) + \frac{2\gamma^K}{(1-\gamma)^2}$$

We start from the following lemma which shows that if for all $t$, $\|f_{t+1} - \mathcal{T}f_t\|_{2,\nu}^2 \leq \varepsilon$, then the greedy policy with respect to $f$ is approaching to $\pi^\star$.

**Lemma 4.4.** *Assume that for all $t$, we have $\|f_{t+1} - \mathcal{T}f_t\|_{2,\nu} \leq \varepsilon$, then for any $k$, we have:*

$$V^{\pi^k} - V^\star \leq \frac{\sqrt{C}\varepsilon}{(1-\gamma)^2} + \frac{\gamma^k V_{\max}}{1-\gamma}.$$

**Proof:** We start from the Performance Difference Lemma:

$$
\begin{aligned}
(1-\gamma)\left(V^\star - V^{\pi_k}\right) &= \mathbb{E}_{s \sim d^{\pi_k}}\left[-A^\star(s, \pi_k(s))\right] \\
&= \mathbb{E}_{s \sim d^{\pi_k}}\left[Q^\star(s, \pi^\star(s)) - Q^\star(s, \pi_k(s))\right] \\
&\leq \mathbb{E}_{s \sim d^{\pi_k}}\left[Q^\star(s, \pi^\star(s)) - f_k(s, \pi^\star(s)) + f_k(s, \pi_k(s)) - Q^\star(s, \pi_k(s))\right] \\
&\leq \|Q^\star - f_k\|_{1, d^{\pi_k} \circ \pi^\star} + \|Q^\star - f_k\|_{1, d^{\pi_k} \circ \pi_k} \\
&\leq \|Q^\star - f_k\|_{2, d^{\pi_k} \circ \pi^\star} + \|Q^\star - f_k\|_{2, d^{\pi_k} \circ \pi_k},
\end{aligned}
$$

where the first inequality comes from the fact that $\pi_k$ is a greedy policy of $f_k$, i.e., $f_k(s, \pi_k(s)) \geq f_k(s, a)$ for any other $a$ including $\pi^\star(s)$. Now we bound each term on the RHS of the above inequality. We do this by consider a state-action distribution $\beta$ that is induced by some policy. We have:

$$
\begin{aligned}
\|Q^\star - f_k\|_{2,\beta} &\leq \|Q^\star - \mathcal{T}f_{k-1}\|_{2,\nu} + \|f_k - \mathcal{T}f_{k-1}\|_{2,\beta} \\
&\leq \gamma\sqrt{\mathbb{E}_{s,a \sim \beta}\left[\left(\mathbb{E}_{s' \sim P(\cdot|s,a)} \max_a Q^\star(s', a) - \max_a f_{k-1}(s', a)\right)^2\right]} + \|f_k - \mathcal{T}f_{k-1}\|_{2,\beta} \\
&\leq \gamma\sqrt{\mathbb{E}_{s,a \sim \beta, s' \sim P(\cdot|s,a)} \max_a \left(Q^\star(s', a) - f_{k-1}(s', a)\right)^2} + \sqrt{C}\|f_k - \mathcal{T}f_{k-1}\|_{2,\nu},
\end{aligned}
$$

where in the last inequality, we use the fact that $(\mathbb{E}[x])^2 \leq \mathbb{E}[x^2]$, $(\max_x f(x) - \max_x g(x))^2 \leq \max_x(f(x) - g(x))^2$ for any two functions $f$ and $g$, and assumption 4.1.

Denote $\beta'(s', a') = \sum_{s,a} \beta(s, a) P(s'|s, a) \mathbf{1}\{a' = \operatorname{argmax}_a \left(Q^\star(s', a) - f_{k-1}(s', a)\right)^2\}$, the above inequality becomes:

$$\|Q^\star - f_k\|_{2,\beta} \leq \gamma\|Q^\star - f_{k-1}\|_{2,\beta'} + \sqrt{C}\|f_k - \mathcal{T}f_{k-1}\|_{2,\nu}.$$

We can recursively repeat the same process for $\|Q^\star - f_{k-1}\|_{2,\nu'}$ till step $k = 0$:

$$\|Q^\star - f_k\|_{2,\beta} \leq \sqrt{C}\sum_{t=0}^{k-1} \gamma^t \|f_{k-t} - \mathcal{T}f_{k-t-1}\|_{2,\nu} + \gamma^k \|Q^\star - f_0\|_{2,\widetilde{\nu}},$$

where $\widetilde{\nu}$ is some valid state-action distribution.

Note that for the first term on the RHS of the above inequality, we can bound it as follow:

$$\sqrt{C}\sum_{t=0}^{k-1} \gamma^t \|f_{k-t} - \mathcal{T}f_{k-t-1}\|_{2,\mu} \leq \sqrt{C}\sum_{t=0}^{k-1} \gamma^k \varepsilon \leq \sqrt{C}\varepsilon\frac{1}{1-\gamma}.$$

For the second term, we have:

$$\gamma\|Q^\star - f_0\|_{2,\widetilde{\nu}} \leq \gamma^k V_{\max}.$$

Thus, we have:

$$\|Q^\star - f_k\|_{2,\beta} = \frac{\sqrt{C}\varepsilon}{1-\gamma} + \gamma^k V_{\max},$$

39

for all $\beta$, including $\beta = d^{\pi_k} \circ \pi^\star$, and $\beta = d^{\pi_k} \circ \pi_k$. This concludes the proof. $\blacksquare$

What we left is to show that least squares control the error $\varepsilon$. Specifically, the following lemma shows that $\varepsilon$ scales in the order of $\sqrt{\epsilon_{approx} + 1/n}$.

**Lemma 4.5** (Bellman error). *With probability at least $1 - \delta$, for all $t = 0, \dots, K$, we have:*

$$\|f_{t+1} - \mathcal{T}f_t\|_{2,\nu}^2 \leq \frac{144V_{\max}^2 \ln(|\mathcal{F}|^2 K/\delta)}{n} + 8\epsilon_{approx}.$$

**Proof:** Let us consider a fixed function $g \in \mathcal{F}$ first, and denote $\hat{f} = \mathrm{argmin}_{f \in \mathcal{F}} \sum_{i=1}^n (f(s_i, a_i) - r_i - \gamma \max_{a' \in \mathcal{A}} g(s_i', a'))^2$ as the corresponding least squares solution. Note that $\hat{f}$ is fully determined by $g$. At the end of this proof, we will apply a union bound over all $g \in \mathcal{F}$.

For any fixed $f \in \mathcal{F}$, let us denote the random variable $z_i^f$:

$$z_i^f := \left( f(s_i, a_i) - r_i - \gamma \max_{a' \in \mathcal{A}} g(s_i', a') \right)^2 - \left( \mathcal{T}g(s_i, a_i) - r_i - \gamma \max_{a' \in \mathcal{A}} g(s_i', a') \right)^2.$$

First note that $z_i^f$ is bounded:

$$|z_i^f| \leq V_{\max}^2,$$

which comes from the assumption that $f(s, a) \in [0, V_{\max}]$ for all $s, a, f$.

Now we compute $\mathbb{E}[z_i^f]$ and the upper bound of the second moment $\mathbb{E}[(z_i^f)^2]$, where the expectation is taken with respect to $s_i, a_i, s_i'$. First we show that $z_i^f$ is an unbiased estimate of $\|f - \mathcal{T}g\|_{2,\nu}^2$:

$$\mathbb{E}_{s_i, a_i \sim \nu, s_i' \sim P(\cdot | s_i, a_i)} \left[ z_i^f \right]$$

$$= \mathbb{E}_{s_i, a_i \sim \nu, s_i' \sim P(\cdot | s_i, a_i)} (f(s_i, a_i) - \mathcal{T}g(s_i, a_i)) \left( f(s_i, a_i) + \mathcal{T}g(s_i, a_i) - 2(r_i + \gamma \max_{a' \in \mathcal{A}} g(s_i', a')) \right)$$

$$= \mathbb{E}_{s_i, a_i \sim \nu} (f(s_i, a_i) - \mathcal{T}g(s_i, a_i)) \left( f(s_i, a_i) + \mathcal{T}g(s_i, a_i) - 2 \left( r_i + \mathbb{E}_{s_i' \sim P(\cdot | s_i, a_i)} \max_{a'} g(s_i', a') \right) \right)$$

$$= \mathbb{E}_{s_i, a_i \sim \nu} (f(s_i, a_i) - \mathcal{T}g(s_i, a_i))^2 = \mathbb{E}_{s, a \sim \nu} (f(s, a) - \mathcal{T}g(s, a))^2 = \|f - \mathcal{T}g\|_{2,\nu}^2,$$

where the third equality uses the definition of Bellman operator, i.e., that $r_i + \mathbb{E}_{s_i' \sim P(\cdot | s_i, a_i)} \max_{a'} g(s_i', a') = \mathcal{T}g(s_i, a_i)$.

Now we calculate the second moment of $z_i^f$. We show that $\mathbb{E}[(z_i^f)^2] \leq 4V_{\max}^2 \|f - \mathcal{T}g\|_{2,\nu}^2$:

$$\mathbb{E}_{s_i, a_i \sim \nu, s_i' \sim P(\cdot | s_i, a_i)} \left[ (z_i^f)^2 \right]$$

$$= \mathbb{E}_{s_i, a_i \sim \nu, s_i' \sim P(\cdot | s_i, a_i)} \left[ (f(s_i, a_i) - \mathcal{T}g(s_i, a_i))^2 \left( f(s_i, a_i) + \mathcal{T}g(s_i, a_i) - 2(r_i + \gamma \max_{a'} g(s_i', a')) \right)^2 \right]$$

$$\leq 4V_{\max}^2 \mathbb{E}_{s_i, a_i \sim \nu} \left[ (f(s_i, a_i) - \mathcal{T}g(s_i, a_i))^2 \right]$$

$$= 4V_{\max}^2 \mathbb{E}_{s, a \sim \nu} \left[ (f(s, a) - \mathcal{T}g(s, a))^2 \right] = 4V_{\max}^2 \|f - \mathcal{T}g\|_{2,\nu}^2$$

where in the inequality, we again use the assumption that $f(s, a) \in [0, V_{\max}]$ for all $s, a, f$.

Now we can use Bernstein's inequality to bound the deviation between the empirical mean $\frac{1}{n} \sum_{i=1}^n z_i^f$ and the expectation $\|f - \mathcal{T}g(s, a)\|_{2,\nu}^2$. Together with a union bound over all $f \in \mathcal{F}$, we have that with probability at least $1 - \delta$,

40

for any $f \in \mathcal{F}$:

$$\|f - \mathcal{T}g\|_{2,\nu}^2 - \frac{1}{n}\sum_{i=1}^n z_i^f \leq \sqrt{\frac{8V_{\max}^2 \|f - \mathcal{T}g\|_{2,\nu}^2 \ln(|\mathcal{F}|/\delta)}{n}} + \frac{4V_{\max}^2 \ln(|\mathcal{F}|/\delta)}{3n}. \tag{0.2}$$

Recall $\hat{f} = \operatorname{argmin}_{f \in \mathcal{F}} z_i^f$. Below we show that $\sum_{i=1}^n z_i^{\hat{f}}/n$ is upper bound by $\frac{\epsilon_{approx}}{2} + \frac{10V_{\max}^2 \ln(1/\delta)}{3n}$.

Let us denote $g' = \operatorname{argmin}_{f \in \mathcal{F}} \|f - \mathcal{T}g\|_{2,\nu}^2$. Note that for $z_i^{g'}$, our previous calculation implies that we have $\mathbb{E}[z_i^{g'}] = \|g' - \mathcal{T}g\|_{2,\nu}^2$, and $\mathbb{E}[(z_i^g)^2] \leq 4V_{\max}^2 \|g' - \mathcal{T}g\|_{2,\nu}^2$.

Thus via a Bernstein's inequality, with probability at least $1 - \delta$, we have:

$$\sum_{i=1}^n z_i^{g'}/n - \mathbb{E}[z_i^{g'}] \leq \sqrt{\frac{8V_{\max}^2 \mathbb{E}[z_i^{g'}] \ln(1/\delta)}{n}} + \frac{4V_{\max}^2 \ln(1/\delta)}{3n}.$$

When $\sum_{i=1}^n z_i^{g'}/n \geq 4V_{\max}^2 \ln(1/\delta)/n$, we can show that the above inequality implies that:

$$\sum_{i=1}^n z_i^{g'}/n \leq \frac{\mathbb{E}[z_i^{g'}]}{2} + \frac{10V_{\max}^2 \ln(1/\delta)}{3n}.$$

Since $\mathbb{E}[z_i^{g'}] \leq \epsilon_{approx}$ by the definition of inherent Bellman error, we can conclude that:

$$\sum_{i=1}^n z_i^{g'}/n \leq \frac{\epsilon_{approx}}{2} + \frac{10V_{\max}^2 \ln(1/\delta)}{3n}.$$

Note that $\hat{f}$ is the least square minimizer, i.e., $\hat{f} = \operatorname{argmin}_{f \in \mathcal{F}} \sum_i z_i^f$, so, we have

$$\sum_{i=1}^n z_i^{\hat{f}}/n \leq \sum_{i=1}^n z_i^{g'}/n \leq \frac{\epsilon_{approx}}{2} + \frac{10V_{\max}^2 \ln(1/\delta)}{3n}.$$

This concludes the upper bound on $\sum_{i=1}^n z_i^{\hat{f}}/n$.

Going back to Eq. 0.2, set $f$ there to be $\hat{f}$, and use the upper bound on $\sum_{i=1}^n z_i^{\hat{f}}/n$ from the above inequality, we have:

$$\|\hat{f} - \mathcal{T}g\|_{2,\nu}^2 \leq \sqrt{\frac{8V_{\max}^2 \|\hat{f} - \mathcal{T}g\|_{2,\nu}^2 \ln(|\mathcal{F}|/\delta)}{n}} + \frac{14V_{\max}^2 \ln(|\mathcal{F}|/\delta)}{3n} + \epsilon_{approx}.$$

Solve for $\|\hat{f} - \mathcal{T}g\|_{2,\nu}^2$ from the above inequality, we get:

$$\|\hat{f} - \mathcal{T}g\|_{2,\nu}^2 \leq \frac{144V_{\max}^2 \ln(|\mathcal{F}|/\delta)}{n} + 8\epsilon_{approx}.$$

Note that the above result holds for a fixed $g \in \mathcal{F}$. With a union bound over all $g \in \mathcal{F}$, we have that with probability at least $1 - \delta$:

$$\forall g \in \mathcal{F}: \quad \|\hat{f} - \mathcal{T}g\|_{2,\nu}^2 \leq \frac{144V_{\max}^2 \ln(|\mathcal{F}|^2/\delta)}{n} + 8\epsilon_{approx}, \tag{0.3}$$

where $\hat{f} := \text{argmin}_{f \in \mathcal{F}} \sum_{i=1}^{n} (f(s_i, a_i) - r_i - \max_{a'} g(s_i', a'))^2$.

Consider iteration $t$, set $g = f_{t+1}$ in Eq. 0.3, and by the definition of $f_t$, we have $\hat{f} = f_t$ in this case, which means that above inequality in Eq. 0.3 holds for the pair $(\hat{f}, g) = (f_t, f_{t+1})$. Now apply a union bound over all $t = 0, \ldots, K$, we conclude the proof. ∎

Now we are ready to prove the main theorem.

**Proof:**[Proof of Theorem 4.3]

From Lemma 4.5, we see that with probability at least $1 - \delta$, we have for all $t \leq K$:

$$\|f_{t+1} - \mathcal{T}f_t\|_{2,\nu}^2 \leq \sqrt{\frac{144 V_{\max}^2 \ln(|\mathcal{F}|^2 K/\delta)}{n}} + \sqrt{8\epsilon_{approx}}.$$

Set $\varepsilon$ in Lemma 4.4 to be $\sqrt{\frac{144 V_{\max}^2 \ln(|\mathcal{F}|^2 K/\delta)}{n}} + \sqrt{8\epsilon_{approx}}$, we conclude the proof. ∎

## 4.2 Fitted Policy-Iteration (FPI)

to be added..

## 4.3 Failure Cases Without Assumption 4.1

## 4.4 FQI for Policy Evaluation

## 4.5 Bibliographic Remarks and Further Readings

The notion of concentrability was developed in [Munos, 2003, 2005] in order to permitting sharper bounds in terms of average case function approximation error, provided that the concentrability coefficient is bounded. These methods also permit sample based fitting methods, with sample size and error bounds, provided there is a data collection policy that induces a bounded concentrability coefficient [Munos, 2005, Szepesvári and Munos, 2005, Antos et al., 2008, Lazaric et al., 2016]. Chen and Jiang [2019] provide a more detailed discussion on this quantity.

# Chapter 5

# Statistical Limits of Generalization

In reinforcement learning, we seek to have learnability results which are applicable to cases where number of states is large (or, possibly, countably or uncountably infinite). This is a question of generalization, which, more broadly, is one of the central challenges in machine learning.

The previous two chapters largely focussed on *sufficient* conditions under which we can obtain sample complexity results which do not explicitly depend on the size of the state (or action) space. This chapter focusses on what are *necessary* conditions for generalization. Here, we can frame our questions by examining the extent to which generalization in RL is similar to (or different from) that in supervised learning. Two most basic settings in supervised learning are: (i) agnostic learning (i.e. finding the best classifier or hypothesis in some class) and (ii) learning with linear models (i.e. learning the best linear regressor or the best linear classifier). This chapter will focus on *lower bounds* with regards to the analogues of these two questions for reinforcement learning:

- (Agnostic learning) Given some hypothesis class (of policies, value functions, or models), what is the sample complexity of finding (nearly) the best hypothesis in this class?

- (Linearly realizable values or policies) Suppose we are given some $d$-dimensional feature mapping where we are guaranteed that either the optimal value function is linear in these given features or that the optimal policy has a linear parameterization. Are we able to obtain sample complexity guarantees that are polynomial in $d$, with little to no explicit dependence on the size of the state or action spaces?

Observe that supervised learning can be viewed as horizon one, $H = 1$, RL problem (where the learner only receives feedback for the "label", i.e. the action, chosen). We can view the second question above as the analogue of linear regression or classification with halfspaces. In supervised learning, both of these settings have postive answers, and they are fundamental in our understanding of generalization. Perhaps surprisingly, we will see negative answers to these questions in the RL setting. The significance of this provides insights as to why our study of generalization in reinforcement learning is substantially more subtle than in supervised learning. Importantly, the insights we develop here will also help us to motivate the more refined assumptions and settings that we consider in subsequent chapters (see Section 5.4 for discussion).

This chapter will work with finite horizon MDPs, where we consider both the episodic setting and the generative model setting. With regards to the first question on agnostic learning, this chapter follows the ideas first introduced in [Kearns et al., 2000]. With regards to the second question on linear realizability, this chapter follows the work in [Du et al., 2019, Weisz et al., 2021, Wang et al., 2021].

## 5.1 Agnostic Learning

Suppose we have a hypothesis class $\mathcal{H}$ (either finite or infinite), where for each $f \in \mathcal{H}$ we have an associated policy $\pi_f : \mathcal{S} \to \mathcal{A}$, which, for simplicity, we assume is deterministic. Here, we could have that:

- $\mathcal{H}$ itself is a class of policies.

- $\mathcal{H}$ is a set of state-action values, where for $f \in \mathcal{H}$, we associate it with the greedy policy $\pi_f(s, h) = \mathrm{argmax}_a\, f(s, h)$, where $s \in \mathcal{S}$ and $h \in [H]$.

- $\mathcal{H}$ could be a class of models (i.e. each $f \in \mathcal{H}$ is an MDP itself). Here, for each $f \in \mathcal{H}$, we can let $\pi_f$ be the optimal policy in the MDP $f$.

We let $\Pi$ denote the induced set of policies from our hypothesis class $\mathcal{H}$, i.e. $\Pi = \{\pi_f | f \in \mathcal{H}\}$.

The goal of agnostic learning can be formulated by the following optimization problem:

$$\max_{\pi \in \Pi} \mathbb{E}_{s_0 \sim \mu} V^\pi(s_0), \tag{0.1}$$

where we are interested in the number of samples required to approximately solve this optimization problem. We will work both in the episodic setting and the generative model setting, and, for simplicity, we will restrict ourselves to finite hypothesis classes.

**Binary classification as an $H = 1$ RL problem:**   Observe that the problem of binary classification can be viewed as learning in an MDP with a horizon of one. In particular, take $H = 1$; take $|\mathcal{A}| = 2$; let the distribution over starting states $s_0 \sim \mu$ correspond to the input distribution; and take the reward function as $r(s, a) = \mathbf{1}(\mathrm{label}(s) = a)$. In other words, we equate our action with the prediction of the binary class membership, and the reward function is determined by if our prediction is correct or not.

### 5.1.1   Review: Binary Classification

One of the most important concepts for learning binary classifiers is that it is possible to *generalize* even when the state space is infinite. Here note that the domain of our classifiers, often denoted by $\mathcal{X}$, is analogous to the state space $\mathcal{S}$. We now briefly review some basics of supervised learning before we turn to the question of generalization in reinforcement learning.

Consider the problem of binary classification with $N$ labeled examples of the form $(x_i, y_i)_{i=1}^N$, with $x_i \in \mathcal{X}$ and $y_i \in \{0, 1\}$. Suppose we have a (finite or infinte) set $\mathcal{H}$ of binary classifiers where each $h \in \mathcal{H}$ is a mapping of the form $h : \mathcal{X} \to \{0, 1\}$. Let $\mathbf{1}(h(x) \neq y)$ be an indicator which takes the value 0 if $h(x) = y$ and 1 otherwise. We assume that our samples are drawn i.i.d. according to a fixed joint distribution $D$ over $(x, y)$.

Define the empirical error and the true error as:

$$\widehat{\mathrm{err}}(h) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(h(x_i) \neq y_i), \quad \mathrm{err}(h) = \mathbb{E}_{(X,Y) \sim D} \mathbf{1}(h(X) \neq Y).$$

For a given $h \in \mathcal{H}$, Hoeffding's inequality implies that with probability at least $1 - \delta$:

$$|\mathrm{err}(h) - \widehat{\mathrm{err}}(h)| \leq \sqrt{\frac{1}{2N} \log \frac{2}{\delta}}.$$

This and the union bound give rise to what is often referred to as the "Occam's razor" bound:

**Proposition 5.1.** *(The "Occam's razor" bound) Suppose $\mathcal{H}$ is finite. Let $\widehat{h} = \arg\min_{h \in \mathcal{H}} \widehat{err}(h)$. With probability at least $1 - \delta$:*

$$err(\widehat{h}) \leq \min_{h \in \mathcal{H}} err(h) + \sqrt{\frac{2}{N} \log \frac{2|\mathcal{H}|}{\delta}}.$$

Hence, provided that

$$N \geq \frac{c \log \frac{2|\mathcal{H}|}{\delta}}{\epsilon^2},$$

then with probability at least $1 - \delta$, we have that:

$$\mathrm{err}(\widehat{h}) \leq \min_{h \in \mathcal{H}} \mathrm{err}(h) + \epsilon.$$

A key observation here is that the our regret — the regret is the left hand side of the above inequality — has *no dependence* on the size of $\mathcal{X}$ (i.e. $\mathcal{S}$) which may be infinite and is only logarithmic in the number of hypothesis in our class.

### 5.1.2 Is agnostic learning possible in RL?

Now let us return to the agnostic learning question in Equation 0.1. We will see that, provided we are willing to be exponential in the horizon $H$, then agnostic learning is possible. Furthermore, it is a straightforward argument to see that we are not able to do better.

**An Occam's Razor Bound for RL**

We now provide a reduction of RL to the supervised learning problem, given only sampling access in the episodic setting. The key issue is how to efficiently reuse data. The idea is that we will simply collect $N$ trajectories by executing a policy which chooses actions uniformly at random; let $\mathrm{Unif}_\mathcal{A}$ denote this policy.

The following shows how we can obtain an unbiased estimate of the value of any policy $\pi$ using this uniform policy $\mathrm{Unif}_\mathcal{A}$:

**Lemma 5.2.** *(Unbiased estimation of $V_0^\pi(\mu)$) Let $\pi$ be any deterministic policy. We have that:*

$$V_0^\pi(\mu) = |\mathcal{A}|^H \cdot \mathbb{E}_{\tau \sim \mathrm{Pr}_{Unif_\mathcal{A}}} \left[ \mathbf{1}\Big(\pi(s_0) = a_0, \ldots, \pi(s_{H-1}) = a_{H-1}\Big) \sum_{h=0}^{H-1} r(s_h, a_h) \right]$$

*where $\mathrm{Pr}_{Unif_\mathcal{A}}$ specifies the distribution over trajectories $\tau = (s_0, a_0, r_0, \ldots s_{H-1}, a_{H-1}, r_{H-1})$ under the policy $\mathrm{Unif}_\mathcal{A}$.*

The proof follows from a standard importance sampling argument (applied to the distribution over the trajectories).
**Proof:** We have that:

$$V_0^\pi(\mu) = \mathbb{E}_{\tau \sim \mathrm{Pr}_\pi} \left[ \sum_{h=0}^{H-1} r_h \right]$$

$$= \mathbb{E}_{\tau \sim \mathrm{Pr}_{Unif_\mathcal{A}}} \left[ \frac{\mathrm{Pr}_\pi(\tau)}{\mathrm{Pr}_{Unif_\mathcal{A}}(\tau)} \sum_{h=0}^{H-1} r_h \right]$$

$$= |\mathcal{A}|^H \cdot \mathbb{E}_{\tau \sim \mathrm{Pr}_{Unif_\mathcal{A}}} \left[ \mathbf{1}\Big(\pi(s_0) = a_0, \ldots, \pi(s_{H-1}) = a_{H-1}\Big) \sum_{h=0}^{H-1} r_h \right]$$

where the last step follows due to that the probability ratio is only nonzero when $\text{Unif}_\mathcal{A}$ choose actions identical to that of $\pi$. ∎

Crudely, the factor of $|\mathcal{A}|^H$ is due to that the estimated reward of $\pi$ on a trajectory is nonzero only when $\pi$ takes exactly identical actions to those taken by $\text{Unif}_\mathcal{A}$ on the trajectory, which occurs with probability $1/|\mathcal{A}|^H$.

Now, given sampling access in the episodic model, we can use $\text{Unif}_\mathcal{A}$ to get any estimate of any other policy $\pi \in \Pi$. Note that the factor of $|\mathcal{A}|^H$ in the previous lemma will lead to this approach being a high variance estimator. Suppose we draw $N$ trajectories under $\text{Unif}_\mathcal{A}$. Denote the $n$-th sampled trajectory by $(s_0^n, a_0^n, r_1^n, s_1^n, \ldots, s_{H-1}^n, a_{H-1}^n, r_{H-1}^n)$. We can then use following to estimate the finite horizon reward of any given policy $\pi$:

$$\widehat{V}_0^\pi(\mu) = \frac{|\mathcal{A}|^H}{N} \sum_{n=1}^N \mathbf{1}\Big(\pi(s_0^n) = a_0^n, \ldots \pi(s_{H-1}^n) = a_{H-1}^n\Big) \sum_{t=0}^{H-1} r(s_t^n, a_t^n).$$

**Proposition 5.3.** *(An "Occam's razor bound" for RL) Let $\delta \geq 0$. Suppose $\Pi$ is a finite and suppose we use the aforementioned estimator, $\widehat{V}_0^\pi(\mu)$, to estimate the value of every $\pi \in \Pi$. Let $\widehat{\pi} = \arg\max_{\pi \in \Pi} \widehat{V}_0^\pi(\mu)$. We have that with probability at least $1 - \delta$,*

$$V_0^{\widehat{\pi}}(\mu) \geq \max_{\pi \in \Pi} V_0^\pi(\mu) - H|\mathcal{A}|^H \sqrt{\frac{2}{N} \log \frac{2|\Pi|}{\delta}}.$$

**Proof:** Note that, fixing $\pi$, our estimator is sum of i.i.d. random variables, where each independent estimator, $|\mathcal{A}|^H \mathbf{1}\Big(\pi(s_0^n) = a_0^n, \ldots \pi(s_{H-1}^n) = a_{H-1}^n\Big) \sum_{t=0}^{H-1} r(s_t^n, a_t^n)$, is bounded by $H|\mathcal{A}|^H$. The remainder of the argument is identical to that used in Proposition 5.1. ∎

Hence, provided that

$$N \geq H|\mathcal{A}|^H \frac{2\log(2|\Pi|/\delta)}{\epsilon^2},$$

then with probability at least $1 - \delta$, we have that:

$$V_0^{\widehat{\pi}}(s_0) \geq \max_{\pi \in \Pi} V_0^\pi(s_0) - \epsilon.$$

This is the analogue of the Occam's razor bound for RL.

Importantly, the above shows that we can avoid dependence on the size of the state space, though this comes at the price of an exponential dependence on the horizon. As we see in the next section, this dependence is unavoidable (without making further assumptions).

**Infinite Policy Classes.** In the supervised learning setting, a crucial observation is that even though a hypothesis set $\mathcal{H}$ of binary classifiers may be infinite, we may still be able to obtain non-trivial generalization bounds. A crucial observation here is that even though the set $\mathcal{H}$ may be infinite, the number of possible behaviors of on a finite set of states is not necessarily exhaustive. The *Vapnik–Chervonenkis* (VC) dimension of $\mathcal{H}$, $VC(\mathcal{H})$, is formal way to characterize this intuition, and, using this concept, we are able to obtain generalization bounds in terms of the $VC(\mathcal{H})$.

With regards to infinite hypothesis classes of policies (say for the case where $|\mathcal{A}| = 2$), extending our Occam's razor bound can be done with precisely the same approach. In particular, when $|\mathcal{A}| = 2$, each $\pi \in \Pi$ can be viewed as Boolean function, and this gives rise to the VC dimension $VC(\Pi)$ of our policy class. The bound in Proposition 5.3 can be replaced with a VC-dimension bound that is analogous to that of binary classification (see Section 5.5).

**Lower Bounds**

Clearly, the drawback of this agnostic learning approach is that we would require a number of samples that is exponential in the problem horizon. We now see that if we desire a sample complexity that scales with $O(\log |\Pi|)$, then an

exponential dependence on the effective horizon is unavoidable, without making further assumptions.

Here, our lower bound permits a (possibly randomized) algorithm to utilize a generative model (which is a more powerful sampling model than the episodic one).

**Proposition 5.4.** *(Lower Bound with a Generative Model) Suppose algorithm $\mathcal{A}$ has access to a generative model. There exists a policy class $\Pi$, where $|\Pi| = |\mathcal{A}|^H$ such that if algorithm $\mathcal{A}$ returns any policy $\pi$ (not necessarily in $\Pi$) such that*

$$V_0^\pi(\mu) \geq \max_{\pi \in \Pi} V_0^\pi(\mu) - 0.5.$$

*with probability greater than $1/2$, then $\mathcal{A}$ must make a number of number calls $N$ to the generative model where:*

$$N \geq c|\mathcal{A}|^H$$

*(where $c$ is an absolute constant).*

**Proof:** The proof is simple. Consider a $|\mathcal{A}|$-ary balanced tree, with $|\mathcal{A}|^H$ states and $|\mathcal{A}|$ actions, where states correspond nodes and actions correspond to edges; actions always move the agent from the root towards a leaf node. We make only one leaf node rewarding, which is unknown to the algorithm. We consider the policy class to be all $|\mathcal{A}|^H$ policies. The theorem now immediately follows since the algorithm gains no knowledge of the rewarding leaf node unless it queries that node. ∎

Note this immediately rules out the possibility that any algorithm which can obtain a $\log |\Pi|$ dependence without paying a factor of $|\mathcal{A}|^H$ in the sample size due to that $\log |\Pi| = H \log |\mathcal{A}|$ in the example above.

## 5.2 Linearly Realizable $Q^\star$

In supervised learning, two of the most widely studied settings are those of linear regression and binary classification with halfspaces. In both settings, we are able to obtain sample complexity results that are polynomial in the feature dimension. We consider the analogue of these assumptions for RL, starting with the analogue for linear regression.

When the state space is large or infinite, we may hope that linearly realizability assumptions on $Q^\star$ may permit a more sample efficient approach. Specifically, we will assume access to a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, and we will assume that a linear function of $\phi$ can accurately represent the $Q^\star$-function. Specifically,

**Assumption 5.5** (Linear $Q^\star$ Realizability). For all $h \in [H]$, assume there exists $\theta_h^* \in \mathbb{R}^d$ such that for all $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$Q_h^*(s, a) = \theta_h^* \cdot \phi(s, a).$$

The hope is that this assumption may permit a sample complexity that is polynomial in $d$ and $H$, with no explicit $|\mathcal{S}|$ or $|\mathcal{A}|$ dependence.

Another assumption that we will use is that the minimum suboptimality gap is lower bounded.

**Assumption 5.6** (Constant Sub-optimality Gap). For any state $s \in \mathcal{S}$, $a \in \mathcal{A}$, the suboptimality gap is defined as $\Delta_h(s, a) := V_h^*(s) - Q_h^*(s, a)$. We assume that

$$\min_{h \in [H], s \in \mathcal{S}, a \in \mathcal{A}} \{\Delta_h(s, a) : \Delta_h(s, a) > 0\} \geq \Delta_{\min}.$$

The hope is that with a "large gap", the identification of the optimal policy itself (as opposed to just estimating its value accurately) may be statistically easier, thus making the problem easier.

47
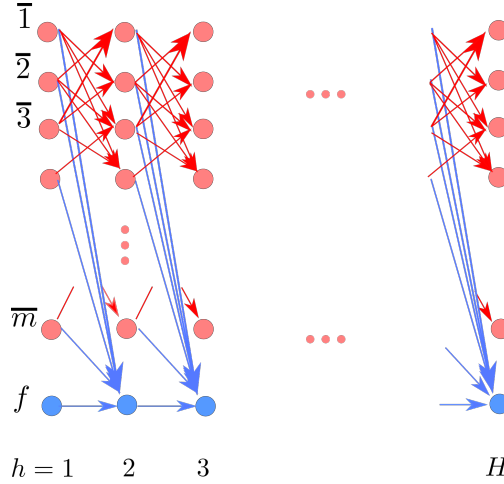
Figure 0.1: **The Leaking Complete Graph Construction.** Illustration of a hard MDP. There are $m + 1$ states in the MDP, where $f$ is an absorbing terminal state. Starting from any non-terminal state $\overline{a}$, regardless of the action, there is at least $\alpha = 1/6$ probability that the next state will be $f$.

### 5.2.1 Linear $Q^\star$ is Statistically Hard

We now present our main hardness results. We start with the case where we have access to a generative model.

**Theorem 5.7.** *(Linear $Q^\star$; Generative Model Case) Consider any algorithm $\mathcal{A}$ which has access to the episodic sampling model and which takes as input the feature mapping $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$. There exists an MDP with a feature mapping $\phi$ satisfying Assumption 5.5 such that if $\mathcal{A}$ (when given $\phi$ as input) finds a policy $\pi$ such that*

$$\mathbb{E}_{s_1 \sim \mu} V^\pi(s_1) \geq \mathbb{E}_{s_1 \sim \mu} V^*(s_1) - 0.05$$

*with probability $0.1$, then $\mathcal{A}$ requires $\min\{2^{cd}, 2^{cH}\}$ samples (c is an absolute constant).*

We now see that in the episodic setting, even if we have a constant gap in the MDP, the hardness of learning is still unavoidable.

**Theorem 5.8.** *(Linear $Q^\star$ Realizability + Gap; Episodic Setting) Consider any algorithm $\mathcal{A}$ which has access to the episodic sampling model and which takes as input the feature mapping $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$. There exists an MDP with a feature mapping $\phi$ satisfying Assumption 5.5 and Assumption 5.6 (where $\Delta_{\min}$ is an absolute constant) such that if $\mathcal{A}$ (using $\phi$ as input) finds a policy $\pi$ such that*

$$\mathbb{E}_{s_1 \sim \mu} V^\pi(s_1) \geq \mathbb{E}_{s_1 \sim \mu} V^*(s_1) - 0.05$$

*with probability $0.1$, then $\mathcal{A}$ requires $\min\{2^{cd}, 2^{cH}\}$ samples (where c is an absolute constant).*

**Linear $Q^\star$ Realizability + Gap; Generative Model Case.** For this case, it is not difficult to see that, with a generative model, it is possible to learn *exactly* learn an optimal policy with a number of samples that is $\mathrm{poly}(d, H, 1/\Delta_{\min})$ (see Section 5.5). The key idea is that, at the last stage, the large gap permits us to learn the optimal policy itself (using linear regression). Furthermore, if we have an optimal policy from timestep $h$ onwards, then we are able to obtain unbiased estimates of $Q^\star_{h-1}(s, a)$, at *any* state action pair, using the generative model. We leave the proof as an exercise to the interested reader.

## 5.2.2 Hard Instance with Constant Suboptimality Gap

The remainder of this section provides the construction of a hard family of MDPs where $Q^*$ is linearly realizable and has constant suboptimality gap and where it takes exponential samples to learn a near-optimal policy. Each of these hard MDPs can roughly be seen as a "leaking complete graph" (see Fig. 0.1). Information about the optimal policy can only be gained by: (1) taking the optimal action; (2) reaching a non-terminal state at level $H$. We will show that when there are exponentially many actions, either events happen with negligible probability unless exponentially many trajectories are played.

Let $m$ be an integer to be determined. The state space is $\{\bar{1}, \cdots, \bar{m}, f\}$. The special state $f$ is called the *terminal state*. The action space is simply $[m]$. Each MDP in this family is specified by an index $a^* \in [m]$ and denoted by $\mathcal{M}_{a^*}$. In other words, there are $m$ MDPs in this family. We will make use of a (large) set of approximately orthogonal vectors, which exists by the Johnson-Lindenstrauss lemma. We state the following lemma without proof:

**Lemma 5.9** (Johnson-Lindenstrauss). *For any $\alpha > 0$, if $m \leq \exp(\frac{1}{8}\alpha^2 d')$, there exists $m$ unit vectors $\{v_1, \cdots, v_m\}$ in $\mathbb{R}^{d'}$ such that $\forall i, j \in [m]$ such that $i \neq j$, $|\langle v_i, v_j \rangle| \leq \alpha$.*

We will set $\alpha = \frac{1}{6}$ and $m = \lfloor \exp(\frac{1}{8}\alpha^2 d) \rfloor$. By Lemma 5.9, we can find such a set of $d$-dimensional unit vectors $\{v_1, \cdots, v_m\}$. For the clarity of presentation, we will use $v_i$ and $v(i)$ interchangeably. The construction of $\mathcal{M}_{a^*}$ is specified below. Note that in our construction, the features, the rewards and the transitions are defined for all $\overline{a_1}, a_2$ with $a_1, a_2 \in [m]$ and $a_1 \neq a_2$. In particular, our construction is properly defined even when $a_1 = a^*$.

**Features.** The feature map, which maps state-action pairs to $d + 1$ dimensional vectors, is defined as follows.

$$\phi(\overline{a_1}, a_2) := \left(0, \left(\left\langle v(a_1), v(a_2) \right\rangle + 2\alpha\right) \cdot v(a_2)\right), \qquad (\forall a_1, a_2 \in [m], a_1 \neq a_2)$$

$$\phi(\overline{a_1}, a_1) := \left(\frac{3}{4}\alpha, \mathbf{0}\right), \qquad (\forall a_1 \in [m])$$

$$\phi(f, 1) = (0, \mathbf{0}),$$
$$\phi(f, a) := (-1, \mathbf{0}). \qquad (\forall a \neq 1)$$

Here $\mathbf{0}$ is the zero vector in $\mathbb{R}^d$. Note that the feature map is independent of $a^*$ and is shared across the MDP family.

**Rewards.** For $h \in [H]$, the rewards are defined as

$$R_h(\overline{a_1}, a^*) := \left\langle v(a_1), v(a^*) \right\rangle + 2\alpha, \qquad (a_1 \neq a^*)$$

$$R_h(\overline{a_1}, a_2) := -2\alpha \left[\left\langle v(a_1), v(a_2) \right\rangle + 2\alpha\right], \qquad (a_2 \neq a^*, a_2 \neq a_1)$$

$$R_h(\overline{a_1}, a_1) := \frac{3}{4}\alpha, \qquad (\forall a_1)$$

$$R_h(f, 1) := 0,$$
$$R_h(f, a) := -1. \qquad (a \neq 1)$$

For $h = H - 1$, $r_{H-1}(s, a) := \langle \phi(s, a), (1, v(a^*)) \rangle$ for every state-action pair.

49

**Transitions.** The initial state distribution $\mu$ is set as a uniform distribution over $\{\bar{1}, \cdots, \bar{m}\}$. The transition probabilities are set as follows.

$$\Pr[f|\overline{a_1}, a^*] = 1,$$
$$\Pr[f|\overline{a_1}, a_1] = 1,$$
$$\Pr[\cdot|\overline{a_1}, a_2] = \begin{cases} \overline{a_2} : \left\langle v(a_1), v(a_2) \right\rangle + 2\alpha \\ f : 1 - \left\langle v(a_1), v(a_2) \right\rangle - 2\alpha \end{cases} \quad, \qquad (a_2 \neq a^*, a_2 \neq a_1)$$
$$\Pr[f|f, \cdot] = 1.$$

After taking action $a_2$, the next state is either $\overline{a_2}$ or $f$. Thus this MDP looks roughly like a "leaking complete graph" (see Fig. 0.1): starting from state $\bar{a}$, it is possible to visit any other state (except for $\overline{a^*}$); however, there is always at least $1 - 3\alpha$ probability of going to the terminal state $f$. The transition probabilities are indeed valid, because

$$0 < \alpha \leq \left\langle v(a_1), v(a_2) \right\rangle + 2\alpha \leq 3\alpha < 1.$$

We now verify that realizability, i.e. Assumption 5.5, is satisfied.

**Lemma 5.10.** *(LInear realizability) In the MDP $\mathcal{M}_{a^*}$, $\forall h \in [H]$, for any state-action pair $(s, a)$, $Q_h^*(s, a) = \langle \phi(s, a), \theta^* \rangle$ with $\theta^* = (1, v(a^*))$.*

**Proof:** We first verify the statement for the terminal state $f$. Observe that at the terminal state $f$, the next state is always $f$ and the reward is either 0 (if action 1 is chosen) or $-1$ (if an action other than 1 is chosen). Hence, we have

$$Q_h^*(f, a) = \begin{cases} 0 & a = 1 \\ -1 & a \neq 1 \end{cases}$$

and

$$V_h^*(f) = 0.$$

This implies $Q_h^*(f, \cdot) = \langle \phi(f, \cdot), (1, v(a^*)) \rangle$.

We now verify realizability for other states via induction on $h = H - 1, \cdots, 0$. The induction hypothesis is that for all $a_1, a_2 \in [m]$, we have

$$Q_h^*(\overline{a_1}, a_2) = \begin{cases} (\langle v(a_1), v(a_2) \rangle + 2\alpha) \cdot \langle v(a_2), v(a^*) \rangle & a_1 \neq a_2 \\ \frac{3}{4}\alpha & a_1 = a_2 \end{cases} \qquad (0.2)$$

and

$$V_h^*(\overline{a_1}) = \begin{cases} \langle v(a_1), v(a^*) \rangle + 2\alpha & a_1 \neq a^* \\ \frac{3}{4}\alpha & a_1 = a^* \end{cases}. \qquad (0.3)$$

Note that (0.2) implies that realizability is satisfied. In the remaining part of the proof we verify Eq. (0.2) and (0.3).

When $h = H - 1$, (0.2) holds by the definition of rewards. Next, note that for all $h \in [H]$, (0.3) follows from (0.2). This is because for all $a_1 \neq a^*$, for all $a_2 \notin \{a_1, a^*\}$.

$$Q_h^*(\overline{a_1}, a_2) = (\langle v(a_1), v(a_2) \rangle + 2\alpha) \cdot \langle v(a_2), v(a^*) \rangle \leq 3\alpha^2,$$

Moreover, for all $a_1 \neq a^*$,

$$Q_h^*(\overline{a_1}, a_1) = \frac{3}{4}\alpha < \alpha.$$

50

Furthermore, for all $a_1 \neq a^*$,

$$Q_h^*(\overline{a_1}, a^*) = \langle v(a_1), v(a^*) \rangle + 2\alpha \geq \alpha > 3\alpha^2.$$

In other words, (0.2) implies that $a^*$ is always the optimal action for all state $\overline{a_1}$ with $a_1 \neq a^*$. Now, for state $\overline{a^*}$, for all $a \neq a^*$, we have

$$Q_h^*(\overline{a^*}, a) = (\langle v(a^*), v(a) \rangle + 2\alpha) \cdot \langle v(a^*), v(a) \rangle \leq 3\alpha^2 < \frac{3}{4}\alpha = Q_h^*(\overline{a^*}, a^*).$$

Hence, (0.2) implies that $a^*$ is always the optimal action for all states $\overline{a}$ with $a \in [m]$.

Thus, it remains to show that (0.2) holds for $h$ assuming (0.3) holds for $h + 1$. Here we only consider the case that $a_2 \neq a_1$ and $a_2 \neq a^*$, since otherwise $\Pr[f | \overline{a_1}, a_2] = 1$ and thus (0.2) holds by the definition of the rewards and the fact that $V_h^*(f) = 0$. When $a_2 \notin \{a_1, a^*\}$, we have

$$\begin{aligned}
Q_h^*(\overline{a_1}, a_2) &= R_h(\overline{a_1}, a_2) + \mathbb{E}_{s_{h+1}} \left[ V_{h+1}^*(s_{h+1}) \middle| \overline{a_1}, a_2 \right] \\
&= -2\alpha \left[ \left\langle v(a_1), v(a_2) \right\rangle + 2\alpha \right] + \Pr[s_{h+1} = \overline{a_2}] \cdot V_{h+1}^*(a_2) + \Pr[s_{h+1} = f] \cdot V_{h+1}^*(f) \\
&= -2\alpha \left[ \left\langle v(a_1), v(a_2) \right\rangle + 2\alpha \right] + \left[ \left\langle v(a_1), v(a_2) \right\rangle + 2\alpha \right] \cdot \left( \left\langle v(a_1), v(a^*) \right\rangle + 2\alpha \right) \\
&= \left( \left\langle v(a_1), v(a_2) \right\rangle + 2\alpha \right) \cdot \left\langle v(a_1), v(a^*) \right\rangle.
\end{aligned}$$

This is exactly (0.2) for $h$. Hence both (0.2) and (0.3) hold for all $h \in [H]$. ∎

We now verify the constant sub-optimality gap, i.e. Assumption 5.6, is satisfied.

**Lemma 5.11.** *(Constant Gap) Assumption 5.6 is satisfied with $\Delta_{\min} = 1/24$.*

**Proof:** From Eq. (0.2) and (0.3), it is easy to see that at state $\overline{a_1} \neq \overline{a^*}$, for $a_2 \neq a^*$, the suboptimality gap is

$$\Delta_h(\overline{a_1}, a_2) := V_h^*(\overline{a_1}) - Q_h^*(\overline{a_1}, a_2) \geq \alpha - \max\left\{ 3\alpha^2, \frac{3}{4}\alpha \right\} = \frac{1}{24}.$$

Moreover, at state $\overline{a^*}$, for $a \neq a^*$, the suboptimality gap is

$$\Delta_h(\overline{a^*}, a) := V_h^*(\overline{a^*}) - Q_h^*(\overline{a^*}, a) \geq \frac{3}{4}\alpha - 3\alpha^2 = \frac{1}{24}.$$

Finally, for the terminal state $f$, the suboptimality gap is obviously 1. Therefore $\Delta_{\min} \geq \frac{1}{24}$ for all MDPs under consideration. ∎

### 5.2.3 The information-theoretic argument

We provide a proof sketch for the lower bound below in Theorem 5.8.

**Proof sketch.** We will show that for any algorithm, there exists $a^* \in [m]$ such that in order to output $\pi$ with

$$\mathbb{E}_{s_1 \sim \mu} V^\pi(s_1) \geq \mathbb{E}_{s_1 \sim \mu} V^*(s_1) - 0.05$$

with probability at least 0.1 for $\mathcal{M}_{a^*}$, the number of samples required is $2^{\Omega(\min\{d, H\})}$.

Observe that the feature map of $\mathcal{M}_{a^*}$ does not depend on $a^*$, and that for $h < H - 1$ and $a_2 \neq a^*$, the reward $R_h(\overline{a_1}, a_2)$ also contains no information about $a^*$. The transition probabilities are also independent of $a^*$, unless the

action $a^*$ is taken. Moreover, the reward at state $f$ is always $0$. Thus, to receive information about $a^*$, the agent either needs to take the action $a^*$, or be at a non-game-over state at the final time step ($h = H - 1$).

However, note that the probability of remaining at a non-terminal state at the next layer is at most

$$\sup_{a_1 \neq a_2} \langle v(a_1), v(a_2) \rangle + 2\alpha \leq 3\alpha \leq \frac{3}{4}.$$

Thus for any algorithm, $\Pr[s_{H-1} \neq f] \leq \left(\frac{3}{4}\right)^H$, which is exponentially small.

In other words, any algorithm that does not know $a^*$ either needs to "be lucky" so that $s_{H-1} = f$, or needs to take $a^*$ "by accident". Since the number of actions is $m = 2^{\Theta(d)}$, either event cannot happen with constant probability unless the number of episodes is exponential in $\min\{d, H\}$.

In order to make this claim rigorous, we can construct a reference MDP $\mathcal{M}_0$ as follows. The state space, action space, and features of $\mathcal{M}_0$ are the same as those of $\mathcal{M}_a$. The transitions are defined as follows:

$$\Pr[\cdot | \overline{a_1}, a_2] = \begin{cases} \overline{a_2} : \left\langle v(a_1), v(a_2) \right\rangle + 2\alpha \\ f : 1 - \left\langle v(a_1), v(a_2) \right\rangle - 2\alpha \end{cases}, \qquad (\forall a_1, a_2 \text{ s.t. } a_1 \neq a_2)$$

$$\Pr[f | f, \cdot] = 1.$$

The rewards are defined as follows:

$$R_h(\overline{a_1}, a_2) := -2\alpha \left[ \left\langle v(a_1), v(a_2) \right\rangle + 2\alpha \right], \qquad (\forall a_1, a_2 \text{ s.t. } a_1 \neq a_2)$$

$$R_h(f, \cdot) := 0.$$

Note that $\mathcal{M}_0$ is identical to $\mathcal{M}_{a^*}$, except when $a^*$ is taken, or when an trajectory ends at a non-terminal state. Since the latter event happens with an exponentially small probability, we can show that for any algorithm the probability of taking $a^*$ in $\mathcal{M}_{a^*}$ is close to the probability of taking $a^*$ in $\mathcal{M}_0$. Since $\mathcal{M}_0$ is independent of $a^*$, unless an exponential number of samples are used, for any algorithm there exists $a^* \in [m]$ such that the probability of taking $a^*$ in $\mathcal{M}_0$ is $o(1)$. It then follows that the probability of taking $a^*$ in $\mathcal{M}_{a^*}$ is $o(1)$. Since $a^*$ is the optimal action for every state, such an algorithm cannot output a near-optimal policy for $\mathcal{M}_{a^*}$.

## 5.3 Linearly Realizable $\pi^\star$

Similar to value-based learning, a natural assumption for policy-based learning is that the optimal policy is realizable by a halfspace.

**Assumption 5.12** (Linear $\pi^*$ Realizability). For any $h \in [H]$, there exists $\theta_h^\star \in \mathbb{R}^d$ that satisfies for any $s \in \mathcal{S}$, we have

$$\pi^* (s) \in \mathrm{argmax}_a \langle \theta_h, \phi(s, a) \rangle.$$

As before (and as is natural in supervised learning), we will also consider the case where that theere is large margin (which is analogous to previous large gap assumption).

**Assumption 5.13** (Constant Margin). Without loss of generality, assume the scalings are such that for all $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, $\|\phi(s, a)\|_2 \leq 1$ and $\|\theta_h^\star\| \leq 1$ Now suppose that for all $s \in \mathcal{S}$ and any $a \notin \pi^*(s)$,

$$\langle \theta_h, \phi(s, \pi^*(s)) \rangle - \langle \theta_h, \phi(s, a) \rangle \geq \Delta_{\min}.$$

Here we restrict the linear coefficients and features to have unit norm for normalization.

We state the following without proof (see Section 5.5):

**Theorem 5.14.** *(Linear $\pi^\star$ Realizability + Margin) Consider any algorithm $\mathcal{A}$ which has access to a generative model and which takes as input the feature mapping $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$. There exists an MDP with a feature mapping $\phi$ satisfying Assumption 5.12 and Assumption 5.13 (where $\Delta_{\min}$ is an absolute constant) such that if $\mathcal{A}$ (with $\phi$ as input) finds a policy $\pi$ such that*

$$\mathbb{E}_{s_1 \sim \mu} V^\pi(s_1) \geq \mathbb{E}_{s_1 \sim \mu} V^*(s_1) - 0.05$$

*with probability $0.1$, then $\mathcal{A}$ requires $\min\{2^{cd}, 2^{cH}\}$ samples, where $c$ is an absolute constant.*

## 5.4 Discussion: Studying Generalization in RL

The previous lower bounds shows that: (i) without further assumptions, agnostic learning (in the standard supervised learning sense) is not possible in RL, unless we can tolerate an exponential dependence on the horizon, and (ii) simple realizability assumptions are also not sufficient for sample efficient RL.

This motivates the study of RL to settings where we either make stronger assumptions or have a means in which the agent can obtain side information. Three examples of these approaches that are considered in this book are:

- Structural (and Modelling) Assumptions: By making stronger assumptions about the how the hypothesis class relates to the underlying MDP, we can move away from agnostic learning lower bound. We have seen one example of this with the stronger Bellman completeness assumption that we considered in Chapter 3. We will see more examples of this in Part 2.

- Distribution Dependent Results (and Distribution Shift): We have seen one example of this approach when we consider the approximate dynamic programming approach in Chapter 4. When we move to policy gradient methods (in Part 3), we will also consider results which depend on the given distribution (i.e. where we suppose our starting state distribution $\mu$ has some reasonable notion of coverage).

- Imitation Learning and Behavior Cloning: here we will consider settings where the agent has input from, effectively, a teacher, and we see how this can circumvent statistical hardness results.

## 5.5 Bibliographic Remarks and Further Readings

The reduction from reinforcement learning to supervised learning was first introduced in [Kearns et al., 2000], which used a different algorithm (the "trajectory tree" algorithm), as opposed to the importance sampling approach presented here. [Kearns et al., 2000] made the connection to the VC dimension of the policy class. The fundamental sample complexity tradeoff — between polynomial dependence on the size of the state space and exponential dependence on the horizon — was discussed in depth in [Kakade, 2003].

Utilizing linear methods for dynamic programming goes back to, at least, [Shannon, 1959, Bellman and Dreyfus, 1959]. Formally considering the linear $Q^\star$ assumption goes back to at least [Wen and Van Roy, 2017]. Resolving the learnability under this assumption was a important open question discussed in [Du et al., 2019], which is now resolved. Here, Theorem 5.7, which resolved this question, is due to [Weisz et al., 2021], and Theorem 5.8, which resolved this question along with a gap, is due to [Wang et al., 2021]. Theorem 5.14, which assumed a linearly realizable optimal policy, is due to [Du et al., 2019].

# Part 2

## Strategic Exploration

# Chapter 6

# Multi-armed & Linear Bandits

For the case, where $\gamma = 0$ (or $H = 1$ in the undiscounted case), the problem of learning in an unknown MDP reduce to the multi-armed bandit problem. The basic algorithms and proof methodologies here are important to understand in their own right, due to that we will have to extend these with more sophisticated variants to handle the exploration-exploitation tradeoff in the more challenging reinforcement learning problem.

This chapter follows analysis of the LinUCB algorithm from the original proof in [Dani et al., 2008], with a simplified concentration analysis due to [Abbasi-Yadkori et al., 2011].

Throughout this chapter, we assume reward is stochastic.

## 6.1 The $K$-Armed Bandit Problem

The setting is where we have $K$ decisions (the "arms"), where when we play arm $a \in \{1, 2, \ldots K\}$ we obtain a random reward $r_a \in [-1, 1]$ from $R(a) \in \Delta([-1, 1])$ which has mean reward:

$$\mathbb{E}_{r_a \sim R(a)}[r_a] = \mu_a$$

where it is easy to see that $\mu_a \in [-1, 1]$.

Every iteration $t$, the learner will pick an arm $I_t \in [1, 2, \ldots K]$. Our cumulative regret is defined as:

$$R_T = T \cdot \max_i \mu_i - \sum_{t=0}^{T-1} \mu_{I_t}$$

We denote $a^\star = \mathrm{argmax}_i \mu_i$ as the optimal arm. We define gap $\Delta_a = \mu_{a^\star} - \mu_a$ for any arm $a$.

**Theorem 6.1.** *There exists an algorithm such that with probability at least $1 - \delta$, we have:*

$$R_T = O\left(\min\left\{\sqrt{KT \cdot \ln(TK/\delta)}, \sum_{a \neq a^\star} \frac{\ln(TK/\delta)}{\Delta_a}\right\} + K\right).$$

### 6.1.1 The Upper Confidence Bound (UCB) Algorithm

We summarize the upper confidence bound (UCB) algorithm in Alg. 2. For simplicity, we allocate the first $K$ rounds to pull each arm once.

**Algorithm 2** UCB
___
1: Play each arm once and denote received reward as $r_a$ for all $a \in \{1, 2, \ldots K\}$
2: **for** $t = 1 \to T - K$ **do**
3:      Execute arm $I_t = \arg\max_{i \in [K]} \left( \hat{\mu}_i^t + \sqrt{\frac{\log(TK/\delta)}{N_i^t}} \right)$
4:      Observe $r_t := r_{I_t}$
5: **end for**
___

where every iteration $t$, we maintain counts of each arm:

$$N_a^t = 1 + \sum_{i=1}^{t-1} \mathbf{1}\{I_i = a\},$$

where $I_t$ is the index of the arm that is picked by the algorithm at iteration $t$. We main the empirical mean for each arm as follows:

$$\hat{\mu}_a^t = \frac{1}{N_a^t} \left( r_a + \sum_{i=1}^{t-1} \mathbf{1}\{I_i = a\} r_i \right).$$

Recall that $r_a$ is the reward of arm $a$ we got during the first $K$ rounds.

We also main the upper confidence bound for each arm as follows:

$$\hat{\mu}_a^t + 2 \sqrt{\frac{\ln(TK/\delta)}{N_a^t}}.$$

The following lemma shows that this is a valid upper confidence bound with high probability.

**Lemma 6.2** (Upper Confidence Bound). *For all $t \in [1, \ldots, T]$ and $a \in [1, 2, \ldots K]$, we have that with probability at least $1 - \delta$,*

$$\left| \hat{\mu}_a^t - \mu_a \right| \leq 2 \sqrt{\frac{\ln(TK/\delta)}{N_a^t}}. \tag{0.1}$$

The proof of the above lemma uses Azuma-Hoeffding's inequality (Theorem A.5) for each arm $a$ and iteration $t$ and then apply a union bound over all $T$ iterations and $K$ arms. The reason one needs to use Azuma-Hoeffding's inequality is that here the number of trials of arm $a$, i.e. $N_a^t$, itself is a random variable while the classic Hoeffding's inequality only applies to the setting where the number of trials is a fixed number.

**Proof:** We first consider a fixed arm $a$. Let us define the following random variables, $X_0 = r_a - \mu_a, X_1 = \mathbf{1}\{I_1 = a\}(r_1 - \mu_a), X_2 = \mathbf{1}\{I_2 = a\}(r_2 - \mu_a), \ldots, X_i = \mathbf{1}\{I_i = a\}(r_i - \mu_a), \ldots$.

Regarding the boundness on these random variables, we have that for $i$ where $\mathbf{1}\{I_i = a\} = 1$, we have $|X_i| \leq 1$, and for $i$ where $\mathbf{1}\{I_i = a\} = 0$, we have $|X_i| = 0$. Now, consider $\mathbb{E}[X_i | \mathcal{H}_{<i}]$, where $\mathcal{H}_{<i}$ is all history up to but not including iteration $i$. Note that we have $\mathbb{E}[X_i | \mathcal{H}_{<i}] = 0$ since condition on history $\mathcal{H}_{<i}$, $\mathbf{1}\{I_i = a\}$ is a deterministic quantity (i.e., UCB algorithm determines whether or not to pull arm $a$ at iteration $i$ only based on the information from $\mathcal{H}_{<i}$). Thus, we can conclude that $\{X_i\}_{i=0}$ is a Martinagle difference sequence. Via Azuma-Hoeffding's inequality, we have that with probability at least $1 - \delta$, for any fixed $t$, we have:

$$\left| \sum_{i=0}^{t-1} X_i \right| = \left| N_a^t \hat{\mu}_a^t - N_a^t \mu_a \right| \leq 2 \sqrt{\ln(1/\delta) N_a^t},$$

where we use the fact that $\sum_{i=0}^{t-1} |X_i|^2 \le N_a^t$. Divide $N_a^t$ on both side, we have:

$$\left|\widehat{\mu}_a^t - \mu_a\right| \le 2\sqrt{\ln(1/\delta)/N_a^t}$$

Apply union bound over all $0 \le t \le T$ and $a \in \{1, \ldots, K\}$, we have that with probability $1 - \delta$, for all $t, a$:

$$\left|\widehat{\mu}_a^t - \mu_a\right| \le 2\sqrt{\ln(1/\delta)/N_a^t} \le 2\sqrt{\ln(TK/\delta)/N_a^t}$$

This concludes the proof. $\blacksquare$

Now we can conclude the proof of the main theorem.

**Proof:** Below we conditioned on the above inequality 0.1 holds. This gives us the following optimism:

$$\mu_a \le \widehat{\mu}_a^t + 2\sqrt{\frac{\ln(TK/\delta)}{N_a^t}}, \forall a, t.$$

Thus, we can upper bound the regret as follows:

$$\mu_{a^\star} - \mu_{I_t} \le \widehat{\mu}_{I_t}^t + 2\sqrt{\frac{\ln(TK/\delta)}{N_{I_t}^t}} - \mu_{I_t} \le 4\sqrt{\frac{\ln(TK/\delta)}{N_{I_t}^t}}.$$

Sum over all iterations, we get:

$$\sum_{t=0}^{T-1} \mu_{a^\star} - \mu_{I_t} \le 4\sqrt{\ln(TK/\delta)} \sum_{t=0}^{T-1} \sqrt{\frac{1}{N_{I_t}^t}}$$

$$= 4\sqrt{\ln(TK/\delta)} \sum_a \sum_{i=1}^{N_a^T} \frac{1}{\sqrt{i}} \le 8\sqrt{\ln(TK/\delta)} \sum_a \sqrt{N_a^T} \le 8\sqrt{\ln(TK/\delta)} \sqrt{K \sum_a N_a^T}$$

$$\le 8\sqrt{\ln(TK/\delta)} \sqrt{KT}.$$

Note that our algorithm has regret $K$ at the first K rounds.

On the other hand, if for each arm $a$, the gap $\Delta_a > 0$, then, we must have:

$$N_a^T \le \frac{4\ln(TK/\delta)}{\Delta_a^2}.$$

which is because after the UCB of an arm $a$ is below $\mu_{a^\star}$, UCB algorithm will never pull this arm $a$ again (the UCB of $a^\star$ is no smaller than $\mu_{a^\star}$).

Thus for the regret calculation, we get:

$$\sum_{t=0}^{T-1} \mu_{a^\star} - \mu_{I_t} \le \sum_{a \ne a^\star} N_a^T \Delta_a \le \sum_{a \ne a^\star} \frac{4\ln(TK/\delta)}{\Delta_a}.$$

Together with the fact that Inequality 0.1 holds with probability at least $1 - \delta$, we conclude the proof. $\blacksquare$

## 6.2 Linear Bandits: Handling Large Action Spaces

Let $D \subset \mathbb{R}^d$ be a compact (but otherwise arbitrary) set of decisions. On each round, we must choose a decision $x_t \in D$. Each such choice results in a reward $r_t \in [-1, 1]$.

**Algorithm 3** The Linear UCB algorithm

**Input:** $\lambda$, $\beta_t$

1: **for** $t = 0, 1 \ldots$ **do**
2:    Execute

$$x_t = \operatorname{argmax}_{x \in D} \max_{\mu \in \text{BALL}_t} \mu \cdot x$$

   and observe the reward $r_t$.
3:    Update $\text{BALL}_{t+1}$ (as specified in Equation 0.2).
4: **end for**

---

We assume that, regardless of the history $\mathcal{H}$ of decisions and observed rewards, the conditional expectation of $r_t$ is a fixed linear function, i.e. for all $x \in D$,

$$\mathbb{E}[r_t | x_t = x] = \mu^\star \cdot x \in [-1, 1],$$

where $x \in D$ is arbitrary. Here, observe that we have assumed the mean reward for any decision is bounded in $[-1, 1]$. Under these assumptions, the *noise sequence*,

$$\eta_t = r_t - \mu^\star \cdot x_t$$

is a martingale difference sequence.

The is problem is essentially a bandit version of a fundamental geometric optimization problem, in which the agent's feedback on each round $t$ is only the observed reward $r_t$ and where the agent does not know $\mu^\star$ apriori.

If $x_0, \ldots x_{T-1}$ are the decisions made in the game, then define the *cumulative regret* by

$$R_T = \mu^\star \cdot x^\star - \sum_{t=0}^{T-1} \mu^\star \cdot x_t$$

where $x^\star \in D$ is an optimal decision for $\mu^\star$, i.e.

$$x^\star \in \operatorname{argmax}_{x \in D} \mu^\star \cdot x$$

which exists since $D$ is compact. Observe that if the mean $\mu^\star$ were known, then the optimal strategy would be to play $x^\star$ every round. Since the expected loss for each decision $x$ equals $\mu^\star \cdot x$, the cumulative regret is just the difference between the expected loss of the optimal algorithm and the expected loss for the actual decisions $x_t$. By the Hoeffding-Azuma inequality (see Lemma A.5), the observed reward $\sum_{t=0}^{T-1} r_t$ will be close to their (conditional) expectations $\sum_{t=0}^{T-1} \mu^\star \cdot x_t$.

Since the sequence of decisions $x_1, \ldots, x_{T-1}$ may depend on the particular sequence of random noise encountered, $R_T$ is a random variable. Our goal in designing an algorithm is to keep $R_T$ as small as possible.

## 6.2.1 The LinUCB algorithm

LinUCB is based on "optimism in the face of uncertainty," which is described in Algorithm 3. At episode $t$, we use all previous experience to define an uncertainty region (an ellipse) $\text{BALL}_t$. The center of this region, $\widehat{\mu}_t$, is the solution of the following regularized least squares problem:

$$\widehat{\mu}_t = \arg \min_\mu \sum_{\tau=0}^{t-1} \|\mu \cdot x_\tau - r_\tau\|_2^2 + \lambda \|\mu\|_2^2$$

$$= \Sigma_t^{-1} \sum_{\tau=0}^{t-1} r_\tau x_\tau,$$

where $\lambda$ is a parameter and where

$$\Sigma_t = \lambda I + \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top, \text{ with } \Sigma_0 = \lambda I.$$

The shape of the region $\text{BALL}_t$ is defined through the feature covariance $\Sigma_t$.

Precisely, the uncertainty region, or confidence ball, is defined as:

$$\text{BALL}_t = \left\{ \mu | (\widehat{\mu}_t - \mu)^\top \Sigma_t (\widehat{\mu}_t - \mu) \leq \beta_t \right\}, \tag{0.2}$$

where $\beta_t$ is a parameter of the algorithm.

**Computation.** Suppose that we have an efficient linear optimization oracle, i.e. that we can efficiently solve the problem:

$$\max_{x \in D} \nu \cdot x$$

for any $\nu$. Even with this, Step 2 of LinUCB may not be computationally tractable. For example, suppose that $D$ is provided to us as a polytope, then the above oracle can be efficiently computed using linear programming, while LinUCB is an NP-hard optimization. Here, we can actually use a wider confidence region, where we can keep track of $\ell_1$ ball which contains $\text{BALL}_t$. See Section 6.4 for further reading.

## 6.2.2 Upper and Lower Bounds

Our main result here is that we have sublinear regret with only a polynomial dependence on the dimension $d$ and, importantly, no dependence on the cardinality of the decision space $D$, i.e. on $|D|$.

**Theorem 6.3.** *Suppose that the noise $\eta_t$ is $\sigma^2$ sub-Gaussian* [1]*, that $\|\mu^\star\| \leq W$, and that $\|x\| \leq B$ for all $x \in D$. Set $\lambda = \sigma^2/W^2$ and*

$$\beta_t := \sigma^2 \left( 2 + 4d \log \left( 1 + \frac{tB^2 W^2}{d} \right) + 8 \log(4/\delta) \right).$$

*We have that with probability greater than $1 - \delta$, that (simultaneously) for all $T \geq 0$,*

$$R_T \leq c\sigma\sqrt{T} \left( d \log \left( 1 + \frac{TB^2 W^2}{d\sigma^2} \right) + \log(4/\delta) \right)$$

*where $c$ is an absolute constant. In other words, we have that $R_T$ is $\tilde{O}(d\sqrt{T})$ with high probability.*

The following shows that no algorithm can do better.

**Theorem 6.4.** *(Lower bound) There exists a distribution over linear bandit problems (i.e. a distribution over $\mu$) with the rewards being bounded by 1 in magnitude and $\sigma^2 \leq 1$, such that for every (randomized) algorithm, we have for $n \geq \max\{256, d^2/16\}$,*

$$\mathbb{E}_\mu \mathbb{E} R_T \geq \frac{1}{2500} d\sqrt{T}.$$

*where the inner expectation is with respect to randomness in the problem and the algorithm.*

---

[1] Roughly speaking, this say that tail probabilities of $\eta_t$ decay no more slowly than a Gaussian distribution. If the noise is bounded, i.e. $|\eta_t| \leq B$ with probability one, then we can take $\sigma^2 = B^2$.

## 6.3 LinUCB Analysis

In establishing the upper bounds there are two main propositions from which the upper bounds follow. The first is in showing that the confidence region is appropriate.

**Proposition 6.5.** *(Confidence) Let $\delta > 0$. We have that*

$$\Pr(\forall t, \; \mu^\star \in \mathrm{BALL}_t) \geq 1 - \delta.$$

Section 6.3.2 is devoted to establishing this confidence bound. In essence, the proof seeks to understand the growth of the quantity $(\widehat{\mu}_t - \mu^\star)^\top \Sigma_t (\widehat{\mu}_t - \mu^\star)$.

The second main step in analyzing LinUCB is to show that as long as the aforementioned high-probability event holds, we have some control on the growth of the regret. Let us define

$$\mathrm{regret}_t = \mu^\star \cdot x^* - \mu^\star \cdot x_t$$

which denotes the instantaneous regret.

The following bounds the sum of the squares of instantaneous regret.

**Proposition 6.6.** *(Sum of Squares Regret Bound) Suppose that $\|x\| \leq B$ for $x \in D$. Suppose $\beta_t$ is increasing and larger than $1$. For LinUCB, if $\mu^\star \in \mathrm{BALL}_t$ for all $t$, then*

$$\sum_{t=0}^{T-1} \mathrm{regret}_t^2 \leq 4\beta_T d \log\left(1 + \frac{TB^2}{d\lambda}\right)$$

This is proven in Section 6.3.1. The idea of the proof involves a potential function argument on the log volume (i.e. the log determinant) of the "precision matrix" $\Sigma_t$ (which tracks how accurate our estimates of $\mu^\star$ are in each direction). The proof involves relating the growth of this volume to the regret.

Using these two results we are able to prove our upper bound as follows:

**Proof:**[Proof of Theorem 6.3] By Propositions 6.5 and 6.6 along with the Cauchy-Schwarz inequality, we have, with probability at least $1 - \delta$,

$$R_T = \sum_{t=0}^{T-1} \mathrm{regret}_t \leq \sqrt{T \sum_{t=0}^{T-1} \mathrm{regret}_t^2} \leq \sqrt{4T\beta_T d \log\left(1 + \frac{TB^2}{d\lambda}\right)}.$$

The remainder of the proof follows from using our chosen value of $\beta_T$ and algebraic manipulations (that $2ab \leq a^2 + b^2$). ∎

We now provide the proofs of these two propositions.

### 6.3.1 Regret Analysis

In this section, we prove Proposition 6.6, which says that the sum of the squares of the instantaneous regrets of the algorithm is small, assuming the evolving confidence balls always contain the true mean $\mu^\star$. An important observation is that on any round $t$ in which $\mu^\star \in \mathrm{BALL}_t$, the instantaneous regret is at most the "width" of the ellipsoid in the direction of the chosen decision. Moreover, the algorithm's choice of decisions forces the ellipsoids to shrink at a rate that ensures that the sum of the squares of the widths is small. We now formalize this.

Unless explicitly stated, all norms refer to the $\ell_2$ norm.

**Lemma 6.7.** *Let $x \in D$. If $\mu \in \mathrm{BALL}_t$ and $x \in D$. Then*

$$|(\mu - \widehat{\mu}_t)^\top x| \leq \sqrt{\beta_t x^\top \Sigma_t^{-1} x}$$

**Proof:** By Cauchy-Schwarz, we have:

$$|(\mu - \widehat{\mu}_t)^\top x| = |(\mu - \widehat{\mu}_t)^\top \Sigma_t^{1/2} \Sigma_t^{-1/2} x| = |(\Sigma_t^{1/2}(\mu - \widehat{\mu}_t))^\top \Sigma_t^{-1/2} x|$$

$$\leq \|\Sigma_t^{1/2}(\mu - \widehat{\mu}_t)\| \|\Sigma_t^{-1/2} x\| = \|\Sigma_t^{1/2}(\mu - \widehat{\mu}_t)\| \sqrt{x^\top \Sigma_t^{-1} x} \leq \sqrt{\beta_t x^\top \Sigma_t^{-1} x}$$

where the last inequality holds since $\mu \in \mathrm{BALL}_t$. $\blacksquare$

Define

$$w_t := \sqrt{x_t^\top \Sigma_t^{-1} x_t}$$

which we interpret as the "normalized width" at time $t$ in the direction of the chosen decision. We now see that the width, $2\sqrt{\beta_t} w_t$, is an upper bound for the instantaneous regret.

**Lemma 6.8.** *Fix $t \leq T$. If $\mu^\star \in \mathrm{BALL}_t$, then*

$$\mathrm{regret}_t \leq 2 \min\left(\sqrt{\beta_t} w_t, 1\right) \leq 2\sqrt{\beta_T} \min\left(w_t, 1\right)$$

**Proof:** Let $\widetilde{\mu} \in \mathrm{BALL}_t$ denote the vector which minimizes the dot product $\widetilde{\mu}^\top x_t$. By choice of $x_t$, we have

$$\widetilde{\mu}^\top x_t = \max_{\mu \in \mathrm{BALL}_t} \max_{x \in D} \mu^\top x \geq (\mu^\star)^\top x^*,$$

where the inequality used the hypothesis $\mu^\star \in \mathrm{BALL}_t$. Hence,

$$\mathrm{regret}_t = (\mu^\star)^\top x^* - (\mu^\star)^\top x_t \leq (\widetilde{\mu} - \mu^\star)^\top x_t$$

$$= (\widetilde{\mu} - \widehat{\mu}_t)^\top x_t + (\widehat{\mu}_t - \mu^\star)^\top x_t \leq 2\sqrt{\beta_t} w_t$$

where the last step follows from Lemma 6.7 since $\widetilde{\mu}$ and $\mu^\star$ are in $\mathrm{BALL}_t$. Since $r_t \in [-1, 1]$, $\mathrm{regret}_t$ is always at most 2 and the first inequality follows. The final inequality is due to that $\beta_t$ is increasing and larger than 1. $\blacksquare$

The following two lemmas prove useful in showing that we can treat the log determinant as a potential function, where can bound the sum of widths independently of the choices made by the algorithm.

**Lemma 6.9.** *We have:*

$$\det \Sigma_T = \det \Sigma_0 \prod_{t=0}^{T-1} (1 + w_t^2).$$

**Proof:** By the definition of $\Sigma_{t+1}$, we have

$$\det \Sigma_{t+1} = \det(\Sigma_t + x_t x_t^\top) = \det(\Sigma_t^{1/2}(I + \Sigma_t^{-1/2} x_t x_t^\top \Sigma_t^{-1/2})\Sigma_t^{1/2})$$

$$= \det(\Sigma_t) \det(I + \Sigma_t^{-1/2} x_t (\Sigma_t^{-1/2} x_t)^\top) = \det(\Sigma_t) \det(I + v_t v_t^\top),$$

where $v_t := \Sigma_t^{-1/2} x_t$. Now observe that $v_t^\top v_t = w_t^2$ and

$$(I + v_t v_t^\top) v_t = v_t + v_t(v_t^\top v_t) = (1 + w_t^2) v_t$$

Hence $(1 + w_t^2)$ is an eigenvalue of $I + v_t v_t^\top$. Since $v_t v_t^\top$ is a rank one matrix, all other eigenvalues of $I + v_t v_t^\top$ equal 1. Hence, $\det(I + v_t v_t^\top)$ is $(1 + w_t^2)$, implying $\det \Sigma_{t+1} = (1 + w_t^2) \det \Sigma_t$. The result follows by induction. $\blacksquare$

**Lemma 6.10.** *("Potential Function" Bound) For any sequence $x_0, \ldots x_{T-1}$ such that, for $t < T$, $\|x_t\|_2 \leq B$, we have:*

$$\log\left(\det \Sigma_{T-1}/\det \Sigma_0\right) = \log\det\left(I + \frac{1}{\lambda}\sum_{t=0}^{T-1} x_t x_t^\top\right) \leq d\log\left(1 + \frac{TB^2}{d\lambda}\right).$$

**Proof:** Denote the eigenvalues of $\sum_{t=0}^{T-1} x_t x_t^\top$ as $\sigma_1, \ldots \sigma_d$, and note:

$$\sum_{i=1}^{d} \sigma_i = \text{Trace}\left(\sum_{t=0}^{T-1} x_t x_t^\top\right) = \sum_{t=0}^{T-1} \|x_t\|^2 \leq TB^2.$$

Using the AM-GM inequality,

$$\log\det\left(I + \frac{1}{\lambda}\sum_{t=0}^{T-1} x_t x_t^\top\right) = \log\left(\prod_{i=1}^{d}(1 + \sigma_i/\lambda)\right)$$

$$= d\log\left(\prod_{i=1}^{d}(1+\sigma_i/\lambda)\right)^{1/d} \leq d\log\left(\frac{1}{d}\sum_{i=1}^{d}(1+\sigma_i/\lambda)\right) \leq d\log\left(1 + \frac{TB^2}{d\lambda}\right),$$

which concludes the proof. ∎

Finally, we are ready to prove that if $\mu^\star$ always stays within the evolving confidence region, then our regret is under control.

**Proof:**[Proof of Proposition 6.6] Assume that $\mu^\star \in \text{BALL}_t$ for all $t$. We have that:

$$\sum_{t=0}^{T-1} \text{regret}_t^2 \leq \sum_{t=0}^{T-1} 4\beta_t \min(w_t^2, 1) \leq 4\beta_T \sum_{t=0}^{T-1} \min(w_t^2, 1)$$

$$\leq 4\beta_T \sum_{t=0}^{T-1} \ln(1 + w_t^2) \leq 4\beta_T \log\left(\det\Sigma_{T-1}/\det\Sigma_0\right) = 4\beta_T d\log\left(1 + \frac{TB^2}{d\lambda}\right)$$

where the first inequality follow from By Lemma 6.8; the second from that $\beta_t$ is an increasing function of $t$; the third uses that for $0 \leq y \leq 1$, $\ln(1+y) \geq y/2$; the final two inequalities follow by Lemmas 6.9 and 6.10. ∎

### 6.3.2 Confidence Analysis

**Proof:**[Proof of Proposition 6.5] Since $r_\tau = x_\tau \cdot \mu^\star + \eta_\tau$, we have:

$$\widehat{\mu}_t - \mu^\star = \Sigma_t^{-1}\sum_{\tau=0}^{t-1} r_\tau x_\tau - \mu^\star = \Sigma_t^{-1}\sum_{\tau=0}^{t-1} x_\tau(x_\tau \cdot \mu^\star + \eta_\tau) - \mu^\star$$

$$= \Sigma_t^{-1}\left(\sum_{\tau=0}^{t-1} x_\tau(x_\tau)^\top\right)\mu^\star - \mu^\star + \Sigma_t^{-1}\sum_{\tau=0}^{t-1}\eta_\tau x_\tau = \lambda\Sigma_t^{-1}\mu^\star + \Sigma_t^{-1}\sum_{\tau=0}^{t-1}\eta_\tau x_\tau$$

For any $0 < \delta_t < 1$, using Lemma A.9, it holds with probability at least $1 - \delta_t$,

$$\sqrt{(\widehat{\mu}_t - \mu^\star)^\top \Sigma_t(\widehat{\mu}_t - \mu^\star)} = \|(\Sigma_t)^{1/2}(\widehat{\mu}_t - \mu^\star)\|$$

$$\leq \left\|\lambda\Sigma_t^{-1/2}\mu^\star\right\| + \left\|\Sigma_t^{-1/2}\sum_{\tau=0}^{t-1}\eta_\tau x_\tau\right\|$$

$$\leq \sqrt{\lambda}\|\mu^\star\| + \sqrt{2\sigma^2\log\left(\det(\Sigma_t)\det(\Sigma^0)^{-1}/\delta_t\right)}.$$

where we have also used the triangle inequality and that $\|\Sigma_t^{-1}\| \leq 1/\lambda$.

We seek to lower bound $\Pr(\forall t, \mu^\star \in \text{BALL}_t)$. Note that at $t = 0$, by our choice of $\lambda$, we have that $\text{BALL}_0$ contains $W^\star$, so $\Pr(\mu^\star \notin \text{BALL}_0) = 0$. For $t \geq 1$, let us assign failure probability $\delta_t = (3/\pi^2)/t^2\delta$ for the $t$-th event, which, using the above, gives us an upper bound on the sum failure probability as

$$1 - \Pr(\forall t, \mu^\star \in \text{BALL}_t) = \Pr(\exists t, \mu^\star \notin \text{BALL}_t) \leq \sum_{t=1}^{\infty} \Pr(\mu^\star \notin \text{BALL}_t) < \sum_{t=1}^{\infty} (1/t^2)(3/\pi^2)\delta = \delta/2.$$

This along with Lemma 6.10 completes the proof. ∎

## 6.4   Bibliographic Remarks and Further Readings

The orignal multi-armed bandit model goes to back to [Robbins, 1952]. The linear bandit model was first introduced in [Abe and Long, 1999]. Our analysis of the LinUCB algorithm follows from the original proof in [Dani et al., 2008], with a simplified concentration analysis due to [Abbasi-Yadkori et al., 2011]. The first sublinear regret bound here was due to [Auer et al., 2002], which used a more complicated algorithm.

The lower bound we present is also due to [Dani et al., 2008], which also shows that LinUCB is minimax optimal.

# Chapter 7

# Strategic Exploration in Tabular MDPs

We now turn to how an agent acting in an unknown MDP can obtain a near-optimal reward over time. Compared with the previous setting with access to a generative model, we no longer have easy access to transitions at each state, but only have the ability to execute trajectories in the MDP. The main complexity this adds to the learning process is that the agent has to engage in exploration, that is, plan to reach new states where enough samples have not been seen yet, so that optimal behavior in those states can be learned.

We will work with finite-horizon MDPs with a fixed start state $s_0$ (see Section 1.2), and we will assume the agent learns in an episodic setting, as introduced in Chapter 2. Here, in in every episode $k$, the learner acts for $H$ step starting from a fixed starting state $s_0$ and, at the end of the $H$-length episode, the state is reset to $s_0$. It is straightforward to extend this setting where the starting state is sampled from a distribution, i.e. $s_0 \sim \mu$.

The goal of the agent is to minimize her expected cumulative regret over $K$ episodes:

$$\text{Regret} := \mathbb{E}\left[ KV^\star(s_0) - \sum_{k=0}^{K-1} \sum_{h=0}^{H-1} r(s_h^k, a_h^k) \right],$$

where the expectation is with respect to the randomness of the MDP environment and, possibly, any randomness of the agent's strategy.

This chapter considers tabular MDPs, where $\mathcal{S}$ and $\mathcal{A}$ are discrete. We will now present a (nearly) sublinear regret algorithm, UCB-Value Iteration. This chapter follows the proof in [Azar et al., 2017]. We first give a simplified analysis of UCB-VI that gives a regret bound in the order of $H^2 S \sqrt{AK}$, followed by a more refined analysis that gives a $H^2 \sqrt{SAK} + H^3 S^2 A$ regret bound.

## 7.1   On The Need for Strategic Exploration

Consider the chain MDP of length $H + 2$ shown in Figure 0.1. The starting state of interest is state $s_0$. We consider a uniformly random policy $\pi_h(s)$ that at any time step $h$ and any state $s$, selects an action from all A many actions uniform randomly. It is easy to see that starting from $s_0$, the probability of such uniform random policy hitting the rewarding state $s_{H+1}$ is exponentially small, i.e., $O\left(A^H\right)$, since it needs to select the right action that moves one step right at every time step h. This example demonstrates that we need to perform strategic exploration in order to avoid the exponential sample complexity.
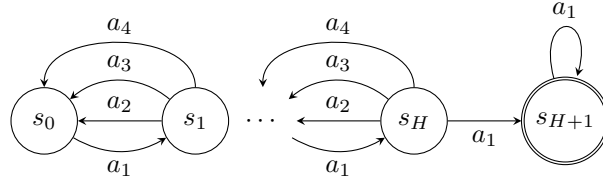
Figure 0.1: A deterministic, chain MDP of length $H + 2$. Rewards are $0$ everywhere other than $r(s_{H+1}, a_1) = 1$. In this example we have $A = 4$.

---

**Algorithm 4** UCBVI

---

**Input:** reward function $r$ (assumed to be known), confidence parameters
1: **for** $k = 0 \ldots K - 1$ **do**
2:     Compute $\widehat{P}_h^k$ as the empirical estimates, for all $h$ (Eq. 0.1)
3:     Compute reward bonus $b_h^k$ for all h (Eq. 0.2)
4:     Run Value-Iteration on $\{\widehat{P}_h^k, r + b_h^k\}_{h=0}^{H-1}$ (Eq. 0.3)
5:     Set $\pi^k$ as the returned policy of VI.
6: **end for**

---

## 7.2 The UCB-VI algorithm

If the learner then executes $\pi^k$ in the underlying MDP to generate a single trajectory $\tau^k = \{s_h^k, a_h^k\}_{h=0}^{H-1}$ with $a_h = \pi_h^k(s_h^k)$ and $s_{h+1}^k \sim P_h(\cdot | s_h^k, a_h^k)$. We first define some notations below. Consider the very beginning of episode $k$. We use the history information up to the end of episode $k - 1$ (denoted as $\mathcal{H}_{<k}$) to form some statistics. Specifically, we define:

$$N_h^k(s, a, s') = \sum_{i=0}^{k-1} \mathbf{1}\{(s_h^i, a_h^i, s_{h+1}^i) = (s, a, s')\},$$

$$N_h^k(s, a) = \sum_{i=0}^{k-1} \mathbf{1}\{(s_h^i, a_h^i) = (s, a)\}, \forall h, s, a.$$

Namely, we maintain counts of how many times $s, a, s'$ and $s, a$ are visited at time step $h$ from the beginning of the learning process to the end of the episode $k - 1$. We use these statistics to form an empirical model:

$$\widehat{P}_h^k(s' | s, a) = \frac{N_h^k(s, a, s')}{N_h^k(s, a)}, \forall h, s, a, s'. \tag{0.1}$$

We will also use the counts to define a *reward bonus*, denoted as $b_h(s, a)$ for all $h, s, a$. Denote $L := \ln(SAHK/\delta)$ ($\delta$ as usual represents the failure probability which we will define later). We define reward bonus as follows:

$$b_h^k(s, a) = 2H\sqrt{\frac{L}{N_h^k(s, a)}}. \tag{0.2}$$

With reward bonus and the empirical model, the learner uses *Value Iteration* on the empirical transition $\widehat{P}_h^k$ and the combined reward $r_h + b_h^k$. Starting at $H$ (note that $H$ is a fictitious extra step as an episode terminates at $H - 1$), we

perform dynamic programming all the way to $h = 0$:

$$
\widehat{V}_H^k(s) = 0, \forall s,
$$
$$
\widehat{Q}_h^k(s,a) = \min \left\{ r_h(s,a) + b_h^k(s,a) + \widehat{P}_h^k(\cdot|s,a) \cdot \widehat{V}_{h+1}^k, \ H \right\},
$$
$$
\widehat{V}_h^k(s) = \max_a \widehat{Q}_h^k(s,a), \pi_h^k(s) = \mathrm{argmax}_a \, \widehat{Q}_h^k(s,a), \forall h, s, a. \tag{0.3}
$$

Note that when using $\widehat{V}_{h+1}^k$ to compute $\widehat{Q}_h^k$, we truncate the value by $H$. This is because we know that due to the assumption that $r(s,a) \in [0,1]$, no policy's Q value will ever be larger than $H$.

Denote $\pi^k = \{\pi_0^k, \ldots, \pi_{H-1}^k\}$. Learner then executes $\pi^k$ in the MDP to get a new trajectory $\tau^k$.

UCBVI repeats the above procedure for $K$ episodes.

## 7.3 Analysis

We will prove the following theorem.

**Theorem 7.1** (Regret Bound of UCBVI). *UCBVI achieves the following regret bound:*

$$
\textit{Regret} := \mathbb{E} \left[ \sum_{k=0}^{K-1} \left( V^\star(s_0) - V^{\pi^k}(s_0) \right) \right] \leq 10 H^2 S \sqrt{AK \cdot \ln(SAH^2K^2)} = \widetilde{O}\left( H^2 S \sqrt{AK} \right)
$$

**Remark** While the above regret is sub-optimal, the algorithm we presented here indeed achieves a sharper bound in the leading term $\widetilde{O}(H^2\sqrt{SAK})$ [Azar et al., 2017], which gives the tight dependency bound on $S, A, K$. In Section 7.4, we will present a refined analysis that indeed achieves $H^2\sqrt{SAK}$ regret. The dependency on $H$ is not tight and tightening the dependency on H requires modifications to the reward bonus (use Bernstein inequality rather than Hoeffding's inequality for reward bonus design).

We prove the above theorem in this section.

We start with bounding the error from the learned model $\widehat{P}_h^k$.

**Lemma 7.2** (State-action wise model error). *Fix $\delta \in (0,1)$. For all $k \in [0, \ldots, K-1], s \in \mathcal{S}, a \in \mathcal{A}, h \in [0, \ldots, H-1]$, with probability at least $1 - \delta$, we have that for any $f : \mathcal{S} \mapsto [0, H]$:*

$$
\left| \left( \widehat{P}_h^k(\cdot|s,a) - P_h^\star(\cdot|s,a) \right)^\top f \right| \leq 8H\sqrt{\frac{S\ln(SAHK/\delta)}{N_h^k(s,a)}}.
$$

**Remark** The proof of the above lemma requires a careful argument using Martingale difference sequence mainly because $N_h^k(s,a)$ itself is a random quantity. We give a proof the above lemma at the end of this section (Sec. 7.3.1). Using the above lemma will result a sub-optimal regret bound in terms of the dependence on $S$. In Section 7.4, we will give a different approach to bound $\left| \left( \widehat{P}_h^k(\cdot|s,a) - P_h^\star(\cdot|s,a) \right)^\top f \right|$ which leads to an improved regret bound.

The following lemma is still about model error, but this time we consider an average model error with respect to $V^\star$ — a deterministic quantity.

**Lemma 7.3** (State-action wise average model error under $V^\star$). *Fix $\delta \in (0,1)$. For all $k \in [1, \dots, K-1], s \in \mathcal{S}, a \in \mathcal{A}, h \in [0, \dots, H-1]$, and consider $V_h^\star : \mathcal{S} \to [0, H]$, with probability at least $1 - \delta$, we have:*

$$\left| \widehat{P}_h^k(\cdot|s,a) \cdot V_{h+1}^\star - P_h^\star(\cdot|s,a) \cdot V_{h+1}^\star \right| \leq 2H \sqrt{\frac{\ln(SAHN/\delta)}{N_h^k(s,a)}}.$$

Note that we can use Lemma 7.2 to bound $\left| \widehat{P}_h^k(\cdot|s,a) \cdot V_{h+1}^\star - P_h^\star(\cdot|s,a) \cdot V_{h+1}^\star \right|$. However, the above lemma shows that by leveraging the fact that $V^\star$ is a deterministic quantity (i.e., independent of the data collected during learning), we can get a tighter upper bound which does not scale polynomially with respect to $S$.

**Proof:** Consider a fixed tuple $s, a, k, h$ first. By the definition of $\widehat{P}_h^k$, we have:

$$\widehat{P}_h^k(\cdot|s,a) \cdot V_{h+1}^\star = \frac{1}{N_h^k(s,a)} \sum_{i=0}^{k-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\} V_{h+1}^\star(s_{h+1}^i).$$

Now denote $\mathcal{H}_{h,i}$ as the entire history from $t = 0$ to iteration $t = i$ where in iteration $i$, $\mathcal{H}_{h,i}$ includes history from time step 0 up to and including time step $h$. We define random variables: $X_i = \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\} V_{h+1}^\star(s_{h+1}^i) - \mathbb{E}\left[\mathbf{1}\{(s_h^i, a_h^i) = (s,a)\} V_{h+1}^\star(s_{h+1}') | \mathcal{H}_{h,i}\right]$ for $i = 0, \dots, K-1$. These random variables have the following properties. First $|X_i| \leq H$ if $\mathbf{1}\{(s_h^i, a_h^i) = (s,a)\} = 1$, else $|X_i| = 0$, which implies that $\sum_{i=0}^k |X_i|^2 \leq N_h^k(s,a)$. Also we have $\mathbb{E}[X_i | \mathcal{H}_{h,i}] = 0$ for all $i$. Thus, $\{X_i\}_{i \geq 0}$ is a Martingale difference sequence. Using Azuma-Hoeffding's inequality, we get that for any fixed $k$, with probability at least $1 - \delta$,

$$\left| \sum_{i=0}^{k-1} X_i \right| = \left| \sum_{i=0}^{k-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\} V_{h+1}^\star(s_{h+1}^i) - N_h^k(s,a) \mathbb{E}_{s' \sim P_h^\star(s,a)} V^\star(s') \right| \leq 2H \sqrt{N_h^k(s,a) \ln(1/\delta)}.$$

Divide $N_h^k(s,a)$ on both sides and use the definition of $\widehat{P}_h(\cdot|s,a)^\top V_{h+1}^\star$, we get:

$$\left| \widehat{P}_h(\cdot|s,a)^\top V_{h+1}^\star - P_h^\star(\cdot|s,a)^\top V_{h+1}^\star \right| \leq 2H \sqrt{\ln(1/\delta)/N_h^k(s,a)}.$$

With a union bound over all $s \in \mathcal{S}, a \in \mathcal{A}, k \in [N], h \in [H]$, we conclude the proof. ∎

We denote the two inequalities in Lemma 7.2 and Lemma 7.3 as event $\mathcal{E}_{model}$. Note that the failure probability of $\mathcal{E}_{model}$ is at most $2\delta$. Below we condition on $\mathcal{E}_{model}$ being true (we deal with failure event at the very end).

Now we study the effect of reward bonus. Similar to the idea in multi-armed bandits, we want to pick a policy $\pi^k$, such that the value of $\pi^k$ in under the combined reward $r_h + b_h^k$ and the empirical model $\widehat{P}_h^k$ is optimistic, i.e., we want $\widehat{V}_0^k(s_0) \geq V_0^\star(s_0)$ for all $s_0$. The following lemma shows that via reward bonus, we are able to achieve this optimism.

**Lemma 7.4** (Optimism). *Assume $\mathcal{E}_{model}$ is true. For all episode $k$, we have:*

$$\widehat{V}_0^k(s_0) \geq V_0^\star(s_0), \forall s_0 \in \mathcal{S};$$

*where $\widehat{V}_h^k$ is computed based on VI in Eq. 0.3.*

**Proof:** We prove via induction. At the additional time step $H$ we have $\widehat{V}_H^k(s) = V_H^\star(s) = 0$ for all $s$.

Starting at $h + 1$, and assuming we have $\widehat{V}_{h+1}^k(s) \geq V_{h+1}^\star(s)$ for all $s$, we move to $h$ below.

Consider any $s, a \in \mathcal{S} \times \mathcal{A}$. First, if $Q_h^k(s,a) = H$, then we have $Q_h^k(s,a) \geq Q_h^\star(s,a)$.

$$\widehat{Q}_h^k(s,a) - Q_h^\star(s,a) = b_h^k(s,a) + \widehat{P}_h^k(\cdot|s,a) \cdot \widehat{V}_{h+1}^k - P_h^\star(\cdot|s,a) \cdot V_{h+1}^\star$$

$$\geq b_h^k(s,a) + \widehat{P}_h^k(\cdot|s,a) \cdot V_{h+1}^\star - P_h^\star(\cdot|s,a) \cdot V_{h+1}^\star$$

$$= b_h^k(s,a) + \left( \widehat{P}_h^k(\cdot|s,a) - P_h^\star(\cdot|s,a) \right) \cdot V_{h+1}^\star$$

$$\geq b_h^k(s,a) - 2H\sqrt{\frac{\ln(SAHK/\delta)}{N_h^k(s,a)}} \geq 0.$$

where the first inequality is from the inductive hypothesis, and the last inequality uses Lemma 7.3 and the definition of bonus.

From $\widehat{Q}_h^k$, one can finish the proof by showing $\widehat{V}_h^k(s) \geq V_h^\star(s), \forall s$. ∎

Before we prove our final result, one more technical lemma will be helpful:

**Lemma 7.5.** *Consider arbitrary $K$ sequence of trajectories $\tau^k = \{s_h^k, a_h^k\}_{h=0}^{H-1}$ for $k = 0, \ldots, K-1$. We have*

$$\sum_{k=0}^{K-1} \sum_{h=0}^{H-1} \frac{1}{\sqrt{N_h^k(s_h^k, a_h^k)}} \leq 2H\sqrt{SAK}.$$

**Proof:** We swap the order of the two summation above:

$$\sum_{k=0}^{K-1} \sum_{h=0}^{H-1} \frac{1}{\sqrt{N_h^k(s_h^k, a_h^k)}} = \sum_{h=0}^{H-1} \sum_{k=0}^{K-1} \frac{1}{\sqrt{N_h^k(s_h^k, a_h^k)}} = \sum_{h=0}^{H-1} \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \sum_{i=1}^{N_h^K(s,a)} \frac{1}{\sqrt{i}}$$

$$\leq 2 \sum_{h=0}^{H-1} \sum_{s,a \in \mathcal{S} \times \mathcal{A}} \sqrt{N_h^K(s,a)} \leq \sum_{h=0}^{H-1} \sqrt{SA \sum_{s,a} N_h^K(s,a)} = H\sqrt{SAK},$$

where in the first inequality we use the fact that $\sum_{i=1}^{N} 1/\sqrt{i} \leq 2\sqrt{N}$, and in the second inequality we use CS inequality. ∎

The proof of our main theorem now follows:

**Proof:**[Proof of Theorem 7.1]

Let us consider episode $k$ and denote $\mathcal{H}_{<k}$ as the history up to the end of episode $k-1$. We consider bounding $V^\star - V^{\pi^k}$. Using optimism and the simulation lemma, we can get the following result:

$$V^\star(s_0) - V^{\pi^k}(s_0) \leq \widehat{V}_0^k(s_0) - V_0^{\pi^k}(s_0) \leq \sum_{h=0}^{H-1} \mathbb{E}_{s_h, a_h \sim d_h^{\pi^k}} \left[ b_h^k(s_h, a_h) + \left( \widehat{P}_h^k(\cdot|s_h, a_h) - P^\star(\cdot|s_h, a_h) \right) \cdot \widehat{V}_{h+1}^{\pi^k} \right]$$

$$(0.4)$$

We prove the above two inequalities in the lecture. We leave the proof of the above inequality (Eq 0.4) as an exercise for readers. Note that this is slightly different from the usual simulation lemma, as here we truncate $\widehat{V}$ by $H$ during VI.

Under $\mathcal{E}_{model}$, we can bound $\left( \widehat{P}_h^k(\cdot|s_h, a_h) - P^\star(\cdot|s_h, a_h) \right) \cdot \widehat{V}_{h+1}^{\pi^k}$ (recall Lemma 7.2) with a Holder's inequality:

$$\left| \left( \widehat{P}_h^k(\cdot|s_h, a_h) - P^\star(\cdot|s_h, a_h) \right) \cdot \widehat{V}_{h+1}^{\pi^k} \right| \leq \left\| \widehat{P}_h^k(\cdot|s_h, a_h) - P^\star(\cdot|s_h, a_h) \right\|_1 \left\| \widehat{V}_{h+1}^{\pi^k} \right\|_\infty$$

$$\leq 8H\sqrt{\frac{SL}{N_h^k(s,a)}},$$

where recall $L := \ln(SAKH/\delta)$. Hence, for the per-episode regret $V^\star(s_0) - V^{\pi^k}(s_0)$, we obtain:

$$V^\star(s_0) - V^{\pi^k}(s_0) \leq \sum_{h=0}^{H-1} \mathbb{E}_{s_h, a_h \sim d_h^{\pi^k}} \left[ b_h^k(s_h, a_h) + 8H\sqrt{SL/N_h^k(s_h, a_h)} \right]$$

$$\leq \sum_{h=0}^{H-1} \mathbb{E}_{s_h, a_h \sim d_h^{\pi^k}} \left[ 10H\sqrt{SL/N_h^k(s_h, a_h)} \right]$$

$$= 10H\sqrt{SL} \cdot \mathbb{E} \left[ \sum_{h=0}^{H-1} \frac{1}{\sqrt{N_h^k(s_h^k, a_h^k)}} \Big| \mathcal{H}_{<k} \right],$$

where in the last term the expectation is taken with respect to the trajectory $\{s_h^k, a_h^k\}$ (which is generated from $\pi^k$) while conditioning on all history $\mathcal{H}_{<k}$ up to and including the end of episode $k-1$.

Now we sum all episodes together and take the failure event into consideration:

$$\mathbb{E}\left[ \sum_{k=0}^{K-1} V^\star(s_0) - V^{\pi^k}(s_0) \right] = \mathbb{E}\left[ \mathbf{1}\{\mathcal{E}_{model}\} \left( \sum_{k=0}^{K-1} V^\star(s_0) - V^{\pi^k}(s_0) \right) \right] + \mathbb{E}\left[ \mathbf{1}\{\overline{\mathcal{E}}_{model}\} \left( \sum_{k=0}^{K-1} V^\star(s_0) - V^{\pi^k}(s_0) \right) \right]$$

$$\leq \mathbb{E}\left[ \mathbf{1}\{\mathcal{E}_{model}\} \left( \sum_{k=0}^{K-1} V^\star(s_0) - V^{\pi^k}(s_0) \right) \right] + 2\delta KH \qquad (0.5)$$

$$\leq 10H\sqrt{S\ln(SAHK/\delta)} \mathbb{E}\left[ \sum_{k=0}^{K-1} \sum_{h=0}^{H-1} \frac{1}{\sqrt{N_h^k(s_h^k, a_h^k)}} \right] + 2\delta KH,$$

where the last inequality uses the law of total expectation.

We can bound the double summation term above using lemma 7.5. We can conclude that:

$$\mathbb{E}\left[ \sum_{k=0}^{K-1} V^\star(s_0) - V^{\pi^k}(s_0) \right] \leq 10H^2 S\sqrt{AK\ln(SAHK/\delta)} + 2\delta KH.$$

Now set $\delta = 1/KH$, we get:

$$\mathbb{E}\left[ \sum_{k=0}^{K-1} V^\star(s_0) - V^{\pi^k}(s_0) \right] \leq 10H^2 S\sqrt{AK\ln(SAH^2K^2)} + 2 = O\left( H^2 S\sqrt{AK\ln(SAH^2K^2)} \right).$$

This concludes the proof of Theorem 7.1. ∎

### 7.3.1 Proof of Lemma 7.2

Here we include a proof of Lemma 7.2. The proof consists of two steps, first we using Martingale concentration to bound $|(\widehat{P}_h^k(\cdot|s,a) - P_h^\star(\cdot|s,a))^\top f|$ for all $s, a, h, k$, but for a fixed $f : \mathcal{S} \mapsto [0, H]$. Note that there are infinitely many such $f$ that maps from $\mathcal{S}$ to $[0, H]$. Thus, the second step is to use a covering argument (i.e., $\epsilon$-net on all $\{f : \mathcal{S} \mapsto [0, H]\}$) to show that $|(\widehat{P}_h^k(\cdot|s,a) - P_h^\star(\cdot|s,a))^\top f|$ is bounded for all $s, a, h, k$, and all $f : \mathcal{S} \mapsto [0, H]$.

**Proof:**[Proof of Lemma 7.2] We consider a fixed tuple $s, a, k, h, f$ first. Recall the definition of $\widehat{P}_h^k(s,a)$, and we have: $\widehat{P}_h^k(\cdot|s,a)^\top f = \sum_{i=0}^{k-1} \mathbf{1}\{s_h^i, a_h^i = s, a\} f(s_{h+1}^i)/N_h^k(s,a)$.

Define $\mathcal{H}_{h,i}$ as the history starting from the beginning of iteration $0$ all the way up to and include time step $h$ at iteration $i$. Define random variables $X_i$ as $X_i = \mathbf{1}\{s_h^i, a_h^i = s, a\}f(s_{h+1}^i) - \mathbf{1}\{s_h^i, a_h^i = s, a\}\mathbb{E}_{s'\sim P_h^\star(s,a)}f(s')$. We now show that $\{X_i\}$ is a Martingale difference sequence. First we have $\mathbb{E}[X_i|\mathcal{H}_{h,i}] = 0$, since $\mathbf{1}\{s_h^i, a_h^i = s, a\}$ is a deterministic quantity given $\mathcal{H}_{h,i}$. Second, we have $|X_i| = 0$ for $(s_h^i, a_h^i) \neq (s,a)$, and $|X_i| \leq H$ when $(s_h^i, a_h^i) = (s,a)$. Thus, we have shown that it is a Martingale difference sequence.

Now we can apply Azuma-Hoeffding's inequality (Theorem A.5). With probability at least $1 - \delta$, we have:

$$\left|\sum_{i=0}^{k-1} X_i\right| = \left|\sum_{i=0}^{k-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}f(s_{h+1}^i) - N_h^k(s,a)\mathbb{E}_{s'\sim P_h^\star(s,a)}f(s')\right| \leq 2H\sqrt{N_h^k(s,a)\ln(1/\delta)}.$$

Now apply union bound over all $s \in \mathcal{S}, a \in \mathcal{A}, h \in [0,\ldots,H-1], k \in [0,\ldots,K-1]$, we have that with probability at least $1 - \delta$, for all $s, a, h, k$, we have:

$$\left|\sum_{i=0}^{k-1} \mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}f(s_{h+1}^i) - N_h^k(s,a)\mathbb{E}_{s'\sim P_h^\star(s,a)}f(s')\right| \leq 2H\sqrt{N_h^k(s,a)\ln(SAKH/\delta)}. \qquad (0.6)$$

Note that the above result only holds for the fixed $f$ that we specified at the beginning of the proof. We are left to perform a covering argument over all functions that map from $\mathcal{S}$ to $[0,H]$. We use a standard $\epsilon$-net argument for that. We abuse notation a bit and denote $f$ as a vector in $[0,H]^S$.

Note that $\|f\|_2 \leq H\sqrt{S}$ for any $f : \mathcal{S} \mapsto [0,H]$. A standard $\epsilon$-net argument shows that we can construct an $\epsilon$-net $\mathcal{N}_\epsilon$ over $[0,H]^S$ with $|\mathcal{N}_\epsilon| \leq (1 + 2H\sqrt{S}/\epsilon)^S$, such that for any $f \in [0,H]^S$, there exists a point $f' \in \mathcal{N}_\epsilon$, such that $\|f - f'\|_2 \leq \epsilon$. Note that the $\epsilon$-net $\mathcal{N}_\epsilon$ is independent of the training data during the learning process.

Now using Eq. 0.6 and a union bound over all $f$ in $\mathcal{N}_\epsilon$, we have that with probability at least $1-\delta$, for all $s, a, k, h, f \in \mathcal{N}_\epsilon$, we have:

$$\left|\frac{\sum_{i=0}^{k-1}\mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}f(s_{h+1}^i)}{N_h^k(s,a)} - \mathbb{E}_{s'\sim P_h^\star(s,a)}f(s')\right| \leq 2H\sqrt{\frac{S\ln(SAKH(1+2H\sqrt{S}/\epsilon)/\delta)}{N_h^k(s,a)}}.$$

Finally, for any $f \in [0,H]^S$, denote its closest point in $\mathcal{N}_\epsilon$ as $f'$, we have:

$$\left|\frac{\sum_{i=0}^{k-1}\mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}f(s_{h+1}^i)}{N_h^k(s,a)} - \mathbb{E}_{s'\sim P_h^\star(s,a)}f(s')\right|$$

$$\leq \left|\frac{\sum_{i=0}^{k-1}\mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}f'(s_{h+1}^i)}{N_h^k(s,a)} - \mathbb{E}_{s'\sim P_h^\star(s,a)}f'(s')\right| + \left|\frac{\sum_{i=0}^{k-1}\mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}(f(s_{h+1}^i) - f'(s_{h+1}^i))}{N_h^k(s,a)}\right|$$

$$+ \left|\mathbb{E}_{s'\sim P_h^\star(s,a)}(f(s') - f'(s'))\right|$$

$$\leq 2H\sqrt{\frac{S\ln\left(SAKH(1+2H\sqrt{S}/\epsilon)/\delta\right)}{N_h^k(s,a)}} + 2\epsilon,$$

where in the last inequality we use the fact that if $\|f - f'\|_2 \leq \epsilon$, then $|f(s) - f'(s)| \leq \epsilon, \forall s \in \mathcal{S}$.

Now we can set $\epsilon = 1/K$ and use the fact that $N_h^k(s,a) \leq K$, we have:

$$\left|\frac{\sum_{i=0}^{k-1}\mathbf{1}\{(s_h^i, a_h^i) = (s,a)\}f(s_{h+1}^i)}{N_h^k(s,a)} - \mathbb{E}_{s'\sim P_h^\star(s,a)}f(s')\right| \leq 2H\sqrt{\frac{S\ln\left(SAKH(1+2HK\sqrt{S})/\delta\right)}{N_h^k(s,a)}} + \frac{2}{K}$$

$$\leq 4H\sqrt{\frac{S\ln(4H^2S^2K^2A/\delta)}{N_h^k(s,a)}} \leq 8H\sqrt{\frac{S\ln(HSKA/\delta)}{N_h^k(s,a)}},$$

73

where the last inequality uses the simple inequality $\ln(4H^2S^2K^2A) \le 4\ln(HSKA)$. This concludes the proof. ∎

## 7.4   An Improved Regret Bound

In this section, we show that UCB-VI also enjoys the regret bound $\tilde{O}\left(H^2\sqrt{SAK} + H^3S^2A\right)$.

**Theorem 7.6** (Improved Regret Bound for UCB-VI). *The total expected regret is bounded as follows:*

$$
\mathbb{E}\left[KV^\star(s_0) - \sum_{k=0}^{K-1} V^{\pi^k}(s_0)\right]
$$

$$
\le 8eH^2\sqrt{SAK} \cdot \sqrt{\ln(SAH^2K^2)} + 2eH^3S^2A \cdot \ln(K)\ln(SAH^2K^2) = \tilde{O}\left(H^2\sqrt{SAK} + H^3S^2A\right).
$$

Note that comparing to the regret bound we had in the previous section, this new regret improves the $S$ dependence in the leading order term (i.e., the term that contains $\sqrt{K}$), but at the cost of introducing a lower order term $H^3S^2A$.

We first give an improved version of the model error presented in Lemma 7.2.

**Lemma 7.7.** *With probability at least $1 - \delta$, for all $h, k, s, a$, and all $f : \mathcal{S} \mapsto [0, H]$, we have:*

$$
\left|\left(\widehat{P}_h^k(\cdot|s, a) - P_h^\star(\cdot|s, a)\right)^\top f\right| \le \frac{\mathbb{E}_{s' \sim P_h^\star(\cdot|s, a)} f(s')}{H} + \frac{3H^2SL}{N_h^k(s, a)}.
$$

Below we prove this lemma. Recall the definition of $\widehat{P}_h^k(s'|s, a)$ in Eq. 0.1. We can use Azume-Bernstein's inequality to bound $|\widehat{P}_h^k(s'|s, a) - P_h^\star(s'|s, a)|$.

**Lemma 7.8.** *With probability at least $1 - \delta$, for all $s, a, k, h$, we have:*

$$
\left|\widehat{P}_h^k(s'|s, a) - P_h^\star(s'|s, a)\right| \le \sqrt{\frac{2P_h^\star(s'|s, a)L}{N_h^k(s, a)}} + \frac{2L}{N_h^k(s, a)},
$$

*where $L := \ln(SAKH/\delta)$.*

**Proof:** Consider a fixed tuple $(s, a, h, k)$. Define $\mathcal{H}_{h,i}$ as the history starting from the beginning of iteration 0 to time step $h$ at iteration $i$ (including step $h$, i.e., up to $s_h^i, a_h^i$). Define random variables $\{X_i\}_{i \ge 0}$ as $X_i = \mathbf{1}\{s_h^i, a_h^i, s_{h+1}^i = s, a, s'\} - \mathbf{1}\{s_h^i, a_h^i = s, a\}P_h^\star(s'|s, a)$. We now show that $\{X_i\}$ is a Martingale difference sequence. Note that $\mathbb{E}[X_i|\mathcal{H}_{h,i}] = 0$. We have $|X_i| \le 1$. To use Azuma-Bernstein's inequality, we note that $\mathbb{E}[X_i^2|\mathcal{H}_{h,i}]$ is bounded as:

$$
\mathbb{E}[X_i^2|\mathcal{H}_{h,i}] = \mathbf{1}\{s_h^i, a_h^i = s, a\}P_h^\star(s'|s, a)(1 - P_h^\star(s'|s, a)),
$$

where we use the fact that the variance of a Bernoulli with parameter $p$ is $p(1-p)$. This means that $\sum_{i=0}^{k-1} \mathbb{E}[X_i^2|\mathcal{H}_{h,i}] = P_h^\star(s'|s, a)(1 - P_h^\star(s'|s, a))N_h^k(s, a)$. Now apply Bernstein's inequality on the martingale difference sequence $\{X_i\}$, we have:

$$
\left|\sum_{i=0}^{k-1} X_i\right| \le \sqrt{\frac{2P_h^\star(s'|s, a)(1 - P_h^\star(s'|s, a))\ln(1/\delta)}{N_h^k(s, a)}} + \frac{2\ln(1/\delta)}{N_h^k(s, a)}
$$

$$
\le \sqrt{\frac{2P_h^\star(s'|s, a)\ln(1/\delta)}{N_h^k(s, a)}} + \frac{2\ln(1/\delta)}{N_h^k(s, a)}
$$

74

Recall the definition of $L := \ln(SAHK/\delta)$. Apply a union bound over all $s \in \mathcal{S}, a \in \mathcal{A}, h \in [0, \ldots, H-1], k \in [0, \ldots, K-1]$, we conclude the proof. ∎

Now we are ready to prove Lemma 7.7.

**Proof:**[Proof of Lemma 7.7] Let us condition on the event in Lemma 7.8 being true, of which the probability is at least $1 - \delta$.

Take any function $f : \mathcal{S} \mapsto [0, H]$, we have:

$$
\left| \left( \widehat{P}_h^k(\cdot|s,a) - P_h^\star(\cdot|s,a) \right)^\top f \right|
$$

$$
\leq \sum_{s' \in \mathcal{S}} \left| \widehat{P}_h^k(s'|s,a) - P_h^\star(\cdot|s,a) \right| f(s') \leq \sum_{s' \in \mathcal{S}} \sqrt{\frac{2L P_h^\star(s'|s,a) f^2(s')}{N_h^k(s,a)}} + \sum_{s' \in \mathcal{S}} \frac{2L f(s')}{N_h^k(s,a)}
$$

$$
\leq \sum_{s' \in \mathcal{S}} \sqrt{\frac{2L P_h^\star(s'|s,a) f^2(s')}{N_h^k(s,a)}} + \frac{2SHL}{N_h^k(s,a)} \leq \sqrt{S} \sqrt{\frac{\sum_{s' \in \mathcal{S}} 2L P_h^\star(s'|s,a) f^2(s')}{N_h^k(s,a)}} + \frac{2SHL}{N_h^k(s,a)}
$$

$$
= \sqrt{\frac{2H^2 SL}{N_h^k(s,a)} \cdot \frac{\sum_{s' \in \mathcal{S}} P_h^\star(s'|s,a) f^2(s')}{H^2}} + \frac{2SHL}{N_h^k(s,a)} \leq \frac{H^2 SL}{N_h^k(s,a)} + \frac{\mathbb{E}_{s' \sim P_h^\star(\cdot|s,a)} f^2(s')}{H^2} + \frac{2SHL}{N_h^k(s,a)}
$$

$$
\leq \frac{\mathbb{E}_{s' \sim P_h^\star(\cdot|s,a)} f(s')}{H} + \frac{H^2 SL}{N_h^k(s,a)} + \frac{2SHL}{N_h^k(s,a)},
$$

where the third inequality uses that condition that $\|f\|_\infty \leq H$, the fourth inequality uses Cauchy-Schwarz inequality, the fifth inequality uses the inequality that $ab \leq (a^2 + b^2)/2$ for any real numbers $a$ and $b$, and the last inequality uses the condition that $\|f\|_\infty \leq H$ again. This concludes the proof. ∎

So far in Lemma 7.7, we have not specify what function $f$ we will use. The following lemma instantiate $f$ to be $\widehat{V}_{h+1}^k - V_{h+1}^\star$. Recall the bonus definition, $b_h^k(s,a) := 2H\sqrt{L/N_h^k(s,a)}$ where $L := \ln(SAHK/\delta)$.

**Lemma 7.9.** *Assume the events in Lemma 7.7 and Lemma 7.3 are true. For all $s, a, k, h$, set $f := \widehat{V}_{h+1}^k - V_{h+1}^\star$, we have:*

$$
\left| \left( \widehat{P}_h^h(\cdot|s,a) - P_h^\star(\cdot|s,a) \right)^\top f \right| \leq \xi_h^k(s,a) + \frac{e}{H} \mathbb{E} \left[ \sum_{\tau=h+1}^{H-1} \left( \xi_\tau^k(s_\tau, a_\tau) + b_\tau^k(s_\tau, a_\tau) \right) \Big| s_h = s, a_h = a, \pi^k \right]
$$

*where $\xi_\tau^i(s,a) = \frac{3H^2 SL}{N_\tau^i(s,a)}$, for any $\tau \in [0, \ldots, H-1], i \in [0, \ldots, K-1]$, and the expectation is with respect to the randomness from the event of following policy $\pi^k$ from $(s,a)$ starting from time step $h$.*

**Proof:** Using Lemma 7.7 and the fact that $f := \widehat{V}_{h+1}^k - V_{h+1}^\star$, we have:

$$
\left| \left( \widehat{P}_h^h(\cdot|s,a) - P_h^\star(\cdot|s,a) \right)^\top f \right| \leq \frac{\mathbb{E}_{s' \sim P_h^\star(\cdot|s,a)} (\widehat{V}_{h+1}^k(s') - V_{h+1}^\star(s'))}{H} + \xi_h^k(s,a). \tag{0.7}
$$

Now consider $\widehat{V}_{h+1}^k(s') - V_{h+1}^\star(s'))$, we have:

$$
\begin{aligned}
\widehat{V}_{h+1}^k(s') - V_{h+1}^\star(s') &\leq \widehat{V}_{h+1}^k(s') - \left(r(s',\pi^k(s')) + P_{h+1}^\star(\cdot|s',\pi^k(s'))^\top V_{h+2}^\star\right) \\
&\leq r(s',\pi^k(s')) + \widehat{P}_{h+1}^k(\cdot|s',\pi^k(s'))^\top \widehat{V}_{h+2}^k - \left(r(s',\pi^k(s')) + P_{h+1}^\star(\cdot|s',\pi^k(s'))^\top V_{h+2}^\star\right) \\
&= \widehat{P}_{h+1}^k(\cdot|s',\pi^k(s'))^\top \widehat{V}_{h+2}^k - P_{h+1}^\star(\cdot|s',\pi^k(s'))^\top \widehat{V}_{h+2}^k + P_{h+1}^\star(\cdot|s',\pi^k(s'))^\top \left(\widehat{V}_{h+2}^k - V_{h+2}^\star\right) \\
&= \left(\widehat{P}_{h+1}^k(\cdot|s',\pi^k(s')) - P_{h+1}^\star(\cdot|s',\pi^k(s'))\right)^\top \left(\widehat{V}_{h+2}^k - V_{h+2}^\star\right) \\
&\quad + \left(\widehat{P}_{h+1}^k(\cdot|s',\pi^k(s')) - P_{h+1}^\star(\cdot|s',\pi^k(s'))\right)^\top V_{h+2}^\star \\
&\quad + P_{h+1}^\star(\cdot|s',\pi^k(s'))^\top \left(\widehat{V}_{h+2}^k - V_{h+2}^\star\right)
\end{aligned}
$$

where the first inequality uses the fact that $V_{h+1}^\star(s') = \max_{a'}(r(s',a') + P_{h+1}^\star(\cdot|s',a')^\top V_{h+2}^\star)$, the second inequality uses the definition of $\widehat{V}_{h+1}^k$ and $\pi^k$. Note that the second term in the RHS of the above inequality can be upper bounded exactly by the bonus using Lemma 7.3, and the first term can be further upper bounded using the same operation that we had in Lemma 7.7, i.e.,

$$
\left(\widehat{P}_{h+1}^k(\cdot|s',\pi^k(s')) - P_{h+1}^\star(\cdot|s',\pi^k(s'))\right)^\top \left(\widehat{V}_{h+2}^k - V_{h+2}^\star\right) \leq \frac{P_{h+1}^\star(\cdot|s',\pi^k(s'))^\top \left(\widehat{V}_{h+2}^k - V_{h+2}^\star\right)}{H} + \xi_{h+1}^k(s',\pi^k(s')).
$$

Combine the above inequalities together, we arrive at:

$$
\widehat{V}_{h+1}^k(s') - V_{h+1}^\star(s') \leq b_{h+1}^k(s',\pi^k(s')) + \xi_{h+1}^k(s',\pi^k(s')) + \left(1 + \frac{1}{H}\right) \mathbb{E}_{s'' \sim P_{h+1}^\star(s',\pi^k(s'))} \left(\widehat{V}_{h+2}^k(s'') - V_{h+2}^\star(s'')\right).
$$

We can recursively apply the same operation on $\widehat{V}_{h+2}^k(s'') - V_{h+2}^\star(s'')$ till step $H$, we have:

$$
\begin{aligned}
\widehat{V}_{h+1}^k(s') - V_{h+1}^\star(s') &\leq \mathbb{E}\left[\sum_{\tau=h+1}^{H-1} \left(1 + \frac{1}{H}\right)^{\tau-h-1} \left(\xi_\tau^k(s_\tau,a_\tau) + b_\tau^k(s_\tau,a_\tau)\right) \Big| s_{h+1} = s', \pi^k\right] \\
&\leq e\mathbb{E}\left[\sum_{\tau=h+1}^{H-1} \left(\xi_\tau^k(s_\tau,a_\tau) + b_\tau^k(s_\tau,a_\tau)\right) \Big| s_{h+1} = s', \pi^k\right],
\end{aligned}
$$

where the last inequality uses the fact that $(1 + 1/H)^x \leq e$ for any $x \in \mathbb{N}^+$. Substitute the above inequality into Eq. 0.7, we get:

$$
\left|\left(\widehat{P}_h^h(\cdot|s,a) - P_h^\star(\cdot|s,a)\right)^\top f\right| \leq \xi_h^k(s,a) + \frac{e}{H}\mathbb{E}\left[\sum_{\tau=h+1}^{H-1} \left(\xi_\tau^k(s_\tau,a_\tau) + b_\tau^k(s_\tau,a_\tau)\right) \Big| s_h = s, a_h = a, \pi^k\right]
$$

$\blacksquare$

Now we are ready to derive a refined upper bound for per-episode regret.

**Lemma 7.10.** *Assume the events in Lemma 7.9 and Lemma 7.4 (optimism) hold. For all $k \in [0, K-1]$, we have:*

$$
V^\star(s_0) - V^{\pi^k}(s_0) \leq e \sum_{h=0}^{H-1} \mathbb{E}_{s_h,a_h \sim d_h^{\pi^k}} \left[2b_h^k(s_h,a_h) + \xi_h^k(s_h,a_h)\right].
$$

**Proof:** Via optimism in Lemma 7.4, we have:

$$V^\star(s_0) - V^{\pi^k}(s_0) \le \widehat{V}_0^k(s_0) - V_0^{\pi^k}(s_0) \le \sum_{h=0}^{H-1} \mathbb{E}_{s_h,a_h \sim d_h^{\pi^k}} \left[ b_h^k(s_h,a_h) + \left( \widehat{P}_h^k(\cdot|s_h,a_h) - P^\star(\cdot|s_h,a_h) \right) \cdot \widehat{V}_{h+1}^{\pi^k} \right]$$

$$= \sum_{h=0}^{H-1} \mathbb{E}_{s_h,a_h \sim d_h^{\pi^k}} \left[ b_h^k(s_h,a_h) + \left( \widehat{P}_h^k(\cdot|s_h,a_h) - P^\star(\cdot|s_h,a_h) \right)^\top \left( \widehat{V}_{h+1}^{\pi^k} - V_{h+1}^\star \right) + \left( \widehat{P}_h^k(\cdot|s_h,a_h) - P^\star(\cdot|s_h,a_h) \right)^\top V_{h+1}^\star \right]$$

$$\le \sum_{h=0}^{H-1} \mathbb{E}_{s_h,a_h \sim d_h^{\pi^k}} \left[ 2b_h^k(s_h,a_h) + \left( \widehat{P}_h^k(\cdot|s_h,a_h) - P^\star(\cdot|s_h,a_h) \right)^\top \left( \widehat{V}_{h+1}^{\pi^k} - V_{h+1}^\star \right) \right]$$

$$\le \sum_{h=0}^{H-1} \mathbb{E}_{s_h,a_h \sim d_h^{\pi^k}} \left[ 2b_h^k(s_h,a_h) + \xi_h^k(s_h,a_h) + \frac{e}{H} \mathbb{E} \left[ \sum_{\tau=h+1}^{H-1} \left( \xi_\tau^k(s_\tau,a_\tau) + b_\tau^k(s_\tau,a_\tau) \right) | \pi^k \right] \right]$$

$$\le e \sum_{h=0}^{H-1} \mathbb{E}_{s_h,a_h \sim d_h^{\pi^k}} \left[ 2b_h^k(s_h,a_h) + \xi_h^k(s_h,a_h) \right]$$

where the first inequality uses optimism, and the fourth inequality uses Lemma 7.9, and the last inequality rearranges by grouping same terms together. ∎

To conclude the final regret bound, let us denote the event $\mathcal{E}_{model}$ as the joint event in Lemma 7.3 and Lemma 7.8, which holds with probability at least $1 - 2\delta$.

**Proof:**[Proof of Theorem 7.6] Following the derivation we have in Eq. 0.5, we decompose the expected regret bound as follows:

$$\mathbb{E} \left[ \sum_{k=0}^{K-1} V^\star(s_0) - V^{\pi^k}(s_0) \right] \le \mathbb{E} \left[ \mathbf{1}\{\mathcal{E}_{model}\} \left( \sum_{k=0}^{K-1} V^\star(s_0) - V^{\pi^k}(s_0) \right) \right] + 2\delta K H$$

$$\le e\mathbb{E} \left[ \sum_{k=0}^{K-1} \sum_{h=0}^{H-1} \left[ 2b_h^k(s_h^k,a_h^k) + \xi_h^k(s_h^k,a_h^k) \right] \right] + 2\delta K H,$$

where the last inequality uses Lemma 7.10. Using Lemma 7.5, we have that:

$$\sum_{k=0}^{K-1} \sum_{h=0}^{H-1} 2b_h^k(s_h^k,a_h^k) \le 4H\sqrt{L} \sum_k \sum_h \frac{1}{\sqrt{N_h^k(s_h^k,a_h^k)}} \le 8H^2\sqrt{SAKL}.$$

For $\sum_k \sum_h \xi_h^k(s_h^k,a_h^k)$, we have:

$$\sum_{k=0}^{K-1} \sum_{h=0}^{H-1} \xi_h^k(s_h^k,a_h^k) = 2H^2 SL \sum_{k=0}^{K-1} \sum_{h=0}^{H-1} \frac{1}{N_h^k(s_h^i,a_h^k)} = 2H^2 SL \sum_{h=0}^{H-1} \sum_{s,a} \sum_{i=1}^{N_h^K(s,a)} \frac{1}{i}$$

$$\le 2H^2 SALHS \ln(K) = 2H^3 S^2 AL \ln(K),$$

where in the last inequality, we use the fact that $N_h^K(s,a) \le K$ for all $s,a,h$, and the inequality $\sum_{i=1}^t 1/i \le \ln(t)$. Putting things together, and setting $\delta = 1/(KH)$, we conclude the proof. ∎

## 7.5 Phased $Q$-learning

*to be added*

## 7.6 Bibliographic Remarks and Further Readings

The first provably correct PAC algorithm for reinforcement learning (which finds a near optimal policy) was due to Kearns and Singh [2002], which provided the $E^3$ algorithm; it achieves polynomial sample complexity in tabular MDPs. Brafman and Tennenholtz [2002] presents the Rmax algorithm which provides a refined PAC analysis over $E^3$. Both are model based approaches [Kakade, 2003] improves the sample complexity to be $O(S^2 A)$. Both $E^3$ and Rmax uses the concept of absorbing MDPs to achieve optimism and balance exploration and exploitation.

Jaksch et al. [2010] provides the first $O(\sqrt{(T)})$ regret bound, where $T$ is the number of timesteps in the MDP ($T$ is proportional to $K$ in our setting); this dependence on $T$ is optimal. Subsequently, Azar et al. [2017], Dann et al. [2017] provide algorithms that, asymptotically, achieve minimax regret bound in tabular MDPs. By this, we mean that for sufficiently large $T$ (for $T \geq \Omega(|\mathcal{S}|^2)$), the results in Azar et al. [2017], Dann et al. [2017] obtain optimal dependencies on $|\mathcal{S}|$ and $|\mathcal{A}|$. The requirement that $T \geq \Omega(|\mathcal{S}|^2)$ before these bounds hold is essentially the requirement that non-trivial model accuracy is required. It is an opent question to remove this dependence.

Lower bounds are provided in [Dann and Brunskill, 2015, Osband and Van Roy, 2016, Azar et al., 2017].

**Further exploration strategies.** Refs and discussion for $Q$-learning, reward free, and thompson sampling to be added...

# Chapter 8

# Linearly Parameterized MDPs

In this chapter, we consider learning and exploration in linearly parameterized MDPs—the linear MDP. Linear MDP generalizes tabular MDPs into MDPs with potentially infinitely many state and action pairs.

This chapter largely follows the model and analysis first provided in [Jin et al., 2020].

## 8.1 Setting

We consider episodic finite horizon MDP with horizon $H$, $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \{r_h\}_h, \{P_h\}_h, H, s_0\}$, where $s_0$ is a fixed initial state, $r_h : \mathcal{S} \times \mathcal{A} \mapsto [0,1]$ and $P_h : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ are time-dependent reward function and transition kernel. Note that for time-dependent finite horizon MDP, the optimal policy will be time-dependent as well. For simplicity, we overload notations a bit and denote $\pi = \{\pi_0, \dots, \pi_{H-1}\}$, where each $\pi_h : \mathcal{S} \mapsto \mathcal{A}$. We also denote $V^\pi := V_0^\pi(s_0)$, i.e., the expected total reward of $\pi$ starting at $h = 0$ and $s_0$.

We define the learning protocol below. Learning happens in an episodic setting. Every episode $k$, learner first proposes a policy $\pi^k$ based on all the history information up to the end of episode $k - 1$. The learner then executes $\pi^k$ in the underlying MDP to generate a single trajectory $\tau^k = \{s_h^k, a_h^k\}_{h=0}^{H-1}$ with $a_h = \pi_h^k(s_h^k)$ and $s_{h+1}^k \sim P_h(\cdot|s_h^k, a_h^k)$. The goal of the learner is to minimize the following cumulative regret over $N$ episodes:

$$\text{Regret} := \mathbb{E}\left[\sum_{k=0}^{K-1}\left(V^\star - V^{\pi^k}\right)\right],$$

where the expectation is with respect to the randomness of the MDP environment and potentially the randomness of the learner (i.e., the learner might make decisions in a randomized fashion).

### 8.1.1 Low-Rank MDPs and Linear MDPs

Note that here we do not assume $\mathcal{S}$ and $\mathcal{A}$ are finite anymore. Indeed in this note, both of them could be continuous. Without any further structural assumption, the lower bounds we saw in the Generalization Lecture forbid us to get a polynomially regret bound.

The structural assumption we make in this note is a linear structure in both reward and the transition.

**Definition 8.1** (Linear MDPs). Consider transition $\{P_h\}$ and $\{r_h\}_h$. A linear MDP has the following structures on $r_h$

and $P_h$:

$$r_h(s,a) = \theta_h^\star \cdot \phi(s,a), \quad P_h(\cdot|s,a) = \mu_h^\star \phi(s,a), \forall h$$

where $\phi$ is a known state-action feature map $\phi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^d$, and $\mu_h^\star \in \mathbb{R}^{|\mathcal{S}| \times d}$. Here $\phi, \theta_h^\star$ are **known** to the learner, while $\mu^\star$ is unknown. We further assume the following norm bound on the parameters: (1) $\sup_{s,a} \|\phi(s,a)\|_2 \leq 1$, (2) $\|v^\top \mu_h^\star\|_2 \leq \sqrt{d}$ for any $v$ such that $\|v\|_\infty \leq 1$, and all $h$, and (3) $\|\theta_h^\star\|_2 \leq W$ for all $h$. We assume $r_h(s,a) \in [0,1]$ for all $h$ and $s,a$.

The model essentially says that the transition matrix $P_h \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}||\mathcal{A}|}$ has rank at most $d$, and $P_h = \mu_h^\star \Phi$. where $\Phi \in \mathbb{R}^{d \times |\mathcal{S}||\mathcal{A}|}$ and each column of $\Phi$ corresponds to $\phi(s,a)$ for a pair $s,a \in \mathcal{S} \times \mathcal{A}$.

**Linear Algebra Notations** For real-valued matrix $A$, we denote $\|A\|_2 = \sup_{x:\|x\|_2=1} \|Ax\|_2$ which denotes the maximum singular value of $A$. We denote $\|A\|_F$ as the Frobenius norm $\|A\|_F^2 = \sum_{i,j} A_{i,j}^2$ where $A_{i,j}$ denotes the $i,j$'th entry of $A$. For any Positive Definite matrix $\Lambda$, we denote $x^\top \Lambda x = \|x\|_\Lambda^2$. We denote $\det(A)$ as the determinant of the matrix $A$. For a PD matrix $\Lambda$, we note that $\det(\Lambda) = \prod_{i=1}^d \sigma_i$ where $\sigma_i$ is the eigenvalues of $\Lambda$. For notation simplicity, during inequality derivation, we will use $\lesssim, \approx$ to suppress all absolute constants. We will use $\widetilde{O}$ to suppress all absolute constants and log terms.

## 8.2 Planning in Linear MDPs

We first study how to do value iteration in linear MDP if $\mu$ is given.

We start from $Q_{H-1}^\star(s,a) = \theta_{H-1}^\star \cdot \phi(s,a)$, and $\pi_{H-1}^\star(s) = \arg\max_a Q_{H-1}^\star(s,a) = \arg\max_a \theta_{H-1}^\star \cdot \phi(s,a)$, and $V_{H-1}^\star(s) = \arg\max_a Q_{H-1}^\star(s,a)$.

Now we do dynamic programming from $h+1$ to $h$:

$$Q_h^\star(s,a) = \theta_h^\star \cdot \phi(s,a) + \mathbb{E}_{s \sim P_h(\cdot|s,a)} V_{h+1}^\star(s') = \theta_h^\star \cdot \phi(s,a) + P_h(\cdot|s,a) \cdot V_{h+1}^\star = \theta_h^\star \cdot \phi(s,a) + (\mu_h^\star \phi(s,a))^\top V_{h+1}^\star \tag{0.1}$$

$$= \phi(s,a) \cdot \left(\theta_h^\star + (\mu_h^\star)^\top V_{h+1}^\star\right) = \phi(s,a) \cdot w_h, \tag{0.2}$$

where we denote $w_h := \theta_h^\star + (\mu_h^\star)^\top V_{h+1}^\star$. Namely we see that $Q_h^\star(s,a)$ is a linear function with respect to $\phi(s,a)$! We can continue by defining $\pi_h^\star(s) = \arg\max_a Q_h^\star(s,a)$ and $V_h^\star(s) = \max_a Q_h^\star(s,a)$.

At the end, we get a sequence of linear $Q^\star$, i.e., $Q_h^\star(s,a) = w_h \cdot \phi(s,a)$, and the optimal policy is also simple, $\pi_h^\star(s) = \arg\max_a w_h \cdot \phi(s,a)$, for all $h = 0, \ldots, H-1$.

One key property of linear MDP is that a Bellman Backup of any function $f : \mathcal{S} \mapsto \mathbb{R}$ is a linear function with respect to $\phi(s,a)$. We summarize the key property in the following claim.

**Claim 8.2.** *Consider any arbitrary function $f : \mathcal{S} \mapsto [0, H]$. At any time step $h \in [0, \ldots H-1]$, there must exist a $w \in \mathbb{R}^d$, such that, for all $s,a \in \mathcal{S} \times \mathcal{A}$:*

$$r_h(s,a) + P_h(\cdot|s,a) \cdot f = w^\top \phi(s,a).$$

The proof of the above claim is essentially the Eq. 0.1.

## 8.3 Learning Transition using Ridge Linear Regression

In this section, we consider the following simple question: given a dataset of state-action-next state tuples, how can we learn the transition $P_h$ for all $h$?

Note that $\mu^\star \in \mathbb{R}^{|\mathcal{S}| \times d}$. Hence explicitly writing down and storing the parameterization $\mu^\star$ takes time at least $|\mathcal{S}|$. We show that we can represent the model in a non-parametric way.

We consider a particular episode $n$. Similar to Tabular-UCBVI, we learn a model at the very beginning of the episode $n$ using all data from the previous episodes (episode 1 to the end of the episode $n - 1$). We denote such dataset as:

$$\mathcal{D}_h^n = \left\{ s_h^i, a_h^i, s_{h+1}^i \right\}_{i=0}^{n-1}.$$

We maintain the following statistics using $\mathcal{D}_h^n$:

$$\Lambda_h^n = \sum_{i=0}^{n-1} \phi(s_h^i, a_h^i) \phi(s_h^i, a_h^i)^\top + \lambda I,$$

where $\lambda \in \mathbb{R}^+$ (it will be set to 1 eventually, but we keep it here for generality).

To get intuition of $\Lambda^n$, think about the tabular setting where $\phi(s, a)$ is a one-hot vector (zeros everywhere except that the entry corresponding to $(s, a)$ is one). Then $\Lambda_h^n$ is a diagonal matrix and the diagonal entry contains $N^n(s, a)$—the number of times $(s, a)$ has been visited.

We consider the following multi-variate linear regression problem. Denote $\delta(s)$ as a one-hot vector that has zero everywhere except that the entry corresponding to $s$ is one. Denote $\epsilon_h^i = P(\cdot|s_h^i, a_h^i) - \delta(s_{h+1}^i)$. Conditioned on history $\mathcal{H}_h^i$ (history $\mathcal{H}_h^i$ denotes all information from the very beginning of the learning process up to and including $(s_h^i, a_h^i)$), we have:

$$\mathbb{E}\left[ \epsilon_h^i | \mathcal{H}_h^i \right] = 0,$$

simply because $s_{h+1}^i$ is sampled from $P_h(\cdot|s_h^i, a_h^i)$ conditioned on $(s_h^i, a_h^i)$. Also note that $\|\epsilon_h^i\|_1 \leq 2$ for all $h, i$.

Since $\mu_h^\star \phi(s_h^i, a_h^i) = P_h(\cdot|s_h^i, a_h^i)$, and $\delta(s_{h+1}^i)$ is an unbiased estimate of $P_h(\cdot|s_h^i, a_h^i)$ conditioned on $s_h^i, a_h^i$, it is reasonable to learn $\mu^\star$ via regression from $\phi(s_h^i, a_h^i)$ to $\delta(s_{h+1}^i)$. This leads us to the following ridge linear regression:

$$\widehat{\mu}_h^n = \mathrm{argmin}_{\mu \in \mathbb{R}^{|\mathcal{S}| \times d}} \sum_{i=0}^{n-1} \left\| \mu \phi(s_h^i, a_h^i) - \delta(s_{h+1}^i) \right\|_2^2 + \lambda \|\mu\|_F^2.$$

Ridge linear regression has the following closed-form solution:

$$\widehat{\mu}_h^n = \sum_{i=0}^{n-1} \delta(s_{h+1}^i) \phi(s_h^i, a_h^i)^\top (\Lambda_h^n)^{-1} \tag{0.3}$$

Note that $\widehat{\mu}_h^n \in \mathbb{R}^{|\mathcal{S}| \times d}$, so we never want to explicitly store it. Note that we will always use $\widehat{\mu}_h^n$ together with a specific $s, a$ pair and a value function $V$ (think about value iteration case), i.e., we care about $\widehat{P}_h^n(\cdot|s, a) \cdot V := (\widehat{\mu}_h^n \phi(s, a)) \cdot V$, which can be re-written as:

$$\widehat{P}_h^n(\cdot|s, a) \cdot V := (\widehat{\mu}_h^n \phi(s, a)) \cdot V = \phi(s, a)^\top \sum_{i=0}^{n-1} (\Lambda_h^n)^{-1} \phi(s_h^i, a_h^i) V(s_{h+1}^i),$$

where we use the fact that $\delta(s)^\top V = V(s)$. Thus the operator $\widehat{P}_h^n(\cdot|s,a) \cdot V$ simply requires storing all data and can be computed via simple linear algebra and the computation complexity is simply $\text{poly}(d,n)$—no poly dependency on $|\mathcal{S}|$.

Let us calculate the difference between $\widehat{\mu}_h^n$ and $\mu_h^\star$.

**Lemma 8.3** (Difference between $\widehat{\mu}_h$ and $\mu_h^\star$). *For all $n$ and $h$, we must have:*

$$\widehat{\mu}_h^n - \mu_h^\star = -\lambda\mu_h^\star\left(\Lambda_h^n\right)^{-1} + \sum_{i=1}^{n-1}\epsilon_h^i\phi(s_h^i,a_h^i)^\top\left(\Lambda_h^n\right)^{-1}.$$

**Proof:** We start from the closed-form solution of $\widehat{\mu}_h^n$:

$$\widehat{\mu}_h^n = \sum_{i=0}^{n-1}\delta(s_{h+1}^i)\phi(s_h^i,a_h^i)^\top(\Lambda_h^n)^{-1} = \sum_{i=0}^{n-1}(P(\cdot|s_h^i,a_h^i) + \epsilon_h^n)\phi(s_h^i,a_h^i)^\top(\Lambda_h^n)^{-1}$$

$$= \sum_{i=0}^{n-1}(\mu_h^\star\phi(s_h^i,a_h^i) + \epsilon_h^i)\phi(s_h^i,a_h^i)^\top(\Lambda_h^n)^{-1} = \sum_{i=0}^{n-1}\mu_h^*\phi(s_h^i,a_h^i)\phi(s_h^i,a_h^i)^\top(\Lambda_h^n)^{-1} + \sum_{i=0}^{n-1}\epsilon_h^i\phi(s_h^i,a_h^i)^\top(\Lambda_h^n)^{-1}$$

$$= \sum_{i=0}^{n-1}\mu_h^\star\phi(s_h^i,a_h^i)\phi(s_h^i,a_h^i)^\top(\Lambda_h^n)^{-1} + \sum_{i=0}^{n-1}\epsilon_h^i\phi(s_h^i,a_h^i)^\top(\Lambda_h^n)^{-1}$$

$$= \mu_h^\star(\Lambda_h^n - \lambda I)(\Lambda_h^n)^{-1} + \sum_{i=0}^{n-1}\epsilon_h^i\phi(s_h^i,a_h^i)^\top(\Lambda_h^n)^{-1} = \mu_h^* - \lambda\mu_h^\star(\Lambda_h^n)^{-1} + \sum_{i=0}^{n-1}\epsilon_h^i\phi(s_h^i,a_h^i)^\top(\Lambda_h^n)^{-1}.$$

Rearrange terms, we conclude the proof. ∎

**Lemma 8.4.** *Fix $V : \mathcal{S} \mapsto [0,H]$. For all $n$ and $s,a \in \mathcal{S} \times \mathcal{A}$, and $h$, with probability at least $1-\delta$, we have:*

$$\left\|\sum_{i=0}^{n-1}\phi(s_h^i,a_h^i)(V^\top\epsilon_h^i)\right\|_{(\Lambda_h^n)^{-1}} \leq 3H\sqrt{\ln\frac{H\det(\Lambda_h^n)^{1/2}\det(\lambda I)^{-1/2}}{\delta}}.$$

**Proof:** We first check the noise terms $\{V^\top\epsilon_h^i\}_{h,i}$. Since $V$ is independent of the data (it's a pre-fixed function), and by linear property of expectation, we have:

$$\mathbb{E}\left[V^\top\epsilon_h^i|\mathcal{H}_h^i\right] = 0, \quad |V^\top\epsilon_h^i| \leq \|V\|_\infty\|\epsilon_h^i\|_1 \leq 2H, \forall h,i.$$

Hence, this is a Martingale difference sequence. Using the Self-Normalized vector-valued Martingale Bound (Lemma A.9), we have that for all $n$, with probability at least $1-\delta$:

$$\left\|\sum_{i=0}^{n-1}\phi(s_h^i,a_h^i)(V^\top\epsilon_h^i)\right\|_{(\Lambda_h^n)^{-1}} \leq 3H\sqrt{\ln\frac{\det(\Lambda_h^n)^{1/2}\det(\lambda I)^{-1/2}}{\delta}}.$$

Apply union bound over all $h \in [H]$, we get that with probability at least $1-\delta$, for all $n,h$:

$$\left\|\sum_{i=0}^{n-1}\phi(s_h^i,a_h^i)(V^\top\epsilon_h^i)\right\|_{(\Lambda_h^n)^{-1}} \leq 3H\sqrt{\ln\frac{H\det(\Lambda_h^n)^{1/2}\det(\lambda I)^{-1/2}}{\delta}}. \tag{0.4}$$

∎

## 8.4 Uniform Convergence via Covering

Now we take a detour first and consider how to achieve a uniform convergence result over a function class $\mathcal{F}$ that contains infinitely many functions. Previously we know how to get uniform convergence if $\mathcal{F}$ is finite—we simply do a union bound. However, when $\mathcal{F}$ contains infinitely many functions, we cannot simply apply a union bound. We will use the covering argument here.

Consider the following ball with radius $R$: $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_2 \leq R \in \mathbb{R}^+\}$. Fix an $\epsilon$. An $\epsilon$-net $\mathcal{N}_\epsilon \subset \Theta$ is a set such that for any $\theta \in \Theta$, there exists a $\theta' \in \mathcal{N}_\epsilon$, such that $\|\theta - \theta'\|_2 \leq \epsilon$. We call the smallest $\epsilon$-net as $\epsilon$-cover. Abuse notations a bit, we simply denote $\mathcal{N}_\epsilon$ as the $\epsilon$-cover.

The $\epsilon$-covering number is the size of $\epsilon$-cover $\mathcal{N}_\epsilon$. We define the covering dimension as $\ln(|\mathcal{N}_\epsilon|)$

**Lemma 8.5.** *The $\epsilon$-covering number of the ball $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_2 \leq R \in \mathbb{R}^+\}$ is upper bounded by $(1 + 2R/\epsilon)^d$.*

We can extend the above definition to a function class. Specifically, we look at the following function. For a triple of $(w, \beta, \Lambda)$ where $w \in \mathbb{R}^d$ and $\|w\|_2 \leq L$, $\beta \in [0, B]$, and $\Lambda$ such that $\sigma_{\min}(\Lambda) \geq \lambda$, we define $f_{w,\beta,\Lambda} : \mathcal{S} \mapsto [0, R]$ as follows:

$$f_{w,\beta,\Lambda}(s) = \min\left\{\max_a\left(w^\top\phi(s,a) + \beta\sqrt{\phi(s,a)^\top\Lambda^{-1}\phi(s,a)}\right), H\right\}, \forall s \in \mathcal{S}. \tag{0.5}$$

We denote the function class $\mathcal{F}$ as:

$$\mathcal{F} = \{f_{w,\beta,\Lambda} : \|w\|_2 \leq L, \beta \in [0, B], \sigma_{\min}(\Lambda) \geq \lambda\}. \tag{0.6}$$

Note that $\mathcal{F}$ contains infinitely many functions as the parameters are continuous. However we will show that it has finite covering number that scales exponentially with respect to the number of parameters in $(w, \beta, \Lambda)$.

Why do we look at $\mathcal{F}$? As we will see later in this chapter $\mathcal{F}$ contains all possible $\widehat{Q}_h$ functions one could encounter during the learning process.

**Lemma 8.6** ($\epsilon$-covering dimension of $\mathcal{F}$)**.** *Consider $\mathcal{F}$ defined in Eq. 0.6. Denote its $\epsilon$-cover as $\mathcal{N}_\epsilon$ with the $\ell_\infty$ norm as the distance metric, i.e., $d(f_1, f_2) = \|f_1 - f_2\|_\infty$ for any $f_1, f_2 \in \mathcal{F}$. We have that:*

$$\ln(|\mathcal{N}_\epsilon|) \leq d\ln(1 + 6L/\epsilon) + \ln(1 + 6B/(\sqrt{\lambda}\epsilon)) + d^2\ln(1 + 18B^2\sqrt{d}/(\lambda\epsilon^2)).$$

Note that the $\epsilon$-covering dimension scales quadratically with respect to $d$.

**Proof:** We start from building a net over the parameter space $(w, \beta, \Lambda)$, and then we convert the net over parameter space to an $\epsilon$-net over $\mathcal{F}$ under the $\ell_\infty$ distance metric.

We pick two functions that corresponding to parameters $(w, \beta, \Lambda)$ and $(\hat{w}, \hat{\beta}, \widehat{\Lambda})$.

$$|f(s) - \hat{f}(s)| \leq \left| \max_a \left( w^\top \phi(s,a) + \beta\sqrt{\phi(s,a)^\top \Lambda^{-1}\phi(s,a)} \right) - \max_a \left( \hat{w}^\top \phi(s,a) + \hat{\beta}\sqrt{\phi(s,a)^\top \hat{\Lambda}^{-1}\phi(s,a)} \right) \right|$$

$$\leq \max_a \left| \left( w^\top \phi(s,a) + \beta\sqrt{\phi(s,a)^\top \Lambda^{-1}\phi(s,a)} \right) - \left( \hat{w}^\top \phi(s,a) + \hat{\beta}\sqrt{\phi(s,a)^\top \hat{\Lambda}^{-1}\phi(s,a)} \right) \right|$$

$$\leq \max_a \left| (w - \hat{w})^\top \phi(s,a) \right| + \max_a \left| (\beta - \hat{\beta})\sqrt{\phi(s,a)^\top \Lambda^{-1}\phi(s,a)} \right|$$

$$\qquad + \max_a \left| \hat{\beta}(\sqrt{\phi(s,a)^\top \Lambda^{-1}\phi(s,a)} - \sqrt{\phi(s,a)^\top \hat{\Lambda}^{-1}\phi(s,a)}) \right|$$

$$\leq \|w - \hat{w}\|_2 + |\beta - \hat{\beta}|/\sqrt{\lambda} + B\sqrt{\left| \phi(s,a)^\top (\Lambda^{-1} - \hat{\Lambda}^{-1})\phi(s,a) \right|}$$

$$\leq \|w - \hat{w}\|_2 + |\beta - \hat{\beta}|/\sqrt{\lambda} + B\sqrt{\|\Lambda^{-1} - \hat{\Lambda}^{-1}\|_F}$$

Note that $\Lambda^{-1}$ is a PD matrix with $\sigma_{\max}(\Lambda^{-1}) \leq 1/\lambda$.

Now we consider the $\epsilon/3$-Net $\mathcal{N}_{\epsilon/3,w}$ over $\{w : \|w\|_2 \leq L\}$, $\sqrt{\lambda}\epsilon/3$-net $\mathcal{N}_{\sqrt{\lambda}\epsilon/3,\beta}$ over interval $[0, B]$ for $\beta$, and $\epsilon^2/(9B^2)$-net $\mathcal{N}_{\epsilon^2/(9B),\Lambda}$ over $\{\Lambda : \|\Lambda\|_F \leq \sqrt{d}/\lambda\}$. The product of these three nets provide a $\epsilon$-cover for $\mathcal{F}$, which means that that size of the $\epsilon$-net $\mathcal{N}_\epsilon$ for $\mathcal{F}$ is upper bounded as:

$$\ln|\mathcal{N}_\epsilon| \leq \ln|\mathcal{N}_{\epsilon/3,w}| + \ln|\mathcal{N}_{\sqrt{\lambda}\epsilon/3,\beta}| + \ln|\mathcal{N}_{\epsilon^2/(9B^2),\Lambda}|$$

$$\leq d\ln(1 + 6L/\epsilon) + \ln(1 + 6B/(\sqrt{\lambda}\epsilon)) + d^2\ln(1 + 18B^2\sqrt{d}/(\lambda\epsilon^2)).$$

$\blacksquare$

**Remark** Covering gives a way to represent the complexity of function class (or hypothesis class). Relating to VC, covering number is upper bound roughly by $\exp(d)$ with $d$ being the VC-dimension. However, there are cases where VC-dimensional is infinite, but covering number if finite.

Now we can build a uniform convergence argument for all $f \in \mathcal{F}$.

**Lemma 8.7** (Uniform Convergence Results). *Set $\lambda = 1$. Fix $\delta \in (0, 1)$. For all $n, h$, all $s, a$, and all $f \in \mathcal{F}$, with probability at least $1 - \delta$, we have:*

$$\left| \left( \widehat{P}_h^n(\cdot|s,a) - P(\cdot|s,a) \right) \cdot f \right| \lesssim H \|\phi(s,a)\|_{(\Lambda_h^n)^{-1}} \left( \sqrt{d\ln(1 + 6L\sqrt{N})} + d\sqrt{\ln(1 + 18B^2\sqrt{d}N)} + \sqrt{\ln\frac{H}{\delta}} \right).$$

**Proof:** Recall Lemma 8.4, we have with probability at least $1 - \delta$, for all $n, h$, for a pre-fixed $V$ (independent of the random process):

$$\left\| \sum_{i=1}^{n-1} \phi(s_h^i, a_h^i)(V^\top \epsilon_h^i) \right\|_{(\Lambda_h^n)^{-1}}^2 \leq 9H^2 \ln \frac{H\det(\Lambda_h^n)^{1/2}\det(\lambda I)^{-1/2}}{\delta} \leq 9H^2 \left( \ln\frac{H}{\delta} + d\ln(1 + N) \right)$$

where we have used the fact that $\|\phi\|_2 \leq 1$, $\lambda = 1$, and $\|\Lambda_h^n\|_2 \leq N + 1$.

Denote the $\epsilon$-cover of $\mathcal{F}$ as $\mathcal{N}_\epsilon$. With an application of a union bound over all functions in $\mathcal{N}_\epsilon$, we have that with probability at least $1 - \delta$, for all $V \in \mathcal{N}_\epsilon$, all $n, h$, we have:

$$\left\| \sum_{i=1}^{n-1} \phi(s_h^i, a_h^i)(V^\top \epsilon_h^i) \right\|_{(\Lambda_h^n)^{-1}}^2 \leq 9H^2 \left( \ln\frac{H}{\delta} + \ln(|\mathcal{N}_\epsilon|) + d\ln(1 + N) \right).$$

Recall Lemma 8.6, substitute the expression of $\ln |\mathcal{N}_\epsilon|$ into the above inequality, we get:

$$\left\|\sum_{i=1}^{n-1}\phi(s_h^i,a_h^i)(V^\top\epsilon_h^i)\right\|_{(\Lambda_h^n)^{-1}}^2 \leq 9H^2\left(\ln\frac{H}{\delta}+d\ln(1+6L/\epsilon)+d^2\ln(1+18B^2\sqrt{d}/\epsilon^2)+d\ln(1+N)\right).$$

Now consider an arbitrary $f \in \mathcal{F}$. By the definition of $\epsilon$-cover, we know that for $f$, there exists a $V \in \mathcal{N}_\epsilon$, such that $\|f-V\|_\infty \leq \epsilon$. Thus, we have:

$$\left\|\sum_{i=1}^{n-1}\phi(s_h^i,a_h^i)(f^\top\epsilon_h^i)\right\|_{(\Lambda_h^n)^{-1}}^2 \leq 2\left\|\sum_{i=1}^{n-1}\phi(s_h^i,a_h^i)(V^\top\epsilon_h^i)\right\|_{(\Lambda_h^n)^{-1}}^2 + 2\left\|\sum_{i=1}^{n-1}\phi(s_h^i,a_h^i)((V-f)^\top\epsilon_h^i)\right\|_{(\Lambda_h^n)^{-1}}^2$$

$$\leq 2\left\|\sum_{i=1}^{n-1}\phi(s_h^i,a_h^i)(V^\top\epsilon_h^i)\right\|_{(\Lambda_h^n)^{-1}}^2 + 8\epsilon^2 N$$

$$\leq 9H^2\left(\ln\frac{H}{\delta}+d\ln(1+6L/\epsilon)+d^2\ln(1+18B^2\sqrt{d}/\epsilon^2)+d\ln(1+N)\right)+8\epsilon^2 N,$$

where in the second inequality we use the fact that $\|\sum_{i=1}^{n-1}\phi(s_h^i,a_h^i)(V-f)^\top\epsilon_h^i\|_{(\Lambda_h^n)^{-1}}^2 \leq 4\epsilon^2 N$, which is from

$$\left\|\sum_{i=1}^{n-1}\phi(s_h^i,a_h^i)(V-f)^\top\epsilon_h^i\right\|_{(\Lambda_h^n)^{-1}}^2 \leq \|\sum_{i=1}^{n-1}\phi(s_h^i,a_h^i)2\epsilon\|_{(\Lambda_h^n)^{-1}}^2 \leq \frac{4\epsilon^2}{\lambda}\|\sum_{i=1}^{n-1}\phi(s_h^i,a_h^i)\|_2 \leq 4\epsilon^2 N.$$

Set $\epsilon = 1/\sqrt{N}$, we get:

$$\left\|\sum_{i=1}^{n-1}\phi(s_h^i,a_h^i)(f^\top\epsilon_h^i)\right\|_{(\Lambda_h^n)^{-1}}^2 \leq 9H^2\left(\ln\frac{H}{\delta}+d\ln(1+6L\sqrt{N})+d^2\ln(1+18B^2\sqrt{d}N)+d\ln(1+N)\right)+8$$

$$\lesssim H^2\left(\ln\frac{H}{\delta}+d\ln(1+6L\sqrt{N})+d^2\ln(1+18B^2\sqrt{d}N)\right),$$

where we recall $\lesssim$ ignores absolute constants.

Now recall that we can express $\left(\widehat{P}_h^n(\cdot|s,a)-P(\cdot|s,a)\right)\cdot f = \phi(s,a)^\top(\widehat{\mu}_h^n-\mu_h^\star)^\top f$. Recall Lemma 8.3, we have:

$$|(\widehat{\mu}_h^n\phi(s,a)-\mu_h^\star\phi(s,a))\cdot f| \leq \left|\lambda\phi(s,a)^\top(\Lambda_h^n)^{-1}(\mu_h^\star)^\top f\right| + \left|\sum_{i=1}^{n-1}\phi(s,a)^\top(\Lambda_h^n)^{-1}\phi(s_h^i,a_h^i)(\epsilon_h^i)^\top f\right|$$

$$\lesssim H\sqrt{d}\|\phi(s,a)\|_{(\Lambda_h^n)^{-1}} + \|\phi(s,a)\|_{(\Lambda_h^n)^{-1}}\sqrt{\left(H^2\left(\ln\frac{H}{\delta}+d\ln(1+6L\sqrt{N})+d^2\ln(1+18B^2\sqrt{d}N)\right)\right)}$$

$$\approx H\|\phi(s,a)\|_{(\Lambda_h^n)^{-1}}\left(\sqrt{\ln\frac{H}{\delta}}+\sqrt{d\ln(1+6L\sqrt{N})}+d\sqrt{\ln(1+18B^2\sqrt{d}N)}\right).$$

$\blacksquare$

## 8.5 Algorithm

Our algorithm, Upper Confidence Bound Value Iteration (UCB-VI) will use reward bonus to ensure optimism. Specifically, we will the following reward bonus, which is motivated from the reward bonus used in linear bandit:

$$b_h^n(s, a) = \beta \sqrt{\phi(s, a)^\top (\Lambda_h^n)^{-1} \phi(s, a)}, \tag{0.7}$$

where $\beta$ contains poly of $H$ and $d$, and other constants and log terms. Again to gain intuition, please think about what this bonus would look like when we specialize linear MDP to tabular MDP.

---

**Algorithm 5** UCBVI for Linear MDPs

1: **Input: parameters** $\beta, \lambda$
2: **for** $n = 1 \dots N$ **do**
3:    Compute $\widehat{P}_h^n$ for all $h$ (Eq. 0.3)
4:    Compute reward bonus $b_h^n$ for all h (Eq. 0.7)
5:    Run Value-Iteration on $\{\widehat{P}_h^n, r_h + b_h^n\}_{h=0}^{H-1}$ (Eq. 0.8)
6:    Set $\pi^n$ as the returned policy of VI.
7: **end for**

---

With the above setup, now we describe the algorithm. Every episode $n$, we learn the model $\widehat{\mu}_h^n$ via ridge linear regression. We then form the quadratic reward bonus as shown in Eq. 0.7. With that, we can perform the following truncated Value Iteration (always truncate the Q function at $H$):

$$\widehat{V}_H^n(s) = 0, \forall s,$$

$$\widehat{Q}_h^n(s, a) = \theta^\star \cdot \phi(s, a) + \beta \sqrt{\phi(s, a)^\top (\Lambda_h^n)^{-1} \phi(s, a)} + \phi(s, a)^\top (\widehat{\mu}_h^n)^\top \widehat{V}_{h+1}^n$$

$$= \beta \sqrt{\phi(s, a)^\top (\Lambda_h^n)^{-1} \phi(s, a)} + (\theta^\star + (\widehat{\mu}_h^n)^\top \widehat{V}_{h+1}^n)^\top \phi(s, a),$$

$$\widehat{V}_h^n(s) = \min\{\max_a \widehat{Q}_h^n(s, a), \ H\}, \quad \pi_h^n(s) = \text{argmax}_a \widehat{Q}_h^n(s, a). \tag{0.8}$$

Note that above $\widehat{Q}_h^n$ contains two components: a quadratic component and a linear component. And $\widehat{V}_h^n$ has the format of $f_{w,\beta,\Lambda}$ defined in Eq. 0.5.

The following lemma bounds the norm of linear weights in $\widehat{Q}_h^n$.

**Lemma 8.8.** *Assume $\beta \in [0, B]$. For all $n, h$, we have $\widehat{V}_h^n$ is in the form of Eq. 0.5, and $\widehat{V}_h^n$ falls into the following class:*

$$\mathcal{V} = \{f_{w,\beta,\Lambda} : \|w\|_2 \leq W + \frac{HN}{\lambda}, \beta \in [0, B], \sigma_{\min}(\Lambda) \geq \lambda\}. \tag{0.9}$$

**Proof:** We just need to show that $\theta^\star + (\widehat{\mu}_h^n)^\top \widehat{V}_{h+1}^n$ has its $\ell_2$ norm bounded. This is easy to show as we always have $\|\widehat{V}_{h+1}^n\|_\infty \leq H$ as we do truncation at Value Iteration:

$$\left\| \theta^\star + (\widehat{\mu}_h^n)^\top \widehat{V}_{h+1}^n \right\|_2 \leq W + \left\| (\widehat{\mu}_h^n)^\top \widehat{V}_{h+1}^n \right\|_2.$$

Now we use the closed-form of $\widehat{\mu}_h^n$ from Eq. 0.3:

$$\left\| (\widehat{\mu}_h^n)^\top \widehat{V}_{h+1}^n \right\|_2 = \left\| \sum_{i=1}^{n-1} \widehat{V}_{h+1}^n(s_{h+1}^i) \phi(s_h^i, a_h^i)^\top (\Lambda_h^n)^{-1} \right\|_2 \leq H \left\| (\Lambda_h^n)^{-1} \sum_{i=0}^{n-1} \phi(s_h^i, a_h^i) \right\|_2 \leq \frac{Hn}{\lambda},$$

where we use the fact that $\|\widehat{V}_{h+1}^n\|_\infty \leq H$, $\sigma_{\max}(\Lambda^{-1}) \leq 1/\lambda$, and $\sup_{s,a} \|\phi(s, a)\|_2 \leq 1$. ∎

## 8.6 Analysis of UCBVI for Linear MDPs

In this section, we prove the following regret bound for UCBVI.

**Theorem 8.9** (Regret Bound). *Set $\beta = \widetilde{O}(Hd)$, $\lambda = 1$. UCBVI (Algorithm 5) achieves the following regret bound:*

$$\mathbb{E}\left[NV^\star - \sum_{i=0}^N V^{\pi^n}\right] \leq \widetilde{O}\left(H^2\sqrt{d^3N}\right)$$

The main steps of the proof are similar to the main steps of UCBVI in tabular MDPs. We first prove optimism via induction, and then we use optimism to upper bound per-episode regret. Finally we use simulation lemma to decompose the per-episode regret.

In this section, to make notation simple, we set $\lambda = 1$ directly.

### 8.6.1 Proving Optimism

Proving optimism requires us to first bound model error which we have built in the uniform convergence result shown in Lemma 8.7, namely, the bound we get for $(\widehat{P}_h^n(\cdot|s,a) - P(\cdot|s,a)) \cdot f$ for all $f \in \mathcal{V}$. Recall Lemma 8.7 but this time replacing $\mathcal{F}$ by $\mathcal{V}$ defined in Eq. 0.9. With probability at least $1 - \delta$, for all $n, h, s, a$ and for all $f \in \mathcal{V}$,

$$\left|(\widehat{P}_h^n(\cdot|s,a) - P(\cdot|s,a)) \cdot f\right| \leq H\|\phi(s,a)\|_{(\Lambda_h^n)^{-1}}\left(\sqrt{\ln\frac{H}{\delta}} + \sqrt{d\ln(1 + 6(W + HN)\sqrt{N})} + d\sqrt{\ln(1 + 18B^2\sqrt{d}N)}\right)$$

$$\lesssim Hd\|\phi(s,a)\|_{(\Lambda_h^n)^{-1}}\left(\sqrt{\ln\frac{H}{\delta}} + \sqrt{\ln(WN + HN^2)} + \sqrt{\ln(B^2dN)}\right)$$

$$\lesssim \|\phi(s,a)\|_{(\Lambda_h^n)^{-1}}\underbrace{Hd\left(\sqrt{\ln\frac{H}{\delta}} + \sqrt{\ln(W + H)} + \sqrt{\ln B} + \sqrt{\ln d} + \sqrt{\ln N}\right)}_{:=\beta}.$$

Denote the above inequality as event $\mathcal{E}_{model}$. Below we are going to condition on $\mathcal{E}_{model}$ being hold. Note that here for notation simplicity, we denote

$$\beta = Hd\left(\sqrt{\ln\frac{H}{\delta}} + \sqrt{\ln(W + H)} + \sqrt{\ln B} + \sqrt{\ln d} + \sqrt{\ln N}\right) = \widetilde{O}(Hd).$$

**remark**  Note that in the definition of $\mathcal{V}$ (Eq. 0.9), we have $\beta \in [0, B]$. And in the above formulation of $\beta$, note that $B$ appears inside a log term. So we need to set $B$ such that $\beta \leq B$ and we can get the correct $B$ by solving the inequality $\beta \leq B$ for $B$.

**Lemma 8.10** (Optimism). *Assume event $\mathcal{E}_{model}$ is true. for all $n$ and $h$,*

$$\widehat{V}_h^n(s) \geq V_h^\star(s), \forall s.$$

**Proof:** We consider a fixed episode $n$. We prove via induction. Assume that $\widehat{V}^n_{h+1}(s) \geq V^\star_{h+1}(s)$ for all $s$. For time step $h$, we have:

$$\hat{Q}^n_h(s,a) - Q^\star_h(s,a)$$
$$= \theta^\star \cdot \phi(s,a) + \beta\sqrt{\phi(s,a)^\top (\Lambda^n_h)^{-1}\phi(s,a)} + \phi(s,a)^\top (\widehat{\mu}^n_h)^\top \widehat{V}^n_{h+1} - \theta^\star \cdot \phi(s,a) - \phi(s,a)^\top (\mu^\star_h)^\top V^\star_{h+1}$$
$$\geq \beta\sqrt{\phi(s,a)^\top (\Lambda^n_h)^{-1}\phi(s,a)} + \phi(s,a)^\top (\widehat{\mu}^n_h - \mu^\star_h)^\top \widehat{V}^n_{h+1},$$

where in the last inequality we use the inductive hypothesis that $\widehat{V}^n_{h+1}(s) \geq V^\star_{h+1}(s)$, and $\mu^\star_h \phi(s,a)$ is a valid distribution (note that $\widehat{\mu}^n_h \phi(s,a)$ is not necessarily a valid distribution). We need to show that the bonus is big enough to offset the model error $\phi(s,a)^\top (\widehat{\mu}^n_h - \mu^\star_h)^\top \widehat{V}^n_{h+1}$. Since we have event $\mathcal{E}_{model}$ being true, we have that:

$$\left| (\widehat{P}^n_h(\cdot|s,a) - P(\cdot|s,a)) \cdot \widehat{V}^n_{h+1} \right| \lesssim \beta\|\phi(s,a)\|_{(\Lambda^n_h)^{-1}},$$

as by the construction of $\mathcal{V}$, we know that $\widehat{V}^n_{h+1} \in \mathcal{V}$.

This concludes the proof. ∎

## 8.6.2 Regret Decomposition

Now we can upper bound the per-episode regret as follows:

$$V^\star - V^{\pi_n} \leq \widehat{V}^n_0(s_0) - V^{\pi_n}_0(s_0).$$

We can further bound the RHS of the above inequality using simulation lemma. Recall Eq. 0.4 that we derived in the note for tabular MDP (Chapter 7:

$$\widehat{V}^n_0(s_0) - V^{\pi_n}_0(s_0) \leq \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d^{\pi_n}_h} \left[ b^n_h(s,a) + \left( \widehat{P}^n_h(\cdot|s,a) - P(\cdot|s,a) \right) \cdot \widehat{V}^n_{h+1} \right].$$

(recall that the simulation lemma holds for any MDPs—it's not specialized to tabular).

In the event $\mathcal{E}_{model}$, we already know that for any $s, a, h, n$, we have $\left( \widehat{P}^n_h(\cdot|s,a) - P(\cdot|s,a) \right) \cdot \widehat{V}^n_{h+1} \lesssim \beta\|\phi(s,a)\|_{(\Lambda^n_h)^{-1}} = b^n_h(s,a)$. Hence, under $\mathcal{E}_{model}$, we have:

$$\widehat{V}^n_0(s_0) - V^{\pi_n}_0(s_0) \leq \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d^{\pi_n}_h} [2b^n_h(s,a)] \lesssim \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d^{\pi_n}_h} [b^n_h(s,a)].$$

Sum over all episodes, we have the following statement.

**Lemma 8.11** (Regret Bound). *Assume the event $\mathcal{E}_{model}$ holds. We have:*

$$\sum_{n=0}^{N-1} (V^\star_0(s_0) - V^{\pi_n}_0(s_0)) \leq \sum_{n=0}^{N-1}\sum_{h=0}^{H-1} \mathbb{E}_{s^n_h,a^n_h\sim d^{\pi_n}_h} [b^n_h(s^n_h,a^n_h)]$$

### 8.6.3 Concluding the Final Regret Bound

We first consider the following elliptical potential argument, which is similar to what we have seen in the linear bandit lecture.

**Lemma 8.12** (Elliptical Potential). *Consider an arbitrary sequence of state action pairs $s_h^i, a_h^i$. Assume $\sup_{s,a} \|\phi(s,a)\|_2 \leq 1$. Denote $\Lambda_h^n = I + \sum_{i=0}^{n-1} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^\top$. We have:*

$$\sum_{i=0}^{N-1} \phi(s_h^i, a_h^i)(\Lambda_h^i)^{-1}\phi(s_h^i, a_h^i) \leq 2\ln\left(\frac{\det(\Lambda_h^{N+1})}{\det(I)}\right) \lesssim 2d\ln(N).$$

**Proof:** By the Lemma 3.7 and 3.8 in the linear bandit lecture note,

$$\sum_{i=1}^{N} \phi(s_h^i, a_h^i)(\Lambda_h^i)^{-1}\phi(s_h^i, a_h^i) \leq 2\sum_{i=1}^{N} \ln(1 + \phi(s_h^i, a_h^i)(\Lambda_h^i)^{-1}\phi(s_h^i, a_h^i))$$

$$\leq 2\ln\left(\frac{\det(\Lambda_h^{N+1})}{\det(I)}\right)$$

$$\leq 2d\ln(1 + \frac{N+1}{d\lambda}) \lesssim 2d\ln(N).$$

where the first inequality uses that for $0 \leq y \leq 1$, $\ln(1+y) \geq y/2$. ∎

Now we use Lemma 8.11 together with the above inequality to conclude the proof.

**Proof:**[Proof of main Theorem 8.9]

We split the expected regret based on the event $\mathcal{E}_{model}$.

$$\mathbb{E}\left[NV^\star - \sum_{n=1}^{N} V^{\pi_n}\right] = \mathbb{E}\left[\mathbf{1}\{\mathcal{E}_{model} \text{ holds}\}\left(NV^\star - \sum_{n=1}^{N} V^{\pi_n}\right)\right]$$

$$+ \mathbb{E}\left[\mathbf{1}\{\mathcal{E}_{model} \text{ doesn't hold}\}\left(NV^\star - \sum_{n=1}^{N} V^{\pi_n}\right)\right]$$

$$\leq \mathbb{E}\left[\mathbf{1}\{\mathcal{E}_{model} \text{ holds}\}\left(NV^\star - \sum_{n=1}^{N} V^{\pi_n}\right)\right] + \delta NH$$

$$\lesssim \mathbb{E}\left[\sum_{n=1}^{N}\sum_{h=0}^{H-1} b_h^n(s_h^n, a_h^n)\right] + \delta NH.$$

Note that:

$$\sum_{n=1}^{N}\sum_{h=0}^{H-1} b_h^n(s_h^n, a_h^n) = \beta \sum_{n=1}^{N}\sum_{h=0}^{H-1} \sqrt{\phi(s_h^n, a_h^n)^\top(\Lambda_h^n)^{-1}\phi(s_h^n, a_h^n)}$$

$$= \beta \sum_{h=0}^{H-1}\sum_{n=1}^{N} \sqrt{\phi(s_h^n, a_h^n)^\top(\Lambda_h^n)^{-1}\phi(s_h^n, a_h^n)}$$

$$\leq \beta \sum_{h=0}^{H-1} \sqrt{N\sum_{n=1}^{N} \phi(s_h^n, a_h^n)(\Lambda_h^n)^{-1}\phi(s_h^n, a_h^n)} \lesssim \beta H\sqrt{Nd\ln(N)}.$$

Recall that $\beta = \widetilde{O}(Hd)$. This concludes the proof. ∎

## 8.7 Bibliographic Remarks and Further Readings

There are number of ways to linearly parameterize an MDP such that it permits for efficient reinforcement learning (both statistically and computationally). The first observation that such assumptions lead to statistically efficient algorithms was due to [Jiang et al., 2017] due to that these models have low Bellman rank (as we shall see in Chapter 9). The first statistically and computationally efficient algorithm for a linearly parameterized MDP model was due to [Yang and Wang, 2019a,b]. Subsequently, [Jin et al., 2020] provided a computationally and statistically efficient algorithm for simplified version of this model, which is the model we consider here. The model of [Modi et al., 2020, Jia et al., 2020, Ayoub et al., 2020, Zhou et al., 2020] provides another linearly parameterized model, which can viewed as parameterizing $P(s'|s,a)$ as a linear combination of feature functions $\phi(s,a,s')$. One notable aspect of the model we choose to present here, where $P_h(\cdot|s,a) = \mu_h^\star \phi(s,a)$, is that this model has a number of free parameters that is $|\mathcal{S}| \cdot d$ (note that $\mu$ is unknown and is of size $|\mathcal{S}| \cdot d$), and yet the statistical complexity does not depend on $|\mathcal{S}|$. Notably, this implies that accurate model estimation request $O(|\mathcal{S}|)$ samples, while the regret for reinforcement learning is only polynomial in $d$. The linearly parameterized models of [Modi et al., 2020, Jia et al., 2020, Ayoub et al., 2020, Zhou et al., 2020] are parameterized by $O(d)$ parameters, and, while $O(d)$ free parameters suggests lower model capacity (where accurate model based estimation requires only polynomial in $d$ samples), these models are incomparable to the linearly parameterized models presented in this chapter;

It is worth observing that all of these models permit statistically efficient estimation due to that they have bounded Bellman rank [Jiang et al., 2017] (and bounded Witness rank [Sun et al., 2019a]), a point which we return to in the next Chapter.

The specific linear model we consider here was originally introduced by [Jin et al., 2020]. The non-parametric model-based algorithm we study here was first introduced by [Lykouris et al., 2019] (but under the context of adversarial attacks).

The analysis we present here does not easily extend to infinite dimensional feature $\phi$ (e.g., RBF kernel); here, [Agarwal et al., 2020a] provide an algorithm and an analysis that extends to infinite dimensional $\phi$, i.e. where we have a Reproducing Kernel Hilbert Space (RKHS) and the regret is based on the concept of Information Gain.

# Chapter 9

# Parametric Models with Bounded Bellman Rank

Our previous lectures on exploration in RL focused on the UCBVI algorithm designed for the tabular MDPs and Linear MDPs. While linear MDPs extends tabular MDPs to the function approximation regime, it is still limited in linear function approximation, and indeed the assumption that a Bellman backup of any function is still a linear function is a strong assumption. In this chapter, we consider the setting beyond tabular and linear representation. We aim to design algorithm with general function approximation that works for a large family of MDPs that subsumes not only tabular MDPs and linear MDPs, but also other models such as Linear function approximation with Bellman Completion (this generalizes linear MDPs), reactive predictive state representation (PSRs), and reactive Partially Observable Markov Decision Process (POMDPs).

## 9.1   Problem setting

We consider finite-horizon episodic time-dependent Markov Decision Process (MDP) $\mathcal{M} = (\{\mathcal{S}\}_h, \{\mathcal{A}\}_h, P_h, s_0, r, H)$ where $\mathcal{S}_h$ is the state space for time step $h$ and we assume $\mathcal{S}_0, \mathcal{S}_2, \ldots, \mathcal{S}_{H-1}$ are disjoint ($\mathcal{A}_0, \mathcal{A}_2, \ldots, \mathcal{A}_{H-1}$ are disjoint as well). We assume $\mathcal{S}_0 = \{s_0\}$ though we can generalize to an arbitrary $\mathcal{S}_0$ with an initial state distribution.

**Remark**   This setting indeed generalizes our previous finite horizon setting where we have a fixed $\mathcal{S}$ and $\mathcal{A}$, but time-dependent $P_h$ and $r_h$, as we just need to add time step $h$ into state space $\mathcal{S}$ to create $\mathcal{S}_h$, i.e., every state $s \in \mathcal{S}_h$ only contains $h$. The benefit of using the above slightly more general notation is that this allows us to ignore $h$ in $P, r$, and in $\pi, V$ and $Q$ as now state action contains time step $h$.

The goal of an agent is to maximize the cumulative expected reward it obtains over $H$ steps:

$$\max_\pi V^\pi(s_0).$$

In this chapter, we focus on a PAC (Probably Approximately Correct) guarantee. Namely, our goal is to find a policy $\hat{\pi}$ such that $V^\pi(s_0) \geq V^\star(s_0)$.

We make the following boundedness assumption on the rewards.

**Assumption 9.1.** Almost surely, for any trajectory $\tau$ and step $h$, $0 \leq r_h \leq 1$. Additionally, $0 \leq \sum_{h=0}^{H-1} r_h \leq 1$ almost surely for any trajectory $\tau$.

While the first part of the assumption is the standard boundedness assumption we have made throughout, the second assumes that the trajectory level rewards are also bounded by 1, instead of $H$, which is helpful for capturing sparse-reward goal-directed problems with rewards only at one point in a successful trajectory. While normalization of the trajectory level reward also keeps the net reward bounded, this makes the total reward only scale as $1/H$ if the rewards are sparse along the trajectory.

## 9.2   Value-function approximation

We consider a model-free, value function based approach here. More specifically, denote $\mathcal{S} = \cup_h \mathcal{S}_h$ and $\mathcal{A} := \cup_h \mathcal{A}_h$, we assume that we are given the following function class:

$$\mathcal{F} = \{f : \mathcal{S} \times \mathcal{A} \to [0, 1]\}.$$

Since we want to learn a near-optimal behavior, we seek to approximate the $Q$-value function of the optimal policy, namely $Q^\star$ using $f \in \mathcal{F}$. To this end, we start with a simplifying assumption that $Q^\star$ lies in $\mathcal{F}$. In practice, this can be weakened to having a good approximation for $Q^\star$ in $\mathcal{F}$, but we focus on exact containment for the cleanest setting. Formally, we make the following *realizability* assumption.

**Assumption 9.2** (Value-function realizability). The function class $\mathcal{F}$ satisfies $Q^\star \in \mathcal{F}$.

Armed with this assumption, we may ask whether we can find $Q^\star$ using a number of samples which does not scale as $|\mathcal{X}|$, trading it off for a statistical complexity measure for $\mathcal{F}$ such as $\ln |\mathcal{F}|$. The next result, adapted from Krishnamurthy et al. [2016] shows that this is not possible.

**Theorem 9.3.** *Fix $H, K \in \mathbb{N}$ with $K \geq 2$ and $\epsilon \in (0, \sqrt{1/8}]$. For any algorithm, there exists an MDP with a horizon of $H$ and $K$ actions, a class of predictors $\mathcal{F}$ with $|\mathcal{F}| = K^H$ and $Q^\star \in \mathcal{F}$ and a constant $c > 0$ such that the probability that the algorithm outputs a policy $\widehat{\pi}$ with $V(\widehat{\pi}) \geq V^\star - \epsilon$ after collecting $T$ trajectories from the MDP is at most $2/3$ for all $T \leq cK^H/\epsilon^2$.*

In words, the theorem says that for any algorithm, there exists an MDP where it cannot find a good policy in fewer than an exponential number of samples in the planning horizon, even when $Q^\star \in \mathcal{F}$ Furthermore, the size of the class $\mathcal{F}$ required for this result is $K^H$, so that a logarithmic dependence on $|\mathcal{F}|$ will not explain the lower bound. The lower bound construction basically uses the binary tree example that we have seen in the Generalization chapter.

The lower bound indicates that in order to learn in polynomial sample complexity, we need additional assumptions. While we have seen that polynomially sample complexity is possible in linear MDPs and tabular MDPs, our goal in this chapter is to significantly weaken the structural assumption on the MDPs.

## 9.3   Bellman Rank

Having concluded that we cannot find a near optimal policy using a reasonable number of samples with just the realizability assumption, it is clear that additional structural assumptions on the problem are required in order to make progress. We now give one example of such a structure, named Bellman rank, which was introduced by Jiang et al. [2017]. In order to motivate and define this quantity, we need some additional notation. For a function $f \in \mathcal{F}$, let us define $\pi_f(x) = \operatorname{argmax}_{a \in \mathcal{A}} f(x, a)$. Namely $\pi_f$ is the greedy action selector with respect to $f$. For a policy $\pi$, function $f \in \mathcal{F}$ and $h \in [H]$, let us also define the *average Bellman error* of the function approximator $f$:

$$\mathcal{E}(f; \pi, h) = \mathbb{E}_{s_h, a_h \sim d_h^\pi} \left[ f(s_h, a_h) - r_h - \mathbb{E}_{s_{h+1} \sim P(\cdot | s_h, a_h)} \max_{a \in \mathcal{A}_{h+1}} f(s_{h+1}, a) \right]. \tag{0.1}$$

This is called the average Bellman error as it is not the error on an individual state action $s, a$, but an expected error under the state-action distribution induced by the policy $\pi$. To see why the definition might be natural, we note that the following property of $Q^\star$ from the Bellman optimality equations.

**Fact 9.4.** $\mathcal{E}(Q^\star, \pi, h) = 0$, for all policies $\pi$ and levels $h$.

The fact holds due to $Q^\star$ satisfying the Bellman optimality condition. We also have seen that for any $f$ such that $f(s, a) = r(s, a) + \mathbb{E}_{s' \sim P(\cdot|s,a)} \max_{a'} f(s', a')$ for all $s, a \in \mathcal{S} \times \mathcal{A}$, then $f = Q^\star$.

Thus, if we ever discover a function $f$ such that $\mathcal{E}(f; \pi, h) \neq 0$ for an arbitrary policy $\pi$ and time step $h$, then we know that $f \neq Q^\star$. The average Bellman error allows us to detect wrong function approximators, i.e., $f$ such that $f \neq Q^\star$.

We now make a structural assumption on average Bellman errors, which allows us to reason about the Bellman errors induced by all policies $\pi_f$ in a sample-efficient manner. For any $h \in [H-1]$, let is define the *Bellman error matrix* $\mathcal{E}_h \in \mathbb{R}^{|\mathcal{F}| \times |\mathcal{F}|}$ as

$$[\mathcal{E}_h]_{g,f} = \mathcal{E}(f; \pi_g, h). \tag{0.2}$$

That is, each entry in the matrix captures the Bellman error of the function indexed by the column $f$ under the greedy policy $\pi_g$ induced by the row $g$ at step $h$. With this notation, we define the Bellman rank of the MDP and a function class $\mathcal{F}$ below.

**Definition 9.5** (Bellman Rank). The Bellman rank of an MDP and a function class $\mathcal{F}$ is the smallest integer $M$ such that $\operatorname{rank}(\mathcal{E}_h) \leq M$ for all $h \in [H-1]$.

Intuitively, if the Bellman rank is small, then for any level $h$, the number of linearly independent rows is small. That is, the average Bellman error for any function under most policies can be expressed as linear combination of the Bellman errors of that function on a small set of policies corresponding to the linearly independent rows. Note that the definition presented here is a modification of the original definition from Jiang et al. [2017] in order to precisely capture the linear MDP example we covered before. [Jiang et al., 2017] showed that Bellman rank can be further upper bounded in terms of latent quantities such as the rank of the transition matrix, or the number of latent states if the MDP has an equivalent formulation as an MDP with a small number of latent states. We refer the reader to Jiang et al. [2017] for detailed examples, as well as connections of Bellman rank with other rank type notions in the RL literature to measure problem complexity.

## 9.3.1 Examples

Before moving on the the algorithm, we first present three examples that admit small Bellman Rank: tabular MDPs, linear MDPs, and Bellman Completion with Linear Function class. We note that the setting of Bellman Completion with Linear Function class subsumes linear MDPs, and linear MDPs subsumes tabular MDPs.

**Tabular MDPs**

For tabular MDPs, we can directly rewrite the Bellman error as follows. Focusing on an arbitrary $h$:

$$\mathcal{E}(f; \pi_g, h) = \sum_{s_h, a_h \in \mathcal{S}_h \times \mathcal{A}_h} d_h^{\pi_g}(s_h, a_h) \left( f(s_h, a_h) - r(s_h, a_h) - \mathbb{E}_{s_{h+1} \sim P(\cdot|s_h, a_h)} \max_a f(s_{h+1}, a) \right)$$

$$= \left\langle d_h^{\pi_g}, f(\cdot, \cdot) - r(\cdot, \cdot) - \mathbb{E}_{s_{h+1} \sim P(\cdot|\cdot, \cdot)} \max_a f(s_{h+1}, a) \right\rangle.$$

Namely, $\mathcal{E}(f; \pi_g, h)$ can be written as an inner product of two vectors whose dimension is $|\mathcal{S}_h||\mathcal{A}_h|$. Note that in tabular MDPs, number of states and number of actions are the natural complexity quantity in sample complexity and regret bounds.

**Linear MDPs**

Recall the linear MDP definition. In linear MDPs, we know that $Q^\star$ is a linear function with respect to feature vector $\phi$, i.e., $Q^\star(s_h, a_h) = (w^\star)^\top \phi(s_h, a_h)$. We will parameterize $\mathcal{F}$ to be the following linear function class:

$$\mathcal{F} = \{w^\top \phi(s, a) : w \in \mathbb{R}^d, \|w\|_2 \leq W\},$$

where we assume $\|w^\star\|_2 \leq W$. Recall that the transition $P$ and reward are also linear, i.e., $P(\cdot|s_h, a_h) = \mu^\star \phi(s_h, a_h), r(s_h, a_h) = \phi(s_h, a_h)^\top \theta^\star$. In this case, for the average Bellman error, we have:

$$
\begin{aligned}
\mathcal{E}(f; \pi_g, h) &= \mathbb{E}_{s_h, a_h \sim d_h^{\pi_g}} \left[ w^\top \phi(s_h, a_h) - (\theta^\star)^\top \phi(s_h, a_h) - \mathbb{E}_{s_{h+1} \sim P(\cdot|s_h, a_h)} \max_{a \in \mathcal{A}_{h+1}} w^\top \phi(s_{h+1}, a) \right] \\
&= \mathbb{E}_{s_h, a_h \sim d_h^{\pi_g}} \left[ w^\top \phi(s_h, a_h) - (\theta^\star)^\top \phi(s_h, a_h) - \phi(s_h, a_h)^\top (\mu^\star)^\top \left( \max_{a \in \mathcal{A}_{h+1}} w^\top \phi(\cdot, a) \right) \right] \\
&= \left\langle w - \theta^\star - (\mu^\star)^\top \left( \max_{a \in \mathcal{A}_{h+1}} w^\top \phi(\cdot, a) \right), \quad \mathbb{E}_{s_h, a_h \sim d_h^{\pi_g}} \phi(s_h, a_h) \right\rangle.
\end{aligned}
$$

Namely, Bellman error is written as the inner product of two vectors whose dimension is $d$. Thus, Bellman rank is at most $d$ for linear MDPs.

**Bellman Completion in Linear Function Approximation**

The last example we study here further generalizes linear MDPs. The setting *Bellman Completion in Linear Function Approximation* is defined as follows.

**Definition 9.6** (Bellman Completion under Linear Function Class). We assume $r(s_h, a_h) = \theta^\star \cdot \phi(s_h, a_h)$ and we consider the following linear function class $\mathcal{F} = \{w^\top \phi(s, a) : w \in \mathbb{R}^d, \|w\|_2 \leq W\}$. Recall the Bellman operator $\mathcal{T}$. We have Bellman completion under $\mathcal{F}$ if and only if for any $f \in \mathcal{F}, \mathcal{T}f \in \mathcal{F}$.

We can verify that linear MDP is a special instance here.

Consider $f(s, a) := w^\top \phi(s, a)$, and denote $\mathcal{T}f(s, a) := \widetilde{w}^\top \phi(s_h, a_h)$. We can rewrite the Bellman error as follows:

$$
\begin{aligned}
\mathcal{E}(f; \pi_g, h) &= \mathbb{E}_{s_h, a_h \sim d_h^{\pi_g}} \left[ w^\top \phi(s_h, a_h) - (\theta^\star)^\top \phi(s_h, a_h) - \mathbb{E}_{s_{h+1} \sim P(\cdot|s_h, a_h)} \max_{a \in \mathcal{A}_{h+1}} w^\top \phi(s_{h+1}, a) \right] \\
&= \mathbb{E}_{s_h, a_h \sim d_h^{\pi_g}} \left[ w^\top \phi(s_h, a_h) - \widetilde{w}^\top \phi(s_h, a_h) \right] \\
&= \left\langle w - \widetilde{w}, \quad \mathbb{E}_{s_h, a_h \sim d_h^{\pi_g}} \left[ \phi(s_h, a_h) \right] \right\rangle.
\end{aligned}
$$

Namely, we can write the bellman error as an inner product of two vectors with dimension $d$. This indicates that the Bellman Rank is at most $d$.

## 9.3.2 Examples that do not have low Bellman Rank

[Jiang et al., 2017] also demonstrates a few more examples (with a slightly modification of the definition of average Bellman error) also admits low-Bellman rank including reactive POMDPs and reactive PSRs. So far in the literature, the only example that doesn't admit low Bellman rank but has polynomial sample complexity algorithms is the factored MDPs [Kearns and Koller, 1999]. Sun et al. [2019a] showed that Bellman rank can be exponential in Factored MDPs.

## 9.4 Algorithm

Having defined our main structural assumption, we now describe an algorithm whose sample complexity depends on the Bellman rank, with no explicit dependence on $|\mathcal{S}|$ and $|\mathcal{A}|$ and only logarithmic scaling with $|\mathcal{F}|$. For ease of presentation, we will assume that all the expectations can be measured exactly with no errors, which serves to illustrate the key idea of *Explore-or-Terminate*. For a more careful analysis with finite samples, we refer the reader to Jiang et al. [2017]. The algorithm, named OLIVE for Optimism Led Iterative Value-function Elimination is an iterative algorithm which successively prunes value functions that have non-zero average Bellman error (recall $Q^\star$ has zero Bellman error at any state-action pair). It then uses the principle of optimism in the face of uncertainty to select its next policy which allows us to detect if a new optimal policy has been found. The algorithm is described in Algorithm 7.

---

**Algorithm 6** The OLIVE algorithm for MDPs with low Bellman rank

---

**Input:** Function class $\mathcal{F}$.

1: Initialize $\mathcal{F}_0 = \mathcal{F}$.
2: **for** $t = 1, 2, \ldots,$ **do**
3:      Define $f_t = \operatorname{argmax}_{f \in \mathcal{F}_{t-1}} \max_a f(s_0, a)$ and $\pi_t = \pi_{f_t}$.
4:      **if** $\max_a f_t(s_0, a) = V^{\pi_t}$ **then return** $\pi_t$.
5:      **else**
6:          Update $\mathcal{F}_t = \{f \in \mathcal{F}_{t-1} \ : \ \mathcal{E}(f; \pi_t, h) = 0, \text{ for all } h \in [H-1]\}$.
7:      **end if**
8: **end for**

---

The key step here is that during elimination procedure (Line 6), we enumerate all remaining $f \in \mathcal{F}_{t-1}$ under a fixed roll-in policy $\pi_t$, i.e., we can estimate $\mathcal{E}(f; \pi_t, h)$ for all $f$ via a dataset that is collected from $\pi_t$. Assume we get the following dataset $\{s_h^i, a_h^i, r_h^i, s_{h+1}^i\}_{i=1}^N$ where $s_h^i, a_h^i \sim d_h^{\pi_t}$, and $s_{h+1}^i \sim P(\cdot|s_h^i, a_h^i)$. For any $f \in \mathcal{F}_{h-1}$, we can form the following empirical estimate of $\mathcal{E}(f; \pi_t, h)$:

$$\widetilde{\mathcal{E}}(f; \pi_t, h) = \frac{1}{N} \sum_{i=1}^N \left( f(s_h^i, a_h^i) - r_h^i - \max_a f(s_{h+1}^i, a) \right).$$

We can apply Hoeffding's inequality (Lemma A.1) and a union bound over $\mathcal{F}_{t-1}$ together here to get a uniform convergence argument, i.e., for any $f \in \mathcal{F}_{t-1}$, we will have $\left| \widetilde{\mathcal{E}}(f; \pi_t, h) - \mathcal{E}(f; \pi_t, h) \right| = \widetilde{O}(\sqrt{\ln(|\mathcal{F}|/\delta)/N})$.

Since we assume that all the expectations are available exactly, the main complexity analysis in OLIVE concerns the number of iterations before it terminates. When we estimate expectations using samples, this iteration complexity is critical as it also scales the sample complexity of the algorithm. We will state and prove the following theorem regarding the iteration complexity of OLIVE.

We need the following lemma to show that performance difference between $\max_a f(s_0, a)$ and the value of its induced greedy policy $V^{\pi_t}$.

**Lemma 9.7** (Performance Difference). *For any $f$, we have:*

$$\max_a f(x_0, a) - V^{\pi_f}(x_0) = \sum_{h=0}^{H-1} \mathbb{E}_{x_h, a_h \sim d_h^{\pi_f}} \left[ f(x_h, a_h) - r(x_h, a_h) - \mathbb{E}_{x_{h+1} \sim P(\cdot|x_h, a_h)} \left[ \max_a f(x_{h+1}, a) \right] \right].$$

We leave the proof of the above lemma in HW2.

**Theorem 9.8.** *For any MDP and $\mathcal{F}$ with Bellman rank $M$, OLIVE terminates in at most $MH$ iterations and outputs $\pi^\star$.*

**Proof:** Consider an iteration $t$ of OLIVE. Due to Assumption 9.2 and Fact 9.4, we know that $Q^\star \in \mathcal{F}_{t-1}$. Suppose OLIVE terminates at this iteration and returns $\pi_t$. Then we have

$$V^{\pi_t} = V_{f_t} = \max_{f \in \mathcal{F}_{t-1}} V_f \geq V_{Q^\star} = V^\star,$$

since $Q^\star \in \mathcal{F}_{t-1}$. So the algorithm correctly outputs an optimal policy when it terminates.

On the other hand, if it does not terminate then $V^{\pi_t} \neq V_{f_t}$ and Lemma 9.7 implies that $\mathcal{E}(f_t, \pi_t, h) > 0$ for some step $h \in [H-1]$. This certainly ensures that $f_t \notin \mathcal{F}_t$, but has significantly stronger implications. Note that $f_t \in \mathcal{F}_{t-1}$ implies that $\mathcal{E}(f_t, \pi_s, h) = 0$ for all $s < t$ and $h \in [H-1]$. Since we just concluded that $\mathcal{E}(f_t, \pi_t, h) > 0$ for some $h$, it must be the case that the row corresponding to $\pi_t$ is linearly independent of those corresponding to $\pi_1, \ldots, \pi_{t-1}$ in the matrix $\mathcal{E}_h$. Consequently, at each non-final iteration, OLIVE finds a row that is linearly independent of all previous identified rows (i.e., indexed by $f_1, f_2, \ldots, f_{t-1}$). Since $\mathcal{E}_h$ has rank $M$, the total number of linearly independent rows one could find is at most $M$! Recall that there are at most $H$ many Bellman error matrices, then the algorithm must terminate in number of rounds $HM$, which gives the statement of the theorem. ∎

The proof of the theorem makes it precise that the factorization underlying Bellman rank really plays the role of an efficient basis for exploration in complex MDPs. Extending these ideas to noisy estimates of expectations requires some care since algebraic notions like rank are not robust to noise. Instead Jiang et al. [2017] use a more general volumetric argument to analyze the noisy case, as well as describe robustness to requirements of exact low-rank factorization and realizability.

Unfortunately, the OLIVE algorithm is not computationally efficient, and a computational hardness result was discovered by Dann et al. [2018]. Developing both statistically and computationally efficient exploration algorithms for RL with rich observations is an area of active research.

However, whether there exists a computationally efficient algorithm for low Bellman rank setting is still an open problem.

## 9.5   Extension to Model-based Setting

We briefly discuss the extension to model-based setting. Different from model-free value-based setting, in model-based learning, we start with a model class that contains possible transitions. To ease presentation, we denote the ground truth model as $P^\star$.

$$\mathcal{P} = \{P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})\}.$$

Again we assume realizability:

**Assumption 9.9.** We assume $P^\star \in \mathcal{P}$.

Given any $P \in \mathcal{P}$, we can define the optimal value function and optimal Q function under $P$. Specifically, we denote $V_P^\star$ and $Q_P^\star$ as the optimal value and Q function under model $P$, and $\pi_P$ as the optimal policy under $P$.

We introduce a class of witness functions (a.k.a discriminators),

$$\mathcal{G} = \{g : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \mapsto \mathbb{R}\}.$$

We denote the *witness model-misfit* for a model $P$ as follows:

$$\mathcal{W}(P; \bar{P}, h, \mathcal{G}) := \sup_{g \in \mathcal{G}} \mathbb{E}_{s_h, a_h \sim d_h^{\pi_{\bar{P}}}} \left[ \mathbb{E}_{s_{h+1} \sim P(\cdot|s_h, a_h)} g(s_h, a_h, s_{h+1}) - \mathbb{E}_{s_{h+1} \sim P^\star(\cdot|s_h, a_h)} g(s_h, a_h, s_{h+1}) \right].$$

Namely, we are using a discriminator class $\mathcal{G}$ to distinguish two distributions $d_h^{\pi_{\bar{P}}} \circ P$ and $d_h^{\pi_{\bar{P}}} \circ P^\star$. Note that for the ground truth model $P^\star$, we always have witness misfit being zero, i.e., $\mathcal{W}(P^\star; \bar{P}, h, \mathcal{G}) = 0$ for any $\bar{P}, h, \mathcal{G}$.

To further compare to Bellman rank, we assume the following assumption on the discriminators:

**Assumption 9.10.** We assume $\mathcal{G}$ contains functions $r + V_P^\star$ for all $P \in \mathcal{P}$, i.e.,

$$\{r(s, a) + V_P^\star(s') : P \in \mathcal{P}\} \subseteq \mathcal{G}.$$

With this assumption, we can show that *Bellman Domination*, i.e., $\mathcal{W}(P; \bar{P}, h, \mathcal{G}) \geq \mathcal{E}(Q_P^\star; \pi_{\bar{P}}, h)$, i.e., the average Bellman error of $Q_P^\star$ under the state-action distribution induced by the optimal policy of $\bar{P}$.

We define Witness rank as follows.

**Definition 9.11** (Witness Rank). Consider the Witness misfit matrix $\mathcal{W}_h \in \mathbb{R}^{|\mathcal{P}| \times |\mathcal{P}|}$, where $[\mathcal{W}_h]_{\bar{P}, P} := \mathcal{W}(P; \bar{P}, h, \mathcal{G})$. Witness rank is the smallest number $M$ such that $\text{rank}(\mathcal{W}_h) \leq M$ for all $h$.

Sun et al. [2019a] provides a more refined definition of Witness rank which is at least as small as the Bellman rank under the value functions $\mathcal{Q} = \{Q_P^\star : P \in \mathcal{P}\}$, and shows that for factored MDPs, Bellman rank with $\mathcal{Q}$ could be exponentially large with respect to horizon $H$ while witness rank remains small and captures the right complexity measure in factored MDPs.

Indeed one can show that there exists a factored MDP, such that no model-free algorithms using $\mathcal{Q}$ as a function class input is able to achieve polynomial sample complexity, while the algorithm OLIME which we present later, achieves polynomially sample complexity [Sun et al., 2019a]. This demonstrates an exponential separation in sample complexity between model-based approaches and model-free approaches.

We close here by providing the algorithm *Optimism-Led Iterative Model Elimination* (OLIME) below.

---
**Algorithm 7** The OLIME algorithm for MDPs with low Witness rank
---
**Input:** model class $\mathcal{P}$.
1: Initialize $\mathcal{P}_0 = \mathcal{P}$.
2: **for** $t = 1, 2, \ldots,$ **do**
3:     Define $P_t = \text{argmax}_{P \in \mathcal{P}_{t-1}} V_P^\star(s_0)$ and $\pi_t = \pi_{P_t}$.
4:     **if** $V_{P_t}^{\pi_t} = V^{\pi_t}$ **then return** $\pi_t$.
5:     **else**
6:         Update $\mathcal{P}_t = \{f \in \mathcal{P}_{t-1} : \mathcal{W}(P; \pi_t, h, \mathcal{G}) = 0, \text{ for all } h \in [H-1]\}$.
7:     **end if**
8: **end for**
---

The algorithm use optimism and maintains $\mathcal{P}_t$ in each iteration. Every iteration, it picks the most optimistic model $P_t$ from the current model class $\mathcal{P}_t$. Again optimism allows us to identify if the algorithm has find the optimal policy. If the algorithm does not terminate, then we are guaranteed to find a model $P_t$ that admits non-zero witness misfit. We then update $\mathcal{P}_t$ by eliminating all models that have non-zero model misfit under the distribution of $\pi_t$. Again, following the similar argument we had for OLIVE, one can show that OLIME terminates in at most $MH$ iterations, with $M$ being the witness rank.

## 9.6   Bibliographic Remarks and Further Readings

Bellman rank was original proposed by Jiang et al. [2017]. Note that the definition of our average Bellman error is a simple modification of the original average Bellman error definition in [Jiang et al., 2017]. Our definition allows it

to capture linear MDPs and the Bellman Completion in Linear Function Approximation, without assuming discrete action space and paying a polynomial dependency on the number of actions.

Witness rank was proposed by Sun et al. [2019a] for model-based setting. Sun et al. [2019a] showed that witness rank captures the correct complexity in Factored MDPs [Kearns and Koller, 1999] while Bellman rank could be exponentially large in $H$ when applied to Factored MDPs. Sun et al. [2019a] also demonstrates an exponential sample complexity gap between model-based algorithms and model-free algorithms with general function approximation.

# Chapter 10

# Deterministic MDPs with Linearly Parameterized $Q^\star$

*To be added*

# Part 3

# Policy Optimization

# Chapter 11

# Policy Gradient Methods and Non-Convex Optimization

For a distribution $\rho$ over states, define:

$$V^\pi(\rho) := \mathbb{E}_{s_0 \sim \rho}[V^\pi(s_0)],$$

where we slightly overload notation. Consider a class of parametric policies $\{\pi_\theta | \theta \in \Theta \subset \mathbb{R}^d\}$. The optimization problem of interest is:

$$\max_{\theta \in \Theta} V^{\pi_\theta}(\rho).$$

We drop the MDP subscript in this chapter.

One immediate issue is that if the policy class $\{\pi_\theta\}$ consists of deterministic policies then $\pi_\theta$ will, in general, not be differentiable. This motivates us to consider policy classes that are stochastic, which permit differentiability.

**Example 11.1** (Softmax policies). It is instructive to explicitly consider a "tabular" policy representation, given by the *softmax policy*:

$$\pi_\theta(a|s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}, \tag{0.1}$$

where the parameter space is $\Theta = \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$. Note that (the closure of) the set of softmax policies contains all stationary and deterministic policies.

**Example 11.2** (Log-linear policies). For any state, action pair $s, a$, suppose we have a feature mapping $\phi_{s,a} \in \mathbb{R}^d$. Let us consider the policy class

$$\pi_\theta(a|s) = \frac{\exp(\theta \cdot \phi_{s,a})}{\sum_{a' \in \mathcal{A}} \exp(\theta \cdot \phi_{s,a'})}$$

with $\theta \in \mathbb{R}^d$.

**Example 11.3** (Neural softmax policies). Here we may be interested in working with the policy class

$$\pi_\theta(a|s) = \frac{\exp\left(f_\theta(s,a)\right)}{\sum_{a' \in \mathcal{A}} \exp\left(f_\theta(s,a')\right)}$$

where the scalar function $f_\theta(s, a)$ may be parameterized by a neural network, with $\theta \in \mathbb{R}^d$.

## 11.1 Policy Gradient Expressions and the Likelihood Ratio Method

Let $\tau$ denote a trajectory, whose unconditional distribution $\Pr_\mu^\pi(\tau)$ under $\pi$ with starting distribution $\mu$, is

$$\Pr_\mu^\pi(\tau) = \mu(s_0)\pi(a_0|s_0)P(s_1|s_0,a_0)\pi(a_1|s_1)\cdots. \tag{0.2}$$

We drop the $\mu$ subscript when it is clear from context.

It is convenient to define the discounted total reward of a trajectory as:

$$R(\tau) := \sum_{t=0}^\infty \gamma^t r(s_t, a_t)$$

where $s_t, a_t$ are the state-action pairs in $\tau$. Observe that:

$$V^{\pi_\theta}(\mu) = \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}}[R(\tau)].$$

**Theorem 11.4.** *(Policy gradients) The following are expressions for $\nabla_\theta V^{\pi_\theta}(\mu)$:*

- *REINFORCE:*

$$\nabla V^{\pi_\theta}(\mu) = \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}}\left[R(\tau)\sum_{t=0}^\infty \nabla \log \pi_\theta(a_t|s_t)\right]$$

- *Action value expression:*

$$\begin{aligned}
\nabla V^{\pi_\theta}(\mu) &= \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\theta}}\left[\sum_{t=0}^\infty \gamma^t Q^{\pi_\theta}(s_t, a_t)\nabla \log \pi_\theta(a_t|s_t)\right] \\
&= \frac{1}{1-\gamma}\mathbb{E}_{s \sim d^{\pi_\theta}}\mathbb{E}_{a \sim \pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,a)\nabla \log \pi_\theta(a|s)\right]
\end{aligned}$$

- *Advantage expression:*

$$\nabla V^{\pi_\theta}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s \sim d^{\pi_\theta}}\mathbb{E}_{a \sim \pi_\theta(\cdot|s)}\left[A^{\pi_\theta}(s,a)\nabla \log \pi_\theta(a|s)\right]$$

The alternative expressions are more helpful to use when we turn to Monte Carlo estimation.

**Proof:** We have:

$$\begin{aligned}
\nabla V^{\pi_\theta}(\mu) &= \nabla \sum_\tau R(\tau)\Pr_\mu^{\pi_\theta}(\tau) \\
&= \sum_\tau R(\tau)\nabla \Pr_\mu^{\pi_\theta}(\tau) \\
&= \sum_\tau R(\tau)\Pr_\mu^{\pi_\theta}(\tau)\nabla \log \Pr_\mu^{\pi_\theta}(\tau) \\
&= \sum_\tau R(\tau)\Pr_\mu^{\pi_\theta}(\tau)\nabla \log \left(\mu(s_0)\pi_\theta(a_0|s_0)P(s_1|s_0,a_0)\pi_\theta(a_1|s_1)\cdots\right) \\
&= \sum_\tau R(\tau)\Pr_\mu^{\pi_\theta}(\tau)\left(\sum_{t=0}^\infty \nabla \log \pi_\theta(a_t|s_t)\right)
\end{aligned}$$

which completes the proof of the first claim.

For the second claim, for any state $s_0$

$$\nabla V^{\pi_\theta}(s_0)$$
$$= \nabla \sum_{a_0} \pi_\theta(a_0|s_0) Q^{\pi_\theta}(s_0, a_0)$$
$$= \sum_{a_0} \left(\nabla \pi_\theta(a_0|s_0)\right) Q^{\pi_\theta}(s_0, a_0) + \sum_{a_0} \pi_\theta(a_0|s_0) \nabla Q^{\pi_\theta}(s_0, a_0)$$
$$= \sum_{a_0} \pi_\theta(a_0|s_0) \left(\nabla \log \pi_\theta(a_0|s_0)\right) Q^{\pi_\theta}(s_0, a_0)$$
$$\quad + \sum_{a_0} \pi_\theta(a_0|s_0) \nabla \left(r(s_0, a_0) + \gamma \sum_{s_1} P(s_1|s_0, a_0) V^{\pi_\theta}(s_1)\right)$$
$$= \sum_{a_0} \pi_\theta(a_0|s_0) \left(\nabla \log \pi_\theta(a_0|s_0)\right) Q^{\pi_\theta}(s_0, a_0) + \gamma \sum_{a_0, s_1} \pi_\theta(a_0|s_0) P(s_1|s_0, a_0) \nabla V^{\pi_\theta}(s_1)$$
$$= \mathbb{E}_{\tau \sim \mathrm{Pr}_{s_0}^{\pi_\theta}} \left[Q^{\pi_\theta}(s_0, a_0) \nabla \log \pi_\theta(a_0|s_0)\right] + \gamma \mathbb{E}_{\tau \sim \mathrm{Pr}_{s_0}^{\pi_\theta}} \left[\nabla V^{\pi_\theta}(s_1)\right].$$

By linearity of expectation,

$$\nabla V^{\pi_\theta}(\mu)$$
$$= \mathbb{E}_{\tau \sim \mathrm{Pr}_\mu^{\pi_\theta}} \left[Q^{\pi_\theta}(s_0, a_0) \nabla \log \pi_\theta(a_0|s_0)\right] + \gamma \mathbb{E}_{\tau \sim \mathrm{Pr}_\mu^{\pi_\theta}} \left[\nabla V^{\pi_\theta}(s_1)\right]$$
$$= \mathbb{E}_{\tau \sim \mathrm{Pr}_\mu^{\pi_\theta}} \left[Q^{\pi_\theta}(s_0, a_0) \nabla \log \pi_\theta(a_0|s_0)\right] + \gamma \mathbb{E}_{\tau \sim \mathrm{Pr}_\mu^{\pi_\theta}} \left[Q^{\pi_\theta}(s_1, a_1) \nabla \log \pi_\theta(a_1|s_1)\right] + \dots .$$

where the last step follows from recursion. This completes the proof of the second claim.

The proof of the final claim is left as an exercise to the reader. ∎

## 11.2 (Non-convex) Optimization

It is worth explicitly noting that $V^{\pi_\theta}(s)$ is non-concave in $\theta$ for the softmax parameterizations, so the standard tools of convex optimization are not applicable.

**Lemma 11.5.** *(Non-convexity) There is an MDP $M$ (described in Figure 0.1) such that the optimization problem $V^{\pi_\theta}(s)$ is not concave for both the direct and softmax parameterizations.*

**Proof:** Recall the MDP in Figure 0.1. Note that since actions in terminal states $s_3$, $s_4$ and $s_5$ do not change the expected reward, we only consider actions in states $s_1$ and $s_2$. Let the "up/above" action as $a_1$ and "right" action as $a_2$. Note that

$$V^\pi(s_1) = \pi(a_2|s_1)\pi(a_1|s_2) \cdot r$$

Now consider

$$\theta^{(1)} = (\log 1, \log 3, \log 3, \log 1), \quad \theta^{(2)} = (-\log 1, -\log 3, -\log 3, -\log 1)$$

where $\theta$ is written as a tuple $(\theta_{a_1,s_1}, \theta_{a_2,s_1}, \theta_{a_1,s_2}, \theta_{a_2,s_2})$. Then, for the softmax parameterization, we have:

$$\pi^{(1)}(a_2|s_1) = \frac{3}{4}; \quad \pi^{(1)}(a_1|s_2) = \frac{3}{4}; \quad V^{(1)}(s_1) = \frac{9}{16}r$$

and

$$\pi^{(2)}(a_2|s_1) = \frac{1}{4}; \quad \pi^{(2)}(a_1|s_2) = \frac{1}{4}; \quad V^{(2)}(s_1) = \frac{1}{16}r.$$
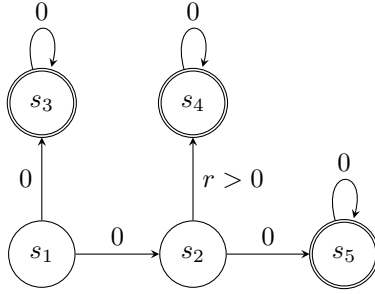
Figure 0.1: (Non-concavity example) A deterministic MDP corresponding to Lemma 11.5 where $V^{\pi_\theta}(s)$ is not concave. Numbers on arrows represent the rewards for each action.

Also, for $\theta^{(\mathrm{mid})} = \frac{\theta^{(1)}+\theta^{(2)}}{2}$,

$$\pi^{(\mathrm{mid})}(a_2|s_1) = \frac{1}{2}; \quad \pi^{(\mathrm{mid})}(a_1|s_2) = \frac{1}{2}; \quad V^{(\mathrm{mid})}(s_1) = \frac{1}{4}r.$$

This gives

$$V^{(1)}(s_1) + V^{(2)}(s_1) > 2V^{(\mathrm{mid})}(s_1),$$

which shows that $V^\pi$ is non-concave. ∎

## 11.2.1 Gradient ascent and convergence to stationary points

Let us say a function $f : \mathbb{R}^d \to \mathbb{R}$ is $\beta$-*smooth* if

$$\|\nabla f(w) - \nabla f(w')\| \le \beta \|w - w'\|, \tag{0.3}$$

where the norm $\|\cdot\|$ is the Euclidean norm. In other words, the derivatives of $f$ do not change too quickly.

Gradient ascent, with a fixed stepsize $\eta$, follows the update rule:

$$\theta_{t+1} = \theta_t + \eta \nabla V^{\pi_{\theta_t}}(\mu).$$

It is convenient to use the shorthand notation:

$$\pi^{(t)} := \pi_{\theta_t}, \qquad V^{(t)} := V^{\pi_{\theta_t}}$$

The next lemma is standard in non-convex optimization.

**Lemma 11.6.** *(Convergence to Stationary Points) Assume that for all $\theta \in \Theta$, $V^{\pi_\theta}$ is $\beta$-smooth and bounded below by $V_*$. Suppose we use the constant stepsize $\eta = 1/\beta$. For all $T$, we have that*

$$\min_{t \le T} \|\nabla V^{(t)}(\mu)\|^2 \le \frac{2\beta(V^*(\mu) - V^{(0)}(\mu))}{T}.$$

## 11.2.2 Monte Carlo estimation and stochastic gradient ascent

One difficulty is that even if we know the MDP $M$, computing the gradient may be computationally intensive. It turns out that we can obtain unbiased estimates of $\pi$ with only simulation based access to our model, i.e. assuming we can obtain sampled trajectories $\tau \sim \mathrm{Pr}_\mu^{\pi_\theta}$.

With respect to a trajectory $\tau$, define:

$$\widehat{Q^{\pi_\theta}}(s_t, a_t) \quad := \quad \sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'})$$

$$\widehat{\nabla V^{\pi_\theta}}(\mu) \quad := \quad \sum_{t=0}^{\infty} \gamma^t \widehat{Q^{\pi_\theta}}(s_t, a_t) \nabla \log \pi_\theta(a_t | s_t)$$

We now show this provides an unbiased estimated of the gradient:

**Lemma 11.7.** *(Unbiased gradient estimate) We have :*

$$\mathbb{E}_{\tau \sim \mathrm{Pr}_\mu^{\pi_\theta}} \left[ \widehat{\nabla V^{\pi_\theta}}(\mu) \right] = \nabla V^{\pi_\theta}(\mu)$$

**Proof:** Observe:

$$
\begin{aligned}
\mathbb{E}[\widehat{\nabla V^{\pi_\theta}}(\mu)] \quad &= \quad \mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t \widehat{Q^{\pi_\theta}}(s_t, a_t) \nabla \log \pi_\theta(a_t | s_t) \right] \\
&\overset{(a)}{=} \quad \mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t \mathbb{E}[\widehat{Q^{\pi_\theta}}(s_t, a_t) | s_t, a_t] \nabla \log \pi_\theta(a_t | s_t) \right] \\
&\overset{(b)}{=} \quad \mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t Q^{\pi_\theta}(s_t, a_t) \nabla \log \pi_\theta(a_t | s_t) \right]
\end{aligned}
$$

where $(a)$ follows from the tower property of the conditional expectations and $(b)$ follows from that the Markov property implies $\mathbb{E}[\widehat{Q^{\pi_\theta}}(s_t, a_t) | s_t, a_t] = Q^{\pi_\theta}(s_t, a_t)$. ∎

Hence, the following procedure is a stochastic gradient ascent algorithm:

1. initialize $\theta_0$.

2. For $t = 0, 1, \ldots$

   (a) Sample $\tau \sim \mathrm{Pr}_\mu^{\pi_\theta}$.
   (b) Update:
   $$\theta_{t+1} = \theta_t + \eta_t \widehat{\nabla V^{\pi_\theta}}(\mu)$$
   where $\eta_t$ is the stepsize and $\widehat{\nabla V^{\pi_\theta}}(\mu)$ estimated with $\tau$.

Note here that we are ignoring that $\tau$ is an infinte length sequence. It can be truncated appropriately so as to control the bias.

The following is standard result with regards to non-convex optimization. Again, with reasonably bounded variance, we will obtain a point $\theta_t$ with small gradient norm.

**Lemma 11.8.** *(Stochastic Convergence to Stationary Points) Assume that for all $\theta \in \Theta$, $V^{\pi_\theta}$ is $\beta$-smooth and bounded below by $V_*$. Suppose the variance is bounded as follows:*

$$\mathbb{E}[\|\widehat{\nabla V^{\pi_\theta}}(\mu) - \nabla V^{\pi_\theta}(\mu)\|^2] \leq \sigma^2$$

*For $t \leq \beta(V^*(\mu) - V^{(0)}(\mu))/\sigma^2$, suppose we use a constant stepsize of $\eta_t = 1/\beta$, and thereafter, we use $\eta_t = \sqrt{2/(\beta T)}$. For all $T$, we have:*

$$\min_{t \leq T} \mathbb{E}[\|\nabla V^{(t)}(\mu)\|^2] \leq \frac{2\beta(V^*(\mu) - V^{(0)}(\mu))}{T} + \sqrt{\frac{2\sigma^2}{T}}.$$

**Baselines and stochastic gradient ascent**

A significant practical issue is that the variance $\sigma^2$ is often large in practice. Here, a form of variance reduction is often critical in practice. A common method is as follows.

Let $f : \mathcal{S} \to \mathbb{R}$.

1. Construct $f$ as an estimate of $V^{\pi_\theta}(\mu)$. This can be done using any previous data.

2. Sample a new trajectory $\tau$, and define:

$$
\widehat{Q^{\pi_\theta}}(s_t, a_t) \;:=\; \sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'})
$$

$$
\widehat{\nabla V^{\pi_\theta}}(\mu) \;:=\; \sum_{t=0}^{\infty} \gamma^t \Big( \widehat{Q^{\pi_\theta}}(s_t, a_t) - f(s_t) \Big) \nabla \log \pi_\theta(a_t|s_t)
$$

We often refer to $f(s)$ as a *baseline* at state $s$.

**Lemma 11.9.** *(Unbiased gradient estimate with Variance Reduction) For any procedure used to construct to the baseline function $f : \mathcal{S} \to \mathbb{R}$, if the samples used to construct $f$ are independent of the trajectory $\tau$, where $\widehat{Q^{\pi_\theta}}(s_t, a_t)$ is constructed using $\tau$, then:*

$$
\mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t \Big( \widehat{Q^{\pi_\theta}}(s_t, a_t) - f(s_t) \Big) \nabla \log \pi_\theta(a_t|s_t) \right] = \nabla V^{\pi_\theta}(\mu)
$$

*where the expectation is with respect to both the random trajectory $\tau$ and the random function $f(\cdot)$.*

**Proof:** For any function $g(s)$,

$$
\mathbb{E}\left[ \nabla \log \pi(a|s) g(s) \right] = \sum_a \nabla \pi(a|s) g(s) = g(s) \sum_a \nabla \pi(a|s) = g(s) \nabla \sum_a \pi(a|s) = g(s) \nabla 1 = 0
$$

Using that $f(\cdot)$ is independent of $\tau$, we have that for all $t$

$$
\mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t f(s_t) \nabla \log \pi_\theta(a_t|s_t) \right] = 0
$$

The result now follow froms Lemma 11.7. ∎

## 11.3  Bibliographic Remarks and Further Readings

The REINFOCE algorithm is due to [Williams, 1992], which is an example of the likelihood ratio method for gradient estimation [Glynn, 1990].

For standard optimization results in non-convex optimization (e.g. Lemma 11.6 and 11.8), we refer the reader to [Beck, 2017]. Our results for convergence rates for SGD to approximate stationary points follow from [Ghadimi and Lan, 2013].

# Chapter 12

# Optimality

We now seek to understand the global convergence properties of policy gradient methods, when given exact gradients. Here, we will largely limit ourselves to the (tabular) softmax policy class in Example 11.1.

Given our a starting distribution $\rho$ over states, recall our objective is:

$$\max_{\theta \in \Theta} V^{\pi_\theta}(\rho).$$

where $\{\pi_\theta | \theta \in \Theta \subset \mathbb{R}^d\}$ is some class of parametric policies.

While we are interested in good performance under $\rho$, we will see how it is helpful to optimize under a different measure $\mu$. Specifically, we consider optimizing $V^{\pi_\theta}(\mu)$, i.e.

$$\max_{\theta \in \Theta} V^{\pi_\theta}(\mu),$$

even though our ultimate goal is performance under $V^{\pi_\theta}(\rho)$.

We now consider the softmax policy parameterization (0.1). Here, we still have a non-concave optimization problem in general, as shown in Lemma 11.5, though we do show that global optimality can be reached under certain regularity conditions. From a practical perspective, the softmax parameterization of policies is preferable to the direct parameterization, since the parameters $\theta$ are unconstrained and standard unconstrained optimization algorithms can be employed. However, optimization over this policy class creates other challenges as we study in this section, as the optimal policy (which is deterministic) is attained by sending the parameters to infinity.

This chapter will study three algorithms for this problem, for the softmax policy class. The first performs direct policy gradient ascent on the objective without modification, while the second adds a log barrier regularizer to keep the parameters from becoming too large, as a means to ensure adequate exploration. Finally, we study the natural policy gradient algorithm and establish a global optimality convergence rate, with no dependence on the dimension-dependent factors.

The presentation in this chapter largely follows the results in [Agarwal et al., 2020d].

## 12.1 Vanishing Gradients and Saddle Points

To understand the necessity of optimizing under a distribution $\mu$ that is different from $\rho$, let us first give an informal argument that some condition on the state distribution of $\pi$, or equivalently $\mu$, is necessary for stationarity to imply
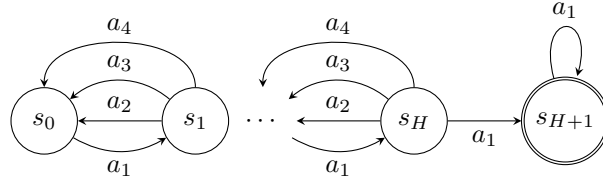
Figure 0.1: (Vanishing gradient example) A deterministic, chain MDP of length $H + 2$. We consider a policy where $\pi(a|s_i) = \theta_{s_i,a}$ for $i = 1, 2, \ldots, H$. Rewards are 0 everywhere other than $r(s_{H+1}, a_1) = 1$. See Proposition 12.1.

optimality. For example, in a sparse-reward MDP (where the agent is only rewarded upon visiting some small set of states), a policy that does not visit *any* rewarding states will have zero gradient, even though it is arbitrarily suboptimal in terms of values. Below, we give a more quantitative version of this intuition, which demonstrates that even if $\pi$ chooses all actions with reasonable probabilities (and hence the agent will visit all states if the MDP is connected), then there is an MDP where a large fraction of the policies $\pi$ have vanishingly small gradients, and yet these policies are highly suboptimal in terms of their value.

Concretely, consider the chain MDP of length $H + 2$ shown in Figure 0.1. The starting state of interest is state $s_0$ and the discount factor $\gamma = H/(H + 1)$. Suppose we work with the direct parameterization, where $\pi_\theta(a|s) = \theta_{s,a}$ for $a = a_1, a_2, a_3$ and $\pi_\theta(a_4|s) = 1 - \theta_{s,a_1} - \theta_{s,a_2} - \theta_{s,a_3}$. Note we do not over-parameterize the policy. For this MDP and policy structure, if we were to initialize the probabilities over actions, say deterministically, then there is an MDP (obtained by permuting the actions) where all the probabilities for $a_1$ will be less than $1/4$.

The following result not only shows that the gradient is exponentially small in $H$, it also shows that many higher order derivatives, up to $O(H/\log H)$, are also exponentially small in $H$.

**Proposition 12.1** (Vanishing gradients at suboptimal parameters). *Consider the chain MDP of Figure 0.1, with $H + 2$ states, $\gamma = H/(H + 1)$, and with the direct policy parameterization (with $3|\mathcal{S}|$ parameters, as described in the text above). Suppose $\theta$ is such that $0 < \theta < 1$ (componentwise) and $\theta_{s,a_1} < 1/4$ (for all states s). For all $k \leq \frac{H}{40 \log(2H)} - 1$, we have $\left\| \nabla_\theta^k V^{\pi_\theta}(s_0) \right\| \leq (1/3)^{H/4}$, where $\nabla_\theta^k V^{\pi_\theta}(s_0)$ is a tensor of the $k_{th}$ order derivatives of $V^{\pi_\theta}(s_0)$ and the norm is the operator norm of the tensor.[1] Furthermore, $V^\star(s_0) - V^{\pi_\theta}(s_0) \geq (H + 1)/8 - (H + 1)^2/3^H$.*

We do not prove this lemma here (see Section 12.5). The lemma illustrates that lack of good exploration can indeed be detrimental in policy gradient algorithms, since the gradient can be small either due to $\pi$ being near-optimal, or, simply because $\pi$ does not visit advantageous states often enough. Furthermore, this lemma also suggests that varied results in the non-convex optimization literature, on escaping from saddle points, do not directly imply global convergence due to that the higher order derivatives are small.

While the chain MDP of Figure 0.1, is a common example where *sample* based estimates of gradients will be 0 under random exploration strategies; there is an exponentially small in $H$ chance of hitting the goal state under a random exploration strategy. Note that this lemma is with regards to *exact* gradients. This suggests that even with exact computations (along with using exact higher order derivatives) we might expect numerical instabilities.

## 12.2 Policy Gradient Ascent

Let us now return to the softmax policy class, from Equation 0.1, where:

$$\pi_\theta(a|s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})},$$

---

[1]The operator norm of a $k_{th}$-order tensor $J \in \mathbb{R}^{d^{\otimes k}}$ is defined as $\sup_{u_1,\ldots,u_k \in \mathbb{R}^d \,:\, \|u_i\|_2 = 1} \langle J, u_1 \otimes \ldots \otimes u_d \rangle$.

where the number of parameters in this policy class is $|\mathcal{S}||\mathcal{A}|$.

Observe that:

$$\frac{\partial \log \pi_\theta(a|s)}{\partial \theta_{s',a'}} = \mathbb{1}\Big[s = s'\Big]\Big(\mathbb{1}\Big[a = a'\Big] - \pi_\theta(a'|s)\Big)$$

where $\mathbb{1}[\mathcal{E}]$ is the indicator of $\mathcal{E}$ being true.

**Lemma 12.2.** *For the softmax policy class, we have:*

$$\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d_\mu^{\pi_\theta}(s)\pi_\theta(a|s)A^{\pi_\theta}(s,a) \tag{0.1}$$

**Proof:** Using the advantage expression for the policy gradient (see Theorem 11.4),

$$
\begin{aligned}
\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta_{s',a'}} &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\Big[A^{\pi_\theta}(s,a)\frac{\partial \log \pi_\theta(a|s)}{\partial \theta_{s',a'}}\Big] \\
&= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\Big[A^{\pi_\theta}(s,a)\mathbb{1}\Big[s = s'\Big]\Big(\mathbb{1}\Big[a = a'\Big] - \pi_\theta(a'|s)\Big)\Big] \\
&= \frac{1}{1-\gamma}d_\mu^{\pi_\theta}(s')\mathbb{E}_{a\sim\pi_\theta(\cdot|s')}\Big[A^{\pi_\theta}(s',a)\Big(\mathbb{1}\Big[a = a'\Big] - \pi_\theta(a'|s')\Big)\Big] \\
&= \frac{1}{1-\gamma}d_\mu^{\pi_\theta}(s')\Big(\mathbb{E}_{a\sim\pi_\theta(\cdot|s')}\Big[A^{\pi_\theta}(s',a)\mathbb{1}\Big[a = a'\Big]\Big] - \pi_\theta(a'|s')\mathbb{E}_{a\sim\pi_\theta(\cdot|s')}\Big[A^{\pi_\theta}(s',a)\Big]\Big) \\
&= \frac{1}{1-\gamma}d_\mu^{\pi_\theta}(s')\pi_\theta(a'|s')A^{\pi_\theta}(s',a') - 0,
\end{aligned}
$$

where the last step uses that for any policy $\sum_a \pi(a|s)A^\pi(s,a) = 0$. ∎

The update rule for gradient ascent is:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta V^{(t)}(\mu). \tag{0.2}$$

Recall from Lemma 11.5 that, even for the case of the softmax policy class (which contains all stationary policies), our optimization problem is non-convex. Furthermore, due to the exponential scaling with the parameters $\theta$ in the softmax parameterization, *any* policy that is nearly deterministic will have gradients close to $0$. Specifically, for any sequence of policies $\pi^{\theta_t}$ that becomes deterministic, $\|\nabla V^{\pi^{\theta_t}}\| \to 0$.

In spite of these difficulties, it turns out we have a positive result that gradient ascent asymptotically converges to the global optimum for the softmax parameterization.

**Theorem 12.3** (Global convergence for softmax parameterization). *Assume we follow the gradient ascent update rule as specified in Equation (0.2) and that the distribution $\mu$ is strictly positive i.e. $\mu(s) > 0$ for all states $s$. Suppose $\eta \le \frac{(1-\gamma)^3}{8}$, then we have that for all states $s$, $V^{(t)}(s) \to V^\star(s)$ as $t \to \infty$.*

The proof is somewhat technical, and we do not provide a proof here (see Section 12.5).

A few remarks are in order. Theorem 12.3 assumed that optimization distribution $\mu$ was *strictly* positive, i.e. $\mu(s) > 0$ for all states $s$. We conjecture that any gradient ascent may not globally converge if this condition is not met. The concern is that if this condition is not met, then gradient ascent may not globally converge due to that $d_\mu^{\pi_\theta}(s)$ effectively scales down the learning rate for the parameters associated with state $s$ (see Equation 0.1).

Furthermore, there is strong reason to believe that the convergence rate for this is algorithm (in the worst case) is exponentially slow in some of the relevant quantities, such as in terms of the size of state space. We now turn to a regularization based approach to ensure convergence at a polynomial rate in all relevant quantities.

## 12.3 Log Barrier Regularization

Due to the exponential scaling with the parameters $\theta$, policies can rapidly become near deterministic, when optimizing under the softmax parameterization, which can result in slow convergence. Indeed a key challenge in the asymptotic analysis in the previous section was to handle the growth of the absolute values of parameters as they tend to infinity. Recall that the relative-entropy for distributions $p$ and $q$ is defined as:

$$\text{KL}(p, q) := \mathbb{E}_{x \sim p}[-\log q(x)/p(x)].$$

Denote the uniform distribution over a set $\mathcal{X}$ by $\text{Unif}_{\mathcal{X}}$, and define the following log barrier regularized objective as:

$$
\begin{aligned}
L_\lambda(\theta) \quad &:= \quad V^{\pi_\theta}(\mu) - \lambda\, \mathbb{E}_{s \sim \text{Unif}_{\mathcal{S}}}\left[\text{KL}(\text{Unif}_{\mathcal{A}}, \pi_\theta(\cdot|s))\right] \\
&= \quad V^{\pi_\theta}(\mu) + \frac{\lambda}{|\mathcal{S}|\,|\mathcal{A}|} \sum_{s,a} \log \pi_\theta(a|s) + \lambda \log |\mathcal{A}|\,,
\end{aligned}
\tag{0.3}
$$

where $\lambda$ is a regularization parameter. The constant (i.e. the last term) is not relevant with regards to optimization. This regularizer is different from the more commonly utilized entropy regularizer, a point which we return to later.

The policy gradient ascent updates for $L_\lambda(\theta)$ are given by:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta L_\lambda(\theta^{(t)}).\tag{0.4}$$

We now see that any approximate first-order stationary points of the entropy-regularized objective is approximately globally optimal, provided the regularization is sufficiently small.

**Theorem 12.4.** *(Log barrier regularization) Suppose $\theta$ is such that:*

$$\|\nabla_\theta L_\lambda(\theta)\|_2 \leq \epsilon_{opt}$$

*and $\epsilon_{opt} \leq \lambda/(2|\mathcal{S}|\,|\mathcal{A}|)$. Then we have that for all starting state distributions $\rho$:*

$$V^{\pi_\theta}(\rho) \quad \geq \quad V^\star(\rho) - \frac{2\lambda}{1-\gamma}\left\|\frac{d_\rho^{\pi^\star}}{\mu}\right\|_\infty.$$

We refer to $\left\|\dfrac{d_\rho^{\pi^\star}}{\mu}\right\|_\infty$ as the *distribution mismatch coefficient*. The above theorem shows the importance of having an appropriate measure $\mu(s)$ in order for the approximate first-order stationary points to be near optimal.

**Proof:** The proof consists of showing that $\max_a A^{\pi_\theta}(s, a) \leq 2\lambda/(\mu(s)|\mathcal{S}|)$ for all states. To see that this is sufficient, observe that by the performance difference lemma (Lemma 1.16),

$$
\begin{aligned}
V^\star(\rho) - V^{\pi_\theta}(\rho) &= \frac{1}{1-\gamma} \sum_{s,a} d_\rho^{\pi^\star}(s) \pi^\star(a|s) A^{\pi_\theta}(s, a) \\
&\leq \frac{1}{1-\gamma} \sum_s d_\rho^{\pi^\star}(s) \max_{a \in \mathcal{A}} A^{\pi_\theta}(s, a) \\
&\leq \frac{1}{1-\gamma} \sum_s 2 d_\rho^{\pi^\star}(s) \lambda/(\mu(s)|\mathcal{S}|) \\
&\leq \frac{2\lambda}{1-\gamma} \max_s \left(\frac{d_\rho^{\pi^\star}(s)}{\mu(s)}\right).
\end{aligned}
$$

which would then complete the proof.

We now proceed to show that $\max_a A^{\pi_\theta}(s, a) \leq 2\lambda/(\mu(s)|\mathcal{S}|)$. For this, it suffices to bound $A^{\pi_\theta}(s, a)$ for any state-action pair $s, a$ where $A^{\pi_\theta}(s, a) \geq 0$ else the claim is trivially true. Consider an $(s, a)$ pair such that $A^{\pi_\theta}(s, a) > 0$. Using the policy gradient expression for the softmax parameterization (see Equation 0.1),

$$\frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d_\mu^{\pi_\theta}(s) \pi_\theta(a|s) A^{\pi_\theta}(s, a) + \frac{\lambda}{|\mathcal{S}|} \left( \frac{1}{|\mathcal{A}|} - \pi_\theta(a|s) \right). \tag{0.5}$$

The gradient norm assumption $\|\nabla_\theta L_\lambda(\theta)\|_2 \leq \epsilon_{\text{opt}}$ implies that:

$$\begin{aligned}
\epsilon_{\text{opt}} \geq \frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} &= \frac{1}{1-\gamma} d_\mu^{\pi_\theta}(s) \pi_\theta(a|s) A^{\pi_\theta}(s, a) + \frac{\lambda}{|\mathcal{S}|} \left( \frac{1}{|\mathcal{A}|} - \pi_\theta(a|s) \right) \\
&\geq \frac{\lambda}{|\mathcal{S}|} \left( \frac{1}{|\mathcal{A}|} - \pi_\theta(a|s) \right),
\end{aligned}$$

where we have used $A^{\pi_\theta}(s, a) \geq 0$. Rearranging and using our assumption $\epsilon_{\text{opt}} \leq \lambda/(2|\mathcal{S}| |\mathcal{A}|)$,

$$\pi_\theta(a|s) \geq \frac{1}{|\mathcal{A}|} - \frac{\epsilon_{\text{opt}} |\mathcal{S}|}{\lambda} \geq \frac{1}{2|\mathcal{A}|}.$$

Solving for $A^{\pi_\theta}(s, a)$ in (0.5), we have:

$$\begin{aligned}
A^{\pi_\theta}(s, a) &= \frac{1-\gamma}{d_\mu^{\pi_\theta}(s)} \left( \frac{1}{\pi_\theta(a|s)} \frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} + \frac{\lambda}{|\mathcal{S}|} \left( 1 - \frac{1}{\pi_\theta(a|s)|\mathcal{A}|} \right) \right) \\
&\leq \frac{1-\gamma}{d_\mu^{\pi_\theta}(s)} \left( 2|\mathcal{A}|\epsilon_{\text{opt}} + \frac{\lambda}{|\mathcal{S}|} \right) \\
&\leq 2\frac{1-\gamma}{d_\mu^{\pi_\theta}(s)} \frac{\lambda}{|\mathcal{S}|} \\
&\leq 2\lambda/(\mu(s)|\mathcal{S}|),
\end{aligned}$$

where the penultimate step uses $\epsilon_{\text{opt}} \leq \lambda/(2|\mathcal{S}| |\mathcal{A}|)$ and the final step uses $d_\mu^{\pi_\theta}(s) \geq (1-\gamma)\mu(s)$. This completes the proof. ∎

The policy gradient ascent updates for $L_\lambda(\theta)$ are given by:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta L_\lambda(\theta^{(t)}). \tag{0.6}$$

By combining the above theorem with the convergence of gradient ascent to first order stationary points (Lemma 11.6), we obtain the following corollary.

**Corollary 12.5.** *(Iteration complexity with log barrier regularization) Let $\beta_\lambda := \frac{8\gamma}{(1-\gamma)^3} + \frac{2\lambda}{|\mathcal{S}|}$. Starting from any initial $\theta^{(0)}$, consider the updates (0.6) with $\lambda = \frac{\epsilon(1-\gamma)}{2\left\|\frac{d_\rho^{\pi^\star}}{\mu}\right\|_\infty}$ and $\eta = 1/\beta_\lambda$. Then for all starting state distributions $\rho$, we have*

$$\min_{t<T} \left\{ V^\star(\rho) - V^{(t)}(\rho) \right\} \leq \epsilon \quad \text{whenever} \quad T \geq \frac{320|\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^6 \epsilon^2} \left\| \frac{d_\rho^{\pi^\star}}{\mu} \right\|_\infty^2.$$

The corollary shows the importance of balancing how the regularization parameter $\lambda$ is set relative to the desired accuracy $\epsilon$, as well as the importance of the initial distribution $\mu$ to obtain global optimality.

**Proof:[of Corollary 12.5]** Let $\beta_\lambda$ be the smoothness of $L_\lambda(\theta)$. A valid upper bound on $\beta_\lambda$ is:

$$\beta_\lambda = \frac{8\gamma}{(1-\gamma)^3} + \frac{2\lambda}{|\mathcal{S}|},$$

where we leave the proof as an exercise to the reader.

Using Theorem 12.4, the desired optimality gap $\epsilon$ will follow if we set $\lambda = \frac{\epsilon(1-\gamma)}{2\left\|\frac{d_\rho^{\pi^\star}}{\mu}\right\|_\infty}$ and if $\|\nabla_\theta L_\lambda(\theta)\|_2 \leq \lambda/(2|\mathcal{S}|\,|\mathcal{A}|)$. In order to complete the proof, we need to bound the iteration complexity of making the gradient sufficiently small.

By Lemma 11.6, after $T$ iterations of gradient ascent with stepsize of $1/\beta_\lambda$, we have

$$\min_{t \leq T} \left\|\nabla_\theta L_\lambda(\theta^{(t)})\right\|_2^2 \leq \frac{2\beta_\lambda(L_\lambda(\theta^\star) - L_\lambda(\theta^{(0)}))}{T} \leq \frac{2\beta_\lambda}{(1-\gamma)\,T}, \tag{0.7}$$

where $\beta_\lambda$ is an upper bound on the smoothness of $L_\lambda(\theta)$. We seek to ensure

$$\epsilon_{\text{opt}} \leq \sqrt{\frac{2\beta_\lambda}{(1-\gamma)\,T}} \leq \frac{\lambda}{2|\mathcal{S}|\,|\mathcal{A}|}$$

Choosing $T \geq \frac{8\beta_\lambda\,|\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)\,\lambda^2}$ satisfies the above inequality. Hence,

$$
\begin{aligned}
\frac{8\beta_\lambda\,|\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)\,\lambda^2} &\leq \frac{64\,|\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^4\,\lambda^2} + \frac{16\,|\mathcal{S}||\mathcal{A}|^2}{(1-\gamma)\,\lambda} \\
&\leq \frac{80\,|\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^4\,\lambda^2} \\
&= \frac{320\,|\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^6\,\epsilon^2}\left\|\frac{d_\rho^{\pi^\star}}{\mu}\right\|_\infty^2
\end{aligned}
$$

where we have used that $\lambda < 1$. This completes the proof. ■

**Entropy vs. log barrier regularization.** A commonly considered regularizer is the entropy, where the regularizer would be:

$$\frac{1}{|\mathcal{S}|}\sum_s H(\pi_\theta(\cdot|s)) = \frac{1}{|\mathcal{S}|}\sum_s\sum_a -\pi_\theta(a|s)\log\pi_\theta(a|s).$$

Note the entropy is far less aggressive in penalizing small probabilities, in comparison to the log barrier, which is equivalent to the relative entropy. In particular, the entropy regularizer is always bounded between $0$ and $\log|\mathcal{A}|$, while the relative entropy (against the uniform distribution over actions), is bounded between $0$ and infinity, where it tends to infinity as probabilities tend to $0$. Here, it can be shown that the convergence rate is asymptotically $O(1\epsilon)$ (see Section 12.5) though it is unlikely that the convergence rate for this method is polynomial in other relevant quantities, including $|\mathcal{S}|$, $|\mathcal{A}|$, $1/(1-\gamma)$, and the distribution mismatch coefficient. The polynomial convergence rate using the log barrier (KL) regularizer crucially relies on the aggressive nature in which the relative entropy prevents small probabilities.

## 12.4 The Natural Policy Gradient

Observe that a policy constitutes a family of probability distributions $\{\pi_\theta(\cdot|s)|s \in \mathcal{S}\}$. We now consider a pre-conditioned gradient descent method based on this family of distributions. Recall that the Fisher information matrix

of a parameterized density $p_\theta(x)$ is defined as $\mathbb{E}_{x \sim p_\theta} \left[ \nabla \log p_\theta(x) \nabla \log p_\theta(x)^\top \right]$. Now we let us define $\mathcal{F}_\rho^\theta$ as an (average) Fisher information matrix on the family of distributions $\{\pi_\theta(\cdot|s)|s \in \mathcal{S}\}$ as follows:

$$\mathcal{F}_\rho^\theta := \mathbb{E}_{s \sim d_\rho^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ (\nabla \log \pi_\theta(a|s)) \nabla \log \pi_\theta(a|s)^\top \right] .$$

Note that the average is under the state-action visitation frequencies. The NPG algorithm performs gradient updates in the geometry induced by this matrix as follows:

$$\theta^{(t+1)} = \theta^{(t)} + \eta F_\rho(\theta^{(t)})^\dagger \nabla_\theta V^{(t)}(\rho), \tag{0.8}$$

where $M^\dagger$ denotes the Moore-Penrose pseudoinverse of the matrix $M$.

Throughout this section, we restrict to using the initial state distribution $\rho \in \Delta(\mathcal{S})$ in our update rule in (0.8) (so our optimization measure $\mu$ and the performance measure $\rho$ are identical). Also, we restrict attention to states $s \in \mathcal{S}$ reachable from $\rho$, since, without loss of generality, we can exclude states that are not reachable under this start state distribution[2].

For the softmax parameterization, this method takes a particularly convenient form; it can be viewed as a soft policy iteration update.

**Lemma 12.6.** *(Softmax NPG as soft policy iteration) For the softmax parameterization* (0.1)*, the NPG updates* (0.8) *take the form:*

$$\theta^{(t+1)} = \theta^{(t)} + \frac{\eta}{1-\gamma} A^{(t)} + \eta v \quad and \quad \pi^{(t+1)}(a|s) = \pi^{(t)}(a|s) \frac{\exp\left(\eta A^{(t)}(s,a)/(1-\gamma)\right)}{Z_t(s)},$$

*where $Z_t(s) = \sum_{a \in \mathcal{A}} \pi^{(t)}(a|s) \exp\left(\eta A^{(t)}(s,a)/(1-\gamma)\right)$. Here, $v$ is only a state dependent offset (i.e. $v_{s,a} = c_s$ for some $c_s \in \mathbb{R}$ for each state $s$), and, owing to the normalization $Z_t(s)$, $v$ has no effect on the update rule.*

It is important to note that while the ascent direction was derived using the gradient $\nabla_\theta V^{(t)}(\rho)$, which depends on $\rho$, the NPG update rule actually has no dependence on the measure $\rho$. Furthermore, there is no dependence on the state distribution $d_\rho^{(t)}$, which is due to the pseudoinverse of the Fisher information cancelling out the effect of the state distribution in NPG.

**Proof:** By definition of the Moore-Penrose pseudoinverse, we have that $(\mathcal{F}_\rho^\theta)^\dagger \nabla V^{\pi_\theta}(\rho) = w_\star$ if an only if $w_\star$ is the minimum norm solution of:

$$\min_w \|\nabla V^{\pi_\theta}(\rho) - \mathcal{F}_\rho^\theta w\|^2 .$$

Let us first evaluate $\mathcal{F}_\rho^\theta w$. For the softmax policy parameterization, Lemma 0.1 implies:

$$w^\top \nabla_\theta \log \pi_\theta(a|s) = w_{s,a} - \sum_{a' \in \mathcal{A}} w_{s,a'} \pi_\theta(a'|s) := w_{s,a} - \overline{w}_s$$

where $\overline{w}_s$ is not a function of $a$. This implies that:

$$\mathcal{F}_\rho^\theta w = \mathbb{E}_{s \sim d_\rho^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \nabla \log \pi_\theta(a|s) \left( w^\top \nabla_\theta \log \pi_\theta(a|s) \right) \right]$$

$$= \mathbb{E}_{s \sim d_\rho^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \nabla \log \pi_\theta(a|s) \left( w_{s,a} - \overline{w}_s \right) \right] = \mathbb{E}_{s \sim d_\rho^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ w_{s,a} \nabla \log \pi_\theta(a|s) \right],$$

where the last equality uses that $\overline{w}_s$ is not a function of $s$. Again using the functional form of derivative of the softmax policy parameterization, we have:

$$\left[ \mathcal{F}_\rho^\theta w \right]_{s',a'} = d^{\pi_\theta}(s') \pi_\theta(a'|s') \left( w_{s',a'} - \overline{w}_{s'} \right).$$

---

[2]Specifically, we restrict the MDP to the set of states $\{s \in \mathcal{S} : \exists \pi \text{ such that } d_\rho^\pi(s) > 0\}$.

This implies:

$$\|\nabla V^{\pi_\theta}(\rho) - \mathcal{F}_\rho^\theta w\|^2 = \sum_{s,a} \left( d^{\pi_\theta}(s)\pi_\theta(a|s) \left( \frac{1}{1-\gamma} A^{\pi_\theta}(s,a) - \left[ \mathcal{F}_\rho^\theta w \right]_{s,a} \right) \right)^2$$

$$= \sum_{s,a} \left( d^{\pi_\theta}(s)\pi_\theta(a|s) \left( \frac{1}{1-\gamma} A^{\pi_\theta}(s,a) - w_{s,a} - \sum_{a'\in\mathcal{A}} w_{s,a'}\pi_\theta(a'|s) \right) \right)^2.$$

Due to that $w = \frac{1}{1-\gamma} A^{\pi_\theta}$ leads to 0 error, the above implies that all 0 error solutions are of the form $w = \frac{1}{1-\gamma} A^{\pi_\theta} + v$, where $v$ is only a state dependent offset (i.e. $v_{s,a} = c_s$ for some $c_s \in \mathbb{R}$ for each state $s$). The first claim follows due to that the minimum norm solution is one of these solutions. The proof of the second claim now follows by the definition of the NPG update rule, along with that $v$ has no effect on the update rule due to the normalization constant $Z_t(s)$. ∎

We now see that this algorithm enjoys a dimension free convergence rate.

**Theorem 12.7** (Global convergence for NPG). *Suppose we run the NPG updates* (0.8) *using $\rho \in \Delta(\mathcal{S})$ and with $\theta^{(0)} = 0$. Fix $\eta > 0$. For all $T > 0$, we have:*

$$V^{(T)}(\rho) \geq V^*(\rho) - \frac{\log|\mathcal{A}|}{\eta T} - \frac{1}{(1-\gamma)^2 T}.$$

Note in the above the theorem that the NPG algorithm is directly applied to the performance measure $V^\pi(\rho)$, and the guarantees are also with respect to $\rho$. In particular, there is no distribution mismatch coefficient in the rate of convergence.

Now setting $\eta \geq (1-\gamma)^2 \log|\mathcal{A}|$, we see that NPG finds an $\epsilon$-optimal policy in a number of iterations that is at most:

$$T \leq \frac{2}{(1-\gamma)^2 \epsilon},$$

which has no dependence on the number of states or actions, despite the non-concavity of the underlying optimization problem.

The proof strategy we take borrows ideas from the classical multiplicative weights algorithm (see Section!12.5).

First, the following improvement lemma is helpful:

**Lemma 12.8** (Improvement lower bound for NPG). *For the iterates $\pi^{(t)}$ generated by the NPG updates* (0.8), *we have for all starting state distributions $\mu$*

$$V^{(t+1)}(\mu) - V^{(t)}(\mu) \geq \frac{(1-\gamma)}{\eta} \mathbb{E}_{s\sim\mu} \log Z_t(s) \geq 0.$$

**Proof:** First, let us show that $\log Z_t(s) \geq 0$. To see this, observe:

$$\log Z_t(s) = \log \sum_a \pi^{(t)}(a|s) \exp(\eta A^{(t)}(s,a)/(1-\gamma))$$

$$\geq \sum_a \pi^{(t)}(a|s) \log \exp(\eta A^{(t)}(s,a)/(1-\gamma)) = \frac{\eta}{1-\gamma} \sum_a \pi^{(t)}(a|s) A^{(t)}(s,a) = 0.$$

where we have used Jensen's inequality on the concave function $\log x$ and that $\sum_a \pi^{(t)}(a|s)A^{(t)}(s,a) = 0$. By the

116

performance difference lemma,

$$V^{(t+1)}(\mu) - V^{(t)}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{(t+1)}}\sum_a \pi^{(t+1)}(a|s)A^{(t)}(s,a)$$

$$= \frac{1}{\eta}\mathbb{E}_{s\sim d_\mu^{(t+1)}}\sum_a \pi^{(t+1)}(a|s)\log\frac{\pi^{(t+1)}(a|s)Z_t(s)}{\pi^{(t)}(a|s)}$$

$$= \frac{1}{\eta}\mathbb{E}_{s\sim d_\mu^{(t+1)}}\mathrm{KL}(\pi_s^{(t+1)}||\pi_s^{(t)}) + \frac{1}{\eta}\mathbb{E}_{s\sim d_\mu^{(t+1)}}\log Z_t(s)$$

$$\geq \frac{1}{\eta}\mathbb{E}_{s\sim d_\mu^{(t+1)}}\log Z_t(s) \geq \frac{1-\gamma}{\eta}\mathbb{E}_{s\sim\mu}\log Z_t(s),$$

where the last step uses that $d_\mu^{(t+1)} \geq (1-\gamma)\mu$ (by (0.6)) and that $\log Z_t(s) \geq 0$. ∎

With this lemma, we now prove Theorem 12.7.

**Proof:[of Theorem 12.7]** Since $\rho$ is fixed, we use $d^\star$ as shorthand for $d_\rho^{\pi^\star}$; we also use $\pi_s$ as shorthand for the vector of $\pi(\cdot|s)$. By the performance difference lemma (Lemma 1.16),

$$V^{\pi^\star}(\rho) - V^{(t)}(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\star}\sum_a \pi^\star(a|s)A^{(t)}(s,a)$$

$$= \frac{1}{\eta}\mathbb{E}_{s\sim d^\star}\sum_a \pi^\star(a|s)\log\frac{\pi^{(t+1)}(a|s)Z_t(s)}{\pi^{(t)}(a|s)}$$

$$= \frac{1}{\eta}\mathbb{E}_{s\sim d^\star}\left(\mathrm{KL}(\pi_s^\star||\pi_s^{(t)}) - \mathrm{KL}(\pi_s^\star||\pi_s^{(t+1)}) + \sum_a \pi^*(a|s)\log Z_t(s)\right)$$

$$= \frac{1}{\eta}\mathbb{E}_{s\sim d^\star}\left(\mathrm{KL}(\pi_s^\star||\pi_s^{(t)}) - \mathrm{KL}(\pi_s^\star||\pi_s^{(t+1)}) + \log Z_t(s)\right),$$

where we have used the closed form of our updates from Lemma 12.6 in the second step.

By applying Lemma 12.8 with $d^\star$ as the starting state distribution, we have:

$$\frac{1}{\eta}\mathbb{E}_{s\sim d^\star}\log Z_t(s) \leq \frac{1}{1-\gamma}\left(V^{(t+1)}(d^\star) - V^{(t)}(d^\star)\right)$$

which gives us a bound on $\mathbb{E}_{s\sim d^\star}\log Z_t(s)$.

Using the above equation and that $V^{(t+1)}(\rho) \geq V^{(t)}(\rho)$ (as $V^{(t+1)}(s) \geq V^{(t)}(s)$ for all states $s$ by Lemma 12.8), we have:

$$V^{\pi^\star}(\rho) - V^{(T-1)}(\rho) \leq \frac{1}{T}\sum_{t=0}^{T-1}(V^{\pi^\star}(\rho) - V^{(t)}(\rho))$$

$$= \frac{1}{\eta T}\sum_{t=0}^{T-1}\mathbb{E}_{s\sim d^\star}(\mathrm{KL}(\pi_s^\star||\pi_s^{(t)}) - \mathrm{KL}(\pi_s^\star||\pi_s^{(t+1)})) + \frac{1}{\eta T}\sum_{t=0}^{T-1}\mathbb{E}_{s\sim d^\star}\log Z_t(s)$$

$$\leq \frac{\mathbb{E}_{s\sim d^\star}\mathrm{KL}(\pi_s^\star||\pi^{(0)})}{\eta T} + \frac{1}{(1-\gamma)T}\sum_{t=0}^{T-1}\left(V^{(t+1)}(d^\star) - V^{(t)}(d^\star)\right)$$

$$= \frac{\mathbb{E}_{s\sim d^\star}\mathrm{KL}(\pi_s^\star||\pi^{(0)})}{\eta T} + \frac{V^{(T)}(d^\star) - V^{(0)}(d^\star)}{(1-\gamma)T}$$

$$\leq \frac{\log|\mathcal{A}|}{\eta T} + \frac{1}{(1-\gamma)^2 T}.$$

The proof is completed using that $V^{(T)}(\rho) \geq V^{(T-1)}(\rho)$. ∎

117

## 12.5 Bibliographic Remarks and Further Readings

The natural policy gradient method was originally presented in [Kakade, 2001]; a number of arguments for this method have been provided based on information geometry [Kakade, 2001, Bagnell and Schneider, 2003, Peters and Schaal, 2008].

The convergence rates in this chapter are largely derived from [Agarwal et al., 2020d]. The proof strategy for the NPG analysis has origins in the online regret framework in changing MDPs [Even-Dar et al., 2009], which would result in a worst rate in comparison to [Agarwal et al., 2020d]. This observation that the proof strategy from [Even-Dar et al., 2009] provided a convergence rate for the NPG was made in [Neu et al., 2017]. The faster NPG rate we present here is due to [Agarwal et al., 2020d]. The analysis of the MD-MPI algorithm [Geist et al., 2019] also implies a $O(1/T)$ rate for the NPG, though with worse dependencies on other parameters.

Building on ideas in [Agarwal et al., 2020d], [Mei et al., 2020] showed that, for the softmax policy class, both the gradient ascent and entropy regularized gradient ascent asymptotically converge at a $O(1/t)$; it is unlikely these methods are have finite rate which are polynomial in other quantities (such as the $|\mathcal{S}|$, $|\mathcal{A}|$, $1/(1-\gamma)$, and the distribution mismatch coefficient).

[Mnih et al., 2016] introduces the entropy regularizer (also see [Ahmed et al., 2019] for a more detailed empirical investigation).

# Chapter 13

# Function Approximation and the NPG

We now analyze the case of using parametric policy classes:

$$\Pi = \{\pi_\theta \mid \theta \in \mathbb{R}^d\},$$

where $\Pi$ may not contain all stochastic policies (and it may not even contain an optimal policy). In contrast with the tabular results in the previous sections, the policy classes that we are often interested in are not fully expressive, e.g. $d \ll |\mathcal{S}||\mathcal{A}|$ (indeed $|\mathcal{S}|$ or $|\mathcal{A}|$ need not even be finite for the results in this section); in this sense, we are in the regime of function approximation.

We focus on obtaining *agnostic* results, where we seek to do as well as the best policy in this class (or as well as some other comparator policy). While we are interested in a solution to the (unconstrained) policy optimization problem

$$\max_{\theta \in \mathbb{R}^d} V^{\pi_\theta}(\rho),$$

(for a given initial distribution $\rho$), we will see that optimization with respect to a different distribution will be helpful, just as in the tabular case,

We will consider variants of the NPG update rule (0.8):

$$\theta \leftarrow \theta + \eta F_\rho(\theta)^\dagger \nabla_\theta V^\theta(\rho). \tag{0.1}$$

Our analysis will leverage a close connection between the NPG update rule (0.8) with the notion of *compatible function approximation*. We start by formalizing this connection. The compatible function approximation error also provides a measure of the expressivity of our parameterization, allowing us to quantify the relevant notion of approximation error for the NPG algorithm.

The main results in this chapter establish the effectiveness of NPG updates where there is error both due to statistical estimation (where we may not use exact gradients) and approximation (due to using a parameterized function class). We will see an precise estimation/approximation decomposition based on the compatible function approximation error.

The presentation in this chapter largely follows the results in [Agarwal et al., 2020d].

## 13.1    Compatible function approximation and the NPG

We now introduce the notion of *compatible function approximation*, which both provides some intuition with regards to policy gradient methods and it will help us later on with regards to characterizing function approximation.

**Lemma 13.1** (Gradients and compatible function approximation). *Let $w^\star$ denote the following minimizer:*

$$w^\star \in \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \left( A^{\pi_\theta}(s,a) - w \cdot \nabla_\theta \log \pi_\theta(a|s) \right)^2 \right],$$

*where the squared error above is referred to as the* compatible function approximation. *Denote the best linear predictor of $A^{\pi_\theta}(s,a)$ using $\nabla_\theta \log \pi_\theta(a|s)$ by $\widehat{A}^{\pi_\theta}(s,a)$, i.e.*

$$\widehat{A}^{\pi_\theta}(s,a) := w^\star \cdot \nabla_\theta \log \pi_\theta(a|s).$$

*We have that:*

$$\nabla_\theta V^{\pi_\theta}(\mu) \;=\; \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \nabla_\theta \log \pi_\theta(a|s) \widehat{A}^{\pi_\theta}(s,a) \right].$$

**Proof:** The first order optimality conditions for $w^\star$ imply

$$\mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \left( A^{\pi_\theta}(s,a) - w^\star \cdot \nabla_\theta \log \pi_\theta(a|s) \right) \nabla_\theta \log \pi_\theta(a|s) \right] = 0 \tag{0.2}$$

Rearranging and using the definition of $\widehat{A}^{\pi_\theta}(s,a)$,

$$\begin{aligned}
\nabla_\theta V^{\pi_\theta}(\mu) &\;=\; \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(s,a) \right] \\
&\;=\; \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \nabla_\theta \log \pi_\theta(a|s) \widehat{A}^{\pi_\theta}(s,a) \right],
\end{aligned}$$

which completes the proof. ∎

The next lemma shows that the weight vector above is precisely the NPG ascent direction. Precisely,

**Lemma 13.2.** *We have that:*

$$F_\rho(\theta)^\dagger \nabla_\theta V^\theta(\rho) = \frac{1}{1-\gamma} w^\star, \tag{0.3}$$

*where $w^\star$ is a minimizer of the following regression problem:*

$$w^\star \in \operatorname{argmin}_w \mathbb{E}_{s \sim d_\rho^{\pi_\theta}, a \sim \pi_\theta(\cdot|s)} \left[ (w^\top \nabla_\theta \log \pi_\theta(\cdot|s) - A^{\pi_\theta}(s,a))^2 \right].$$

**Proof:** The above is a straightforward consequence of the first order optimality conditions (see Equation 0.2). Specifically, Equation 0.2, along with the advantage expression for the policy gradient (see Theorem 11.4), imply that $w^\star$ must satisfy:

$$\nabla_\theta V^\theta(\rho) = \frac{1}{1-\gamma} F_\rho(\theta) w^\star$$

which completes the proof. ∎

This lemma implies that we might write the NPG update rule as:

$$\theta \leftarrow \theta + \frac{\eta}{1-\gamma} w^\star.$$

where $w^\star$ is minimizer of the compatible function approximation error (which depends on $\theta$.

The above regression problem can be viewed as "compatible" function approximation: we are approximating $A^{\pi_\theta}(s,a)$ using the $\nabla_\theta \log \pi_\theta(\cdot|s)$ as features. We also consider a variant of the above update rule, $Q$-NPG, where instead of using advantages in the above regression we use the $Q$-values. This viewpoint provides a methodology for approximate updates, where we can solve the relevant regression problems with samples.

## 13.2 Examples: NPG and $Q$-NPG

In practice, the most common policy classes are of the form:

$$\Pi = \left\{ \pi_\theta(a|s) = \frac{\exp\left(f_\theta(s,a)\right)}{\sum_{a' \in \mathcal{A}} \exp\left(f_\theta(s,a')\right)} \,\Big|\, \theta \in \mathbb{R}^d \right\}, \tag{0.4}$$

where $f_\theta$ is a differentiable function. For example, the tabular softmax policy class is one where $f_\theta(s,a) = \theta_{s,a}$. Typically, $f_\theta$ is either a linear function or a neural network. Let us consider the NPG algorithm, and a variant $Q$-NPG, in each of these two cases.

### 13.2.1 Log-linear Policy Classes and Soft Policy Iteration

For any state-action pair $(s,a)$, suppose we have a feature mapping $\phi_{s,a} \in \mathbb{R}^d$. Each policy in the log-linear policy class is of the form:

$$\pi_\theta(a|s) = \frac{\exp(\theta \cdot \phi_{s,a})}{\sum_{a' \in \mathcal{A}} \exp(\theta \cdot \phi_{s,a'})},$$

with $\theta \in \mathbb{R}^d$. Here, we can take $f_\theta(s,a) = \theta \cdot \phi_{s,a}$.

With regards to compatible function approximation for the log-linear policy class, we have:

$$\nabla_\theta \log \pi_\theta(a|s) = \overline{\phi}_{s,a}^{\,\theta}, \text{where} \quad \overline{\phi}_{s,a}^{\,\theta} = \phi_{s,a} - \mathbb{E}_{a' \sim \pi_\theta(\cdot|s)}[\phi_{s,a'}],$$

that is, $\overline{\phi}_{s,a}^{\,\theta}$ is the centered version of $\phi_{s,a}$. With some abuse of notation, we accordingly also define $\overline{\phi}^\pi$ for any policy $\pi$. Here, using (0.3), the NPG update rule (0.1) is equivalent to:

$$\text{NPG:} \quad \theta \leftarrow \theta + \eta w_\star, \qquad w_\star \in \mathrm{argmin}_w \, \mathbb{E}_{s \sim d_\rho^{\pi_\theta}, a \sim \pi_\theta(\cdot|s)} \left[ \left( A^{\pi_\theta}(s,a) - w \cdot \overline{\phi}_{s,a}^{\,\theta} \right)^2 \right].$$

(We have rescaled the learning rate $\eta$ in comparison to (0.1)). Note that we recompute $w_\star$ for every update of $\theta$. Here, the compatible function approximation error measures the expressivity of our parameterization in how well linear functions of the parameterization can capture the policy's advantage function.

We also consider a variant of the NPG update rule (0.1), termed *Q-NPG*, where:

$$Q\text{-NPG:} \quad \theta \leftarrow \theta + \eta w_\star, \qquad w_\star \in \mathrm{argmin}_w \, \mathbb{E}_{s \sim d_\rho^{\pi_\theta}, a \sim \pi_\theta(\cdot|s)} \left[ \left( Q^{\pi_\theta}(s,a) - w \cdot \phi_{s,a} \right)^2 \right].$$

Note we do not center the features for $Q$-NPG; observe that $Q^\pi(s,a)$ is also not 0 in expectation under $\pi(\cdot|s)$, unlike the advantage function.

(NPG/$Q$-NPG and Soft-Policy Iteration) We now see how we can view both NPG and $Q$-NPG as an incremental (soft) version of policy iteration, just as in Lemma 12.6 for the tabular case. Rather than writing the update rule in terms of the parameter $\theta$, we can write an equivalent update rule directly in terms of the (log-linear) policy $\pi$:

$$\text{NPG:} \quad \pi(a|s) \leftarrow \pi(a|s) \exp(w_\star \cdot \phi_{s,a})/Z_s, \quad w_\star \in \mathrm{argmin}_w \, \mathbb{E}_{s \sim d_\rho^\pi, a \sim \pi(\cdot|s)} \left[ \left( A^\pi(s,a) - w \cdot \overline{\phi}_{s,a}^{\,\pi} \right)^2 \right],$$

where $Z_s$ is normalization constant. While the policy update uses the original features $\phi$ instead of $\overline{\phi}^\pi$, whereas the quadratic error minimization is terms of the centered features $\overline{\phi}^\pi$, this distinction is not relevant due to that we may also instead use $\overline{\phi}^\pi$ (in the policy update) which would result in an equivalent update; the normalization makes the update invariant to (constant) translations of the features. Similarly, an equivalent update for $Q$-NPG, where we update $\pi$ directly rather than $\theta$, is:

$$Q\text{-NPG:} \quad \pi(a|s) \leftarrow \pi(a|s) \exp(w_\star \cdot \phi_{s,a})/Z_s, \quad w_\star \in \mathrm{argmin}_w \, \mathbb{E}_{s \sim d_\rho^\pi, a \sim \pi(\cdot|s)} \left[ \left( Q^\pi(s,a) - w \cdot \phi_{s,a} \right)^2 \right].$$

(On the equivalence of NPG and $Q$-NPG) If it is the case that the compatible function approximation error is $0$, then it straightforward to verify that the NPG and $Q$-NPG are equivalent algorithms, in that their corresponding policy updates will be equivalent to each other.

### 13.2.2 Neural Policy Classes

Now suppose $f_\theta(s, a)$ is a neural network parameterized by $\theta \in \mathbb{R}^d$, where the policy class $\Pi$ is of form in (0.4). Observe:

$$\nabla_\theta \log \pi_\theta(a|s) = g_\theta(s, a), \text{where} \quad g_\theta(s, a) = \nabla_\theta f_\theta(s, a) - \mathbb{E}_{a' \sim \pi_\theta(\cdot|s)}[\nabla_\theta f_\theta(s, a')],$$

and, using (0.3), the NPG update rule (0.1) is equivalent to:

$$\text{NPG:} \quad \theta \leftarrow \theta + \eta w_\star, \qquad w_\star \in \operatorname{argmin}_w \mathbb{E}_{s \sim d_\rho^{\pi_\theta}, a \sim \pi_\theta(\cdot|s)} \left[ \left( A^{\pi_\theta}(s, a) - w \cdot g_\theta(s, a) \right)^2 \right]$$

(Again, we have rescaled the learning rate $\eta$ in comparison to (0.1)).

The $Q$-NPG variant of this update rule is:

$$\text{$Q$-NPG:} \quad \theta \leftarrow \theta + \eta w_\star, \qquad w_\star \in \operatorname{argmin}_w \mathbb{E}_{s \sim d_\rho^{\pi_\theta}, a \sim \pi_\theta(\cdot|s)} \left[ \left( Q^{\pi_\theta}(s, a) - w \cdot \nabla_\theta f_\theta(s, a) \right)^2 \right].$$

## 13.3 The NPG "Regret Lemma"

It is helpful for us to consider NPG more abstractly, as an update rule of the form

$$\theta^{(t+1)} = \theta^{(t)} + \eta w^{(t)}. \tag{0.5}$$

We will now provide a lemma where $w^{(t)}$ is an *arbitrary* (bounded) sequence, which will be helpful when specialized.

Recall a function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be $\beta$-smooth if for all $x, x' \in \mathbb{R}^d$:

$$\|\nabla f(x) - \nabla f(x')\|_2 \leq \beta \|x - x'\|_2,$$

and, due to Taylor's theorem, recall that this implies:

$$\left| f(x') - f(x) - \nabla f(x) \cdot (x' - x) \right| \leq \frac{\beta}{2} \|x' - x\|_2^2. \tag{0.6}$$

The following analysis of NPG is draws close connections to the mirror-descent approach used in online learning (see Section 13.6), which motivates us to refer to it as a "regret lemma".

**Lemma 13.3.** *(NPG Regret Lemma) Fix a comparison policy $\widetilde{\pi}$ and a state distribution $\rho$. Assume for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ that $\log \pi_\theta(a|s)$ is a $\beta$-smooth function of $\theta$. Consider the update rule (0.5), where $\pi^{(0)}$ is the uniform distribution (for all states) and where the sequence of weights $w^{(0)}, \ldots, w^{(T)}$, satisfies $\|w^{(t)}\|_2 \leq W$ (but is otherwise arbitrary). Define:*

$$\text{err}_t = \mathbb{E}_{s \sim \widetilde{d}} \mathbb{E}_{a \sim \widetilde{\pi}(\cdot|s)} \left[ A^{(t)}(s, a) - w^{(t)} \cdot \nabla_\theta \log \pi^{(t)}(a|s) \right].$$

*We have that:*

$$\min_{t < T} \left\{ V^{\widetilde{\pi}}(\rho) - V^{(t)}(\rho) \right\} \leq \frac{1}{1 - \gamma} \left( \frac{\log |\mathcal{A}|}{\eta T} + \frac{\eta \beta W^2}{2} + \frac{1}{T} \sum_{t=0}^{T-1} \text{err}_t \right).$$

122

This lemma is the key tool in understanding the role of function approximation of various algorithms. We will consider one example in detail with regards to the log-linear policy class (from Example 11.2).

Note that when $\mathrm{err}_t = 0$, as will be the case with the (tabular) softmax policy class with exact gradients, we obtain a convergence rate of $O(\sqrt{1/T})$ using a learning rate of $\eta = O(\sqrt{1/T})$. Note that this is slower than the faster rate of $O(1/T)$, provided in Theorem 12.7. Obtaining a bound that leads to a faster rate in the setting with errors requires more complicated dependencies on $\mathrm{err}_t$ than those stated above.

**Proof:** By smoothness (see (0.6)),

$$
\begin{aligned}
\log \frac{\pi^{(t+1)}(a|s)}{\pi^{(t)}(a|s)} &\geq \nabla_\theta \log \pi^{(t)}(a|s) \cdot \left(\theta^{(t+1)} - \theta^{(t)}\right) - \frac{\beta}{2}\|\theta^{(t+1)} - \theta^{(t)}\|_2^2 \\
&= \eta \nabla_\theta \log \pi^{(t)}(a|s) \cdot w^{(t)} - \eta^2 \frac{\beta}{2}\|w^{(t)}\|_2^2.
\end{aligned}
$$

We use $\widetilde{d}$ as shorthand for $d_\rho^{\widetilde{\pi}}$ (note $\rho$ and $\widetilde{\pi}$ are fixed); for any policy $\pi$, we also use $\pi_s$ as shorthand for the vector $\pi(\cdot|s)$. Using the performance difference lemma (Lemma 1.16),

$$
\begin{aligned}
&\mathbb{E}_{s \sim \widetilde{d}}\left(\mathrm{KL}(\widetilde{\pi}_s || \pi_s^{(t)}) - \mathrm{KL}(\widetilde{\pi}_s || \pi_s^{(t+1)})\right) \\
&= \mathbb{E}_{s \sim \widetilde{d}} \mathbb{E}_{a \sim \widetilde{\pi}(\cdot|s)}\left[\log \frac{\pi^{(t+1)}(a|s)}{\pi^{(t)}(a|s)}\right] \\
&\geq \eta \mathbb{E}_{s \sim \widetilde{d}} \mathbb{E}_{a \sim \widetilde{\pi}(\cdot|s)}\left[\nabla_\theta \log \pi^{(t)}(a|s) \cdot w^{(t)}\right] - \eta^2 \frac{\beta}{2}\|w^{(t)}\|_2^2 \qquad \text{(using previous display)} \\
&= \eta \mathbb{E}_{s \sim \widetilde{d}} \mathbb{E}_{a \sim \widetilde{\pi}(\cdot|s)}\left[A^{(t)}(s,a)\right] - \eta^2 \frac{\beta}{2}\|w^{(t)}\|_2^2 \\
&\quad + \eta \mathbb{E}_{s \sim \widetilde{d}} \mathbb{E}_{a \sim \widetilde{\pi}(\cdot|s)}\left[\nabla_\theta \log \pi^{(t)}(a|s) \cdot w^{(t)} - A^{(t)}(s,a)\right] \\
&= (1-\gamma)\eta\left(V^{\widetilde{\pi}}(\rho) - V^{(t)}(\rho)\right) - \eta^2 \frac{\beta}{2}\|w^{(t)}\|_2^2 - \eta \, \mathrm{err}_t
\end{aligned}
$$

Rearranging, we have:

$$
V^{\widetilde{\pi}}(\rho) - V^{(t)}(\rho) \leq \frac{1}{1-\gamma}\left(\frac{1}{\eta}\mathbb{E}_{s \sim \widetilde{d}}\left(\mathrm{KL}(\widetilde{\pi}_s || \pi_s^{(t)}) - \mathrm{KL}(\widetilde{\pi}_s || \pi_s^{(t+1)})\right) + \frac{\eta\beta}{2}W^2 + \mathrm{err}_t\right)
$$

Proceeding,

$$
\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}(V^{\widetilde{\pi}}(\rho) - V^{(t)}(\rho)) &\leq \frac{1}{\eta T(1-\gamma)}\sum_{t=0}^{T-1}\mathbb{E}_{s \sim \widetilde{d}}(\mathrm{KL}(\widetilde{\pi}_s || \pi_s^{(t)}) - \mathrm{KL}(\widetilde{\pi}_s || \pi_s^{(t+1)})) \\
&\quad + \frac{1}{T(1-\gamma)}\sum_{t=0}^{T-1}\left(\frac{\eta\beta W^2}{2} + \mathrm{err}_t\right) \\
&\leq \frac{\mathbb{E}_{s \sim \widetilde{d}}\mathrm{KL}(\widetilde{\pi}_s || \pi^{(0)})}{\eta T(1-\gamma)} + \frac{\eta\beta W^2}{2(1-\gamma)} + \frac{1}{T(1-\gamma)}\sum_{t=0}^{T-1}\mathrm{err}_t \\
&\leq \frac{\log|\mathcal{A}|}{\eta T(1-\gamma)} + \frac{\eta\beta W^2}{2(1-\gamma)} + \frac{1}{T(1-\gamma)}\sum_{t=0}^{T-1}\mathrm{err}_t,
\end{aligned}
$$

which completes the proof. ∎

## 13.4 $Q$-NPG: Performance Bounds for Log-Linear Policies

For a state-action distribution $\upsilon$, define:

$$L(w; \theta, \upsilon) := \mathbb{E}_{s,a \sim \upsilon}\left[\left(Q^{\pi_\theta}(s,a) - w \cdot \phi_{s,a}\right)^2\right].$$

The iterates of the $Q$-NPG algorithm can be viewed as minimizing this loss under some (changing) distribution $\upsilon$.

We now specify an approximate version of $Q$-NPG. It is helpful to consider a slightly more general version of the algorithm in the previous section, where instead of optimizing under a starting state distribution $\rho$, we have a different starting *state-action* distribution $\nu$. The motivation for this is similar in spirit to our log barrier regularization: we seek to maintain exploration (and estimation) over the action space even if the current policy does not have coverage over the action space.

Analogous to the definition of the state-action visitation measure $d^\pi_{s_0}$ (see Equation 0.5), we define another state-action visitation measure, $d^\pi_{s_0,a_0}$, as follows:

$$d^\pi_{s_0,a_0}(s,a) := (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathrm{Pr}^\pi(s_t = s, a_t = a | s_0, a_0) \tag{0.7}$$

where $\mathrm{Pr}^\pi(s_t = s, a_t = a | s_0, a_0)$ is the probability that $s_t = s$ and $a_t = a$, after starting at state $s_0$, taking action $a_0$, and following $\pi$ thereafter. Again, we overload notation and write:

$$d^\pi_\nu(s,a) = \mathbb{E}_{s_0,a_0 \sim \nu}[d_{s_0,a_0}]$$

where $\nu$ is a distribution over $\mathcal{S} \times \mathcal{A}$.

$Q$-NPG will be defined with respect to the *on-policy* state action measure starting with $s_0, a_0 \sim \nu$. As per our convention, we define

$$d^{(t)} := d^{\pi^{(t)}}_\nu.$$

The approximate version of this algorithm is:

$$\text{Approx. Q-NPG: } \theta^{(t+1)} = \theta^{(t)} + \eta w^{(t)}, \qquad w^{(t)} \approx \mathrm{argmin}_{\|w\|_2 \leq W} L(w; \theta^{(t)}, d^{(t)}), \tag{0.8}$$

where the above update rule also permits us to constrain the norm of the update direction $w^{(t)}$ (alternatively, we could use $\ell_2$ regularization as is also common in practice). The exact minimizer is denoted as:

$$w^{(t)}_\star \in \mathrm{argmin}_{\|w\|_2 \leq W} L(w; \theta^{(t)}, d^{(t)}).$$

Note that $w^{(t)}_\star$ depends on the current parameter $\theta^{(t)}$.

Our analysis will take into account both the *excess risk* (often also referred to as estimation error) and the *approximation error*. The standard approximation-estimation error decomposition is as follows:

$$L(w^{(t)}; \theta^{(t)}, d^{(t)}) \quad = \quad \underbrace{L(w^{(t)}; \theta^{(t)}, d^{(t)}) - L(w^{(t)}_\star; \theta^{(t)}, d^{(t)})}_{\text{Excess risk}} + \underbrace{L(w^{(t)}_\star; \theta^{(t)}, d^{(t)})}_{\text{Approximation error}}$$

Using a sample based approach, we would expect $\epsilon_{\text{stat}} = O(1/\sqrt{N})$ or better, where $N$ is the number of samples used to estimate. $w^{(t)}_\star$ In constrast, the approximation error is due to modeling error, and does not tend to 0 with more samples. We will see how these two errors have strikingly different impact on our final performance bound.

Note that we have already considered two cases where $\epsilon_{\text{approx}} = 0$. For the tabular softmax policy class, it is immediate that $\epsilon_{\text{approx}} = 0$. A more interesting example (where the state and action space could be infinite) is provided by

the linear parameterized MDP model from Chapter 8. Here, provided that we use the log-linear policy class (see Section 13.2.1) with features corresponding to the linear MDP features, it is straightforward to see that $\epsilon_{\text{approx}} = 0$ for this log-linear policy class. More generally, we will see the effect of model misspecification in our performance bounds.

We make the following assumption on these errors:

**Assumption 13.4** (Approximation/estimation error bounds). Let $w^{(0)}, w^{(1)}, \ldots w^{(T-1)}$ be the sequence of iterates used by the $Q$-NPG algorithm Suppose the following holds for all $t < T$:

1. (*Excess risk*) Suppose the estimation error is bounded as follows:

$$L(w^{(t)}; \theta^{(t)}, d^{(t)}) - L(w_\star^{(t)}; \theta^{(t)}, d^{(t)}) \leq \epsilon_{\text{stat}}$$

2. (*Approximation error*) Suppose the approximation error is bounded as follows:

$$L(w_\star^{(t)}; \theta^{(t)}, d^{(t)}) \leq \epsilon_{\text{approx}}.$$

We will also see how, with regards to our estimation error, we will need a far more mild notion of coverage. Here, with respect to any state-action distribution $\upsilon$, define:

$$\Sigma_\upsilon = \mathbb{E}_{s,a\sim\upsilon} \left[ \phi_{s,a} \phi_{s,a}^\top \right].$$

We make a the following conditioning assumption:

**Assumption 13.5** (Relative condition number). Fix a state distribution $\rho$ (this will be what ultimately be the performance measure that we seek to optimize). Consider an arbitrary comparator policy $\pi^\star$ (not necessarily an optimal policy). With respect to $\pi^\star$, define the state-action measure $d^\star$ as

$$d^\star(s, a) = d_\rho^{\pi^\star}(s) \cdot \text{Unif}_{\mathcal{A}}(a)$$

i.e. $d^\star$ samples states from the comparators state visitation measure, $d_\rho^{\pi^\star}$ and actions from the uniform distribution. Define

$$\sup_{w\in\mathbb{R}^d} \frac{w^\top \Sigma_{d^\star} w}{w^\top \Sigma_\nu w} = \kappa,$$

and assume that $\kappa$ is finite.

We later discuss why it is reasonable to expect that $\kappa$ is not a quantity related to the size of the state space.

The main result of this chapter shows how the approximation error, the excess risk, and the conditioning, determine the final performance.

**Theorem 13.6.** *Fix a state distribution $\rho$; a state-action distribution $\nu$; an arbitrary comparator policy $\pi^\star$ (not necessarily an optimal policy). Suppose Assumption 13.5 holds with respect to these choices and that $\|\phi_{s,a}\|_2 \leq B$ for all $s, a$. Suppose the Q-NPG update rule (in (0.8)) starts with $\theta^{(0)} = 0$, $\eta = \sqrt{2\log|\mathcal{A}|/(B^2 W^2 T)}$, and the (random) sequence of iterates satisfies Assumption 13.4. We have that:*

$$\mathbb{E}\left[ \min_{t<T} \left\{ V^{\pi^\star}(\rho) - V^{(t)}(\rho) \right\} \right] \leq \frac{BW}{1-\gamma} \sqrt{\frac{2\log|\mathcal{A}|}{T}} + \sqrt{\frac{4|\mathcal{A}|}{(1-\gamma)^3} \left( \kappa \cdot \epsilon_{\text{stat}} + \left\| \frac{d^\star}{\nu} \right\|_\infty \cdot \epsilon_{\text{approx}} \right)}.$$

Note when $\epsilon_{\text{approx}} = 0$, our convergence rate is $O(\sqrt{1/T})$ plus a term that depends on the excess risk; hence, provided we obtain enough samples, then $\epsilon_{\text{stat}}$ will also tend to $0$, and we will be competitive with the comparison policy $\pi^\star$.

The above also shows the striking difference between the effects of estimation error and approximation error. A few remarks are now in order.

**Transfer learning, distribution shift, and the approximation error.** In large scale problems, the worst case distribution mismatch factor $\left\| \frac{d^\star}{\nu} \right\|_\infty$ is unlikely to be small. However, this factor is ultimately due to transfer learning. Our approximation error is with respect to the fitting distribution $d^{(t)}$, where we assume that $L(w_\star^{(t)}; \theta^{(t)}, d^{(t)}) \le \epsilon_{\text{approx}}$. As the proof will show, the relevant notion of approximation error will be $L(w_\star^{(t)}; \theta^{(t)}, d^\star)$, where $d^\star$ is the *fixed* comparators measure. In others words, to get a good performance bound we need to successfully have low transfer learning error to the fixed measure $d^\star$. Furthermore, in many modern machine learning applications, this error is often is favorable, in that it is substantially better than worst case theory might suggest.

See Section 13.6 for further remarks on this point.

**Dimension dependence in $\kappa$ and the importance of $\nu$.** It is reasonable to think about $\kappa$ as being dimension dependent (or worse), but it is not necessarily related to the size of the state space. For example, if $\|\phi_{s,a}\|_2 \le B$, then $\kappa \le \frac{B^2}{\sigma_{\min}(\mathbb{E}_{s,a \sim \nu}[\phi_{s,a} \phi_{s,a}^\top])}$ though this bound may be pessimistic. Here, we also see the importance of choice of $\nu$ in having a small (relative) condition number; in particular, this is the motivation for considering the generalization which allows for a starting state-action distribution $\nu$ vs. just a starting state distribution $\mu$ (as we did in the tabular case). Roughly speaking, we desire a $\nu$ which provides good coverage over the features. As the following lemma shows, there always exists a universal distribution $\nu$, which can be constructed only with knowledge of the feature set (without knowledge of $d^\star$), such that $\kappa \le d$.

**Lemma 13.7.** *($\kappa \le d$ is always possible) Let $\Phi = \{\phi(s,a)|(s,a) \in \mathcal{S} \times \mathcal{A}\} \subset \mathbb{R}^d$ and suppose $\Phi$ is a compact set. There always exists a state-action distribution $\nu$, which is supported on at most $d^2$ state-action pairs and which can be constructed only with knowledge of $\Phi$ (without knowledge of the MDP or $d^\star$), such that:*

$$\kappa \le d.$$

**Proof:** The distribution can be found through constructing the minimal volume ellipsoid containing $\Phi$, i.e. the Löwner-John ellipsoid. **To be added...** ∎

**Direct policy optimization vs. approximate value function programming methods** Part of the reason for the success of the direct policy optimization approaches is to due their more mild dependence on the approximation error. Here, our theoretical analysis has a dependence on a distribution mismatch coefficient, $\left\| \frac{d^\star}{\nu} \right\|_\infty$, while approximate value function methods have even worse dependencies. See Chapter 4. As discussed earlier and as can be seen in the regret lemma (Lemma 13.3), the distribution mismatch coefficient is due to that the relevant error for NPG is a transfer error notion to a *fixed* comparator distribution, while approximate value function methods have more stringent conditions where the error has to be small under, essentially, the distribution of *any* other policy.

### 13.4.1 Analysis

**Proof:** (of Theorem 13.6) For the log-linear policy class, due to that the feature mapping $\phi$ satisfies $\|\phi_{s,a}\|_2 \le B$, then it is not difficult to verify that $\log \pi_\theta(a|s)$ is a $B^2$-smooth function. Using this and the NPG regret lemma (Lemma 13.3), we have:

$$\min_{t < T} \left\{ V^{\pi^\star}(\rho) - V^{(t)}(\rho) \right\} \le \frac{BW}{1 - \gamma} \sqrt{\frac{2 \log |\mathcal{A}|}{T}} + \frac{1}{(1 - \gamma)T} \sum_{t=0}^{T-1} \text{err}_t.$$

where we have used our setting of $\eta$.

We make the following decomposition of $\mathrm{err}_t$:

$$\mathrm{err}_t = \mathbb{E}_{s \sim d_\rho^\star, a \sim \pi^\star(\cdot|s)}\left[A^{(t)}(s,a) - w_\star^{(t)} \cdot \nabla_\theta \log \pi^{(t)}(a|s)\right]$$
$$+ \mathbb{E}_{s \sim d_\rho^\star, a \sim \pi^\star(\cdot|s)}\left[\left(w_\star^{(t)} - w^{(t)}\right) \cdot \nabla_\theta \log \pi^{(t)}(a|s)\right].$$

For the first term, using that $\nabla_\theta \log \pi_\theta(a|s) = \phi_{s,a} - \mathbb{E}_{a' \sim \pi_\theta(\cdot|s)}[\phi_{s,a'}]$ (see Section 13.2.1), we have:

$$\mathbb{E}_{s \sim d_\rho^\star, a \sim \pi^\star(\cdot|s)}\left[A^{(t)}(s,a) - w_\star^{(t)} \cdot \nabla_\theta \log \pi^{(t)}(a|s)\right]$$
$$= \mathbb{E}_{s \sim d_\rho^\star, a \sim \pi^\star(\cdot|s)}\left[Q^{(t)}(s,a) - w_\star^{(t)} \cdot \phi_{s,a}\right] - \mathbb{E}_{s \sim d_\rho^\star, a' \sim \pi^{(t)}(\cdot|s)}\left[Q^{(t)}(s,a') - w_\star^{(t)} \cdot \phi_{s,a'}\right]$$
$$\leq \sqrt{\mathbb{E}_{s \sim d_\rho^\star, a \sim \pi^\star(\cdot|s)}\left(Q^{(t)}(s,a) - w_\star^{(t)} \cdot \phi_{s,a}\right)^2} + \sqrt{\mathbb{E}_{s \sim d_\rho^\star, a' \sim \pi^{(t)}(\cdot|s)}\left(Q^{(t)}(s,a') - w_\star^{(t)} \cdot \phi_{s,a'}\right)^2}$$
$$\leq 2\sqrt{|\mathcal{A}|\mathbb{E}_{s \sim d_\rho^\star, a \sim \mathrm{Unif}_\mathcal{A}}\left[\left(Q^{(t)}(s,a) - w_\star^{(t)} \cdot \phi_{s,a}\right)^2\right]} = 2\sqrt{|\mathcal{A}|L(w_\star^{(t)}; \theta^{(t)}, d^\star)}.$$

where we have used the definition of $d^\star$ and $L(w_\star^{(t)}; \theta^{(t)}, d^\star)$ in the last step. Using following crude upper bound,

$$L(w_\star^{(t)}; \theta^{(t)}, d^\star) \leq \left\|\frac{d^\star}{d^{(t)}}\right\|_\infty L(w_\star^{(t)}; \theta^{(t)}, d^{(t)}) \leq \frac{1}{1-\gamma}\left\|\frac{d^\star}{\nu}\right\|_\infty L(w_\star^{(t)}; \theta^{(t)}, d^{(t)}),$$

(where the last step uses the defintion of $d^{(t)}$, see Equation 0.7), we have that:

$$\mathbb{E}_{s \sim d_\rho^\star, a \sim \pi^\star(\cdot|s)}\left[A^{(t)}(s,a) - w_\star^{(t)} \cdot \nabla_\theta \log \pi^{(t)}(a|s)\right] \leq 2\sqrt{\frac{|\mathcal{A}|}{1-\gamma}\left\|\frac{d^\star}{\nu}\right\|_\infty L(w_\star^{(t)}; \theta^{(t)}, d^{(t)})}. \tag{0.9}$$

For the second term, let us now show that:

$$\mathbb{E}_{s \sim d_\rho^\star, a \sim \pi^\star(\cdot|s)}\left[\left(w_\star^{(t)} - w^{(t)}\right) \cdot \nabla_\theta \log \pi^{(t)}(a|s)\right]$$
$$\leq 2\sqrt{\frac{|\mathcal{A}|\kappa}{1-\gamma}\left(L(w^{(t)}; \theta^{(t)}, d^{(t)}) - L(w_\star^{(t)}; \theta^{(t)}, d^{(t)})\right)} \tag{0.10}$$

To see this, first observe that a similar argument to the above leads to:

$$\mathbb{E}_{s \sim d_\rho^\star, a \sim \pi^\star(\cdot|s)}\left[\left(w_\star^{(t)} - w^{(t)}\right) \cdot \nabla_\theta \log \pi^{(t)}(a|s)\right]$$
$$= \mathbb{E}_{s \sim d_\rho^\star, a \sim \pi^\star(\cdot|s)}\left[\left(w_\star^{(t)} - w^{(t)}\right) \cdot \phi_{s,a}\right] - \mathbb{E}_{s \sim d_\rho^\star, a' \sim \pi^{(t)}(\cdot|s)}\left[\left(w_\star^{(t)} - w^{(t)}\right) \cdot \phi_{s,a'}\right]$$
$$\leq 2\sqrt{|\mathcal{A}|\mathbb{E}_{s,a \sim d^\star}\left[\left(\left(w_\star^{(t)} - w^{(t)}\right) \cdot \phi_{s,a}\right)^2\right]} = 2\sqrt{|\mathcal{A}| \cdot \|w_\star^{(t)} - w^{(t)}\|_{\Sigma_{d^\star}}^2},$$

where we use the notation $\|x\|_M^2 := x^\top M x$ for a matrix $M$ and a vector $x$. From the definition of $\kappa$,

$$\|w_\star^{(t)} - w^{(t)}\|_{\Sigma_{d^\star}}^2 \leq \kappa \|w_\star^{(t)} - w^{(t)}\|_{\Sigma_\nu}^2 \leq \frac{\kappa}{1-\gamma}\|w_\star^{(t)} - w^{(t)}\|_{\Sigma_{d^{(t)}}}^2$$

using that $(1-\gamma)\nu \leq d_\nu^{\pi^{(t)}}$ (see (0.7)). Due to that $w_\star^{(t)}$ minimizes $L(w; \theta^{(t)}, d^{(t)})$ over the set $\mathcal{W} := \{w : \|w\|_2 \leq W\}$, for any $w \in \mathcal{W}$ the first-order optimality conditions for $w_\star^{(t)}$ imply that:

$$(w - w_\star^{(t)}) \cdot \nabla L(w_\star^{(t)}; \theta^{(t)}, d^{(t)}) \geq 0.$$

Therefore, for any $w \in \mathcal{W}$,

$$L(w; \theta^{(t)}, d^{(t)}) - L(w_\star^{(t)}; \theta^{(t)}, d^{(t)})$$
$$= \mathbb{E}_{s,a \sim d^{(t)}} \left[ \left( w \cdot \phi(s,a) - w_\star \cdot \phi(s,a) + w_\star \cdot \phi(s,a) - Q^{(t)}(s,a) \right)^2 \right] - L(w_\star^{(t)}; \theta^{(t)}, d^{(t)})$$
$$= \mathbb{E}_{s,a \sim d^{(t)}} \left[ \left( w \cdot \phi(s,a) - w_\star \cdot \phi(s,a) \right)^2 \right] + 2(w - w_\star) \mathbb{E}_{s,a \sim d^{(t)}} \left[ \phi(s,a) \left( w_\star \cdot \phi(s,a) - Q^{(t)}(s,a) \right) \right]$$
$$= \|w - w_\star^{(t)}\|_{\Sigma_{d^{(t)}}}^2 + (w - w_\star^{(t)}) \cdot \nabla L(w_\star^{(t)}; \theta^{(t)}, d^{(t)})$$
$$\geq \|w - w_\star^{(t)}\|_{\Sigma_{d^{(t)}}}^2.$$

Noting that $w^{(t)} \in \mathcal{W}$ by construction in Algorithm 0.8 yields the claimed bound on the second term in (0.10).

Using the bounds on the first and second terms in (0.9) and (0.10), along with concavity of the square root function, we have that:

$$\mathrm{err}_t \leq 2\sqrt{\frac{|\mathcal{A}|}{1-\gamma} \left\| \frac{d^\star}{\nu} \right\|_\infty L(w_\star^{(t)}; \theta^{(t)}, d^{(t)})} + 2\sqrt{\frac{|\mathcal{A}|\kappa}{1-\gamma} \left( L(w^{(t)}; \theta^{(t)}, d^{(t)}) - L(w_\star^{(t)}; \theta^{(t)}, d^{(t)}) \right)}.$$

The proof is completed by substitution and using our assumptions on $\epsilon_{\mathrm{stat}}$ and $\epsilon_{\mathrm{bias}}$. ∎

## 13.5 $Q$-NPG Sample Complexity

**To be added...**

## 13.6 Bibliographic Remarks and Further Readings

The notion of *compatible function approximation* was due to [Sutton et al., 1999], which also proved the claim in Lemma 13.1. The close connection of the NPG update rule to compatible function approximation (Lemma 0.3) was noted in [Kakade, 2001].

The regret lemma (Lemma 13.3) for the NPG analysis has origins in the online regret framework in changing MDPs [Even-Dar et al., 2009]. The convergence rates in this chapter are largely derived from [Agarwal et al., 2020d]. The Q-NPG algorithm for the log-linear policy classes is essentially the same algorithm as POLITEX [Abbasi-Yadkori et al., 2019], with a distinction that it is important to use a state action measure over the initial distribution. The analysis and error decomposition of Q-NPG is from [Agarwal et al., 2020d], which has a more general analysis of NPG with function approximation under the regret lemma. This more general approach also permits the analysis of neural policy classes, as shown in [Agarwal et al., 2020d]. Also, [Liu et al., 2019] provide an analysis of the TRPO algorithm [Schulman et al., 2015] (essentially the same as NPG) for neural network parameterizations in the somewhat restrictive linearized "neural tangent kernel" regime.

# Chapter 14

# CPI, TRPO, and More

In this chapter, we consider conservative policy iteration (CPI) and trust-region constrained policy optimization (TRPO). Both CPI and TRPO can be understood as making small incremental update to the policy by forcing that the new policy's state action distribution is not too far away from the current policy's. We will see that CPI achieves that by forming a new policy that is a mixture of the current policy and a local greedy policy, while TRPO forcing that by explicitly adding a KL constraint (over polices' induced trajectory distributions space) in the optimization procedure. We will show that TRPO gives an equivalent update procedure as Natural Policy Gradient.

Along the way, we discuss the benefit of incremental policy update, by contrasting it to another family of policy update procedure called *Approximate Policy Iteration* (API), which performs local greedy policy search and could potentially lead to abrupt policy change. We show that API in general fails to converge or make local improvement, unless under a much stronger concentrability ratio assumption.

The algorithm and analysis of CPI is adapted from the original one in [Kakade and Langford, 2002], and the we follow the presentation of TRPO from [Schulman et al., 2015], while making a connection to the NGP algorithm.

## 14.1 Conservative Policy Iteration

As the name suggests, we will now describe a more conservative version of the policy iteration algorithm, which shifts the next policy away from the current policy with a small step size to prevent drastic shifts in successive state distributions.

We consider the discounted MDP here $\{\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho\}$ where $\rho$ is the initial state distribution. Similar to Policy Gradient Methods, we assume that we have a restart distribution $\mu$ (i.e., the $\mu$-restart setting). Throughout this section, for any policy $\pi$, we denote $d_\mu^\pi$ as the state-action visitation starting from $s_0 \sim \mu$ instead of $\rho$, and $d^\pi$ the state-action visitation starting from the true initial state distribution $\rho$, i.e., $s_0 \sim \rho$. Similarly, we denote $V_\mu^\pi$ as expected discounted total reward of policy $\pi$ starting at $\mu$, while $V^\pi$ as the expected discounted total reward of $\pi$ with $\rho$ as the initial state distribution. We assume $\mathcal{A}$ is discrete but $\mathcal{S}$ could be continuous.

CPI is based on the concept of *Reduction to Supervised Learning*. Specifically we will use the Approximate Greedy Policy Selector defined in Chapter 4 (Definition **??**). We recall the definition of the $\varepsilon$-approximate Greedy Policy Selector $\mathcal{G}_\varepsilon(\pi, \Pi, \mu)$ below. Given a policy $\pi$, policy class $\Pi$, and a restart distribution $\mu$, denote $\widehat{\pi} = \mathcal{G}_\varepsilon(\pi, \Pi, \mu)$, we have that:

$$\mathbb{E}_{s \sim d_\mu^\pi} A^\pi(s, \widehat{\pi}(s)) \geq \max_{\widetilde{\pi} \in \Pi} \mathbb{E}_{s \sim d_\mu^\pi} A^\pi(s, \widetilde{\pi}(s)) - \varepsilon.$$

Recall that in Chapter 4 we explained two approach to implement such approximate oracle: one with a reduction to classification oracle, and the other one with a reduction to regression oracle.

### 14.1.1 The CPI Algorithm

CPI, summarized in Alg. 8, will iteratively generate a sequence of policies $\pi^i$. Note we use $\pi_\alpha = (1-\alpha)\pi + \alpha\pi'$ to refer to a randomized policy which at any state $s$, chooses an action according to $\pi$ with probability $1 - \alpha$ and according to $\pi'$ with probability $\alpha$. The greedy policy $\pi'$ is computed using the $\varepsilon$-approximate greedy policy selector $\mathcal{G}_\varepsilon(\pi^t, \Pi, \mu)$. The algorithm is terminate when there is no significant one-step improvement over $\pi^t$, i.e., $\mathbb{E}_{s\sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi'(s))) \leq \varepsilon$.

---

**Algorithm 8** Conservative Policy Iteration (CPI)

---

**Input:** Initial policy $\pi^0 \in \Pi$, accuracy parameter $\varepsilon$.
  1: **for** $t = 0, 1, 2 \ldots$ **do**
  2:     $\pi' = \mathcal{G}_\varepsilon(\pi^t, \Pi, \mu)$
  3:     **if** $\mathbb{E}_{s\sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi'(s)) \leq \varepsilon$ **then**
  4:         Return $\pi^t$
  5:     **end if**
  6:     Update $\pi^{t+1} = (1 - \alpha)\pi^t + \alpha\pi'$
  7: **end for**

---

The main intuition behind the algorithm is that the stepsize $\alpha$ controls the difference between state distributions of $\pi^t$ and $\pi^{t+1}$. Let us look into the performance difference lemma to get some intuition on this conservative update. From PDL, we have:

$$V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}} A^{\pi^t}(s, \pi^{t+1}(s)) = \frac{\alpha}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}} A^{\pi^t}(s, \pi'(s)),$$

where the last equality we use the fact that $\pi^{t+1} = (1 - \alpha)\pi^t + \alpha\pi'$ and $A^{\pi^t}(s, \pi^t(s)) = 0$ for all $s$. Thus, if we can search for a policy $\pi' \in \Pi$ that maximizes $\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}} A^{\pi^t}(s, \pi'(s))$ and makes $\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}} A^{\pi^t}(s, \pi'(s)) > 0$, then we can guarantee policy improvement. However, at episode $t$, we do not know the state distribution of $\pi^{t+1}$ and all we have access to is $d_\mu^{\pi^t}$. Thus, we explicitly make the policy update procedure to be conservative such that $d_\mu^{\pi^t}$ and the new policy's distribution $d_\mu^{\pi^{t+1}}$ is guaranteed to be not that different. Thus we can hope that $\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}} A^{\pi^t}(s, \pi'(s))$ is close to $\mathbb{E}_{s\sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi'(s))$, and the latter is something that we can manipulate using the greedy policy selector.

Below we formalize the above intuition and show that with small enough $\alpha$, we indeed can ensure monotonic policy improvement.

We start from the following lemma which shows that $\pi^{t+1}$ and $\pi^t$ are close to each other in terms of total variation distance at any state, and $d_\mu^{\pi^{t+1}}$ and $d_\mu^{\pi^t}$ are close as well.

**Lemma 14.1** (Similar Policies imply similar state visitations). *Consider any $t$, we have that:*

$$\left\|\pi^{t+1}(\cdot|s) - \pi^t(\cdot|s)\right\|_1 \leq 2\alpha, \forall s;$$

*Further, we have:*

$$\left\|d_\mu^{\pi^{t+1}} - d_\mu^{\pi^t}\right\|_1 \leq \frac{2\alpha\gamma}{1-\gamma}.$$

**Proof:** The first claim in the above lemma comes from the definition of policy update:

$$\left\|\pi^{t+1}(\cdot|s) - \pi^t(\cdot|s)\right\|_1 = \alpha\left\|\pi^t(\cdot|s) - \pi'(\cdot|s)\right\|_1 \leq 2\alpha.$$

Denote $\mathbb{P}_h^\pi$ as the state distribution resulting from $\pi$ at time step $h$ with $\mu$ as the initial state distribution. We consider bounding $\|\mathbb{P}_h^{\pi^{t+1}} - \mathbb{P}_h^{\pi^t}\|_1$ with $h \geq 1$.

$$\mathbb{P}_h^{\pi^{t+1}}(s') - \mathbb{P}_h^{\pi^t}(s') = \sum_{s,a}\left(\mathbb{P}_{h-1}^{\pi^{t+1}}(s)\pi^{t+1}(a|s) - \mathbb{P}_{h-1}^{\pi^t}(s)\pi^t(a|s)\right)P(s'|s,a)$$

$$= \sum_{s,a}\left(\mathbb{P}_{h-1}^{\pi^{t+1}}(s)\pi^{t+1}(a|s) - \mathbb{P}_{h-1}^{\pi^{t+1}}(s)\pi^t(a|s) + \mathbb{P}_{h-1}^{\pi^{t+1}}(s)\pi^t(a|s) - \mathbb{P}_{h-1}^{\pi^t}(s)\pi^t(a|s)\right)P(s'|s,a)$$

$$= \sum_s \mathbb{P}_{h-1}^{\pi^{t+1}}(s)\sum_a\left(\pi^{t+1}(a|s) - \pi^t(a|s)\right)P(s'|s,a)$$

$$+ \sum_s\left(\mathbb{P}_{h-1}^{\pi^{t+1}}(s) - \mathbb{P}_{h-1}^{\pi^t}(s)\right)\sum_a \pi^t(a|s)P(s'|s,a).$$

Take absolute value on both sides, we get:

$$\sum_{s'}\left|\mathbb{P}_h^{\pi^{t+1}}(s') - \mathbb{P}_h^{\pi^t}(s')\right| \leq \sum_s \mathbb{P}_{h-1}^{\pi^{t+1}}(s)\sum_a\left|\pi^{t+1}(a|s) - \pi^t(a|s)\right|\sum_{s'}P(s'|s,a)$$

$$+ \sum_s\left|\mathbb{P}_{h-1}^{\pi^{t+1}}(s) - \mathbb{P}_{h-1}^{\pi^t}(s)\right|\sum_{s'}\sum_a \pi^t(a|s)P(s'|s,a)$$

$$\leq 2\alpha + \|\mathbb{P}_{h-1}^{\pi^{t+1}} - \mathbb{P}_{h-1}^{\pi^t}\|_1 \leq 4\alpha + \|\mathbb{P}_{h-2}^{\pi^{t+1}} - \mathbb{P}_{h-2}^{\pi^t}\|_1 = 2h\alpha.$$

Now use the definition of $d_\mu^\pi$, we have:

$$d_\mu^{\pi^{t+1}} - d_\mu^{\pi^t} = (1-\gamma)\sum_{h=0}^\infty \gamma^h\left(\mathbb{P}_h^{\pi^{t+1}} - \mathbb{P}_h^{\pi^t}\right).$$

Add $\ell_1$ norm on both sides, we get:

$$\left\|d_\mu^{\pi^{t+1}} - d_\mu^{\pi^t}\right\|_1 \leq (1-\gamma)\sum_{h=0}^\infty \gamma^h 2h\alpha$$

It is not hard to verify that $\sum_{h=0}^\infty \gamma^h h = \frac{\gamma}{(1-\gamma)^2}$. Thus, we can conclude that:

$$\left\|d_\mu^{\pi^{t+1}} - d_\mu^{\pi^t}\right\|_1 \leq \frac{2\alpha\gamma}{1-\gamma}.$$

$\blacksquare$

The above lemma states that if $\pi^{t+1}$ and $\pi^t$ are close in terms of total variation distance for every state, then the total variation distance between the resulting state visitations from $\pi^{t+1}$ and $\pi^t$ will be small up to a effective horizon $1/(1-\gamma)$ amplification.

The above lemma captures the key of the conservative policy update. Via the conservative policy update, we make sure that $d_\mu^{\pi^{t+1}}$ and $d_\mu^{\pi^t}$ are close to each other in terms of total variation distance. Now we use the above lemma to show a monotonic policy improvement.

**Theorem 14.2** (Monotonic Improvement in CPI). *Consider any episode $t$. Denote $\mathbb{A} = \mathbb{E}_{s\sim d_\mu^{\pi^t}}A^{\pi^t}(s,\pi'(s))$. We have:*

$$V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} \geq \frac{\alpha}{1-\gamma}\left(\mathbb{A} - \frac{2\alpha\gamma}{(1-\gamma)^2}\right)$$

131

*Set $\alpha = \frac{\mathbb{A}(1-\gamma)^2}{4\gamma}$, we get:*

$$V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} \geq \frac{\mathbb{A}^2(1-\gamma)}{8\gamma}.$$

The above lemma shows that as long as we still have positive one-step improvement, i.e., $\mathbb{A} > 0$, then we guarantee that $\pi^{t+1}$ is strictly better than $\pi^t$.

**Proof:** Via PDL, we have:

$$V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\mu^{\pi^{t+1}}} \alpha A^{\pi^t}(s, \pi'(s)).$$

Recall Lemma 14.1, we have:

$$
\begin{aligned}
(1-\gamma)\left(V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t}\right) &= \mathbb{E}_{s \sim d_\mu^{\pi^t}} \alpha A^{\pi^t}(s, \pi'(s)) + \mathbb{E}_{s \sim d_\mu^{\pi^{t+1}}} \alpha A^{\pi^t}(s, \pi'(s)) - \mathbb{E}_{s \sim d_\mu^{\pi^t}} \alpha A^{\pi^t}(s, \pi'(s)) \\
&\geq \mathbb{E}_{s \sim d_\mu^{\pi^t}} \alpha A^{\pi^t}(s, \pi'(s)) - \alpha \sup_{s,a,\pi} |A^\pi(s,a)| \|d_\mu^{\pi^t} - d_\mu^{\pi^{t+1}}\|_1 \\
&\geq \mathbb{E}_{s \sim d_\mu^{\pi^t}} \alpha A^{\pi^t}(s, \pi'(s)) - \frac{\alpha}{1-\gamma} \|d_\mu^{\pi^t} - d_\mu^{\pi^{t+1}}\|_1 \\
&\geq \mathbb{E}_{s \sim d_\mu^{\pi^t}} \alpha A^{\pi^t}(s, \pi'(s)) - \frac{2\alpha^2\gamma}{(1-\gamma)^2} = \alpha\left(\mathbb{A} - \frac{2\alpha\gamma}{(1-\gamma)^2}\right)
\end{aligned}
$$

where the first inequality we use the fact that for any two distributions $P_1$ and $P_2$ and any function $f$, $|\mathbb{E}_{x \sim P_1} f(x) - \mathbb{E}_{x \sim P_2} f(x)| \leq \sup_x |f(x)| \|P_1 - P_2\|_1$, for the second inequality, we use the fact that $|A^\pi(s,a)| \leq 1/(1-\gamma)$ for any $\pi, s, a$, and the last inequality uses Lemma 14.1.

For the second part of the above theorem, note that we want to maximum the policy improvement as much as possible by choosing $\alpha$. So we can pick $\alpha$ which maximizes $\alpha(\mathbb{A} - 2\alpha\gamma/(1-\gamma)^2)$. This gives us the $\alpha$ we claimed in the lemma. Plug in $\alpha$ back into $\alpha(\mathbb{A} - 2\alpha\gamma/(1-\gamma)^2)$, we conclude the second part of the theorem.

$\blacksquare$

The above theorem indicates that with the right choice of $\alpha$, we guarantee that the policy is making improvement as long as $\mathbb{A} > 0$. Recall the termination criteria in CPI where we terminate CPI when $\mathbb{A} \leq \varepsilon$. Putting these results together, we obtain the following overall convergence guarantee for the CPI algorithm.

**Theorem 14.3** (Local optimality of CPI). *Algorithm 8 terminates in at most $8\gamma/\epsilon^2$ steps and outputs a policy $\pi^t$ satisfying $\max_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi(s)) \leq 2\varepsilon$.*

**Proof:** Note that our reward is bounded in $[0, 1]$ which means that $V_\mu^\pi \in [0, 1/(1-\gamma)]$. Note that we have shown in Theorem 14.2, every iteration $t$, we have policy improvement at least $\frac{\mathbb{A}^2(1-\gamma)}{8\gamma}$, where recall $\mathbb{A}$ at episode $t$ is defined as $\mathbb{A} = \mathbb{E}_{s \sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi'(s))$. If the algorithm does not terminate at episode $t$, then we guarantee that:

$$V_\mu^{\pi^{t+1}} \geq V_\mu^{\pi^t} + \frac{\varepsilon^2(1-\gamma)}{8\gamma}$$

Since $V_\mu^\pi$ is upper bounded by $1/(1-\gamma)$, so we can at most make improvement $8\gamma/\epsilon^2$ many iterations.

Finally, recall that $\varepsilon$-approximate greedy policy selector $\pi' = \mathcal{G}_\varepsilon(\pi^t, \Pi, \mu)$, we have:

$$\max_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi(s)) \leq \mathbb{E}_{s \sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi'(s)) + \varepsilon \leq 2\varepsilon$$

This concludes the proof. ∎

Theorem 14.3 can be viewed as a local optimality guarantee in a sense. It shows that when CPI terminates, we cannot find a policy $\pi \in \Pi$ that achieves local improvement over the returned policy more than $\varepsilon$. However, this does not necessarily imply that the value of $\pi$ is close to $V^\star$. However, similar to the policy gradient analysis, we can turn this local guarantee into a global one when the restart distribution $\mu$ covers $d^{\pi^\star}$. We formalize this intuition next.

**Theorem 14.4** (Global optimality of CPI). *Upon termination, we have a policy $\pi$ such that:*

$$V^\star - V^\pi \leq \frac{2\varepsilon + \epsilon_\Pi}{(1-\gamma)^2} \left\| \frac{d^{\pi^\star}}{\mu} \right\|_\infty,$$

*where $\epsilon_\Pi := \mathbb{E}_{s \sim d^\pi_\mu}[\max_{a \in \mathcal{A}} A^\pi(s,a)] - \max_{\pi \in \Pi} \mathbb{E}_{s \sim d^\pi_\mu}[A^\pi(s, \pi(s))]$.*

In other words, if our policy class is rich enough to approximate the policy $\max_{a \in \mathcal{A}} A^\pi(s,a)$ under $d^\pi_\mu$, i.e., $\epsilon_\Pi$ is small, and $\mu$ covers $d^{\pi^\star}$ in a sense that $\left\| \frac{d^{\pi^\star}}{\mu} \right\|_\infty \leq \infty$, CPI guarantees to find a near optimal policy.

**Proof:** By the performance difference lemma,

$$
\begin{aligned}
V^\star - V^\pi &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d^{\pi^\star}} A^\pi(s, \pi^\star(s)) \\
&\leq \frac{1}{1-\gamma} \mathbb{E}_{s \sim d^{\pi^\star}} \max_{a \in \mathcal{A}} A^\pi(s, a) \\
&\leq \frac{1}{1-\gamma} \left\| \frac{d^{\pi^\star}}{d^\pi_\mu} \right\|_\infty \mathbb{E}_{s \sim d^\pi_\mu} \max_{a \in \mathcal{A}} A^\pi(s, a) \\
&\leq \frac{1}{(1-\gamma)^2} \left\| \frac{d^{\pi^\star}}{\mu} \right\|_\infty \mathbb{E}_{s \sim d^\pi_\mu} \max_{a \in \mathcal{A}} A^\pi(s, a) \\
&=\leq \frac{1}{(1-\gamma)^2} \left\| \frac{d^{\pi^\star}}{\mu} \right\|_\infty \left[ \max_{\hat\pi \in \Pi} \mathbb{E}_{s \sim d^\pi_\mu} A^\pi(s, \hat\pi(s)) - \max_{\hat\pi \in \Pi} \mathbb{E}_{s \sim d^\pi_\mu} A^\pi(s, \hat\pi(s)) + \mathbb{E}_{s \sim d^\pi_\mu} \max_{a \in \mathcal{A}} A^\pi(s, a) \right] \\
&\leq \frac{1}{(1-\gamma)^2} \left\| \frac{d^{\pi^\star}}{\mu} \right\|_\infty (2\varepsilon + \epsilon_\Pi),
\end{aligned}
$$

where the second inequality holds due to the fact that $\max_a A^\pi(s,a) \geq 0$, the third inequality uses the fact that $d^\pi_\mu(s) \geq (1-\gamma)\mu(s)$ for any $s$ and $\pi$, and the last inequality uses the definition $\epsilon_\Pi$ and Theorem 14.3. ∎

It is informative to contrast CPI and policy gradient algorithms due to the similarity of their guarantees. Both provide local optimality guarantees. For CPI, the local optimality always holds, while for policy gradients it requires a smooth value function as a function of the policy parameters. If the distribution mismatch between an optimal policy and the output of the algorithm is not too large, then both algorithms further yield a near optimal policy. The similarities are not so surprising. Both algorithms operate by making local improvements to the current policy at each iteration, by inspecting its advantage function. The changes made to the policy are controlled using a stepsize parameter in both the approaches. It is the actual mechanism of the improvement which differs in the two cases. Policy gradients assume that the policy's reward is a differentiable function of the parameters, and hence make local improvements through gradient ascent. The differentiability is certainly an assumption and does not necessarily hold for all policy classes. An easy example is when the policy itself is not an easily differentiable function of its parameters. For instance, if the policy is parametrized by regression trees, then performing gradient updates can be challenging.

In CPI, on the other hand, the basic computational primitive required on the policy class is the ability to maximize the advantage function relative to the current policy. Notice that Algorithm 8 does not necessarily restrict to a policy class, such as a set of parametrized policies as in policy gradients. Indeed, due to the reduction to supervised learning approach (e.g., using the weighted classification oracle CO), we can parameterize policy class via decision tree, for

instance. This property makes CPI extremely attractive. Any policy class over which efficient supervised learning algorithms exist can be adapted to reinforcement learning with performance guarantees.

A second important difference between CPI and policy gradients is in the notion of locality. Policy gradient updates are local in the parameter space, and we hope that this makes small enough changes to the state distribution that the new policy is indeed an improvement on the older one (for instance, when we invoke the performance difference lemma between successive iterates). While this is always true in expectation for correctly chosen stepsizes based on properties of stochastic gradient ascent on smooth functions, the variance of the algorithm and lack of robustness to suboptimal stepsizes can make the algorithm somewhat finicky. Indeed, there are a host of techniques in the literature to both lower the variance (through control variates) and explicitly control the state distribution mismatch between successive iterates of policy gradients (through trust region techniques). On the other hand, CPI explicitly controls the amount of perturbation to the state distribution by carefully mixing policies in a manner which does not drastically alter the trajectories with high probability. Indeed, this insight is central to the proof of CPI, and has been instrumental in several follow-ups, both in the direct policy improvement as well as policy gradient literature.

## 14.2 Trust Region Methods and Covariant Policy Search

So far we have seen policy gradient methods and CPI which all uses a small step-size to ensure incremental update in policies. Another popular approach for incremental policy update is to explicitly forcing small change in policies' distribution via a trust region constraint. More specifically, let us go back to the general policy parameterization $\pi_\theta$. At iteration $t$ with the current policy $\pi_{\theta_t}$, we are interested in the following local trust-region constrained optimization:

$$\max_\theta \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s, a)$$

$$\text{s.t., } KL\left(\Pr_\mu^{\pi_{\theta_t}} || \Pr_\mu^{\pi_\theta}\right) \le \delta,$$

where recall $\Pr_\mu^\pi(\tau)$ is the trajectory distribution induced by $\pi$ starting at $s_0 \sim \mu$, and $KL(P_1||P_2)$ are KL-divergence between two distribution $P_1$ and $P_2$. Namely we explicitly perform local policy search with a constraint forcing the new policy not being too far away from $\Pr_\mu^{\pi_{\theta_t}}$ in terms of KL divergence.

As we are interested in small local update in parameters, we can perform sequential quadratic programming here, i.e., we can further linearize the objective function at $\theta_t$ and quadratize the KL constraint at $\theta_t$ to form a local quadratic programming:

$$\max_\theta \left\langle \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \mathbb{E}_{a \sim \pi_{\theta_t}(\cdot|s)} \nabla_\theta \ln \pi_{\theta_t}(a|s) A^{\pi_{\theta_t}}(s, a), \theta \right\rangle \tag{0.1}$$

$$\text{s.t., } \left\langle \nabla_\theta KL\left(\Pr_\mu^{\pi_{\theta_t}} || \Pr_\mu^{\pi_\theta}\right)|_{\theta=\theta_t}, \theta - \theta_t \right\rangle + \frac{1}{2}(\theta - \theta_t)^\top \left(\nabla_\theta^2 KL\left(\Pr_\mu^{\pi_{\theta_t}} || \Pr_\mu^{\pi_\theta}\right)|_{\theta=\theta_t}\right)(\theta - \theta_t) \le \delta, \tag{0.2}$$

where we denote $\nabla^2 KL|_{\theta=\theta_t}$ as the Hessian of the KL constraint measured at $\theta_t$. Note that KL divergence is not a valid metric as it is not symmetric. However, its local quadratic approximation can serve as a valid local distance metric, as we prove below that the Hessian $\nabla^2 KL|_{\theta=\theta_t}$ is a positive semi-definite matrix. Indeed, we will show that the Hessian of the KL constraint is exactly equal to the fisher information matrix, and the above quadratic programming exactly reveals a Natural Policy Gradient update. Hence Natural policy gradient can also be interpreted as performing sequential quadratic programming with KL constraint over policy's trajectory distributions.

To match to the practical algorithms in the literature (e.g., TRPO), below we focus on episode finite horizon setting again (i.e., an MDP with horizon $H$).

**Claim 14.5.** *Consider a finite horizon MDP with horizon $H$. Consider any fixed $\theta_t$. We have:*

$$\nabla_\theta KL\left(\mathrm{Pr}_\mu^{\pi^{\theta_t}}||\mathrm{Pr}_\mu^{\pi_\theta}\right)|_{\theta=\theta_t} = 0,$$

$$\nabla_\theta^2 KL\left(\mathrm{Pr}_\mu^{\pi^{\theta_t}}||\mathrm{Pr}_\mu^{\pi_\theta}\right)|_{\theta=\theta_t} = H\mathbb{E}_{s,a\sim d^{\pi_{\theta_t}}}\nabla\ln\pi_{\theta_t}(a|s)\left(\nabla\ln\pi_{\theta_t}(a|s)\right)^\top.$$

**Proof:** We first recall the trajectory distribution in finite horizon setting.

$$\mathrm{Pr}_\mu^\pi(\tau) = \mu(s_0)\prod_{h=0}^{H-1}\pi(a_h|s_h)P(s_{h+1}|s_h,a_h).$$

We first prove that the gradient of KL is zero. First note that:

$$KL\left(\mathrm{Pr}_\mu^{\pi^{\theta_t}}||\mathrm{Pr}_\mu^{\pi_\theta}\right) = \sum_\tau \mathrm{Pr}_\mu^{\pi^{\theta_t}}(\tau)\ln\frac{\mathrm{Pr}_\mu^{\pi^{\theta_t}}(\tau)}{\mathrm{Pr}_\mu^{\pi^\theta}(\tau)}$$

$$= \sum_\tau \mathrm{Pr}_\mu^{\pi^{\theta_t}}(\tau)\left(\sum_{h=0}^{H-1}\ln\frac{\pi_{\theta_t}(a_h|s_h)}{\pi_\theta(a_h|s_h)}\right) = \sum_{h=0}^{H-1}\mathbb{E}_{s_h,a_h\sim\mathbb{P}_h^{\pi_{\theta_t}}}\ln\frac{\pi_{\theta_t}(a_h|s_h)}{\pi_\theta(a_h|s_h)}.$$

Thus, for $\nabla_\theta KL\left(\mathrm{Pr}_\mu^{\pi^{\theta_t}}||\mathrm{Pr}_\mu^{\pi_\theta}\right)$, we have:

$$\nabla KL\left(\mathrm{Pr}_\mu^{\pi^{\theta_t}}||\mathrm{Pr}_\mu^{\pi_\theta}\right)|_{\theta=\theta_t} = \sum_{h=0}^{H-1}\mathbb{E}_{s_h,a_h\sim\mathbb{P}_h^{\pi_{\theta_t}}} - \nabla\ln\pi_{\theta_t}(a_h|s_h)$$

$$= -\sum_{h=0}^{H-1}\mathbb{E}_{s_h\sim\mathbb{P}_h^{\pi_{\theta_t}}}\mathbb{E}_{a_h\sim\pi_{\theta_t}(\cdot|s_h)}\nabla\ln\pi_{\theta_t}(a_h|s_h) = 0,$$

where we have seen the last step when we argue the unbiased nature of policy gradient with an action independent baseline.

Now we move to the Hessian.

$$\nabla^2 KL\left(\mathrm{Pr}_\mu^{\pi^{\theta_t}}||\mathrm{Pr}_\mu^{\pi_\theta}\right)|_{\theta=\theta_t} = -\sum_{h=0}^{H-1}\mathbb{E}_{s_h,a_h\sim\mathbb{P}_h^{\pi_{\theta_t}}}\nabla^2\ln\pi_\theta(a_h|s_h)|_{\theta=\theta_t}$$

$$= -\sum_{h=0}^{H-1}\mathbb{E}_{s_h,a_h\sim\mathbb{P}_h^{\pi_{\theta_t}}}\left(\nabla\left(\frac{\nabla\pi_\theta(a|s)}{\pi_\theta(a|s)}\right)\right)$$

$$= -\sum_{h=0}^{H-1}\mathbb{E}_{s_h,a_h\sim\mathbb{P}_h^{\pi_{\theta_t}}}\left(\frac{\nabla^2\pi_\theta(a_h|s_h)}{\pi_\theta(a_h|s_h)} - \frac{\nabla\pi_\theta(a_h|s_h)\nabla\pi_\theta(a_h|s_h)^\top}{\pi_\theta^2(a_h|s_h)}\right)$$

$$= \sum_{h=0}^{H-1}\mathbb{E}_{s_h,a_h\sim\mathbb{P}_h^{\pi_{\theta_t}}}\nabla_\theta\ln\pi_\theta(a_h|s_h)\nabla_\theta\ln\pi_\theta(a_h|s_h)^\top,$$

where in the last equation we use the fact that $\mathbb{E}_{s_h,a_h\sim\mathbb{P}_h^{\pi_{\theta_t}}}\frac{\nabla^2\pi_\theta(a_h|s_h)}{\pi_\theta(a_h|s_h)} = 0$. ∎

The above claim shows that a second order taylor expansion of the KL constraint over trajectory distribution gives a local distance metric at $\theta_t$:

$$(\theta-\theta_t)F_{\theta_t}(\theta-\theta_t),$$

where again $F_{\theta_t} := H\mathbb{E}_{s,a\sim d^{\pi_{\theta_t}}} \nabla \ln \pi_{\theta_t}(a|s) \left(\nabla \ln \pi_{\theta_t}(a|s)\right)^\top$ is proportional to the fisher information matrix. Note that $F_{\theta_t}$ is a PSD matrix and thus $d(\theta, \theta_t) := (\theta - \theta_t)F_{\theta_t}(\theta - \theta_t)$ is a valid distance metric. By sequential quadratic programming, we are using local geometry information of the trajectory distribution manifold induced by the parameterization $\theta$, rather the naive Euclidean distance in the parameter space. Such a method is sometimes referred to as *Covariant Policy Search*, as the policy update procedure will be invariant to linear transformation of parameterization (See Section 14.3 for further discussion).

Now using the results from Claim 14.5, we can verify that the local policy optimization procedure in Eq. 0.2 exactly recovers the NPG update, where the step size is based on the trust region parameter $\delta$. Denote $\Delta = \theta - \theta_t$, we have:

$$\max_\Delta \langle \Delta, \nabla_\theta V^{\pi_{\theta_t}} \rangle,$$
$$s.t., \Delta^\top F_{\theta_t} \Delta^\top \leq \delta,$$

which gives the following update procedure:

$$\theta_{t+1} = \theta_t + \Delta = \theta_t + \sqrt{\frac{\delta}{(\nabla V^{\pi_{\theta_t}})^\top F_{\theta_t}^{-1} \nabla V^{\pi_{\theta_t}}}} \cdot F_{\theta_t}^{-1} \nabla V^{\pi_{\theta_t}},$$

where note that we use the self-normalized learning rate computed using the trust region parameter $\delta$.


## 14.2.1 Proximal Policy Optimization

Here we consider an $\ell_\infty$ style trust region constraint:

$$\max_\theta \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \mathbb{E}_{a\sim\pi^\theta(\cdot|s)} A^{\pi^{\theta_t}}(s,a) \tag{0.3}$$

$$s.t., \sup_s \left\| \pi^\theta(\cdot|s) - \pi^{\theta_t}(\cdot|s) \right\|_{tv} \leq \delta, \tag{0.4}$$

Namely, we restrict the new policy such that it is close to $\pi^{\theta_t}$ at every state $s$ under the total variation distance. Recall the CPI's update, CPI indeed makes sure that the new policy will be close the old policy at every state. In other words, the new policy computed by CPI is a feasible solution of the constraint Eq. 0.4, but is not the optimal solution of the above constrained optimization program. Also one downside of the CPI algorithm is that one needs to keep all previous learned policies around, which requires large storage space when policies are parameterized by large deep neural networks.

Proximal Policy Optimization (PPO) aims to directly optimize the objective Eq. 0.3 using multiple steps of gradient updates, and approximating the constraints Eq. 0.4 via a clipping trick. We first rewrite the objective function using importance weighting:

$$\max_\theta \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \mathbb{E}_{a\sim\pi^{\theta_t}(\cdot|s)} \frac{\pi^\theta(a|s)}{\pi^{\theta_t}(\cdot|s)} A^{\pi^{\theta_t}}(s,a), \tag{0.5}$$

where we can easily approximate the expectation $\mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \mathbb{E}_{a\sim\pi^{\theta_t}(\cdot|s)}$ via finite samples $s \sim d^{\pi_{\theta_t}}, a \sim \pi^{\theta_t}(\cdot|s)$.

To make sure $\pi^\theta(a|s)$ and $\pi^{\theta_t}(a|s)$ are not that different, PPO modifies the objective function by clipping the density ratio $\pi^\theta(a|s)$ and $\pi^{\theta_t}(a|s)$:

$$L(\theta) := \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \mathbb{E}_{a\sim\pi^{\theta_t}(\cdot|s)} \min\left\{ \frac{\pi^\theta(a|s)}{\pi^{\theta_t}(\cdot|s)} A^{\pi^{\theta_t}}(s,a), \text{clip}\left(\frac{\pi^\theta(a|s)}{\pi^{\theta_t}(\cdot|s)}; 1-\epsilon, 1+\epsilon\right) A^{\pi^{\theta_t}}(s,a) \right\}, \tag{0.6}$$

where $\text{clip}\left(x; 1-\epsilon, 1+\epsilon\right) = \begin{cases} 1-\epsilon & x \le 1-\epsilon \\ 1+\epsilon & x \ge 1+\epsilon. \\ x & \text{else} \end{cases}$ The clipping operator ensures that for $\pi_\theta$, at any state action pair

where $\pi^\theta(a|s)/\pi^{\theta_t}(a|s) \notin [1-\epsilon, 1+\epsilon]$, we get zero gradient, i.e., $\nabla_\theta \left[\text{clip}\left(\frac{\pi^\theta(a|s)}{\pi^{\theta_t}(\cdot|s)}; 1-\epsilon, 1+\epsilon\right) A^{\pi^{\theta_t}}(s,a)\right] = 0$. The outer min makes sure the objective function $L(\theta)$ is a lower bound of the original objective. PPO then proposes to collect a dataset $(s, a)$ with $s \sim d_\mu^{\pi_{\theta_t}}$ and $a \sim \pi^{\theta_t}(\cdot|s)$, and then perform multiple steps of mini-batch stochastic gradient ascent on $L(\theta)$.

One of the key difference between PPO and other algorithms such as NPG is that PPO targets to optimizes objective Eq. 0.5 via multiple steps of mini-batch stochastic gradient ascent with mini-batch data from $d_\mu^{\pi_{\theta_t}}$ and $\pi^{\theta_t}$, while algorithm such as NPG indeed optimizes the first order taylor expansion of Eq. 0.5 at $\theta_t$, i.e.,

$$\max_\theta \left\langle (\theta - \theta^t),\ \mathbb{E}_{s,a \sim d_\mu^{\pi_{\theta_t}}} \nabla_\theta \ln \pi^{\theta_t}(a|s) A^{\pi^{\theta_t}}(s,a) \right\rangle,$$

upper to some trust region constraints (e.g., $\|\theta - \theta_t\|_2 \le \delta$ in policy gradient, and $\|\theta - \theta_t\|_{F_{\theta^t}}^2 \le \delta$ for NPG ).

## 14.3   Bibliographic Remarks and Further Readings

The analysis of CPI is adapted from the original one in [Kakade and Langford, 2002]. There have been a few further interpretations of CPI. One interesting perspective is that CPI can be treated as a boosting algorithm [Scherrer and Geist, 2014].

More generally, CPI and NPG are part of family of *incremental* algorithms, including Policy Search by Dynamic Programming (PSDP) [Bagnell et al., 2004] and MD-MPI [Geist et al., 2019]. PSDP operates in a finite horizon setting and optimizes a sequence of time-dependent policies; from the last time step to the first time step, every iteration of, PSDP only updates the policy at the current time step while holding the future policies fixed — thus making incremental update on the policy. See [Scherrer, 2014] for more a detailed discussion and comparison of some of these approaches. Mirror Descent-Modified Policy Iteration (MD-MPI) algorithm [Geist et al., 2019] is a family of actor-critic style algorithms which is based on regularization and is incremental in nature; with negative entropy as the Bregman divergence (for the tabular case), MD-MPI recovers the NPG the tabular case (for the softmax parameterization).

Broadly speaking, these incremental algorithms can improve upon the stringent concentrability conditions for approximate value iteration methods, presented in Chapter 4. Scherrer [2014] provide a more detailed discussion of bounds which depend on these density ratios. As discussed in the last chapter, the density ratio for NPG can be interpreted as a factor due to transfer learning to a single, *fixed* distribution.

The interpretation of NPG as Covariant Policy Search is due to [Bagnell and Schneider, 2003], as the policy update procedure will be invariant to linear transformations of the parameterization; see [Bagnell and Schneider, 2003] for a more detailed discussion on this.

The TRPO algorithm is due to [Schulman et al., 2015]. The original TRPO analysis provides performance guarantees, largely relying on a reduction to the CPI guarantees. In this Chapter, we make a sharper connection of TRPO to NPG, which was subsequently observed by a number of researchers; this connection provides a sharper analysis for the generalization and approximation behavior of TRPO (e.g. via the results presented in Chapter 13). In practice, a popular variant is the Proximal Policy Optimization (PPO) algorithm [Schulman et al., 2017].

# Part 4

## Further Topics

# Chapter 15

# Imitation Learning

In this chapter, we study imitation learning. Unlike the Reinforcement Learning setting, in Imitation Learning, we do not have access to the ground truth reward function (or cost function), but instead, we have expert demonstrations. We often assume that the expert is a policy that approximately optimizes the underlying reward (or cost) functions. The goal is to leverage the expert demonstrations to learn a policy that performs as good as the expert.

We consider three settings of imitation learning: (1) pure offline where we only have expert demonstrations and no more real world interaction is allowed, (2) hybrid where we have expert demonstrations, and also is able to interact with the real world (e.g., have access to the ground truth transition dynamics); (3) interactive setting where we have an interactive expert and also have access to the underlying reward (cost) function.

## 15.1   Setting

We will focus on finite horizon MDPs $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, r, \mu, P, H\}$ where $r$ is the reward function but is unknown to the learner. We represent the expert as a closed-loop policy $\pi^\star : \mathcal{S} \mapsto \Delta(\mathcal{A})$. For analysis simplicity, we assume the expert $\pi^\star$ is indeed the optimal policy of the original MDP $\mathcal{M}$ with respect to the ground truth reward $r$. Again, our goal is to learn a policy $\widehat{\pi} : \mathcal{S} \mapsto \Delta(\mathcal{A})$ that performs as well as the expert, i.e., $V^{\widehat{\pi}}$ needs to be close to $V^\star$, where $V^\pi$ denotes the expected total reward of policy $\pi$ under the MDP $\mathcal{M}$. We denote $d^\pi$ as the state-action visitation of policy $\pi$ under $\mathcal{M}$.

We assume we have a pre-collected expert dataset in the format of $\{s_i^\star, a_i^\star\}_{i=1}^M$ where $s_i^\star, a_i^\star \sim d^{\pi^\star}$.

## 15.2   Offline IL: Behavior Cloning

We study offline IL here. Specifically, we study the classic Behavior Cloning algorithm.

We consider a policy class $\Pi = \{\pi : \mathcal{S} \mapsto \Delta(A)\}$. We consider the following realizable assumption.

**Assumption 15.1.** We assume $\Pi$ is rich enough such that $\pi^\star \in \Pi$.

For analysis simplicity, we assume $\Pi$ is discrete. But our sample complexity will only scale with respect to $\ln(|\Pi|)$.

Behavior cloning is one of the simplest Imitation Learning algorithm which only uses the expert data $\mathcal{D}^\star$ and does not require any further interaction with the MDP. It computes a policy via a reduction to supervised learning. Specifically,

we consider a reduction to Maximum Likelihood Estimation (MLE):

$$\text{Behavior Cloning (BC):} \quad \widehat{\pi} = \text{argmax}_{\pi \in \Pi} \sum_{i=1}^{N} \ln \pi(a_i^\star | s_i^\star). \tag{0.1}$$

Namely we try to find a policy from $\Pi$ that has the maximum likelihood of fitting the training data. As this is a reduction to MLE, we can leverage the existing classic analysis of MLE (e.g., Chapter 7 in [Geer and van de Geer, 2000]) to analyze the performance of the learned policy.

**Theorem 15.2** (MLE Guarantee). *Consider the MLE procedure (Eq. 0.1). With probability at least $1 - \delta$, we have:*

$$\mathbb{E}_{s \sim d^{\pi^\star}} \|\widehat{\pi}(\cdot|s) - \pi^\star(\cdot|s)\|_{tv}^2 \leq \frac{\ln(|\Pi|/\delta)}{M}.$$

We refer readers to Section E, Theorem 21 in [Agarwal et al., 2020b] for detailed proof of the above MLE guarantee.

Now we can transfer the average divergence between $\widehat{\pi}$ and $\pi^\star$ to their performance difference $V^{\widehat{\pi}}$ and $V^{\pi^\star}$.

One thing to note is that BC only ensures that the learned policy $\widehat{\pi}$ is close to $\pi^\star$ under $d^{\pi^\star}$. Outside of $d^{\pi^\star}$'s support, we have no guarantee that $\widehat{\pi}$ and $\pi^\star$ will be close to each other. The following theorem shows that a compounding error occurs when we study the performance of the learned policy $V^{\widehat{pi}}$.

**Theorem 15.3** (Sample Complexity of BC). *With probability at least $1 - \delta$, BC returns a policy $\widehat{\pi}$ such that:*

$$V^\star - V^{\widehat{\pi}} \leq \frac{2}{(1-\gamma)^2} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}.$$

**Proof:** We start with the performance difference lemma and the fact that $\mathbb{E}_{a \sim \pi(\cdot|s)} A^\pi(s, a) = 0$:

$$(1 - \gamma)\left(V^\star - V^{\widehat{\pi}}\right) = \mathbb{E}_{s \sim d^{\pi^\star}} \mathbb{E}_{a \sim \pi^\star(\cdot|s)} A^{\widehat{\pi}}(s, a)$$

$$= \mathbb{E}_{s \sim d^{\pi^\star}} \mathbb{E}_{a \sim \pi^\star(\cdot|s)} A^{\widehat{\pi}}(s, a) - \mathbb{E}_{s \sim d^{\pi^\star}} \mathbb{E}_{a \sim \widehat{\pi}(\cdot|s)} A^{\widehat{\pi}}(s, a)$$

$$\leq \mathbb{E}_{s \sim d^{\pi^\star}} \frac{1}{1 - \gamma} \|\pi^\star(\cdot|s) - \widehat{\pi}(\cdot|s)\|_1$$

$$\leq \frac{1}{1 - \gamma} \sqrt{\mathbb{E}_{s \sim d^{\pi^\star}} \|\pi^\star(\cdot|s) - \widehat{\pi}(\cdot|s)\|_1^2}$$

$$= \frac{1}{1 - \gamma} \sqrt{4\mathbb{E}_{s \sim d^{\pi^\star}} \|\pi^\star(\cdot|s) - \widehat{\pi}(\cdot|s)\|_{tv}^2}.$$

where we use the fact that $\sup_{s,a,\pi} |A^\pi(s,a)| \leq \frac{1}{1-\gamma}$, and the fact that $(\mathbb{E}[x])^2 \leq \mathbb{E}[x^2]$.

Using Theorem 15.2 and rearranging terms conclude the proof. ∎

For Behavior cloning, from the supervised learning error the quadratic polynomial dependency on the effect horizon $1/(1 - \gamma)$ is not avoidable in worst case [Ross and Bagnell, 2010]. This is often referred as the distribution shift issue in the literature. Note that $\widehat{\pi}$ is trained under $d^{\pi^\star}$, but during execution, $\widehat{\pi}$ makes prediction on states that are generated by itself, i.e., $d^{\widehat{\pi}}$, instead of the training distribution $d^{\pi^\star}$.

## 15.3 The Hybrid Setting: Statistical Benefit and Algorithm

The question we want to answer here is that if we know the underlying MDP's transition $P$ (but the reward is still unknown), can we improve Behavior Cloning? In other words:

Instead of a quadratic dependency on the effective horizon, we should expect a linear dependency on the effective horizon. The key benefit of the known transition is that we can test our policy using the known transition to see how far away we are from the expert's demonstrations, and then use the known transition to plan to move closer to the expert demonstrations.

In this section, we consider a statistical efficient, but computationally intractable algorithm, which we use to demonstrate that informationally theoretically, by interacting with the underlying known transition, we can do better than Behavior Cloning. In the next section, we will introduce a popular and computationally efficient algorithm Maximum Entropy Inverse Reinforcement Learning (MaxEnt-IRL) which operates under this setting (i.e., expert demonstrations with a known transition).

We start from the same policy class $\Pi$ as we have in the BC algorithm, and again we assume realizability (Assumption 15.1) and $\Pi$ is discrete.

Since we know the transition $P$, information theoretically, for any policy $\pi$, we have $d^\pi$ available (though it is computationally intractable to compute $d^\pi$ for large scale MDPs). We have $(s_i^\star, a_i^\star)_{i=1}^M \sim d^{\pi^\star}$.

Below we present an algorithm which we called Distribution Matching with Scheffé Tournament (DM-ST).

For any two policies $\pi$ and $\pi'$, we denote $f_{\pi,\pi'}$ as the following witness function:

$$f_{\pi,\pi'} := \mathrm{argmax}_{f:\|f\|_\infty \leq 1} \left[ \mathbb{E}_{s,a\sim d^\pi} f(s,a) - \mathbb{E}_{s,a\sim d^{\pi'}} f(s,a) \right].$$

We denote the set of witness functions as:

$$\mathcal{F} = \{f_{\pi,\pi'} : \pi, \pi' \in \Pi, \pi \neq \pi'\}.$$

Note that $|\mathcal{F}| \leq |\Pi|^2$.

DM-ST selects $\widehat{\pi}$ using the following procedure:

$$\text{DM-ST:} \quad \widehat{\pi} \in \mathrm{argmin}_{\pi\in\Pi} \left[ \max_{f\in\mathcal{F}} \left[ \mathbb{E}_{s,a\sim d^\pi} f(s,a) - \frac{1}{M}\sum_{i=1}^M f(s_i^\star, a_i^\star) \right] \right].$$

The following theorem captures the sample complexity of DM-ST.

**Theorem 15.4** (Sample Complexity of DM-ST). *With probability at least $1-\delta$, DM-ST finds a policy $\widehat{\pi}$ such that:*

$$V^\star - V^{\widehat{\pi}} \leq \frac{4}{1-\gamma}\sqrt{\frac{2\ln(|\Pi|) + \ln\left(\frac{1}{\delta}\right)}{M}}.$$

**Proof:** The proof basically relies on a uniform convergence argument over $\mathcal{F}$ of which the size is $|\Pi|^2$. First we note that for all policy $\pi \in \Pi$:

$$\max_{f\in\mathcal{F}} \left[ \mathbb{E}_{s,a\sim d^\pi} f(s,a) - \mathbb{E}_{s,a\sim d^\star} f(s,a) \right] = \max_{f:\|f\|_\infty\leq 1} \left[ \mathbb{E}_{s,a\sim d^\pi} f(s,a) - \mathbb{E}_{s,a\sim d^\star} f(s,a) \right] = \left\| d^\pi - d^{\pi^\star} \right\|_1,$$

where the first equality comes from the fact that $\mathcal{F}$ includes $\arg\max_{f:\|f\|_\infty\leq 1} \left[ \mathbb{E}_{s,a\sim d^\pi} f(s,a) - \mathbb{E}_{s,a\sim d^\star} f(s,a) \right]$.

Via Hoeffding's inequality and a union bound over $\mathcal{F}$, we get that with probability at least $1-\delta$, for all $f \in \mathcal{F}$:

$$\left| \frac{1}{M}\sum_{i=1}^M f(s_i^\star, a_i^\star) - \mathbb{E}_{s,a\sim d^\star} f(s,a) \right| \leq 2\sqrt{\frac{\ln(|\mathcal{F}|/\delta)}{M}} := \epsilon_{stat}.$$

Denote $\widehat{f} := \arg\max_{f \in \mathcal{F}} \left[ \mathbb{E}_{s,a \sim d^{\widehat{\pi}}} f(s,a) - \mathbb{E}_{s,a \sim d^\star} f(s,a) \right]$, and $\widetilde{f} := \arg\max_{f \in \mathcal{F}} \mathbb{E}_{s,a \sim d^{\widehat{\pi}}} f(s,a) - \frac{1}{M} \sum_{i=1}^{M} f(s_i, a_i)$. Hence, for $\widehat{\pi}$, we have:

$$\left\| d^{\widehat{\pi}} - d^\star \right\|_1 = \mathbb{E}_{s,a \sim d^{\widehat{\pi}}} \widehat{f}(s,a) - \mathbb{E}_{s,a \sim d^\star} \widehat{f}(s,a) \le \mathbb{E}_{s,a \sim d^{\widehat{\pi}}} \widehat{f}(s,a) - \frac{1}{M} \sum_{i=1}^{M} \widehat{f}(s_i^\star, a_i^\star) + \epsilon_{stat}$$

$$\le \mathbb{E}_{s,a \sim d^{\widehat{\pi}}} \widetilde{f}(s,a) - \frac{1}{M} \sum_{i=1}^{M} \widetilde{f}(s_i, a_i) + \epsilon_{stat}$$

$$\le \mathbb{E}_{s,a \sim d^{\pi^\star}} \widetilde{f}(s,a) - \frac{1}{M} \sum_{i=1}^{M} \widetilde{f}(s_i, a_i) + \epsilon_{stat}$$

$$\le \mathbb{E}_{s,a \sim d^{\pi^\star}} \widetilde{f}(s,a) - \mathbb{E}_{s,a \sim d^\star} \widetilde{f}(s,a) + 2\epsilon_{stat} = 2\epsilon_{stat},$$

where in the third inequality we use the optimality of $\widehat{\pi}$.

Recall that $V^\pi = \mathbb{E}_{s,a \sim d^\pi} r(s,a)/(1 - \gamma)$, we have:

$$V^{\widehat{\pi}} - V^\star = \frac{1}{1 - \gamma} \left( \mathbb{E}_{s,a \sim d^{\widehat{\pi}}} r(s,a) - \mathbb{E}_{s,a \sim d^\star} r(s,a) \right) \le \frac{\sup_{s,a} |r(s,a)|}{1 - \gamma} \left\| d^{\widehat{\pi}} - d^\star \right\|_1 \le \frac{2}{1 - \gamma} \epsilon_{stat}.$$

This concludes the proof. ∎

Note that above theorem confirms the statistical benefit of having the access to a known transition: comparing to the classic Behavior Cloning algorithm, the approximation error of DM-ST only scales linearly with respect to horizon $1/(1 - \gamma)$ instead of quadratically.

### 15.3.1 Extension to Agnostic Setting

So far we focused on agnostic learning setting. What would happen if $\pi^\star \notin \Pi$? We can still run our DM-ST algorithm as is. We state the sample complexity of DM-ST in agnostic setting below.

**Theorem 15.5** (Agnostic Guarantee of DM-ST). *Assume $\Pi$ is finite, but $\pi^\star \notin \Pi$. With probability at least $1 - \delta$, DM-ST learns a policy $\widehat{\pi}$ such that:*

$$V^\star - V^{\widehat{\pi}} \le \frac{1}{1 - \gamma} \| d^\star - d^{\widetilde{\pi}} \|_1 \le \frac{3}{1 - \gamma} \min_{\pi \in \Pi} \| d^\pi - d^\star \|_1 + \widetilde{O} \left( \frac{1}{1 - \gamma} \sqrt{\frac{\ln(|\Pi| + \ln(1/\delta))}{M}} \right).$$

**Proof:** We first define some terms below. Denote $\widetilde{\pi} := \mathrm{argmin}_{\pi \in \Pi} \| d^\pi - d^\star \|_1$. Let us denote:

$$\widetilde{f} = \mathrm{argmax}_{f \in \mathcal{F}} \left[ \mathbb{E}_{s,a \sim d^{\widehat{\pi}}} f(s,a) - \mathbb{E}_{s,a \sim d^{\widetilde{\pi}}} f(s,a) \right],$$

$$\overline{f} = \mathrm{argmax}_{f \in \mathcal{F}} \left[ \mathbb{E}_{s,a \sim d^{\widehat{\pi}}} f(s,a) - \frac{1}{M} \sum_{i=1}^{M} f(s_i^\star, a_i^\star) \right],$$

$$f' = \arg\max_{f \in \mathcal{F}} \left[ \mathbb{E}_{s,a \sim d^{\widetilde{\pi}}} [f(s,a)] - \frac{1}{M} \sum_{i=1}^{M} f(s_i^\star, a_i^\star) \right].$$

144

Starting with a triangle inequality, we have:

$$
\begin{aligned}
\left\| d^{\widehat{\pi}} - d^{\pi^\star} \right\|_1 &\leq \left\| d^{\widehat{\pi}} - d^{\widetilde{\pi}} \right\|_1 + \left\| d^{\widetilde{\pi}} - d^{\pi^\star} \right\|_1 \\
&= \mathbb{E}_{s,a\sim d^{\widehat{\pi}}}\left[ \widetilde{f}(s,a) \right] - \mathbb{E}_{s,a\sim d^{\widetilde{\pi}}}\left[ \widetilde{f}(s,a) \right] + \left\| d^{\widetilde{\pi}} - d^{\pi^\star} \right\|_1 \\
&= \mathbb{E}_{s,a\sim d^{\widehat{\pi}}}\left[ \widetilde{f}(s,a) \right] - \frac{1}{M}\sum_{i=1}^{M} \widetilde{f}(s_i^\star, a_i^\star) + \frac{1}{M}\sum_{i=1}^{M} \widetilde{f}(s_i^\star, a_i^\star) - \mathbb{E}_{s,a\sim d^{\widetilde{\pi}}}\left[ \widetilde{f}(s,a) \right] + \left\| d^{\widetilde{\pi}} - d^{\pi^\star} \right\|_1 \\
&\leq \mathbb{E}_{s,a\sim d^{\widehat{\pi}}}\left[ \overline{f}(s,a) \right] - \frac{1}{M}\sum_{i=1}^{M} \overline{f}(s_i^\star, a_i^\star) + \frac{1}{M}\sum_{i=1}^{M} \widetilde{f}(s_i^\star, a_i^\star) - \mathbb{E}_{s,a\sim d^\star} \widetilde{f}(s,a) \\
&\quad + \left[ \mathbb{E}_{s,a\sim d^\star} \widetilde{f}(s,a) - \mathbb{E}_{s,a\sim d^{\widetilde{\pi}}}\left[ \widetilde{f}(s,a) \right] \right] + \left\| d^{\widetilde{\pi}} - d^{\pi^\star} \right\|_1 \\
&\leq \mathbb{E}_{s,a\sim d^{\widetilde{\pi}}}\left[ f'(s,a) \right] - \frac{1}{M}\sum_{i=1}^{M} f'(s_i^\star, a_i^\star) + 2\sqrt{\frac{\ln(|\mathcal{F}|/\delta)}{M}} + 2\| d^{\widetilde{\pi}} - d^\star \|_1 \\
&\leq \mathbb{E}_{s,a\sim d^{\widetilde{\pi}}}\left[ f'(s,a) \right] - \mathbb{E}_{s,a\sim d^\star}\left[ f'(s,a) \right] + 4\sqrt{\frac{\ln(|\mathcal{F}|/\delta)}{M}} + 2\| d^{\widetilde{\pi}} - d^\star \|_1 \\
&\leq 3\| d^\star - d^{\widetilde{\pi}} \|_1 + 4\sqrt{\frac{\ln(|\mathcal{F}|/\delta)}{M}},
\end{aligned}
$$

where the first inequality uses the definition of $\overline{f}$, the second inequality uses the fact that $\widehat{\pi}$ is the minimizer of $\max_{f\in\mathcal{F}} \mathbb{E}_{s,a\sim d^\pi} f(s,a) - \frac{1}{M}\sum_{i=1}^{M} f(s_i^\star, a_i^\star)$, along the way we also use Heoffding's inequality where $\forall f \in \mathcal{F}$, $|\mathbb{E}_{s,a\sim d^\star} f(s,a) - \sum_{i=1}^{M} f(s_i^\star, a_i^\star)| \leq 2\sqrt{\ln(|\mathcal{F}|/\delta)/M}$, with probability at least $1-\delta$. ∎

As we can see, comparing to the realizable setting, here we have an extra term that is related to $\min_{\pi\in\Pi} \| d^\pi - d^\star \|_1$. Note that the dependency on horizon also scales linearly in this case. In general, the constant 3 in front of $\min_{\pi\in\Pi} \| d^\pi - d^\star \|_1$ is not avoidable in Scheffé estimator [Devroye and Lugosi, 2012].


## 15.4 Maximum Entropy Inverse Reinforcement Learning


Similar to Behavior cloning, we assume we have a dataset of state-action pairs from expert $\mathcal{D}^\star = \{ s_i^\star, a_i^\star \}_{i=1}^{N}$ where $s_i^\star, a_i^\star \sim d^{\pi^\star}$. Different from Behavior cloning, here we assume that we have access to the underlying MDP's transition, i.e., we assume transition is known and we can do planning if we were given a cost function.

We assume that we are given a state-action feature mapping $\phi : \mathcal{S}\times\mathcal{A} \mapsto \mathbb{R}^d$ (this can be extended infinite dimensional feature space in RKHS, but we present finite dimensional setting for simplicity). We assume the true ground truth cost function as $c(s,a) := \theta^\star \cdot \phi(s,a)$ with $\|\theta^\star\|_2 \leq 1$ and $\theta^\star$ being unknown.

The goal of the learner is to compute a policy $\pi : \mathcal{S} \mapsto \Delta(A)$ such that when measured under the true cost function, it performs as good as the expert, i.e., $\mathbb{E}_{s,a\sim d^\pi} \theta^\star \cdot \phi(s,a) \approx \mathbb{E}_{s,a\sim d^{\pi^\star}} \theta^\star \cdot \phi(s,a)$.

We will focus on finite horizon MDP setting in this section. We denote $\rho^\pi(\tau)$ as the trajectory distribution induced by $\pi$ and $d^\pi$ as the average state-action distribution induced by $\pi$.

### 15.4.1 MaxEnt IRL: Formulation and The Principle of Maximum Entropy

MaxEnt IRL uses the principle of Maximum Entropy and poses the following policy optimization optimization program:

$$\max_{\pi} -\sum_{\tau} \rho^{\pi}(\tau) \ln \rho^{\pi}(\tau),$$

$$s.t., \mathbb{E}_{s,a\sim d^{\pi}} \phi(s,a) = \sum_{i=1}^{N} \phi(s_i^{\star}, a_i^{\star})/N.$$

Note that $\sum_{i=1}^{N} \phi(s_i^{\star}, a_i^{\star})/N$ is an unbiased estimate of $\mathbb{E}_{s,a\sim d^{\pi^{\star}}} \phi(s,a)$. MaxEnt-IRL searches for a policy that maximizes the entropy of its trajectory distribution subject to a moment matching constraint.

Note that there could be many policies that satisfy the moment matching constraint, i.e., $\mathbb{E}_{s,a\sim d^{\pi}} \phi(s,a) = \mathbb{E}_{s,a\sim d^{\pi^{\star}}} \phi(s,a)$, and any feasible solution is guaranteed to achieve the same performance of the expert under the ground truth cost function $\theta^{\star} \cdot \phi(s,a)$. The maximum entropy objective ensures that the solution is unique.

Using the Markov property, we notice that:

$$\text{argmax}_{\pi} -\sum_{\tau} \rho^{\pi}(\tau) \ln \rho^{\pi}(\tau) = \text{argmax}_{\pi} -\mathbb{E}_{s,a\sim d^{\pi}} \ln \pi(a|s).$$

This, we can rewrite the MaxEnt-IRL as follows:

$$\min_{\pi} \mathbb{E}_{s,a\sim d^{\pi}} \ln \pi(a|s), \tag{0.2}$$

$$s.t., \mathbb{E}_{s,a\sim d^{\pi}} \phi(s,a) = \sum_{i=1}^{N} \phi(s_i^{\star}, a_i^{\star})/N. \tag{0.3}$$

### 15.4.2 Algorithm

To better interpret the objective function, below we replace $\sum_i \phi(s_i^{\star}, a_i^{\star})/N$ by its expectation $\mathbb{E}_{s,a\sim d^{\pi^{\star}}} \phi(s,a)$. Note that we can use standard concentration inequalities to bound the difference between $\sum_i \phi(s_i^{\star}, a_i^{\star})/N$ and $\mathbb{E}_{s,a\sim d^{\pi^{\star}}} \phi(s,a)$.

Using Lagrange multipliers, we can rewrite the constrained optimization program in Eq. 0.2 as follows:

$$\min_{\pi} \mathbb{E}_{s,a\sim d^{\pi}} \ln \pi(a|s) + \max_{\theta} \mathbb{E}_{s,a\sim d^{\pi}} \theta^{\top} \phi(s,a) - \mathbb{E}_{s,a\sim d^{\pi^{\star}}} \theta^{\top} \phi(s,a).$$

The above objective conveys a clear goal of our imitation learning problem: we are searching for $\pi$ that minimizes the MMD between $d^{\pi}$ and $d^{\pi^{\star}}$ with a (negative) entropy regularization on the trajectory distribution of policy $\pi$.

To derive an algorithm that optimizes the above objective, we first again swap the min max order via the minimax theorem:

$$\max_{\theta} \left( \min_{\pi} \mathbb{E}_{s,a\sim d^{\pi}} \theta^{\top} \phi(s,a) - \mathbb{E}_{s,a\sim d^{\pi^{\star}}} \theta^{\top} \phi(s,a) + \mathbb{E}_{s,a\sim d^{\pi}} \ln \pi(a|s) \right).$$

The above objective proposes a natural algorithm where we perform projected gradient ascent on $\theta$, while perform best response update on $\pi$, i.e., given $\theta$, we solve the following planning problem:

$$\text{argmin}_{\pi} \mathbb{E}_{s,a\sim d^{\pi}} \theta^{\top} \phi(s,a) + \mathbb{E}_{s,a\sim d^{\pi}} \ln \pi(a|s). \tag{0.4}$$

Note that the above objective can be understood as planning with cost function $\theta^\top \phi(s, a)$ with an additional negative entropy regularization on the trajectory distribution. On the other hand, given $\pi$, we can compute the gradient of $\theta$, which is simply the difference between the expected features:

$$\mathbb{E}_{s,a\sim d^\pi} \phi(s, a) - \mathbb{E}_{s,a\sim d^{\pi^\star}} \phi(s, a),$$

which gives the following gradient ascent update on $\theta$:

$$\theta := \theta + \eta \left( \mathbb{E}_{s,a\sim d^\pi} \phi(s, a) - \mathbb{E}_{s,a\sim d^{\pi^\star}} \phi(s, a) \right). \tag{0.5}$$

**Algorithm of MaxEnt-IRL**  MaxEnt-IRL updates $\pi$ and $\theta$ alternatively using Eq. 0.4 and Eq. 0.5, respectively. We summarize the algorithm in Alg. 9. Note that for the stochastic gradient of $\theta_t$, we see that it is the empirical average feature difference between the current policy $\pi_t$ and the expert policy $\pi^\star$.

---

**Algorithm 9** MaxEnt-IRL

---

**Input:** Expert data $\mathcal{D}^\star = \{s_i^\star, a_i^\star\}_{i=1}^M$, MDP $\mathcal{M}$, parameters $\beta, \eta, N$.
1:  Initialize $\theta_0$ with $\|\theta_0\|_2 \leq 1$.
2:  **for** $t = 1, 2, \ldots,$ **do**
3:      Entropy-regularized Planning with cost $\theta_t^\top \phi(s, a)$: $\pi_t \in \mathrm{argmin}_\pi \mathbb{E}_{s,a\sim d^\pi} \left[ \theta_t^\top \phi(s, a) + \beta \ln \pi(a|s) \right]$.
4:      Draw samples $\{s_i, a_i\}_{i=1}^N \sim d^{\pi_t}$ by executing $\pi_t$ in $\mathcal{M}$.
5:      Stochastic Gradient Update: $\theta' = \theta_t + \eta \left[ \frac{1}{N} \sum_{i=1}^N \phi(s_i, a_i) - \frac{1}{M} \sum_{i=1}^M \phi(s_i^\star, a_i^\star) \right]$.
6:  **end for**

---

Note that Alg. 9 uses an entropy-regularized planning oracle. Below we discuss how to implement such entropy-regularized planning oracle via dynamic programming.

## 15.4.3   Maximum Entropy RL: Implementing the Planning Oracle in Eq. 0.4

The planning oracle in Eq. 0.4 can be implemented in a value iteration fashion using Dynamic Programming. We denote $c(s, a) := \theta \cdot \phi(s, a)$.

We are interested in implementing the following planning objective:

$$\mathrm{argmin}_\pi \mathbb{E}_{s,a\sim d^\pi} \left[ c(s, a) + \ln \pi(a|s) \right]$$

The subsection has its own independent interests beyond the framework of imitation learning. This maximum entropy regularized planning formulation is widely used in RL as well and it is well connected to the framework of RL as Inference.

As usually, we start from the last time step $H - 1$. For any policy $\pi$ and any state-action $(s, a)$, we have the cost-to-go $Q_{H-1}^\pi(s, a)$:

$$Q_{H-1}^\pi(s, a) = c(s, a) + \ln \pi(a|s), \quad V_{H-1}^\pi(s) = \sum_a \pi(a|s) \left( c(s, a) + \ln \pi(a|s) \right).$$

We have:

$$V_{H-1}^\star(s) = \min_{\rho \in \Delta(\mathcal{A})} \sum_a \rho(a) c(s, a) + \rho(a) \ln \rho(a). \tag{0.6}$$

Take gradient with respect to $\rho$, set it to zero and solve for $\rho$, we get:

$$\pi_{H-1}^\star(a|s) \propto \exp\left( -c(s, a) \right), \forall s, a.$$

Substitute $\pi^\star_{H-1}$ back to the expression in Eq. 0.6, we get:

$$V^\star_{H-1}(s) = -\ln\left(\sum_a \exp\left(-c(s,a)\right)\right),$$

i.e., we apply a softmin operator rather than the usual min operator (recall here we are minimizing cost).

With $V^\star_{h+1}$, we can continue to $h$. Denote $Q^\star(s,a)$ as follows:

$$Q^\star_h(s,a) = r(s,a) + \mathbb{E}_{s'\sim P(\cdot|s,a)}V^\star_{h+1}(s').$$

For $V^\star_h$, we have:

$$V^\star_h(s) = \min_{\rho\in\Delta(\mathcal{A})}\sum_a \rho(a)\left(c(s,a) + \ln\rho(a) + \mathbb{E}_{s'\sim P(\cdot|s,a)}V^\star_{h+1}(s')\right).$$

Again we can show that the minimizer of the above program has the following form:

$$\pi^\star_h(a|s) \propto \exp\left(-Q^\star_h(s,a)\right). \tag{0.7}$$

Substitute $\pi^\star_h$ back to $Q^\star_h$, we can show that:

$$V^\star_h(s) = -\ln\left(\sum_a \exp\left(-Q^\star_h(s,a)\right)\right),$$

where we see again that $V^\star_h$ is based on a softmin operator on $Q^\star$.

Thus the *soft value iteration* can be summarized below:

Soft Value Iteration:

$$Q^\star_{H-1}(s,a) = c(s,a), \quad \pi^\star_h(a|s) \propto \exp(-Q^\star_h(s,a)), \quad V^\star_h(s) = -\ln\left(\sum_a \exp(-Q^\star_h(s,a))\right), \forall h.$$

We can continue the above procedure to $h = 0$, which gives us the optimal policies, all in the form of Eq. 0.7

## 15.5 Interactive Imitation Learning: AggreVaTe and Its Statistical Benefit over Offline IL Setting

We study the Interactive Imitation Learning setting where we have an expert policy that can be queried at any time during training, and we also have access to the ground truth reward signal.

We present AggreVaTe (Aggregate Values to Imitate) first and then analyze its sample complexity under the realizable setting.

Again we start with a realizable policy class $\Pi$ that is discrete. We denote $\Delta(\Pi)$ as the convex hull of $\Pi$ and each policy $\pi \in \Delta(\Pi)$ is a mixture policy represented by a distribution $\rho \in \mathbb{R}^{|\Pi|}$ with $\rho[i] \geq 0$ and $\|\rho\|_1 = 1$. With this parameterization, our decision space simply becomes $\Delta(|\Pi|)$, i.e., any point in $\Delta(|\Pi|)$ corresponds to a mixture policy. Notation wise, given $\rho \in \Delta(|\Pi|)$, we denote $\pi_\rho$ as the corresponding mixture policy. We denote $\pi_i$ as the i-th policy in $\Pi$. We denote $\rho[i]$ as the $i$-th element of the vector $\rho$.

**Algorithm 10** AggreVaTe

---
**Input:** The interactive expert, regularization $\lambda$
1: Initialize $\rho_0$ to be a uniform distribution.
2: **for** $t = 0, 2, \ldots,$ **do**
3:     Sample $s_t \sim d^{\pi_{\rho_t}}$
4:     Query expert to get $A^\star(s_t, a)$ for all $a \in \mathcal{A}$
5:     Policy update: $\rho_{t+1} = \text{argmax}_{\rho \in \Delta(|\Pi|)} \sum_{j=0}^{t} \sum_{i=1}^{|\Pi|} \rho[i] \left( \mathbb{E}_{a \sim \pi_i(\cdot|s_t)} A^\star(s_t, a) \right) - \lambda \sum_{i=1}^{|\Pi|} \rho[i] \ln(\rho[i])$
6: **end for**

---

AggreVaTe assumes an interactive expert from whom I can query for action feedback. Basically, given a state $s$, let us assume that expert returns us the advantages of all actions, i.e., *one query of expert oracle at any state $s$ returns $A^\star(s, a), \forall a \in \mathcal{A}$.*[1]

The algorithm in summarized in Alg. 10.

To analyze the algorithm, let us introduce some additional notations. Let us denote $\ell_t(\rho) = \sum_{i=1}^{|\Pi|} \rho[i] \mathbb{E}_{a \sim \pi_i(\cdot|s)} A^\star(s_t, a)$, which is dependent on state $s_t$ generated at iteration $t$, and is a linear function with respect to decision variable $\rho$. AggreVaTe is essentially running the specific online learning algorithm, Follow-the-Regularized Leader (FTRL) (e.g., see [Shalev-Shwartz, 2011]) with Entropy regularization:

$$\rho_{t+1} = \text{argmax}_{\rho \in \Delta(|\Pi|)} \sum_{i=0}^{t} \ell_t(\rho) + \lambda \text{Entropy}(\rho).$$

Denote $c = \max_{s,a} A^\pi(s, a)$. This implies that $\sup_{\rho, t} \|\ell_t(\rho)\| \leq c$. FTRL with linear loss functions and entropy regularization gives the following deterministic regret guarantees ([Shalev-Shwartz, 2011]):

$$\max_\rho \sum_{t=0}^{T-1} \ell_t(\rho) - \sum_{t=0}^{T-1} \ell_t(\rho_t) \leq c\sqrt{\log(|\Pi|)T}. \tag{0.8}$$

We will analyze AggreVaTe's sample complexity using the above result.

**Theorem 15.6** (Sample Complexity of AggreVaTe). *Denote $c = \sup_{s,a} |A^\star(s, a)|$. Let us denote $\epsilon_\Pi$ and $\epsilon_{stat}$ as follows:*

$$\epsilon_\Pi := \max_{\rho \in \Delta(|\Pi|)} \frac{1}{M} \sum_{t=0}^{M-1} \ell_t(\rho^\star), \quad \epsilon_{stat} := \sqrt{\frac{\ln(|\Pi|)/\delta}{M}} + 2\sqrt{\frac{\ln(1/\delta)}{M}}.$$

*With probability at least $1 - \delta$, after $M$ iterations (i.e., $M$ calls of expert oracle), AggreVaTe finds a policy $\widehat{\pi}$ such that:*

$$V^\star - V^{\widehat{\pi}} \leq \frac{c}{1 - \gamma} \epsilon_{stat} - \frac{1}{1 - \gamma} \epsilon_\Pi.$$

**Proof:** At each iteration $t$, let us define $\tilde{\ell}_t(\rho_t)$ as follows:

$$\tilde{\ell}_t(\rho_t) = \mathbb{E}_{s \sim d^{\pi_{\rho_t}}} \sum_{i=1}^{|\Pi|} \rho_t[i] \mathbb{E}_{a \sim \pi_i(\cdot|s)} A^\star(s, a)$$

---

[1] Technically one cannot use one query to get $A^\star(s, a)$ for all $a$. But one can use importance weighting to get an unbiased estimate of $A^\star(s, a)$ for all $a$ via just one expert roll-out. For analysis simplicity, we assume one expert query at $s$ returns the whole vector $A^\star(s, \cdot) \in \mathbb{R}^{|\mathcal{A}|}$.

Denote $\mathbb{E}_t[\ell_t(\rho_t)]$ as the conditional expectation of $\ell_t(\rho_t)$, conditioned on all history up and including the end of iteration $t-1$. Thus, we have $\mathbb{E}_t[\ell_t](\rho_t) = \tilde{\ell}_t(\rho_t)$ as $\rho_t$ only depends on the history up to the end of iteration $t-1$. Also note that $|\ell_t(\rho)| \leq c$. Thus by Azuma-Hoeffding inequality (Theorem A.5), we get:

$$\left| \frac{1}{M} \sum_{t=0}^{M-1} \tilde{\ell}_t(\rho_t) - \frac{1}{T} \sum_{t=0}^{M-1} \ell_t(\rho_t) \right| \leq 2c\sqrt{\frac{\ln(1/\delta)}{M}},$$

with probability at least $1 - \delta$. Now use Eq. 0.8, and denote $\rho^\star = \operatorname{argmin}_{\rho \in \Delta(\Pi)} \frac{1}{M} \sum_{t=0}^{M-1} \ell_t(\rho)$, i.e., the best minimizer that minimizes the average loss $\sum_{t=0}^{M-1} \ell_t(\rho)/M$. we get:

$$\frac{1}{M} \sum_{t=0}^{M-1} \tilde{\ell}_t(\rho_t) \geq \frac{1}{M} \sum_{t=0}^{M-1} \ell_t(\rho_t) - 2c\sqrt{\frac{\ln(1/\delta)}{M}} \geq \frac{1}{M} \sum_{t=0}^{M-1} \ell_t(\rho^\star) - c\sqrt{\frac{\ln(|\Pi|)}{M}} - 2c\sqrt{\frac{\ln(1/\delta)}{M}},$$

which means that there must exist a $\hat{t} \in \{0, \ldots, M-1\}$, such that:

$$\tilde{\ell}_{\hat{t}}(\rho_{\hat{t}}) \geq \frac{1}{M} \sum_{t=0}^{M-1} \ell_t(\rho^\star) - c\sqrt{\frac{\ln(|\Pi|)}{M}} - 2c\sqrt{\frac{\ln(1/\delta)}{M}}.$$

Now use Performance difference lemma, we have:

$$(1 - \gamma)\left(-V^\star + V^{\pi_{\rho_{\hat{t}}}}\right) = \mathbb{E}_{s \sim d^{\pi_{\rho_{\hat{t}}}}} \sum_{i=1}^{|\Pi|} \rho_{\hat{t}}[i] \mathbb{E}_{a \sim \pi_i(\cdot|s)} A^\star(s,a) = \ell_{\hat{t}}(\rho_{\hat{t}})$$

$$\geq \frac{1}{M} \sum_{t=0}^{M-1} \ell_t(\rho^\star) - c\sqrt{\frac{\ln(|\Pi|)}{T}} - 2c\sqrt{\frac{\ln(1/\delta)}{T}}.$$

Rearrange terms, we get:

$$V^\star - V^{\pi_{\rho_{\hat{t}}}} \leq -\frac{1}{1-\gamma}\left[\frac{1}{M} \sum_{t=0}^{M-1} \ell_t(\rho^\star)\right] + \frac{c}{1-\gamma}\left[\sqrt{\frac{\ln(|\Pi|)/\delta}{M}} + 2\sqrt{\frac{\ln(1/\delta)}{M}}\right],$$

which concludes the proof. ∎

**Remark** We analyze the $\epsilon_\Pi$ by discussing realizable setting and non-realizable setting separately. When $\Pi$ is realizable, i.e., $\pi^\star \in \Pi$, by the definition of our loss function $\ell_t$, we immediately have $\sum_{i=1}^{M} \ell_t(\rho^\star) \geq 0$ since $A^\star(s, \pi^\star(s)) = 0$ for all $s \in \mathcal{S}$. Moreover, when $\pi^\star$ is not the globally optimal policy, it is possible that $\epsilon_\Pi := \sum_{i=1}^{M} \ell_t(\rho^\star)/M > 0$, which implies that when $M \to \infty$ (i.e., $\epsilon_{stat} \to 0$), AggreVaTe indeed can learn a policy that outperforms the expert policy $\pi^\star$. In general when $\pi^\star \notin \Pi$, there might not exist a mixture policy $\rho$ that achieves positive advantage against $\pi^\star$ under the $M$ training samples $\{s_0, \ldots, s_{M-1}\}$. In this case, $\epsilon_\Pi < 0$.

Does AggreVaTe avoids distribution shift? Under realizable setting, note that the bound explicitly depends on $\sup_{s,a} |A^\star(s,a)|$ (i.e., it depends on $|\ell_t(\rho_t)|$ for all $t$). In worst case, it is possible that $\sup_{s,a} |A^\star(s,a)| = \Theta(1/(1-\gamma))$, which implies that AggreVaTe could suffer a quadratic horizon dependency, i.e., $1/(1-\gamma)^2$. Note that DM-ST provably scales linearly with respect to $1/(1-\gamma)$, but DM-ST requires a stronger assumption that the transition $P$ is known.

When $\sup_{s,a} |A^\star(s,a)| = o(1/(1-\gamma))$, then AggreVaTe performs strictly better than BC. This is possible when the MDP is mixing quickly under $\pi^\star$, or $Q^\star$ is L- Lipschitz continuous, i.e., $|Q^\star(s,a) - Q^\star(s,a')| \leq L\|a - a'\|$, with bounded range in action space, e.g., $\sup_{a,a'} \|a - a'\| \leq \beta \in \mathbb{R}^+$. In this case, if $L$ and $\beta$ are independent of $1/(1-\gamma)$, then $\sup_{s,a} |A^\star(s,a)| \leq L\beta$, which leads to a $\frac{\beta L}{1-\gamma}$ dependency.

When $\pi^\star \notin \Pi$, the agnostic result of AggreVaTe and the agnostic result of DM-ST is not directly comparable. Note that the model class error that DM-ST suffers is algorithmic-independent, i.e., it is $\min_{\pi \in \Pi} \|d^\pi - d^{\pi^\star}\|_1$ and it only depends on $\pi^\star$ and $\Pi$, while the model class $\epsilon_\Pi$ in AggreVaTe is algorithmic path-dependent, i.e., in additional to $\pi^\star$ and $\Pi$, it depends the policies $\pi_1, \pi_2, \ldots$ computed during the learning process. Another difference is that $\min_{\pi \in \Pi} \|d^\pi - d^{\pi^\star}\|_1 \in [0, 2]$, while $\epsilon_\Pi$ indeed could scale linearly with respect to $1/(1 - \gamma)$, i.e., $\epsilon_\pi \in [-\frac{1}{1-\gamma}, 0]$ (assume $\pi^\star$ is the globally optimal policy for simplicity).

## 15.6 Bibliographic Remarks and Further Readings

Behavior cloning was used in autonomous driving back in 1989 [Pomerleau, 1989], and the distribution shift and compounding error issue was studied by Ross and Bagnell [2010]. Ross and Bagnell [2010], Ross et al. [2011] proposed using interactive experts to alleviate the distribution shift issue.

MaxEnt-IRL was proposed by Ziebart et al. [2008]. The original MaxEnt-IRL focuses on deterministic MDPs and derived distribution over trajectories. Later Ziebart et al. [2010] proposed the Principle of Maximum Causal Entropy framework which captures MDPs with stochastic transition. MaxEnt-IRL has been widely used in real applications (e.g., [Kitani et al., 2012, Ziebart et al., 2009]).

To the best of our knowledge, the algorithm Distribution Matching with Scheffé Tournament introduced in this chapter is new here and is the first to demonstrate the statistical benefit (in terms of sample complexity of the expert policy) of the hybrid setting over the pure offline setting with general function approximation.

Recently, there are approaches that extend the linear cost functions used in MaxEnt-IRL to deep neural networks which are treated as discriminators to distinguish between experts datasets and the datasets generated by the policies. Different distribution divergences have been proposed, for instance, JS-divergence [Ho and Ermon, 2016], general f-divergence which generalizes JS-divergence [Ke et al., 2019], and Integral Probability Metric (IPM) which includes Wasserstein distance [Sun et al., 2019b]. While these adversarial IL approaches are promising as they fall into the hybrid setting, it has been observed that empirically, adversarial IL sometimes cannot outperform simple algorithms such as Behavior cloning which operates in pure offline setting (e.g., see experiments results on common RL benchmarks from Brantley et al. [2019]).

Another important direction in IL is to combine expert demonstrations and reward functions together (i.e., perform Imitation together with RL). There are many works in this direction, including learning with pre-collected expert data [Hester et al., 2017, Rajeswaran et al., 2017, Cheng et al., 2018, Le et al., 2018, Sun et al., 2018], learning with an interactive expert [Daumé et al., 2009, Ross and Bagnell, 2014, Chang et al., 2015, Sun et al., 2017, Cheng et al., 2020, Cheng and Boots, 2018]. The algorithm AggreVaTe was originally introduced in [Ross and Bagnell, 2014]. A policy gradient and natural policy gradient version of AggreVaTe are introduced in [Sun et al., 2017]. The precursor of AggreVaTe is the algorithm Data Aggregation (DAgger) [Ross et al., 2011], which leverages interactive experts and a reduction to no-regret online learning, but without assuming access to the ground truth reward signals.

The maximum entropy RL formulation has been widely used in RL literature as well. For instance, Guided Policy Search (GPS) (and its variants) use maximum entropy RL formulation as its planning subroutine [Levine and Koltun, 2013, Levine and Abbeel, 2014]. We refer readers to the excellent survey from Levine [2018] for more details of Maximum Entropy RL formulation and its connections to the framework of RL as Probabilistic Inference.

# Chapter 16

# Linear Quadratic Regulators

This chapter will introduce some of the fundamentals of optimal control for the linear quadratic regulator model. This model is an MDP, with continuous states and actions. While the model itself is often inadequate as a global model, it can be quite effective as a locally linear model (provided our system does not deviate away from the regime where our linear model is reasonable approximation).

The basics of optimal control theory can be found in any number of standards text Anderson and Moore [1990], Evans [2005], Bertsekas [2017]. The treatment of Gauss-Newton and the NPG algorithm are due to Fazel et al. [2018].

## 16.1   The LQR Model

In the standard optimal control problem, a dynamical system is described as

$$x_{t+1} = f_t(x_t, u_t, w_t),$$

where $f_t$ maps a state $x_t \in \mathbb{R}^d$, a control (the action) $u_t \in \mathbb{R}^k$, and a disturbance $w_t$, to the next state $x_{t+1} \in \mathbb{R}^d$, starting from an initial state $x_0$. The objective is to find the control policy $\pi$ which minimizes the long term cost,

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}_\pi \left[ \sum_{t=0}^{H} c_t(x_t, u_t) \right] \\ \text{such that} \quad & x_{t+1} = f_t(x_t, u_t, w_t) \ \ t = 0, \ldots, H. \end{aligned}$$

where $H$ is the time horizon (which can be finite or infinite).

In practice, this is often solved by considering the linearized control (sub-)problem where the dynamics are approximated by

$$x_{t+1} = A_t x_t + B_t u_t + w_t,$$

with the matrices $A_t$ and $B_t$ are derivatives of the dynamics $f$ and where the costs are approximated by a quadratic function in $x_t$ and $u_t$.

This chapter focuses on an important special case: finite and infinite horizon problem referred to as the linear quadratic regulator (LQR) problem. We can view this model as being an local approximation to non-linear model. However, we will analyze these models under the assumption that they are globally valid.

**Finite Horizon LQRs.** The finite horizon LQR problem is given by

$$\text{minimize} \quad \mathbb{E}\left[x_H^\top Q x_H + \sum_{t=0}^{H-1}(x_t^\top Q x_t + u_t^\top R u_t)\right]$$

$$\text{such that} \quad x_{t+1} = A_t x_t + B_t u_t + w_t, \quad x_0 \sim \mathcal{D}, \ w_t \sim N(0, \sigma^2 I),$$

where initial state $x_0 \sim \mathcal{D}$ is assumed to be randomly distributed according to distribution $\mathcal{D}$; the disturbance $w_t \in \mathbb{R}^d$ follows the law of a multi-variate normal with covariance $\sigma^2 I$; the matrices $A_t \in \mathbb{R}^{d \times d}$ and $B_t \in \mathbb{R}^{d \times k}$ are referred to as system (or transition) matrices; $Q \in \mathbb{R}^{d \times d}$ and $R \in \mathbb{R}^{k \times k}$ are both positive definite matrices that parameterize the quadratic costs. Note that this model is a finite horizon MDP, where the $\mathcal{S} = \mathbb{R}^d$ and $\mathcal{A} = \mathbb{R}^k$.

**Infinite Horizon LQRs.** We also consider the infinite horizon LQR problem:

$$\text{minimize} \quad \lim_{H \to \infty} \frac{1}{H} \mathbb{E}\left[\sum_{t=0}^{H}(x_t^\top Q x_t + u_t^\top R u_t)\right]$$

$$\text{such that} \quad x_{t+1} = A x_t + B u_t + w_t, \quad x_0 \sim \mathcal{D}, \ w_t \sim N(0, \sigma^2 I).$$

Note that here we are assuming the dynamics are time homogenous. We will assume that the optimal objective function (i.e. the optimal average cost) is finite; this is referred to as the system being *controllable*. This is a special case of an MDP with an average reward objective.

Throughout this chapter, we assume $A$ and $B$ are such that the optimal cost is finite. Due to the geometric nature of the system dynamics (say for a controller which takes controls $u_t$ that are linear in the state $x_t$), there may exists linear controllers with infinite costs. This instability of LQRs (at least for some $A$ and $B$ and some controllers) leads to that the theoretical analysis often makes various assumptions on $A$ and $B$ in order to guarantee some notion of stability. In practice, the finite horizon setting is more commonly used in practice, particularly due to that the LQR model is only a good local approximation of the system dynamics, where the infinite horizon model tends to be largely of theoretical interest. See Section 16.6.

**The infinite horizon discounted case?** The infinite horizon discounted case tends not to be studied for LQRs. This is largely due to that, for the undiscounted case (with the average cost objective), we have may infinte costs (due to the aforementioned geometric nature of the system dynamics); in such cases, discounting will not necessarily make the average cost finite.

# 16.2 Bellman Optimality: Value Iteration & The Algebraic Riccati Equations

A standard result in optimal control theory shows that the optimal control input can be written as a linear function in the state. As we shall see, this is a consequence of the Bellman equations.

## 16.2.1 Planning and Finite Horizon LQRs

Slightly abusing notation, it is convenient to define the value function and state-action value function with respect to the costs as follows: For a policy $\pi$, a state $x$, and $h \in \{0, \ldots H-1\}$, we define the value function $V_h^\pi : \mathbb{R}^d \to \mathbb{R}$ as

$$V_h^\pi(x) = \mathbb{E}\left[x_H^\top Q x_H + \sum_{t=h}^{H-1}(x_t^\top Q x_t + u_t^\top R u_t) \mid \pi, x_h = x\right],$$

where again expectation is with respect to the randomness of the trajectory, that is, the randomness in state transitions. Similarly, the state-action value (or Q-value) function $Q_h^\pi : \mathbb{R}^d \times \mathbb{R}^k \to \mathbb{R}$ is defined as

$$Q_h^\pi(x, u) = \mathbb{E}\left[x_H^\top Q x_H + \sum_{t=h}^{H-1}(x_t^\top Q x_t + u_t^\top R u_t) \mid \pi, x_h = x, u_h = u\right].$$

We define $V^\star$ and $Q^\star$ analogously.

The following theorem provides a characterization of the optimal policy, via the algebraic Riccati equations. These equations are simply the value iteration algorithm for the special case of LQRs.

**Theorem 16.1.** *(Value Iteration and the Riccati Equations). Suppose $R$ is positive definite. The optimal policy is a linear controller specified by:*

$$\pi^\star(x_t) = -K_t^\star x_t$$

*where*

$$K_t^\star = (B_t^\top P_{t+1} B_t + R)^{-1} B_t^\top P_{t+1} A_t.$$

*Here, $P_t$ can be computed iteratively, in a backwards manner, using the following algebraic Riccati equations, where for $t \in [H]$,*

$$
\begin{aligned}
P_t &:= A_t^\top P_{t+1} A_t + Q - A_t^\top P_{t+1} B_t (B_t^\top P_{t+1} B_t + R)^{-1} B_t^\top P_{t+1} A_t \\
&= A_t^\top P_{t+1} A_t + Q - (K_{t+1}^\star)^\top (B_t^\top P_{t+1} B_t + R) K_{t+1}^\star
\end{aligned}
$$

*and where $P_H = Q$. (The above equation is simply the value iteration algorithm).*

*Furthermore, for $t \in [H]$, we have that:*

$$V_t^\star(x) = x^\top P_t x + \sigma^2 \sum_{h=t+1}^{H} \text{Trace}(P_h).$$

We often refer to $K_t^\star$ as the *optimal control gain* matrices. It is straightforward to generalize the above when $\sigma \geq 0$.

We have assumed that $R$ is strictly positive definite, to avoid have to working with the pseudo-inverse; the theorem is still true when $R = 0$, provided we use a pseudo-inverse.

**Proof:** By the Bellman optimality conditions (Theorem **??** for episodic MDPs), we know the optimal policy (among all possibly history dependent, non-stationary, and randomized policies), is given by a deterministic stationary policy which is only a function of $x_t$ and $t$. We have that:

$$
\begin{aligned}
Q_{H-1}(x, u) &= \mathbb{E}\left[(A_{H-1}x + B_{H-1}u + w_{H-1})^\top Q (A_{H-1}x + B_{H-1}u + w_{H-1})\right] + x^\top Q x + u^\top R u \\
&= (A_{H-1}x + B_{H-1}u)^\top Q (A_{H-1}x + B_{H-1}u) + \sigma^2 \text{Trace}(Q) + x^\top Q x + u^\top R u
\end{aligned}
$$

due to that $x_H = A_{H-1}x + B_{H-1}u + w_{H-1}$, and $\mathbb{E}[w_{H-1}^\top Q w_{H-1}] = \text{Trace}(\sigma^2 Q)$. Due to that this is a quadratic function of $u$, we can immediately derive that the optimal control is given by:

$$\pi_{H-1}^\star(x) = -(B_{H-1}^\top Q B_{H-1} + R)^{-1} B_{H-1}^\top Q A_{H-1} x = -K_{H-1}^\star x,$$

where the last step uses that $P_H := Q$.

For notational convenience, let $K = K_{H-1}^\star$, $A = A_{H-1}$, and $B = B_{H-1}$. Using the optimal control at $x$, i.e.

$u = -K^\star_{H-1}x$, we have:

$$
\begin{aligned}
V^\star_{H-1}(x) &= Q_{H-1}(x, -K^\star_{H-1}x) \\
&= x^\top\left((A-BK)^\top Q(A-BK) + Q + K^\top RK\right)x + \sigma^2\mathrm{Trace}(Q) \\
&= x^\top\left(AQA + Q - 2K^\top B^\top QA + K^\top(B^\top QB + R)K\right)x + \sigma^2\mathrm{Trace}(Q) \\
&= x^\top\left(AQA + Q - 2K^\top(B^\top QB + R)K + K^\top(B^\top QB + R)K\right)x + \sigma^2\mathrm{Trace}(Q) \\
&= x^\top\left(AQA + Q - K^\top(B^\top QB + R)K\right)x + \sigma^2\mathrm{Trace}(Q) \\
&= x^\top P_{H-1}x + \sigma^2\mathrm{Trace}(Q).
\end{aligned}
$$

where the fourth step uses our expression for $K = K^\star_{H-1}$. This proves our claim for $t = H - 1$.

This implies that:

$$
\begin{aligned}
Q^\star_{H-2}(x, u) &= \mathbb{E}[V^\star_{H-1}(A_{H-2}x + B_{H-2}u + w_{H-2})] + x^\top Qx + u^\top Ru \\
&= (A_{H-2}x + B_{H-2}u)^\top P_{H-1}(A_{H-2}x + B_{H-2}u) + \sigma^2\mathrm{Trace}(P_{H-1}) + \sigma^2\mathrm{Trace}(Q) + x^\top Qx + u^\top Ru.
\end{aligned}
$$

The remainder of the proof follows from a recursive argument, which can be verified along identical lines to the $t = H - 1$ case. ∎

### 16.2.2  Planning and Infinite Horizon LQRs

**Theorem 16.2.** *Suppose that the optimal cost is finite and that $R$ is positive definite. Let $P$ be a solution to the following algebraic Riccati equation:*

$$P = A^T PA + Q - A^T PB(B^T PB + R)^{-1}B^T PA. \tag{0.1}$$

*(Note that $P$ is a positive definite matrix). We have that the optimal policy is:*

$$\pi^\star(x) = -K^\star x$$

*where the optimal control gain is:*

$$K^* = -(B^T PB + R)^{-1}B^T PA. \tag{0.2}$$

*We have that $P$ is unique and that the optimal average cost is $\sigma^2\mathrm{Trace}(P)$.*

As before, $P$ parameterizes the optimal value function. We do not prove this theorem here though it follows along similar lines to the previous proof, via a limiting argument.

To find $P$, we can again run the recursion:

$$P \leftarrow Q + A^T PA - A^T PB(R + B^T PB)^{-1}B^T PA$$

starting with $P = Q$, which can be shown to converge to the unique positive semidefinite solution of the Riccati equation (since one can show the fixed-point iteration is contractive). Again, this approach is simply value iteration.

## 16.3  Convex Programs to find $P$ and $K^\star$

For the infinite horizon LQR problem, the optimization may be formulated as a convex program. In particular, the LQR problem can also be expressed as a semidefinite program (SDP) with variable $P$, for the infinite horizon case. We now present this primal program along with the dual program.

156

Note that these programs are the analogues of the linear programs from Section **??** for MDPs. While specifying these linear programs for an LQR (as an LQR is an MDP) would result in infinite dimensional linear programs, the special structure of the LQR implies these primal and dual programs have a more compact formulation when specified as an SDP.

### 16.3.1   The Primal for Infinite Horizon LQR

The primal optimization problem is given as:

$$\text{maximize} \quad \sigma^2 \text{Trace}(P)$$
$$\text{subject to} \quad \begin{bmatrix} A^T P A + Q - I & A^\top P B \\ B^T P A & B^\top P B + R \end{bmatrix} \succeq 0, \quad P \succeq 0,$$

where the optimization variable is $P$. This SDP has a unique solution, $P^\star$, which satisfies the algebraic Riccati equations (Equation 0.1 ); the optimal average cost of the infinite horizon LQR is $\sigma^2 \text{Trace}(P^\star)$; and the optimal policy is given by Equation 0.2.

The SDP can be derived by relaxing the equality in the Riccati equation to an inequality, then using the Schur complement lemma to rewrite the resulting Riccati inequality as linear matrix inequality. In particular, we can consider the relaxation where $P$ must satisfy:

$$P \succeq A^T P A + Q - A^T P B (B^T P B + R)^{-1} B^T P A.$$

That the solution to this relaxed optimization problem leads to the optimal $P^\star$ is due to the Bellman optimality conditions.

Now the Schur complement lemma for positive semi-definiteness is as follows: define

$$X = \begin{bmatrix} D & E \\ E^\top & F \end{bmatrix}.$$

(for matrices $D$, $E$, and $F$ of appropriate size, with $D$ and $F$ being square symmetric matrices). We have that $X$ is PSD if and only if

$$F - E^\top D^{-1} E \succeq 0.$$

This shows that the constraint set is equivalent to the above relaxation.

### 16.3.2   The Dual

The dual optimization problem is given as:

$$\text{minimize} \quad \text{Trace} \left( \Sigma \cdot \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \right)$$
$$\text{subject to} \quad \Sigma_{xx} = (A \ B) \Sigma (A \ B)^\top + \sigma^2 I, \quad \Sigma \succeq 0,$$

where the optimization variable is a symmetric matrix $\Sigma$, which is a $(d+k) \times (d+k)$ matrix with the block structure:

$$\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xu} \\ \Sigma_{ux} & \Sigma_{uu} \end{bmatrix}.$$

The interpretation of $\Sigma$ is that it is the covariance matrix of the stationary distribution. This analogous to the state-visitation measure for an MDP.

This SDP has a unique solution, say $\Sigma^\star$. The optimal gain matrix is then given by:

$$K^\star = -\Sigma^\star_{ux} (\Sigma^\star_{xx})^{-1}.$$

## 16.4 Policy Iteration, Gauss Newton, and NPG

Note that the noise variance $\sigma^2$ does not impact the optimal policy, in either the discounted case or in the infinite horizon case.

Here, when we examine local search methods, it is more convenient to work with case where $\sigma = 0$. In this case, we can work with cumulative cost rather the average cost. Precisely, when $\sigma = 0$, the infinite horizon LQR problem takes the form:

$$\min_K C(K), \text{ where } C(K) = \mathbb{E}_{x_0 \sim \mathcal{D}} \left[ \sum_{t=0}^{H} (x_t^\top Q x_t + u_t^\top R u_t) \right],$$

where they dynamics evolves as

$$x_{t+1} = (A - BK)x_t.$$

Note that we have directly parameterized our policy as a linear policy in terms of the gain matrix $K$, due to that we know the optimal policy is linear in the state. Again, we assume that $C(K^\star)$ is finite; this assumption is referred to as the system being *controllable*.

We now examine local search based approaches, where we will see a close connection to policy iteration. Again, we have a non-convex optimization problem:

**Lemma 16.3.** *(Non-convexity) If $d \geq 3$, there exists an LQR optimization problem, $\min_K C(K)$, which is not convex or quasi-convex.*

Regardless, we will see that gradient based approaches are effective. For local search based approaches, the importance of (some) randomization, either in $x_0$ or noise through having a disturbance, is analogous to our use of having a wide-coverage distribution $\mu$ (for MDPs).

### 16.4.1 Gradient Expressions

Gradient descent on $C(K)$, with a fixed stepsize $\eta$, follows the update rule:

$$K \leftarrow K - \eta \nabla C(K).$$

It is helpful to explicitly write out the functional form of the gradient. Define $P_K$ as the solution to:

$$P_K = Q + K^\top R K + (A - BK)^\top P_K (A - BK).$$

and, under this definition, it follows that $C(K)$ can be written as:

$$C(K) = \mathbb{E}_{x_0 \sim \mathcal{D}} \, x_0^\top P_K x_0.$$

Also, define $\Sigma_K$ as the (un-normalized) state correlation matrix, i.e.

$$\Sigma_K = \mathbb{E}_{x_0 \sim \mathcal{D}} \sum_{t=0}^{\infty} x_t x_t^\top.$$

**Lemma 16.4.** *(Policy Gradient Expression) The policy gradient is:*

$$\nabla C(K) = 2 \left( (R + B^\top P_K B)K - B^\top P_K A \right) \Sigma_K$$

*For convenience, define $E_K$ to be*

$$E_K = \left( (R + B^\top P_K B)K - B^\top P_K A \right),$$

*as a result the gradient can be written as $\nabla C(K) = 2E_K \Sigma_K$.*

**Proof:** Observe:

$$C_K(x_0) = x_0^\top P_K x_0 = x_0^\top \left(Q + K^\top R K\right) x_0 + x_0^\top (A - BK)^\top P_K (A - BK) x_0$$
$$= x_0^\top \left(Q + K^\top R K\right) x_0 + C_K((A - BK)x_0).$$

Let $\nabla$ denote the gradient with respect to $K$; note that $\nabla C_K((A - BK)x_0)$ has two terms as function of $K$, one with respect to $K$ in the subscript and one with respect to the input $(A - BK)x_0$. This implies

$$\nabla C_K(x_0) = 2RK x_0 x_0^\top - 2B^\top P_K (A - BK) x_0 x_0^\top + \nabla C_K(x_1)|_{x_1 = (A-BK)x_0}$$
$$= 2\left((R + B^\top P_K B)K - B^\top P_K A\right) \sum_{t=0}^{\infty} x_t x_t^\top$$

where we have used recursion and that $x_1 = (A - BK)x_0$. Taking expectations completes the proof. ∎

**The natural policy gradient.** Let us now motivate a version of the natural gradient. The natural policy gradient follows the update:

$$\theta \leftarrow \theta - \eta\, F(\theta)^{-1} \nabla C(\theta), \text{ where } F(\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \nabla \log \pi_\theta(u_t|x_t) \nabla \log \pi_\theta(u_t|x_t)^\top\right],$$

where $F(\theta)$ is the Fisher information matrix. A natural special case is using a linear policy with additive Gaussian noise, i.e.

$$\pi_K(x, u) = \mathcal{N}(Kx, \sigma^2 I) \tag{0.3}$$

where $K \in \mathbb{R}^{k \times d}$ and $\sigma^2$ is the noise variance. In this case, the natural policy gradient of $K$ (when $\sigma$ is considered fixed) takes the form:

$$K \leftarrow K - \eta \nabla C(\pi_K) \Sigma_K^{-1} \tag{0.4}$$

Note a subtlety here is that $C(\pi_K)$ is the randomized policy.

To see this, one can verify that the Fisher matrix of size $kd \times kd$, which is indexed as $[G_K]_{(i,j),(i',j')}$ where $i, i' \in \{1, \ldots k\}$ and $j, j' \in \{1, \ldots d\}$, has a block diagonal form where the only non-zeros blocks are $[G_K]_{(i,\cdot),(i,\cdot)} = \Sigma_K$ (this is the block corresponding to the $i$-th coordinate of the action, as $i$ ranges from 1 to $k$). This form holds more generally, for any diagonal noise.

### 16.4.2 Convergence Rates

We consider three exact rules, where we assume access to having exact gradients. As before, we can also estimate these gradients through simulation. For gradient descent, the update is

$$K_{n+1} = K_n - \eta \nabla C(K_n). \tag{0.5}$$

For natural policy gradient descent, the direction is defined so that it is consistent with the stochastic case, as per Equation 0.4, in the exact case the update is:

$$K_{n+1} = K_n - \eta \nabla C(K_n) \Sigma_{K_n}^{-1} \tag{0.6}$$

One show that the Gauss-Newton update is:

$$K_{n+1} = K_n - \eta (R + B^\top P_{K_n} B)^{-1} \nabla C(K_n) \Sigma_{K_n}^{-1}. \tag{0.7}$$

(Gauss-Newton is non-linear optimization approach which uses a certain Hessian approximation. It can be show that this leads to the above update rule.) Interestingly, for the case when $\eta = 1$, the Gauss-Newton method is equivalent to the policy iteration algorithm, which optimizes a one-step deviation from the current policy.

The Gauss-Newton method requires the most complex oracle to implement: it requires access to $\nabla C(K)$, $\Sigma_K$, and $R + B^\top P_K B$; as we shall see, it also enjoys the strongest convergence rate guarantee. At the other extreme, gradient descent requires oracle access to only $\nabla C(K)$ and has the slowest convergence rate. The natural policy gradient sits in between, requiring oracle access to $\nabla C(K)$ and $\Sigma_K$, and having a convergence rate between the other two methods.

In this theorem, $\|M\|_2$ denotes the spectral norm of a matrix $M$.

**Theorem 16.5.** *(Global Convergence of Gradient Methods) Suppose $C(K_0)$ is finite and, for $\mu$ defined as*

$$\mu := \sigma_{min}(\mathbb{E}_{x_0 \sim \mathcal{D}} x_0 x_0^\top),$$

*suppose $\mu > 0$.*

- *Gauss-Newton case: Suppose $\eta = 1$, the Gauss-Newton algorithm (Equation 0.7) enjoys the following performance bound:*
$$C(K_N) - C(K^*) \le \epsilon, \quad for \quad N \ge \frac{\|\Sigma_{K^*}\|_2}{\mu} \log \frac{C(K_0) - C(K^*)}{\epsilon}.$$

- *Natural policy gradient case: For a stepsize $\eta = 1/(\|R\|_2 + \frac{\|B\|_2^2 C(K_0)}{\mu})$, natural policy gradient descent (Equation 0.6) enjoys the following performance bound:*
$$C(K_N) - C(K^*) \le \epsilon, \quad for \quad N \ge \frac{\|\Sigma_{K^*}\|_2}{\mu} \left( \frac{\|R\|_2}{\sigma_{min}(R)} + \frac{\|B\|_2^2 C(K_0)}{\mu \sigma_{min}(R)} \right) \log \frac{C(K_0) - C(K^*)}{\epsilon}.$$

- *Gradient descent case: For any starting policy $K_0$, there exists a (constant) stepsize $\eta$ (which could be a function of $K_0$), such that:*
$$C(K_N) \to C(K^*), \quad as \quad N \to \infty.$$

### 16.4.3   Gauss-Newton Analysis

We only provide a proof for the Gauss-Newton case (see 16.6 for further readings).

We overload notation and let $K$ denote the policy $\pi(x) = Kx$. For the infinite horizon cost function, define:

$$
\begin{aligned}
V_K(x) &:= \sum_{t=0}^{\infty} \left( x_t^\top Q x_t + u_t^\top R u_t \right) \\
&= x^\top P_K x,
\end{aligned}
$$

and

$$Q_K(x, u) := x^\top Q x + u^\top R u + V_K(Ax + Bu),$$

and

$$A_K(x, u) = Q_K(x, u) - V_K(x).$$

The next lemma is identical to the performance difference lemma.

**Lemma 16.6.** *(Cost difference lemma) Suppose $K$ and $K'$ have finite costs. Let $\{x_t'\}$ and $\{u_t'\}$ be state and action sequences generated by $K'$, i.e. starting with $x_0' = x$ and using $u_t' = -K'x_t'$. It holds that:*

$$V_{K'}(x) - V_K(x) = \sum_t A_K(x_t', u_t').$$

*Also, for any $x$, the advantage is:*

$$A_K(x, K'x) = 2x^\top (K' - K)^\top E_K x + x^\top (K' - K)^\top (R + B^\top P_K B)(K' - K)x. \tag{0.8}$$

**Proof:** Let $c'_t$ be the cost sequence generated by $K'$. Telescoping the sum appropriately:

$$V_{K'}(x) - V_K(x) = \sum_{t=0} c'_t - V_K(x) = \sum_{t=0}(c'_t + V_K(x'_t) - V_K(x'_t)) - V_K(x)$$

$$= \sum_{t=0}(c'_t + V_K(x'_{t+1}) - V_K(x'_t)) = \sum_{t=0} A_K(x'_t, u'_t)$$

which completes the first claim (the third equality uses the fact that $x = x_0 = x'_0$).

For the second claim, observe that:

$$V_K(x) = x^\top \left(Q + K^\top R K\right) x + x^\top (A - BK)^\top P_K (A - BK) x$$

And, for $u = K'x$,

$$
\begin{aligned}
A_K(x, u) &= Q_K(x, u) - V_K(x) \\
&= x^\top \left(Q + (K')^\top R K'\right) x + x^\top (A - BK')^\top P_K (A - BK') x - V_K(x) \\
&= x^\top \left(Q + (K' - K + K)^\top R(K' - K + K)\right) x + \\
&\quad x^\top (A - BK - B(K' - K))^\top P_K (A - BK - B(K' - K)) x - V_K(x) \\
&= 2x^\top (K' - K)^\top \left((R + B^\top P_K B)K - B^\top P_K A\right) x + \\
&\quad x^\top (K' - K)^\top (R + B^\top P_K B)(K' - K)) x,
\end{aligned}
$$

which completes the proof. ∎

We have the following corollary, which can be viewed analogously to a smoothness lemma.

**Corollary 16.7.** *("Almost" smoothness)* $C(K)$ satisfies:

$$C(K') - C(K) = -2\mathrm{Trace}(\Sigma_{K'}(K - K')^\top E_K) + \mathrm{Trace}(\Sigma_{K'}(K - K')^\top (R + B^\top P_K B)(K - K'))$$

To see why this is related to smoothness (recall the definition of a smooth function in Equation 0.6), suppose $K'$ is sufficiently close to $K$ so that:

$$\Sigma_{K'} \approx \Sigma_K + O(\|K - K'\|) \tag{0.9}$$

and the leading order term $2\mathrm{Trace}(\Sigma_{K'}(K' - K)^\top E_K)$ would then behave as $\mathrm{Trace}((K' - K)^\top \nabla C(K))$.

**Proof:** The claim immediately results from Lemma 16.6, by using Equation 0.8 and taking an expectation. ∎

We now use this cost difference lemma to show that $C(K)$ is gradient dominated.

**Lemma 16.8.** *(Gradient domination) Let $K^*$ be an optimal policy. Suppose $K$ has finite cost and $\mu > 0$. It holds that:*

$$
\begin{aligned}
C(K) - C(K^*) &\leq \|\Sigma_{K^*}\| \mathrm{Trace}(E_K^\top (R + B^\top P_K B)^{-1} E_K) \\
&\leq \frac{\|\Sigma_{K^*}\|}{\mu^2 \sigma_{min}(R)} \mathrm{Trace}(\nabla C(K)^\top \nabla C(K))
\end{aligned}
$$

**Proof:** From Equation 0.8 and by completing the square,

$$
\begin{aligned}
A_K(x, K'x) &= Q_K(x, K'x) - V_K(x) \\
&= 2\mathrm{Trace}(xx^\top (K' - K)^\top E_K) + \mathrm{Trace}(xx^\top (K' - K)^\top (R + B^\top P_K B)(K' - K)) \\
&= \mathrm{Trace}(xx^\top \left(K' - K + (R + B^\top P_K B)^{-1} E_K\right)^\top (R + B^\top P_K B)\left(K' - K + (R + B^\top P_K B)^{-1} E_K\right)) \\
&\quad - \mathrm{Trace}(xx^\top E_K^\top (R + B^\top P_K B)^{-1} E_K) \\
&\geq -\mathrm{Trace}(xx^\top E_K^\top (R + B^\top P_K B)^{-1} E_K) \tag{0.10}
\end{aligned}
$$

with equality when $K' = K - (R + B^\top P_K B)^{-1} E_K$.

Let $x_t^*$ and $u_t^*$ be the sequence generated under $K_*$. Using this and Lemma 16.6,

$$C(K) - C(K^*) = -\mathbb{E}\sum_t A_K(x_t^*, u_t^*) \leq \mathbb{E}\sum_t \text{Trace}(x_t^*(x_t^*)^\top E_K^\top (R + B^\top P_K B)^{-1} E_K)$$

$$= \text{Trace}(\Sigma_{K^*} E_K^\top (R + B^\top P_K B)^{-1} E_K) \leq \|\Sigma_{K^*}\| \text{Trace}(E_K^\top (R + B^\top P_K B)^{-1} E_K)$$

which completes the proof of first inequality. For the second inequality, observe:

$$\text{Trace}(E_K^\top (R + B^\top P_K B)^{-1} E_K) \leq \|(R + B^\top P_K B)^{-1}\| \text{Trace}(E_K^\top E_K) \leq \frac{1}{\sigma_{\min}(R)} \text{Trace}(E_K^\top E_K)$$

$$= \frac{1}{\sigma_{\min}(R)} \text{Trace}(\Sigma_K^{-1} \nabla C(K)^\top \nabla C(K) \Sigma_K^{-1}) \leq \frac{1}{\sigma_{min}(\Sigma_K)^2 \sigma_{\min}(R)} \text{Trace}(\nabla C(K)^\top \nabla C(K))$$

$$\leq \frac{1}{\mu^2 \sigma_{\min}(R)} \text{Trace}(\nabla C(K)^\top \nabla C(K))$$

which completes the proof of the upper bound. Here the last step is because $\Sigma_K \succeq \mathbb{E}[x_0 x_0^\top]$. ∎

The next lemma bounds the one step progress of Gauss-Newton.

**Lemma 16.9.** *(Gauss-Newton Contraction) Suppose that:*

$$K' = K - \eta(R + B^\top P_K B)^{-1} \nabla C(K) \Sigma_K^{-1} , .$$

*If $\eta \leq 1$, then*

$$C(K') - C(K^*) \leq \left(1 - \frac{\eta\mu}{\|\Sigma_{K^*}\|}\right)(C(K) - C(K^*))$$

**Proof:** Observe that for PSD matrices $A$ and $B$, we have that $\text{Trace}(AB) \geq \sigma_{\min}(A)\text{Trace}(B)$. Also, observe $K' = K - \eta(R + B^\top P_K B)^{-1} E_K$. Using Lemma 16.7 and the condition on $\eta$,

$$\begin{aligned} C(K') - C(K) &= -2\eta\text{Trace}(\Sigma_{K'} E_K^\top (R + B^\top P_K B)^{-1} E_K) + \eta^2 \text{Trace}(\Sigma_{K'} E_K^\top (R + B^\top P_K B)^{-1} E_K) \\ &\leq -\eta\text{Trace}(\Sigma_{K'} E_K^\top (R + B^\top P_K B)^{-1} E_K) \\ &\leq -\eta\sigma_{\min}(\Sigma_{K'})\text{Trace}(E_K^\top (R + B^\top P_K B)^{-1} E_K) \\ &\leq -\eta\mu\text{Trace}(E_K^\top (R + B^\top P_K B)^{-1} E_K) \\ &\leq -\eta\frac{\mu}{\|\Sigma_{K^*}\|}(C(K) - C(K^*)), \end{aligned}$$

where the last step uses Lemma 16.8.

∎

With this lemma, the proof of the convergence rate of the Gauss Newton algorithm is immediate.

**Proof:** (of Theorem 16.5, Gauss-Newton case) The theorem is due to that $\eta = 1$ leads to a contraction of $1 - \frac{\mu}{\|\Sigma_{K^*}\|}$ at every step. ∎

## 16.5   System Level Synthesis for Linear Dynamical Systems

We demonstrate another parameterization of controllers which admits convexity. Specifically in this section, we consider finite horizon setting, i.e.,

$$x_{t+1} = Ax_t + Bu_t + w_t, \quad x_0 \sim \mathcal{D}, w_t \sim \mathcal{N}(0, \sigma^2 I), \quad t \in [0, 1, \ldots, H-1].$$

Instead of focusing on quadratic cost function, we consider general convex cost function $c(x_t, u_t)$, and the goal is to optimize:

$$\mathbb{E}_\pi \left[ \sum_{t=0}^{H-1} c_t(x_t, u_t) \right],$$

where $\pi$ is a time-dependent policy.

While we relax the assumption on the cost function from quadratic cost to general convex cost, we still focus on linearly parameterized policy, i.e., we are still interested in searching for a sequence of time-dependent linear controllers $\{-K_t^\star\}_{t=0}^{H-1}$ with $u_t = -K_t^\star x_t$, such that it minimizes the above objective function. We again assume the system is controllable, i.e., the expected total cost under $\{-K_t^\star\}_{t=0}^{H-1}$ is finite.

Note that due to the fact that now the cost function is not quadratic anymore, the Riccati equation we studied before does not hold , and it is not necessarily true that the value function of any linear controllers will be a quadratic function.

We present another parameterization of linear controllers which admit convexity in the objective function with respect to parameterization (recall that the objective function is non-convex with respect to linear controllers $-K_t$). With the new parameterization, we will see that due to the convexity, we can directly apply gradient descent to find the globally optimal solution.

Consider any fixed time-dependent linear controllers $\{-K_t\}_{t=0}^{H-1}$. We start by rolling out the linear system under the controllers. Note that during the rollout, once we observe $x_{t+1}$, we can compute $w_t$ as $w_t = x_{t+1} - Ax_t - Bu_t$. With $x_0 \sim \mu$, we have that at time step $t$:

$$
\begin{aligned}
u_t = -K_t x_t &= -K_t(Ax_{t-1} - BK_{t-1}x_{t-1} + w_{t-1}) \\
&= -K_t w_{t-1} - K_t(A - BK_{t-1})x_{t-1} \\
&= -K_t w_{t-1} - K_t(A - BK_{t-1})(Ax_{t-2} - BK_{t-2}x_{t-2} + w_{t-2}) \\
&= \underbrace{-K_t}_{:=M_{t-1;t}} w_{t-1} \underbrace{-K_t(A - BK_{t-1})}_{:=M_{t-2;t}} w_{t-2} \underbrace{-K_t(A - BK_{t-1})(A - BK_{t-2})}_{:=M_{t-3;t}} x_{t-2} \\
&= -K_t \left( \prod_{\tau=1}^{t}(A - BK_{t-\tau}) \right) x_0 - \sum_{\tau=0}^{t-1} K_t \prod_{h=1}^{t-1-\tau}(A - BK_{t-h})w_\tau
\end{aligned}
$$

Note that we can equivalently write $u_t$ using $x_0$ and the noises $w_0, \ldots, w_{t-1}$. Hence for time step $t$, let us do the following re-parameterization. Denote

$$
M_t := -K_t \left( \prod_{\tau=1}^{t}(A - BK_{t-\tau}) \right), \quad M_{\tau;t} := -K_t \prod_{h=1}^{t-1-\tau}(A - BK_{t-h}), \text{ where } \tau = [0, \ldots, t-1].
$$

Now we can express the control $u_t$ using $M_{\tau;t}$ for $\tau \in [0, \ldots, t-1]$ and $M_t$ as follows:

$$
u_t = M_t x_0 + \sum_{\tau=0}^{t-1} M_{\tau;t} w_\tau,
$$

which is equal to $u_t = -K_t x_t$. Note that above we only reasoned time step $t$. We can repeat the same calculation for all $t = 0 \to H - 1$.

The above calculation basically proves the following claim.

**Claim 16.10.** *For any linear controllers $\pi := \{-K_0, \ldots, -K_{H-1}\}$, there exists a parameterization, $\widetilde{\pi} := \{\{M_t, M_{0;t}, \ldots M_{t-1;t}\}\}_{t=0}^{H-1}$, such that when execute $\pi$ and $\widetilde{\pi}$ under any initialization $x_0$, and any sequence of noises $\{w_t\}_{t=0}^{H-1}$, they generate exactly the same state-control trajectory.*

We can execute $\widetilde{\pi} := \{\{M_t, M_{0;t}, \ldots M_{t-1;t}\}\}_{t=0}^{H-1}$ in the following way. Given any $x_0$, we execute $u_0 = M_0 x_0$ and observe $x_1$; at time step t with the observed $x_t$, we calculate $w_{t-1} = x_t - Ax_{t-1} - Bu_{t-1}$, and execute the control $u_t = M_t x_0 + \sum_{\tau=0}^{t-1} M_{\tau;t} w_\tau$. We repeat until we execute the last control $u_{H-1}$ and reach $x_H$.

What is the benefit of the above parameterization? Note that for any $t$, this is clearly over-parameterized: the simple linear controllers $-K_t$ has $d \times k$ many parameters, while the new controller $\{M_t; \{M_{\tau;t}\}_{\tau=0}^{t-1}\}$ has $t \times d \times k$ many parameters. The benefit of the above parameterization is that the objective function now is convex! The following claim formally shows the convexity.

**Claim 16.11.** *Given* $\widetilde{\pi} := \{\{M_t, M_{0;t}, \ldots M_{t-1;t}\}\}_{t=0}^{H-1}$ *and denote its expected total cost as* $J(\widetilde{\pi}) := \mathbb{E}\left[\sum_{t=0}^{H-1} c(x_t, u_t)\right]$, *where the expectation is with respect to the noise* $w_t \sim \mathcal{N}(0, \sigma^2 I)$ *and the initial state* $x_0 \sim \mu$, *and c is any convex function with respect to* $x, u$. *We have that* $J(\widetilde{\pi})$ *is convex with respect to parameters* $\{\{M_t, M_{0;t}, \ldots M_{t-1;t}\}\}_{t=0}^{H-1}$.

**Proof:** We consider a fixed $x_0$ and a fixed sequence of noises $\{w_t\}_{t=0}^{H-1}$. Recall that for $u_t$, we can write it as:

$$u_t = M_t x_0 + \sum_{\tau=0}^{t-1} M_{\tau;t} w_\tau,$$

which is clearly linear with respect to the parameterization $\{M_t; M_{0;t}, \ldots M_{t-1;t}\}$.

For $x_t$, we can show by induction that it is linear with respect to $\{M_\tau; M_{0;\tau}, \ldots M_{\tau-1;\tau}\}_{\tau=0}^{t-1}$. Note that it is clearly true for $x_0$ which is independent of any parameterizations. Assume this claim holds for $x_t$ with $t \geq 0$, we now check $x_{t+1}$. We have:

$$x_{t+1} = Ax_t + Bu_t = Ax_t + BM_t x_0 + \sum_{\tau=0}^{t-1} BM_{\tau;t} w_\tau$$

Note that by inductive hypothesis $x_t$ is linear with respect to $\{M_\tau; M_{0;\tau}, \ldots M_{\tau-1;\tau}\}_{\tau=0}^{t-1}$, and the part $BM_t x_0 + \sum_{\tau=0}^{t-1} BM_{\tau;t} w_\tau$ is clearly linear with respect to $\{M_t; M_{0;t}, \ldots M_{t-1;t}\}$. Together, this implies that $x_{t+1}$ is linear with respect to $\{M_\tau; M_{0;\tau}, \ldots M_{\tau-1;\tau}\}_{\tau=0}^{t}$, which concludes that for any $x_t$ with $t = 0, \ldots, H-1$, we have $x_t$ is linear with respect to $\{M_\tau; M_{0;\tau}, \ldots M_{\tau-1;\tau}\}_{\tau=0}^{H-1}$.

Note that the trajectory total cost is $\sum_{t=0}^{H-1} c(x_t, u_t)$. Since $c_t$ is convex with respect to $x_t$ and $u_t$, and $x_t$ and $u_t$ are linear with respect to $\{M_\tau; M_{0;\tau}, \ldots M_{\tau-1;\tau}\}_{\tau=0}^{H-1}$, we have that $\sum_{t=0}^{H-1} c_t(x_t, u_t)$ is convex with respect to $\{M_\tau; M_{0;\tau}, \ldots M_{\tau-1;\tau}\}_{\tau=0}^{H-1}$.

In last step we can simply take expectation with respect to $x_0$ and $w_1, \ldots, w_{H-1}$. By linearity of expectation, we conclude the proof. ∎

The above immediately suggests that (sub) gradient-based algorithms such as projected gradient descent on parameters $\{M_t, M_{0;t}, \ldots M_{t-1;t}\}_{t=0}^{H-1}$ can converge to the globally optimal solutions for any convex function $c_t$. Recall that the best linear controllers $\{-K_t^\star\}_{t=0}^{H-1}$ has its own corresponding parameterization $\{M_t^\star, M_{0;t}^\star, \ldots, M_{t-1;t}^\star\}_{t=0}^{H-1}$. Thus gradient-based optimization (with some care of the boundness of the parameters $\{M_t^\star, M_{0;t}^\star, \ldots, M_{t-1;t}^\star\}_{t=0}^{H-1}$) can find a solution that at least as good as the best linear controllers $\{-K_t^\star\}_{t=0}^{H-1}$.

**Remark** The above claims easily extend to time-dependent transition and cost function, i.e., $A_t, B_t, c_t$ are time-dependent. One can also extend it to episodic online control setting with adversarial noises $w_t$ and adversarial cost functions using no-regret online convex programming algorithms [Shalev-Shwartz, 2011]. In episodic online control, every episode $k$, an adversary determines a sequence of bounded noises $w_0^k, \ldots, w_{H-1}^k$, and cost function $c^k(x, u)$; the learner proposes a sequence of controllers $\widetilde{\pi}^k = \{M_t^k, M_{0;t}^k, \ldots, M_{t-1;t}^k\}_{t=0}^{H-1}$ and executes them (learner does

not know the cost function $c^k$ until the end of the episode, and the noises are revealed in a way when she observes $x_t^k$ and calculates $w_{t-1}^k$ as $x_t^k - Ax_{t-1}^k - Bu_{t-1}^k$); at the end of the episode, learner observes $c^k$ and suffers total cost $\sum_{t=0}^{H-1} c^k(x_t^k, u_t^k)$. The goal of the episodic online control is to be no-regret with respect to the best linear controllers in hindsight:

$$\sum_{k=0}^{K-1} \sum_{t=0}^{H-1} c^k(x_t^k, u_t^k) - \min_{\{-K_t^\star\}_{t=0}^{H-1}} \sum_{k=0}^{K-1} J^k(\{-K_t^\star\}_{t=0}^{H-1}) = o(K),$$

where we denote $J^k(\{-K_t^\star\}_{t=0}^{H-1})$ as the total cost of executing $\{-K_t^\star\}_{t=0}^{H-1}$ in episodic $k$ under $c^k$ and noises $\{w_t^k\}$, i.e., $\sum_{t=0}^{H-1} c(x_t, u_t)$ with $u_t = -K_t^\star x_t$ and $x_{t+1} = Ax_t + Bu_t + w_t^k$, for all $t$.

## 16.6 Bibliographic Remarks and Further Readings

The basics of optimal control theory can be found in any number of standards text [Anderson and Moore, 1990, Evans, 2005, Bertsekas, 2017]. The primal and dual formulations for the continuous time LQR are derived in Balakrishnan and Vandenberghe [2003], while the dual formulation for the discrete time LQR, that we us here, is derived in Cohen et al. [2019].

The treatment of Gauss-Newton, NPG, and PG algorithm are due to Fazel et al. [2018]. We have only provided the proof for the Gauss-Newton case based; the proofs of convergence rates for NPG and PG can be found in Fazel et al. [2018].

For many applications, the finite horizon LQR model is widely used as a model of locally linear dynamics, e.g. [Ahn et al., 2007, Todorov and Li, 2005, Tedrake, 2009, Perez et al., 2012]. The issue of instabilities are largely due to model misspecification and in the accuracy of the Taylor's expansion; it is less evident how the infinite horizon LQR model captures these issues. In contrast, for MDPs, practice tends to deal with stationary (and discounted) MDPs, due that stationary policies are more convenient to represent and learn; here, the non-stationary, finite horizon MDP model tends to be more of theoretical interest, due to that it is straightforward (and often more effective) to simply incorporate temporal information into the state. Roughly, if our policy is parametric, then practical representational constraints leads us to use stationary policies (incorporating temporal information into the state), while if we tend to use non-parametric policies (e.g. through some rollout based procedure, say with "on the fly" computations like in model predictive control, e.g. [Williams et al., 2017]), then it is often more effective to work with finite horizon, non-stationary models.

**Sample Complexity and Regret for LQRs.** We have not treated the sample complexity of learning in an LQR (see [Dean et al., 2017, Simchowitz et al., 2018, Mania et al., 2019] for rates). Here, the basic analysis follows from the online regression approach, which was developed in the study of linear bandits [Dani et al., 2008, Abbasi-Yadkori et al., 2011]; in particular, the self-normalized bound for vector-valued martingales [Abbasi-Yadkori et al., 2011] (see Theorem A.9) provides a direct means to obtain sharp confidence intervals for estimating the system matrices $A$ and $B$ from data from a single trajectory (e.g. see [Simchowitz and Foster, 2020]).

Another family of work provides regret analyses of online LQR problems [Abbasi-Yadkori and Szepesvári, 2011, Dean et al., 2018, Mania et al., 2019, Cohen et al., 2019, Simchowitz and Foster, 2020]. Here, naive random search suffices for sample efficient learning of LQRs [Simchowitz and Foster, 2020]. For the learning and control of more complex nonlinear dynamical systems, one would expect this is insufficient, where strategic exploration is required for sample efficient learning; just as in the case for MDPs (e.g. the UCB-VI algorithm).

**Convex Parameterization of Linear Controllers** The convex parameterization in Section 16.5 is based on [Agarwal et al., 2019] which is equivalent to the System Level Synthesis (SLS) parameterization [Wang et al., 2019]. Agar-

wal et al. [2019] uses SLS parameterization in the infinite horizon online control setting and leverages a reduction to online learning with memory. Note that in episodic online control setting, we can just use classic no-regret online learner such as projected gradient descent [Zinkevich, 2003]. For partial observable linear systems, the classic Youla parameterization[Youla et al., 1976] introduces a convex parameterization. We also refer readers to [Simchowitz et al., 2020] for more detailed discussion about Youla parameterization and the generalization of Youla parameterization. Moreover, using the performance difference lemma, the exact optimal control policy for adversarial noise with full observations can be exactly characterized Foster and Simchowitz [2020], Goel and Hassibi [2020] and yields a form reminiscent of the SLS parametrization Wang et al. [2019].

# Chapter 17

# Partially Observable Markov Decision Processes

**To be added...**

# Bibliography

Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Conference on Learning Theory*, pages 1–26, 2011.

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. POLITEX: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pages 3692–3702, 2019.

Naoki Abe and Philip M. Long. Associative reinforcement learning using linear probabilistic concepts. In *Proc. 16th International Conf. on Machine Learning*, pages 3–11. Morgan Kaufmann, San Francisco, CA, 1999.

Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *NeurIPS*, 2020a.

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *NeurIPS*, 2020b.

Alekh Agarwal, Sham Kakade, and Lin F. Yang. Model-based reinforcement learning with a generative model is minimax optimal. In *COLT*, volume 125, pages 67–83, 2020c.

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift, 2020d.

Naman Agarwal, Brian Bullins, Elad Hazan, Sham M Kakade, and Karan Singh. Online control with adversarial disturbances. *arXiv preprint arXiv:1902.08721*, 2019.

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans, editors. *Understanding the impact of entropy on policy optimization*, 2019. URL https://arxiv.org/abs/1811.11214.

Hyo-Sung Ahn, YangQuan Chen, and Kevin L. Moore. Iterative learning control: Brief survey and categorization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1099–1121, 2007.

Brian D. O. Anderson and John B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1990. ISBN 0-13-638560-5.

András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47 (2-3):235–256, 2002. ISSN 0885-6125.

Alex Ayoub, Zeyu Jia, Csaba Szepesvári, Mengdi Wang, and Lin F. Yang. Model-based reinforcement learning with value-targeted regression. *arXiv preprint arXiv:2006.01107*, 2020. URL `https://arxiv.org/abs/2006.01107`.

Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J Kappen. Minimax pac bounds on the sample complexity of reinforcement lear ning with a generative model. *Machine learning*, 91(3):325–349, 2013.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of Machine Learning Research*, volume 70, pages 263–272, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

J. Andrew Bagnell and Jeff Schneider. Covariant policy search. *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1019–1024, 2003. URL `http://dl.acm.org/citation.cfm?id=1630659.1630805`.

J Andrew Bagnell, Sham M Kakade, Jeff G Schneider, and Andrew Y Ng. Policy search by dynamic programming. In *Advances in neural information processing systems*, pages 831–838, 2004.

V. Balakrishnan and L. Vandenberghe. Semidefinite programming duality and linear time-invariant systems. *IEEE Transactions on Automatic Control*, 48(1):30–41, 2003.

Keith Ball. An elementary introduction to modern convex geometry. *Flavors of geometry*, 31:1–58, 1997.

A. Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi: 10.1137/1.9781611974997.

Richard Bellman. Dynamic programming and Lagrange multipliers. *Proceedings of the National Academy of Sciences*, 42(10):767–769, 1956.

Richard Bellman and Stuart Dreyfus. Functional approximations and dynamic programming. *Mathematical Tables and Other Aids to Computation*, 13(68):247–251, 1959.

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2017.

Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.

Kianté Brantley, Wen Sun, and Mikael Henaff. Disagreement-regularized imitation learning. In *International Conference on Learning Representations*, 2019.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume, and John Langford. Learning to search better than your teacher. In *International Conference on Machine Learning*, pages 2058–2066. PMLR, 2015.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051, 2019.

Ching-An Cheng and Byron Boots. Convergence of value aggregation for imitation learning. *arXiv preprint arXiv:1801.07292*, 2018.

Ching-An Cheng, Xinyan Yan, Nolan Wagener, and Byron Boots. Fast policy learning through imitation and reinforcement. *arXiv preprint arXiv:1805.10413*, 2018.

Ching-An Cheng, Andrey Kolobov, and Alekh Agarwal. Policy improvement from multiple experts. *arXiv preprint arXiv:2007.00795*, 2020.

Alon Cohen, Tomer Koren, and Yishay Mansour. Learning linear-quadratic regulators efficiently with only sqrtT regret. In *International Conference on Machine Learning*, pages 1300–1309, 2019.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, pages 355–366, 2008.

Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2818–2826, 2015.

Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5713–5723, 2017.

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E. Schapire. On oracle-efficient PAC reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems 31*, 2018.

Hal Daumé, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine learning*, 75(3):297–325, 2009.

S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu. On the sample complexity of the linear quadratic regulator. *ArXiv e-prints*, 2017.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. Regret bounds for robust adaptive control of the linear quadratic regulator. In *Advances in Neural Information Processing Systems*, pages 4188–4197, 2018.

Luc Devroye and Gábor Lugosi. *Combinatorial methods in density estimation*. Springer Science & Business Media, 2012.

Simon S Du, Sham M Kakade, Ruosong Wang, and Lin F Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2019.

Lawrence C. Evans. An introduction to mathematical optimal control theory. *University of California, Department of Mathematics*, page 126, 2005. ISSN 14712334.

Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. Online Markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

Maryam Fazel, Rong Ge 0001, Sham M. Kakade, and Mehran Mesbahi. Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1466–1475. PMLR, 2018.

Dylan J Foster and Max Simchowitz. Logarithmic regret for adversarial online control. *arXiv preprint arXiv:2003.00189*, 2020.

Sara A Geer and Sara van de Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. *arXiv preprint arXiv:1901.11275*, 2019.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Peter W. Glynn. Likelihood ratio gradient estimation for stochastic systems. *Commun. ACM*, 33(10):75–84, 1990. ISSN 0001-0782.

Gautam Goel and Babak Hassibi. The power of linear controllers in lqr control. *arXiv preprint arXiv:2002.02574*, 2020.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Gabriel Dulac-Arnold, et al. Deep q-learning from demonstrations. *arXiv preprint arXiv:1704.03732*, 2017.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

Daniel Hsu, Sham Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78, 11 2008.

Daniel J. Hsu, S. Kakade, and Tong Zhang. Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14:569–600, 2012.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Zeyu Jia, Lin Yang, Csaba Szepesvari, and Mengdi Wang. Model-based reinforcement learning with value-targeted regression. *Proceedings of Machine Learning Research*, 120:666–686, 2020.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E. Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, 2017.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143, 2020.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *arXiv preprint arXiv:2102.00815*, 2021.

S. Kakade. A natural policy gradient. In *NIPS*, 2001.

Sham Kakade and John Langford. Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the 19th International Conference on Machine Learning*, volume 2, pages 267–274, 2002.

Sham Machandranath Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University of College London, 2003.

Liyiming Ke, Matt Barnes, Wen Sun, Gilwoo Lee, Sanjiban Choudhury, and Siddhartha Srinivasa. Imitation learning as $f$-divergence minimization. *arXiv preprint arXiv:1905.12888*, 2019.

Michael Kearns and Daphne Koller. Efficient reinforcement learning in factored mdps. In *IJCAI*, volume 16, pages 740–747, 1999.

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

Michael J Kearns and Satinder P Singh. Finite-sample convergence rates for q-learning and indirect algorithms. In *Advances in neural information processing systems*, pages 996–1002, 1999.

Michael J. Kearns, Yishay Mansour, and Andrew Y. Ng. Approximate planning in large pomdps via reusable trajectories. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 1001–1007. MIT Press, 2000.

Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European Conference on Computer Vision*, pages 201–214. Springer, 2012.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. PAC reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, pages 1840–1848, 2016.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Analysis of classification-based policy iteration algorithms. *The Journal of Machine Learning Research*, 17(1):583–612, 2016.

Hoang M Le, Nan Jiang, Alekh Agarwal, Miroslav Dudík, Yisong Yue, and Hal Daumé III. Hierarchical imitation and reinforcement learning. *arXiv preprint arXiv:1803.00590*, 2018.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems*, pages 1071–1079, 2014.

Sergey Levine and Vladlen Koltun. Guided policy search. In *International Conference on Machine Learning*, pages 1–9, 2013.

Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Breaking the sample size barrier in model-based reinforcement learning with a generative model. *CoRR*, abs/2005.12900, 2020.

Boyi Liu, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural proximal/trust region policy optimization attains globally optimal policy. *CoRR*, abs/1906.10306, 2019. URL http://arxiv.org/abs/1906.10306.

Thodoris Lykouris, Max Simchowitz, Aleksandrs Slivkins, and Wen Sun. Corruption robust exploration in episodic reinforcement learning. *arXiv preprint arXiv:1911.08689*, 2019.

Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalent control of LQR is efficient. *arXiv preprint arXiv:1902.07826*, 2019.

Yishay Mansour and Satinder Singh. On the complexity of policy iteration. *UAI*, 01 1999.

C. McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics*, pages 148–188. Cambridge University Press, 1989.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods, 2020.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

Aditya Modi, Nan Jiang, Ambuj Tewari, and Satinder P. Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 108 of *Proceedings of Machine Learning Research*, 2020.

Rémi Munos. Error bounds for approximate policy iteration. In *ICML*, volume 3, pages 560–567, 2003.

Rémi Munos. Error bounds for approximate value iteration. In *AAAI*, 2005.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *CoRR*, abs/1705.07798, 2017.

Ian Osband and Benjamin Van Roy. On lower bounds for regret in reinforcement learning. *ArXiv*, abs/1608.02732, 2016.

Alejandro Perez, Robert Platt, George Konidaris, Leslie Kaelbling, and Tomas Lozano-Perez. LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics. In *IEEE International Conference on Robotics and Automation*, pages 2537–2542, 2012.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomput.*, 71(7-9):1180–1190, 2008. ISSN 0925-2312.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

H. Robbins. Some aspects of the sequential design of experiments. In *Bulletin of the American Mathematical Society*, volume 55, 1952.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668, 2010.

Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

Bruno Scherrer. Approximate policy iteration schemes: a comparison. In *International Conference on Machine Learning*, pages 1314–1322, 2014.

Bruno Scherrer and Matthieu Geist. Local policy search in a convex space and conservative policy iteration as boosted policy search. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2014.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

John Schulman, F. Wolski, Prafulla Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

Claude E. Shannon. Programming a computer playing chess. *Philosophical Magazine*, Ser.7, 41(312), 1959.

Aaron Sidford, Mengdi Wang, Xian Wu, Lin F. Yang, and Yinyu Ye. Near-optimal time and sample complexities for for solving discounted markov decision process with a generative model. In *Advances in Neural Information Processing Systems 31*, 2018.

Max Simchowitz and Dylan J Foster. Naive exploration is optimal for online LQR. *arXiv preprint arXiv:2001.09576*, 2020.

Max Simchowitz, Horia Mania, Stephen Tu, Michael I Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *COLT*, 2018.

Max Simchowitz, Karan Singh, and Elad Hazan. Improper learning for non-stochastic control. *arXiv preprint arXiv:2001.09254*, 2020.

Satinder Singh and Richard Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.

Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. *arXiv preprint arXiv:1703.01030*, 2017.

Wen Sun, J Andrew Bagnell, and Byron Boots. Truncated horizon policy search: Combining reinforcement learning & imitation learning. *arXiv preprint arXiv:1805.11240*, 2018.

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, pages 2898–2933, 2019a.

Wen Sun, Anirudh Vemula, Byron Boots, and J Andrew Bagnell. Provably efficient imitation learning from observation alone. *arXiv preprint arXiv:1905.10948*, 2019b.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 99, pages 1057–1063, 1999.

Csaba Szepesvári and Rémi Munos. Finite time bounds for sampling based fitted value iteration. In *Proceedings of the 22nd international conference on Machine learning*, pages 880–887. ACM, 2005.

Russ Tedrake. LQR-trees: Feedback motion planning on sparse randomized trees. *The International Journal of Robotics Research*, 35, 2009.

Michael J Todd. *Minimum-Volume Ellipsoids: Theory and Algorithms*, volume 23. SIAM, 2016.

Emanuel Todorov and Weiwei Li. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *American Control Conference*, pages 300–306, 2005.

Yuanhao Wang, Ruosong Wang, and Sham M. Kakade. An exponential lower bound for linearly-realizable mdps with constant suboptimality gap. *CoRR*, abs/2103.12690, 2021. URL https://arxiv.org/abs/2103.12690.

Yuh-Shyang Wang, Nikolai Matni, and John C Doyle. A system-level approach to controller synthesis. *IEEE Transactions on Automatic Control*, 64(10):4079–4093, 2019.

Gellért Weisz, Philip Amortila, and Csaba Szepesvári. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. In *Algorithmic Learning Theory, 16-19 March 2021, Virtual Conference, Worldwide*, volume 132 of *Proceedings of Machine Learning Research*, 2021.

Zheng Wen and Benjamin Van Roy. Efficient reinforcement learning in deterministic systems with value function generalization. *Mathematics of Operations Research*, 42(3):762–782, 2017.

Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, 2017.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Lin F Yang and Mengdi Wang. Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. *arXiv preprint arXiv:1905.10389*, 2019a.

Lin F. Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004, 2019b.

Yinyu Ye. A new complexity result on solving the markov decision problem. *Math. Oper. Res.*, 30:733–749, 08 2005.

Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Math. Oper. Res.*, 36(4):593–603, 2011.

Ming Yin, Yu Bai, and Yu-Xiang Wang. Near optimal provable uniform convergence in off-policy evaluation for reinforcement learning. *ArXiv*, abs/2007.03760, 2021.

Dante Youla, Hamid Jabr, and Jr Bongiorno. Modern wiener-hopf design of optimal controllers–part ii: The multivariable case. *IEEE Transactions on Automatic Control*, 21(3):319–338, 1976.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR, 2020.

Dongruo Zhou, Jiafan He, and Quanquan Gu. Provably efficient reinforcement learning for discounted mdps with feature mapping. *arXiv preprint arXiv:2006.13165*, 2020.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, pages 1433–1438, 2008.

Brian D Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J Andrew Bagnell, Martial Hebert, Anind K Dey, and Siddhartha Srinivasa. Planning-based prediction for pedestrians. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3931–3936. IEEE, 2009.

Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. Modeling interaction via the principle of maximum causal entropy. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. Carnegie Mellon University, 2010.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

# Appendix A

# Concentration

**Lemma A.1.** *(Hoeffding's inequality) Suppose $X_1, X_2, \ldots X_n$ are a sequence of independent, identically distributed (i.i.d.) random variables with mean $\mu$. Let $\bar{X}_n = n^{-1} \sum_{i=1}^{n} X_i$. Suppose that $X_i \in [b_-, b_+]$ with probability 1, then*

$$P(\bar{X}_n \geq \mu + \epsilon) \leq e^{-2n\epsilon^2/(b_+ - b_-)^2}.$$

*Similarly,*

$$P(\bar{X}_n \leq \mu - \epsilon) \leq e^{-2n\epsilon^2/(b_+ - b_-)^2}.$$

The Chernoff bound implies that with probability $1 - \delta$:

$$\bar{X}_n - EX \leq (b_+ - b_-)\sqrt{\ln(1/\delta)/(2n)}.$$

Now we introduce the definition of subGaussian random variables.

**Definition A.2** (Sub-Gaussian Random Variable). A random variable $X$ is $\sigma$-subGaussian if for all $\lambda \in \mathbb{R}$, it holds that $\mathbb{E}\left[\exp(\lambda X)\right] \leq \exp\left(\lambda^2 \sigma^2 / 2\right)$.

One can show that a Gaussian random variable with zero mean and standard deviation $\sigma$ is a $\sigma$-subGaussian random variable.

The following theorem shows that the tails of a $\sigma$-subGaussian random variable decay approximately as fast as that of a Gaussian variable with zero mean and standard deviation $\sigma$.

**Theorem A.3.** *If $X$ is a $\sigma$-subGaussian, then for any $\epsilon > 0$, we have $\Pr\left(X \geq \epsilon\right) \leq \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$.*

The following lemma shows that the sum of independent sub-Gaussian variables is still sub-Gaussian.

**Lemma A.4.** *Suppose that $X_1$ and $X_2$ are independent and $\sigma_1$ and $\sigma_2$ subGaussian, respectively. Then for any $c \in \mathbb{R}$, we have $cX$ being $|c|\sigma$-subGaussian. We also have $X_1 + X_2$ being $\sqrt{\sigma_1^2 + \sigma_2^2}$-subGaussian.*

**Theorem A.5** (Hoeffding-Azuma Inequality). *Suppose $X_1, \ldots, X_T$ is a martingale difference sequence where each $X_t$ is a $\sigma_t$ sub-Gaussian, Then, for all $\epsilon > 0$ and all positive integer $N$,*

$$P\left(\sum_{i=1}^{N} X_i \geq \epsilon\right) \leq \exp\left(\frac{-\epsilon^2}{2\sum_{i=1}^{N} \sigma_i^2}\right).$$

**Lemma A.6.** *(Bernstein's inequality) Suppose $X_1, \dots, X_n$ are independent random variables. Let $\bar{X}_n = n^{-1} \sum_{i=1}^{n} X_i$, $\mu = \mathbb{E}\bar{X}_n$, and $Var(X_i)$ denote the variance of $X_i$. If $X_i - EX_i \leq b$ for all $i$, then*

$$P(\bar{X}_n \geq \mu + \epsilon) \leq \exp\left[-\frac{n^2 \epsilon^2}{2 \sum_{i=1}^{n} Var(X_i) + 2nb\epsilon/3}\right].$$

If all the variances are equal, the Bernstein inequality implies that, with probability at least $1 - \delta$,

$$\bar{X}_n - EX \leq \sqrt{2\mathrm{Var}(X) \ln(1/\delta)/n} + \frac{2b \ln(1/\delta)}{3n}.$$

**Lemma A.7** (Bernstein's Inequality for Martingales). *Suppose $X_1, X_2 \dots$ is a martingale difference sequence where $|X_i| \leq M \in \mathbb{R}^+$ almost surely. Then for all positive $t$ and $n \in \mathbb{N}^+$, we have:*

$$P\left(\sum_{i=1}^{n} X_i \geq t\right) \leq \exp\left(-\frac{t^2/2}{\sum_{i=1}^{n} \mathbb{E}\left[X_i^2 | X_{<i}\right] + Mt/3}\right).$$

The following concentration bound is a simple application of the McDiarmid's inequality [McDiarmid, 1989] (e.g. see [Hsu et al., 2008] for proof).

**Proposition A.8.** *(Concentration for Discrete Distributions) Let $z$ be a discrete random variable that takes values in $\{1, \dots, d\}$, distributed according to $q$. We write $q$ as a vector where $\vec{q} = [\Pr(z = j)]_{j=1}^{d}$. Assume we have $N$ iid samples, and that our empirical estimate of $\vec{q}$ is $[\widehat{q}]_j = \sum_{i=1}^{N} \mathbf{1}[z_i = j]/N$.*

*We have that $\forall \epsilon > 0$:*
$$\Pr\left(\|\widehat{q} - \vec{q}\|_2 \geq 1/\sqrt{N} + \epsilon\right) \leq e^{-N\epsilon^2}.$$

*which implies that:*
$$\Pr\left(\|\widehat{q} - \vec{q}\|_1 \geq \sqrt{d}(1/\sqrt{N} + \epsilon)\right) \leq e^{-N\epsilon^2}.$$

**Lemma A.9** (Self-Normalized Bound for Vector-Valued Martingales; [Abbasi-Yadkori et al., 2011]). *Let $\{\varepsilon_i\}_{i=1}^{\infty}$ be a real-valued stochastic process with corresponding filtration $\{\mathcal{F}_i\}_{i=1}^{\infty}$ such that $\varepsilon_i$ is $\mathcal{F}_i$ measurable, $\mathbb{E}[\varepsilon_i|\mathcal{F}_{i-1}] = 0$, and $\varepsilon_i$ is conditionally $\sigma$-sub-Gaussian with $\sigma \in \mathbb{R}^+$. Let $\{X_i\}_{i=1}^{\infty}$ be a stochastic process with $X_i \in \mathcal{H}$ (some Hilbert space) and $X_i$ being $\mathcal{F}_t$ measurable. Assume that a linear operator $\Sigma : \mathcal{H} \to \mathcal{H}$ is positive definite, i.e., $x^\top \Sigma x > 0$ for any $x \in \mathcal{H}$. For any $t$, define the linear operator $\Sigma_t = \Sigma_0 + \sum_{i=1}^{t} X_i X_i^\top$ (here $xx^\top$ denotes outer-product in $\mathcal{H}$). With probability at least $1 - \delta$, we have for all $t \geq 1$:*

$$\left\|\sum_{i=1}^{t} X_i \varepsilon_i\right\|_{\Sigma_t^{-1}}^2 \leq \sigma^2 \log\left(\frac{\det(\Sigma_t)\det(\Sigma_0)^{-1}}{\delta^2}\right).$$

**Theorem A.10** (OLS with a fixed design). *Consider a fixed dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ where $x_i \in \mathbb{R}^d$, and $y_i = (\theta^\star)^\top x_i + \epsilon_i$, where $\{\epsilon_i\}_{i=1}^{N}$ are independent $\sigma$-sub-Gaussian random variables, and $\mathbb{E}[\epsilon_i] = 0$. Denote $\Lambda = \sum_{i=1}^{N} x_i x_i^\top$, and we further assume that $\Lambda$ is full rank such that $\Lambda^{-1}$ exists. Denote $\hat{\theta}$ as the least squares solution, i.e., $\hat{\theta} = \Lambda^{-1} \sum_{i=1}^{N} x_i y_i$. Pick $\delta \in (0, 1)$, with probability at least $1 - \delta$, we have:*

$$\left\|\hat{\theta} - \theta^\star\right\|_{\Lambda}^2 \leq \sigma^2(d + \sqrt{d \ln(1/\delta)} + 2\ln(1/\delta)).$$

The proof of the above theorem is standard (e.g. the proof can be easily adapted from Lemma 6 in [Hsu et al., 2012]).