Reinforcement Learning: Theory and Algorithms

Alekh Agarwal Nan Jiang Sham M. Kakade

September 29, 2020

WORKING DRAFT

V2: This version is changing to use un-normalized values.

Contents

Notation

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

- 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 way $v \leq v'$, 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:

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

- 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^π to be the transition matrix on state-action pairs induced by a deterministic policy π. 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,

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

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

$$\begin{aligned} \pi_Q(s) &:= \mathrm{argmax}_{a\in\mathcal{A}}\, Q(s,a) \\ V_Q(s) &:= \max_{a\in\mathcal{A}} Q(s,a).. \end{aligned}$$

- 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 and Computational Complexity

1.1 Markov Decision Processes

In reinforcement learning, the interactions between the agent and the environment are often described by a 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.
- An action space $\mathcal{A}$, which also may be discrete or infinite.
- 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.
- A discount factor $\gamma \in [0, 1)$, which defines a horizon for the problem.
- An initial state distribution $\mu \in \Delta(\mathcal{S})$, which species 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. μ is a distribution supported only on s_0.

1.1.1 Interaction protocol

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,

$$\tau_t = (s_0, a_0, r_1, s_1, \ldots, s_t),$$

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

1.1.2 The objective, policies, and values

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 mapping from a trajectory to an action, i.e. $\pi : \mathcal{H} \to \mathcal{A}$ where $\mathcal{H}$ is the set of all possibly trajectories. A deterministic, *stationary* policy $\pi : \mathcal{S} \to \mathcal{A}$ specifies a decision-making strategy in which the agent chooses actions adaptively based on the current state, i.e., $a_t = \pi(s_t)$. The agent may also choose actions according to a stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, and, overloading notation, we write $a_t \sim \pi(\cdot|s_t)$. A deterministic policy is a special case when $\pi(s)$ is a point mass for all $s \in \mathcal{S}$.

For a fixed policy and a starting state $s_0 = s$, we define the value function $V^\pi_M : \mathcal{S} \to \mathbb{R}$ as the discounted sum of future rewards

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

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

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

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

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

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

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

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^π 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 γ^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.

1.1.3 Bellman consistency equations for stationary policies

By definition, V^π and Q^π satisfy the following *Bellman consistency equations*: for all $s \in \mathcal{S}, a \in \mathcal{A}$,

$$\begin{aligned} V^\pi(s) &= Q^\pi(s,\pi(s)). \\ Q^\pi(s,a) &= r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[V^\pi(s')\right], \end{aligned} \tag{0.2}$$

where we are treating π as a deterministic policy.

It is helpful to view V^π as vector of length $\mathcal{S}$ and Q^π 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^π to be the transition matrix on state-action pairs induced by a deterministic policy π. In particular,

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

For a randomized stationary policy, we have

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

With this notation, it is straightforward to verify:

$$\begin{aligned} Q^\pi &= r + \gamma P V^\pi \\ Q^\pi &= r + \gamma P^\pi Q^\pi \,. \end{aligned}$$

The above implies that:

$$Q^\pi = (I - \gamma P^\pi)^{-1} r \tag{0.3}$$

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

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

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

1.1.4 Bellman optimality equations

Due to the Markov structure, there exists a stationary and deterministic policy that simultaneously maximizes $V^\pi(s)$ for all $s \in \mathcal{S}$ and maximizes $Q^\pi(s,a)$ for all $s \in \mathcal{S}, a \in \mathcal{A}$; we denote this *optimal policy* as $\pi^\star_M$ (or $\pi^\star$). This is formalized in the following theorem:

Theorem 1.4. *Let Π be the set of all non-stationary and randomized policies. There exists a stationary and deterministic policy π such that, for all $s \in \mathcal{S}$,*

$$V^{\pi}(s) = \max_{\pi' \in \Pi} V_M^{\pi'}(s).$$

We refer to such a π as an optimal policy.

Proof: We will show the deterministic and stationary policy $\pi(s) = \mathrm{argmax}_{a \in \mathcal{A}} \max_{\pi' \in \Pi} Q^{\pi'}(s,a)$ is an optimal policy. By definition of π, there exists a policy $\widetilde{\pi}$, such that $\widetilde{\pi}$ chooses the same action as $\pi(s)$ at time step 0 and that, for all $s \in \mathcal{S}$,

$$V^{\widetilde{\pi}}(s) = \max_{\pi' \in \Pi} V_M^{\pi'}(s).$$

In other words, $\widetilde{\pi}$ is an optimal policy.

Define the policy π_τ to be a policy which acts according to π before time τ and, at time τ, it starts to execute the policy $\widetilde{\pi}$, starting from state s_τ. We now show that π_τ is optimal for all $\tau \geq 0$. By construction $\pi_0 = \widetilde{\pi}$ is optimal. Let us now show that π_τ is optimal given that $\pi_{\tau-1}$ is optimal. Observe that:

$$\begin{aligned}
& V^{\pi_{\tau-1}}(s) \\
= \;& \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_\tau, a_\tau) \;\Big|\; s_0 = s\Big] \\
= \;& \mathbb{E}\left[r(s_0,a_0) + \dots \gamma^{\tau-1} r(s_{\tau-1}, a_{\tau-1}) + \gamma^\tau \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_{t+\tau}, a_{t+\tau}) | s_\tau = s\right] \;\Big|\; s_0 = s, a_0 = a \right] \\
\leq \;& \mathbb{E}\left[r(s_0,a_0) + \dots \gamma^{\tau-1} r(s_{\tau-1}, a_{\tau-1}) + \gamma^\tau \max_{\pi' \in \Pi} \left(\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_{t+\tau}, a_{t+\tau}) | \pi', s_\tau = s\right]\right) \;\Big|\; s_0 = s, a_0 = a \right] \\
= \;& \mathbb{E}\Big[r(s_0,a_0) + \dots \gamma^{\tau-1} r(s_{\tau-1}, a_{\tau-1}) + \gamma^\tau V^{\widetilde{\pi}}(s_\tau) \;\Big|\; s_0 = s \Big] \\
= \;& V^{\pi_\tau}(s)\,.
\end{aligned}$$

Now since $V^{\pi_{\tau-1}}(s)$ was optimal, then we have that $V^{\pi_\tau}(s)$ is also optimal. This completes the proof. ■

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.

Theorem 1.5. *Let $Q^\star$ by defined as the vector $Q^\star(s,a) = \max_{\pi \in \Pi} Q^\pi(s,a)$ where Π is the space of all (non-stationary and randomized) policies. We have that a vector $Q \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ is equal to $Q^\star$ if and only if it satisfies:*

$$Q^\star(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\max_{a' \in \mathcal{A}} Q^\star(s',a') \right]. \tag{0.4}$$

Before we prove this claim, we will provide a few definitions. We use $V^\star$ and $Q^\star$ as a shorthand for $V^{\pi^\star}$ and $Q^{\pi^\star}$, respectively. We let π_Q denote the greedy policy with respect to a vector $Q \in \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|}$, i.e

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

where ties are broken in some arbitrary (and deterministic) manner. With this notation, the optimal policy $\pi^\star$ is obtained by choosing actions greedily (with arbitrary tie-breaking mechanisms) with respect to Q, i.e.

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

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

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

The *Bellman optimality operator* $\mathcal{T}_M : \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ is defined by the follows: for a vector $Q \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$,

$$\mathcal{T}Q := r + \gamma P V_Q \,. \tag{0.5}$$

This allows us to rewrite Equation 0.4 in the concise form:

$$Q^\star = \mathcal{T}Q^\star$$

i.e. $Q^\star$ is a fixed point of the operator $\mathcal{T}$.

Proof: We first show that $Q^\star$ (the state-action value of an optimal policy) satisfies $Q^\star = \mathcal{T}Q^\star$. For an optimal value function, we have that $V^\star(s) = \max_a Q^\star(s,a)$. For all actions $a \in \mathcal{A}$, we have:

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

Here the equality (a) follows from Theorem 1.4 due to that there exists a policy that is optimal for every starting state.

For the converse, suppose $Q = \mathcal{T}Q$ for some Q. For $\pi = \pi_Q$, this implies that $Q = r + \gamma P^{\pi_Q} Q$. This implies:

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

using Equation 0.3 in the last step. In other words, Q is the action value of the policy π_Q. Now observe for any other policy π':

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

The proof is completed by noting that $(P^{\pi'} - P^\pi)Q^\pi \leq 0$. To see this, observe that:

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

where we use $\pi = \pi_Q$ in the last step. ■

1.2 Computational Complexity

The remainder of this section will be concerned with computing an optimal policy when given knowledge of the MDP $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$. 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, γ) 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, γ) 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 $L(P, r, \gamma)$).

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

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 γ are value and policy iteration polynomial time algorithms; otherwise, they are not polynomial time algorithms. Similarly, only for a fixed value of γ 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 1.5 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 Iterative Methods

Planning refers to the problem of computing $\pi^\star_M$ given the MDP specification $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$. This section reviews classical planning algorithms that compute $Q^\star$.

1.3.1 Value Iteration

A simple algorithm 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 1.6. *(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 1.7. *(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:

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

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

Theorem 1.8. *(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 1.6 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 γ and using Lemma 1.7. ■

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 starts from an arbitrary policy π_0, and repeat the following iterative procedure: for $k = 0, 1, 2, \ldots$

1. *Policy evaluation.* Compute Q^{π_k}

2. *Policy improvement.* Update the policy:

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

In each iteration, we compute the Q-value function of π_k, using the analytical form given in Equation 0.3, 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 1.9. *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,

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

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 π_{k+1} is the greedy policy in the first inequality and recursion in the second inequality. Using this,

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

which completes the proof of the first claim.

For the second claim,

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

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 1.6 in the last step. ■

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

Theorem 1.10. *(policy iteration convergence). Let π_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 γ, 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 1.5 for references.

1.4 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 γ 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.

1.4.1 The Primal LP and A Polynomial Time Algorithm

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

$$\begin{aligned} \min \quad & \sum_s \mu(s) V(s) \\ \text{subject to} \quad & V(s) \geq r(s,a) + \gamma \sum_{s'} P(s'|s,a) V(s') \quad \forall a \in \mathcal{A},\, s \in \mathcal{S} \end{aligned}$$

Here, 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 1.5 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].

1.4.2 The Dual LP and the State-Action Polytope

For a fixed (possibly stochastic) policy π, let us define the state-action visitation distribution ν_μ^π as:

$$\nu_\mu^\pi(s,a) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathrm{Pr}^\pi(s_t = s, a_t = a)$$

where $\mathrm{Pr}^\pi(s_t = s, a_t = a)$ is the state-action visitation probability, where we execute π in M starting at state $s_0 \sim \mu$.

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

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

Let us define the state-action polytope as follows:

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

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

Proposition 1.11. We have that $\mathcal{K}$ is equal to the set of all feasible state-action distributions, i.e. $\nu \in \mathcal{K}$ if and only if there exists a stationary (and possibly randomized) policy π such that $\nu_\mu^\pi = \nu$.

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

$$\begin{array}{ll} \max & \frac{1}{1-\gamma} \sum_{s,a} \nu(s,a) r(s,a) \\ \text{subject to} & \nu \in \mathcal{K} \end{array}$$

Note that $\mathcal{K}$ 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 $\nu^\star$ is the solution to this LP, then we have that:

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

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

1.5 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 1.11

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 1.7 is due to Singh and Yee [1994].

[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

Let us now look at the statistical complexity of learning a near optimal policy. Here, we look at a more abstract sampling model, a generative model, which allows us study the minimum number of transitions we need to observe. This chapter characterizes the minimax optimal sample complexity of estimating $Q^\star$ and learning a near optimal policy.

In this chapter, we will assume that the reward function is known (and deterministic). This is often a mild assumption, particularly due to that much of the difficulty in RL is due to the uncertainty in the transition model P. This will also not effect the minimax sample complexity.

A *generative model* provides us with a sample $s' \sim P(\cdot|s,a)$ upon input of a state action pair (s,a). 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 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 $|\mathcal{S}||\mathcal{A}|N$.

The generative model setting is a reasonable abstraction to understand the statistical limit, without having to directly address exploration.

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 $\widehat{V}^\pi$, $\widehat{Q}^\pi$, $\widehat{Q}^\star$ $\widehat{\pi}^\star$ denote the value function, action value function, and optimal policy in $\widehat{M}$.

A key question here is:

Do we require an accurate model of the world in order to find a near optimal policy?

Let's us first start by looking at the naive approach where we build an accurate model of world, which will be sufficient for learning a near optimal policy. In particular, as we shall see $O(|\mathcal{S}|^2|\mathcal{A}|)$ is sufficient to provide us with an accurate model [1] The question is if we can improve upon this and find a near optimal policy with a number of samples that is *sub-linear* in the model size, i.e. use a number of samples that is smaller than $O(|\mathcal{S}|^2|\mathcal{A}|)$. Furthermore, we also wish to characterize the minimax dependence on the effective horizon, i.e. on the dependence on $1/(1-\gamma)$.

[1] Note that this is consistent with parameter counting since P is specified by $O|\mathcal{S}|^2|\mathcal{A}|$ parameters.

2.1 Warmup: a naive model-based approach

Note that since P has a $|\mathcal{S}|^2|\mathcal{A}|$ parameters, a naive approach would be to estimate P accurately and then use our accurate model $\widehat{P}$ for planning.

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

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

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

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

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

- (Uniform value accuracy) For all policies π,

$$\|Q^\pi - \widehat{Q}^\pi\|_\infty \leq \epsilon/2$$

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

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

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

Lemma 2.2. *(Simulation Lemma) For all π we have that:*

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

Proof: Using our matrix equality for Q^π (see Equation 0.3), 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 π, 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

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

where the final inequality follows since $P^\pi w$ is an average of the elements of w by the definition of P^π 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 ℓ_1 norm (Lemma A.4), 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 δ and c appropriately).

For the second claim, we have that:

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

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

The proof for the final claim immediately follows from the second claim. ■

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 minimax optimal sample complexity, which we will see is sublinear in the model size.

We wills state our results in terms of N, and recall that N is the # of call to the generative models per state actin 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 minimax optimal rate.

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

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

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 1.7, we loose an additional horizon factor and have:

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

We return to this point in Corollary 2.7 and Theorem 2.8.

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

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

$$\begin{array}{rcl} Q^\star - \widehat{Q}^\star & \leq & \gamma(I-\gamma \widehat{P}^{\pi^\star})^{-1}(P-\widehat{P})V^\star \\ Q^\star - \widehat{Q}^\star & \geq & \gamma(I-\gamma \widehat{P}^{\widehat{\pi}^\star})^{-1}(P-\widehat{P})V^\star \end{array}$$

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,

$$\begin{array}{rcl} Q^\star - \widehat{Q}^\star & = & Q^{\pi^\star} - \widehat{Q}^{\widehat{\pi}^\star} \\ & = & (1-\gamma)\Big((I-\gamma P^{\pi^\star})^{-1}r - (I-\gamma \widehat{P}^{\widehat{\pi}^\star})^{-1}r\Big) \\ & = & (I-\gamma \widehat{P}^{\pi^\star})^{-1}((I-\gamma \widehat{P}^{\widehat{\pi}^\star}) - (I-\gamma P^{\pi^\star}))Q^\star \\ & = & \gamma(I-\gamma \widehat{P}^{\pi^\star})^{-1}(P^{\pi^\star} - \widehat{P}^{\widehat{\pi}^\star})Q^\star \\ & \leq & \gamma(I-\gamma \widehat{P}^{\pi^\star})^{-1}(P^{\pi^\star} - \widehat{P}^{\pi^\star})Q^\star \\ & = & \gamma(I-\gamma \widehat{P}^{\pi^\star})^{-1}(P-\widehat{P})V^\star\,, \end{array}$$

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 Minimax Optimal Sample Complexity with the Model Based Approach

We now refine the crude bound on $\widehat{Q}^\star$ to be optimal:

Theorem 2.6. *(Value estimation) For $\delta \geq 0$ and with probability greater than $1-\delta$,*

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

where c is an absolute constant.

Corollary 2.7. *Provided that $\epsilon \leq 1$, we have that if*

$$N \geq \frac{c}{(1-\gamma)^3} \frac{|\mathcal{S}||\mathcal{A}| \log(c|\mathcal{S}||\mathcal{A}|/\delta)}{\epsilon^2},$$

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

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

Note that this implies $\|Q^\star - Q^{\widehat{\pi}^\star}\|_\infty \leq \epsilon/(1-\gamma)$.

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 $Q^\star$. The above corollary shows is not sharp with regards to finding a near optimal policy. The following Theorem shows that in fact both value estimation and policy estimation have the same rate.

Theorem 2.8. *Provided that $\epsilon \leq \sqrt{\frac{1}{1-\gamma}}$, we have that if*

$$N \geq \frac{c}{(1-\gamma)^3} \frac{|\mathcal{S}||\mathcal{A}| \log(c|\mathcal{S}||\mathcal{A}|/\delta)}{\epsilon^2},$$

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

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

We state this improved theorem without proof due to it being more involved, and only prove Theorem 2.6. See Section 2.5 for further discussion.

2.3.1 Lower Bounds

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

Theorem 2.9. *There exists ϵ_0, δ_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 (ϵ,δ)-good on all $M \in \mathcal{M}$, then $\mathcal{A}$ must use a number of samples that is lower bounded as follows*

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

2.3.2 Variance Lemmas

The key to the shaper 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 $\delta \geq 0$. With probability greater than $1 - \delta$,*

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

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 Σ^π_M as the variance of the discounted reward, i.e.

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

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

ex The following lemma shows that Σ^π_M satisfies a Bellman consistency condition.

Lemma 2.11. *(Bellman consistency of Σ) For any MDP M,*

$$\Sigma^\pi_M = \gamma^2\mathrm{Var}_P(V^\pi_M) + \gamma^2 P^\pi \Sigma^\pi_M \tag{0.1}$$

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 π and MDP M,*

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

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 ν (where ν 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, $\Sigma^\pi_M = \gamma^2(I - \gamma^2 P^\pi)^{-1}\mathrm{Var}_P(V^\pi_M)$, so taking $v = \mathrm{Var}_P(V^\pi_M)$ and using that $\|\Sigma^\pi_M\|_\infty \leq \gamma^2/(1-\gamma)^2$ completes the proof.

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

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

which proves the claim. ■

2.3.3 Completing the proof

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

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

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,

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

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

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

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

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

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

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

where $\Delta_{\delta,N}$ is defined in Proposition 2.4. To obtain a cumulative probability of error less than δ, we replace δ 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.

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

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

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. ■

2.4 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 2.8). In other words, if we had normalized the value functions [2], then for additive accuracy (on our normalized value functions) our sample size would scale linearly with the effective horizon.

2.5 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).

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 $\widehat{\pi}^\star$. The improvement in Theorem 2.8 is due to [Agarwal et al., 2020], which shows that the model based approach is near optimal, without any amplification of $1/(1-\gamma)$.

Finally, we remark that we may hope for the bounds on our value estimation to hold up to $\epsilon \leq 1/(1-\gamma)$, which would be consistent with the lower bounds. 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 an open question if the naive model based approach also achieves the non-asymptotic statistical limit.

[2] 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, this book it is helpful to understand sample size with regards to normalized quantities.

Chapter 3

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.

3.1 The K-Armed Bandit Problem

The setting is where we have K decisions (the "arms"), where when we play arm $i \in \{1, 2, \dots K\}$ we obtain a random reward r_i which has mean reward:

$$E[r_i] = \mu_i$$

where we assume $\mu_i \in [-1, 1]$.

Our cumulative regret is defined as:

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

Theorem 3.1. *To be added...*

3.2 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 $r_t \in [-1, 1]$.

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*,

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

Algorithm 1 The Linear UCB algorithm

```
Input: $\lambda$, $\beta_t$
for $t = 0, 1 \ldots$ do
    Execute
        $x_t = \mathrm{argmax}_{x \in D} \max_{\mu \in \mathrm{BALL}_t} \mu \cdot x$
    and observe the reward $r_t$.
    Update $\mathrm{BALL}_{t+1}$ (as specified in Equation 0.1).
end for
```

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.

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

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.2), 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.

3.2.1 The LinUCB algorithm

LinUCB is based on "optimism in the face of uncertainty," which is described in Algorithm 1. At episode t, we use all previous experience to define an uncertainty region (an ellipse) BALL_t. The center of this region, $\widehat{\mu}_t$, is the solution of the following regularized least squares problem:

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

where λ 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 BALL_t is defined through the feature covariance Σ_t.

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

$$\textsc{Ball}_t = \left\{ (\widehat{\mu}_t - \mu^\star)^\top \Sigma_t (\widehat{\mu}_t - \mu^\star) \leq \beta_t \right\}, \tag{0.1}$$

where β_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 ν. 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 ℓ_1 ball which contains $\textsc{Ball}_t$. See Section 3.4 for further reading.

3.2.2 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 3.2. *Suppose that the noise η_t is σ^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*

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

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

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

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

3.3 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 3.3. (Confidence) Let $\delta > 0$. We have that

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

Section 3.3.2 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

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

[1] Roughly speaking, this say that tail probabilities of η_t decay no more slowly than a Gaussian distribution. If the noise is bounded, i.e. $|\eta_t| \leq B$

which denotes the instantaneous regret.

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

Proposition 3.4. (Sum of Squares Regret Bound) Suppose that $\|x\| \leq B$ for $x \in D$. Suppose β_t is increasing and larger than 1. For LinUCB, if $\mu^\star \in \textsc{Ball}_t$ for all t, then

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

This is proven in Section 3.3.1. The idea of the proof involves a potential function argument on the log volume (i.e. the log determinant) of the "precision matrix" Σ_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 3.2] By Propositions 3.3 and 3.4 along with the Cauchy-Schwarz inequality, we have, with probability at least $1 - \delta$,

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

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

We now provide the proofs of these two propositions.

3.3.1 Regret Analysis

In this section, we prove Proposition 3.4, 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 $\mu^\star \in \textsc{Ball}_t$, 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 ℓ_2 norm.

Lemma 3.5. *Let $x \in D$. If $\mu \in \textsc{Ball}_t$ 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:

$$\begin{aligned} |(\mu - \widehat{\mu}_t)^\top x| &= |(\mu - \widehat{\mu}_t)^\top \Sigma_t^{1/2} \Sigma_t^{-1/2} x| = |(\Sigma_t^{1/2}(\mu - \widehat{\mu}_t))^\top \Sigma_t^{-1/2} x| \\ &\leq \|\Sigma_t^{1/2}(\mu - \widehat{\mu}_t)\| \|\Sigma_t^{-1/2} x\| = \|\Sigma_t^{1/2}(\mu - \widehat{\mu}_t)\| \sqrt{x^\top \Sigma_t^{-1} x} \leq \sqrt{\beta_t x^\top \Sigma_t^{-1} x} \end{aligned}$$

where the last inequality holds since $\mu \in \textsc{Ball}_t$. ■

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 3.6. *Fix $t \leq T$. If $\mu^\star \in \textsc{Ball}_t$, then*

$$\text{regret}_t \leq 2\min\left(\sqrt{\beta_t}w_t, 1\right) \leq 2\sqrt{\beta_T}\min\left(w_t, 1\right)$$

Proof: Let $\widetilde{\mu} \in \textsc{Ball}_t$ denote the vector which minimizes the dot product $\widetilde{\mu}^\top x_t$. By choice of x_t, we have

$$\widetilde{\mu}^\top x_t = \max_{\mu \in \textsc{Ball}_t} \max_{x \in D} \mu^\top x \geq (\mu^\star)^\top x^*,$$

where the inequality used the hypothesis $\mu^\star \in \textsc{Ball}_t$. Hence,

$$\begin{aligned} \text{regret}_t &= (\mu^\star)^\top x^* - (\mu^\star)^\top x_t \leq (\widetilde{\mu} - \mu^\star)^\top x_t \\ &= (\widetilde{\mu} - \widehat{\mu}_t)^\top x_t + (\widehat{\mu}_t - \mu^\star)^\top x_t \leq 2\sqrt{\beta_t}w_t \end{aligned}$$

where the last step follows from Lemma 3.5 since $\widetilde{\mu}$ and $\mu^\star$ are in $\textsc{Ball}_t$. Since $r_t \in [-1, 1]$, regret_t is always at most 2 and the first inequality follows. The final inequality is due to that β_t is increasing and larger than 1. ■

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 3.7. *We have:*

$$\det \Sigma_T = \det \Sigma_0 \prod_{t=0}^{T-1}(1 + w_t^2).$$

Proof: By the definition of Σ_{t+1}, we have

$$\begin{aligned} \det \Sigma_{t+1} &= \det(\Sigma_t + x_t x_t^\top) = \det(\Sigma_t^{1/2}(I + \Sigma_t^{-1/2}x_t x_t^\top \Sigma_t^{-1/2})\Sigma_t^{1/2}) \\ &= \det(\Sigma_t)\det(I + \Sigma_t^{-1/2}x_t(\Sigma_t^{-1/2}x_t)^\top) = \det(\Sigma_t)\det(I + v_t v_t^\top), \end{aligned}$$

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. ■

Lemma 3.8. *("Potential Function" Bound) For any sequence $x_0, \ldots x_{T-1}$ such that, for $t < T$, $\|x_t\|_2 \leq B$, we have:*

$$\log\Big(\det \Sigma_{T-1}/\det \Sigma_0\Big) = \log\det\left(I + \frac{1}{\lambda}\sum_{t=0}^{T-1} x_t x_t^\top\right) \leq d\log\left(1 + \frac{TB^2}{d\lambda}\right).$$

Proof: Denote the eigenvalues of $\sum_{t=0}^{T-1} x_t x_t^\top$ as $\sigma_1, \ldots \sigma_d$, and note:

$$\sum_{i=1}^{d} \sigma_i = \text{Trace}\left(\sum_{t=0}^{T-1} x_t x_t^\top\right) = \sum_{t=0}^{T-1}\|x_t\|^2 \leq TB^2.$$

with probability one, then we can take $\sigma^2 = B^2$.

Using the AM-GM inequality,

$$\log\det\left(I+\frac{1}{\lambda}\sum_{t=0}^{T-1}x_t x_t^\top\right)=\log\left(\prod_{i=1}^{d}(1+\sigma_i/\lambda)\right)$$
$$=d\log\left(\prod_{i=1}^{d}(1+\sigma_i/\lambda)\right)^{1/d}\le d\log\left(\frac{1}{d}\sum_{i=1}^{d}(1+\sigma_i/\lambda)\right)\le d\log\left(1+\frac{TB^2}{d\lambda}\right),$$

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 3.4] Assume that $\mu^\star\in\textsc{Ball}_t$ for all t. We have that:

$$\sum_{t=0}^{T-1}\text{regret}_t^2\le\sum_{t=0}^{T-1}4\beta_t\min(w_t^2,1)\le 4\beta_T\sum_{t=0}^{T-1}\min(w_t^2,1)$$
$$\le 4\beta_T\sum_{t=0}^{T-1}\ln(1+w_t^2)\le 4\beta_T\log\Big(\det\Sigma_{T-1}/\det\Sigma_0\Big)=4\beta_T d\log\left(1+\frac{TB^2}{d\lambda}\right)$$

where the first inequality follow from By Lemma 3.6; the second from that β_t is an increasing function of t; the third uses that for $0\le y\le 1$, $\ln(1+y)\ge y/2$; the final two inequalities follow by Lemmas 3.7 and 3.8. ■

3.3.2 Confidence Analysis

Proof:[Proof of Proposition 3.3] 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.5, it holds with probability at least $1-\delta_t$,

$$\sqrt{(\widehat{\mu}_t-\mu^\star)^\top\Sigma_t(\widehat{\mu}_t-\mu^\star)}=\|(\Sigma_t)^{1/2}(\widehat{\mu}_t-\mu^\star)\|^2$$
$$\le\left\|\lambda(\Sigma_t)^{-1/2}\mu^\star\right\|+\left\|\Sigma_t^{-1}\sum_{\tau=0}^{t-1}\eta_\tau x_\tau\right\|$$
$$\le\sqrt{\lambda}\|\mu^\star\|+\sqrt{2\sigma^2\log\left(\det(\Sigma_t)\det(\Sigma^0)^{-1}/\delta_t\right)}.$$

where we have also used the triangle inequality.

We seek to lower bound $\Pr(\forall t,\,\mu^\star\in\textsc{Ball}_t)$. Note that at $t=0$, by our choice of λ, we have that $\textsc{Ball}_0$ contains $W^\star$, so $\Pr(\mu^\star\notin\textsc{Ball}_0)=0$. For $t\ge 1$, let us assign failure probability $\delta_t=(3/\pi^2)/t^2$ for the t-th event, which, using the above, gives us an upper bound on the sum failure probability as

$$1-\Pr(\forall t,\,\mu^\star\in\textsc{Ball}_t)=\Pr(\exists t,\,\mu^\star\notin\textsc{Ball}_t)\le\sum_{t=1}^{\infty}\Pr(\mu^\star\notin\textsc{Ball}_t)<\sum_{t=1}^{\infty}(1/t^2)(3/\pi^2)=1/2.$$

This along with Lemma 3.8 completes the proof. ■

3.4 Further References and 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 sub-linear regret bound here was due to [Auer et al., 2002], which used a more complicated algorithm.

Chapter 4

Generalization

Up to now we have focussed on "tabular" MDPs. While studying this setting is theoretically important, we ultimately 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.

A fundamental question here is:

To what extent is generalization in RL similar to (or different from) that in supervised learning?

This is the focus of this chapter. Understanding this question is crucial in how we study (and design) scalable algorithms. These insights will also help us to motivate the various more refined assumptions (and settings) that we will consider in subsequent chapters.

In supervised learning (and binary classification in particular), it is helpful to distinguish between two different objectives: First, it is not difficult to see that, in general, it is not possible to learn the Bayes optimal classifier in a sample efficient manner without strong underlying assumptions on the data generating process.[1] Alternatively, given some restricted set of classifiers (our hypothesis class $\mathcal{H}$, which may not contain the Bayes optimal classifier), we may hope to do as well as the best classifier in this set, i.e. we seek low (statistical) *regret*. This objective is referred to as agnostic learning; here, obtaining low regret is possible, provided some measure of the complexity of our hypothesis set is not too large.

With regards to reinforcement learning, we may ask a similar question. It is not difficult to see that in order to provably learn the truly optimal policy in a sample efficient manner (say that does not depend on the number of states $|\mathcal{S}|$), then we must rely on quite strong assumptions. Analogous to the agnostic learning question in supervised learning, we may ask the following question: given some restricted (and low complexity) policy class Π (which may not contain the optimal policy $\pi^\star$), what is the sample complexity of doing nearly as well as the best policy in this class?

Before we address this question, a few remarks are in order.

Binary classification as a $\gamma = 0$ RL problem. Let us observe that the problem of binary classification can be thought of as learning in an MDP: take $\gamma = 0$ (i.e. the effective horizon is 1); suppose we have a distribution of starting states $s_0 \sim \mu$; suppose $|\mathcal{A}| = 2$; and the reward function is $r(s, a) = \mathbf{1}(\text{label}(s) = a)$. In other words, we equate our action with the prediction of the binary class, and the reward function is 1 or 0, determined by if our prediction is correct.

[1] Such impossibility results are often referred to as a "No free lunch theorem" theorems.

Sampling model In this chapter, we consider a weaker (and more realistic) sampling model where we have a starting state distribution μ over states. We assume sampling access to the MDP where we start at a state $s_0 \sim \mu$; we can rollout a policy π of our choosing; and we can terminate the trajectory at will. We are interested in learning with a small number of observed trajectories.

4.1 Review: Binary Classification and Generalization

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:

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

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

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

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

Proposition 4.1. (The "Occam's razor" bound) Suppose $\mathcal{H}$ is finite. Let $\widehat{h} = \arg\min_{h\in\mathcal{H}} \widehat{\text{err}}(h)$ and $h^\star = \arg\min_{h\in\mathcal{H}} \text{err}(h)$. With probability at least $1 - \delta$:

$$\text{err}(\widehat{h}) - \text{err}(h^\star) \leq \sqrt{\frac{2}{N}\log\frac{2|\mathcal{H}|}{\delta}}.$$

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:

$$\text{err}(\widehat{h}) - \text{err}(h^\star) \leq \epsilon.$$

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.

In the supervised learning setting, a crucial observation is that even though a hypothesis set $\mathcal{H}$ may be infinite, the number of possible behaviors of on a finite set of states is not necessarily exhaustive. Let us review the definition of the VC dimension for a hypothesis set of boolean functions. We say that the set $\{x_1, x_2, \ldots x_d\}$ is shattered if there exists an $h \in \mathcal{H}$ that can realize any of the possible 2^d labellings. The *Vapnik–Chervonenkis* (VC) dimension is the size of the largest shattered set. If $d = VC(\mathcal{H})$, then the Sauer–Shelah lemma states the number of possible labellings on a set of N points by functions in $\mathcal{H}$ is at most $\left(\frac{eN}{d}\right)^d$. For $d << N$, this is much less than 2^N.

The following classical bound highlights how generalization is possible on infinite hypothesis classes with VC dimension.

Proposition 4.2. (VC dimension and generalization) Let $\widehat{h} = \arg\min_{h\in\mathcal{H}} \widehat{\text{err}}(h)$ and $h^\star = \arg\min_{h\in\mathcal{H}} \text{err}(h)$. Suppose $\mathcal{H}$ has a bounded VC dimension. For $m \geq \text{VC}(\mathcal{H})$, we have that with probability at least $1-\delta$:

$$\text{err}(\widehat{h}) - \text{err}(h^\star) \leq \sqrt{\frac{c}{N}\left(\text{VC}(\mathcal{H})\log\frac{2N}{\text{VC}(\mathcal{H})} + \log\frac{2}{\delta}\right)},$$

where c is an absolute constant

4.2 Generalization and Agnostic Learning in RL

Now consider the case where we have a set of policies Π (either finite or infinite). For example, Π could be a parametric set. Alternatively, we could have a set of parametric value functions $\mathcal{V} = \{f_\theta : \mathcal{S}\times\mathcal{A}\to\mathbb{R} |\, \theta\in\mathbb{R}^d\}$, and Π could be the set of policies that are greedy with respect to values in $\mathcal{V}$.

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

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

As before, we only hope to perform favorably against the best policy in Π. Recall that in our aforementioned sampling model we have the ability to obtain trajectories from $s_0\sim\mu$ under policies of our choosing. As we have seen, agnostic learning is possible in the supervised learning setting, with regret bounds that have no dependence on the size of the domain — the size of domain is analogous to the size the state space $|\mathcal{S}|$.

4.2.1 Upper Bounds: Data Reuse and Importance Sampling

We now provide a reduction of RL to the supervised learning problem. The key issue is how to efficiently reuse data. Here, we will simply collect N trajectories by executing a policy which chooses samples uniformly at random; let π_{uar} denote this policy. For simplicity, we only consider deterministic policies.

The following shows how we can obtain a nearly unbiased estimate of the reward with this uniform policy:

Lemma 4.3. *(Near unbiased estimation of $V^\pi(s_0)$) We have that:*

$$|\mathcal{A}|^H\, \mathbb{E}_{\pi_{\text{uar}}}\left[\mathbf{1}\Big(\pi(s_0)=a_0,\,\ldots,\pi(s_H)=a_H\Big)\sum_{t=0}^{H}\gamma^t r(s_t,a_t)\Big| s_0\right] = \mathbb{E}_\pi\left[\sum_{t=0}^{H}\gamma^t r(s_t,a_t)\Big| s_0\right].$$

(Truncation) We also have that:

$$|V^\pi(s_0) - \mathbb{E}_\pi\left[\sum_{t=0}^{H}\gamma^t r(s_t,a_t)\right]| \leq \gamma^H/(1-\gamma),$$

which implies that for $H = \frac{\log\left(1/\left(\epsilon(1-\gamma)\right)\right)}{1-\gamma}$ we will have an ϵ approximation to $V^\pi(s_0)$.

In other words, the estimated reward of π on a trajectory is nonzero only when π takes exactly identical actions to those taken by π_{uar} on the trajectory, in which case the estimated value of π is $|\mathcal{A}|^H$ times that of π_{uar}. Note the factor of $|\mathcal{A}|^H$, due to importance sampling, leads this being a high variance estimate. We will return to this point in the next section.

Proof: To be added... ■

Denote the n-th sampled trajectory by $(s_0^n, a_0^n, r_1^n, s_1^n, \ldots, s_H^n)$, where H is the cutoff time where the trajectory ends. We can then use following to estimate the γ-discounted reward of any given policy π:

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

Proposition 4.4. (Generalization in RL) Suppose Π is a finite set of policies. Let $\widehat{\pi} = \arg\max_{\pi\in\Pi} \widehat{V}^\pi(s_0)$. Using $H = \frac{\log\big(2/\big(\epsilon(1-\gamma)\big)\big)}{1-\gamma}$ we have that with probability at least $1-\delta$:

$$V^{\widehat{\pi}}(s_0) \geq \arg\max_{\pi\in\Pi} V^\pi(s_0) - \frac{\epsilon}{2} - |\mathcal{A}|^H \sqrt{\frac{2}{N}\log\frac{2|\Pi|}{\delta}}.$$

Hence, provided that

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

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

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

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).

With regards to infinite hypothesis classes of policies, extending the Occam's razor bound can be done with standard approaches from statistical learning theory. For example, consider the case where $|\mathcal{A}| = 2$, where Π is class of deterministic policies. Here, as each $\pi \in \Pi$ can be viewed as Boolean function, VC(Π) is defined in the usual manner. Here, we have:

Proposition 4.5. (Bounded VC dimension) Suppose $|\mathcal{A}| = 2$ and that suppose Π has a bounded VC dimension. Let $\widehat{\pi} = \arg\max_{\pi\in\Pi} \widehat{V}^\pi(s_0)$. Using $H = \frac{\log\big(2/\big(\epsilon(1-\gamma)\big)\big)}{1-\gamma}$ and for $N \geq \text{VC}(\Pi)$, we have that with probability at least $1-\delta$:

$$V^{\widehat{\pi}}(s_0) \geq \arg\max_{\pi\in\Pi} V^\pi(s_0) - \frac{\epsilon}{2} - 2^H \sqrt{\frac{c}{N}\left(\text{VC}(\Pi)\log\frac{2N}{\text{VC}(\Pi)} + \log\frac{2}{\delta}\right)},$$

where c is an absolute constant.

We do not prove this result here, which follows a standard argument using results in statistical learning theory. The key observation here is that, the Sauer–Shelah lemma bounds the number of possible labellings on a set of N trajectories (each of length H) by $\left(\frac{eNH}{d}\right)^d$, where $d = \text{VC}(\Pi)$.

See Section 3.4.

4.2.2 Lower Bounds

Clearly, the drawback of these bounds are that they are 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.

An algorithm is a procedure which sequentially samples trajectories and then returns some policy π (we often say the algorithm is *proper* if it returns a $\pi \in \Pi$). An algorithm is deterministic if it executes a policy (to obtain a trajectory) in manner that is a deterministic function of the data that it has collected. We only consider deterministic algorithms in this section, which does not quantitatively change the conclusions.

First, let us present the following simple observation, which already shows that avoiding an $\exp(1/(1-\gamma))$ dependence is not possible.

Proposition 4.6. (Lower Bound for The Complete Policy Class) Suppose $|\mathcal{A}| = 2$ and $|\mathcal{S}| = 2^H$, where $H = \lfloor \frac{\log(2)}{1-\gamma} \rfloor$. Let Π be the set of all 2^H policies. There exists a family of MDPs such that if a deterministic algorithm $\mathcal{A}$ is guaranteed to find a policy π such that:

$$V^{\widehat{\pi}}(s_0) \geq \arg\max_{\pi \in \Pi} V^{\pi}(s_0) - 1/4.$$

then $\mathcal{A}$ must use $N \geq 2^H$ trajectories.

Observe that $\log |\Pi| = H \log(2)$, so this already rules out the possibility of logarithmic dependence on the size of the policy class, without having an exponential dependence on H. The proof is straightforward, where we consider a family of binary trees where the rewards are at one of the terminal leaf nodes.

Proof: Consider a family of deterministic MDPs, where each in each MDP the dynamics are specified by a binary tree of depth H, with $H = \lfloor \frac{\log(2)}{1-\gamma} \rfloor$ and where there is a reward at one of the terminal leaf nodes. Note that for setting of H, $\gamma^H \leq \exp(-(1-\gamma)H) \geq 1/2$. Since Π is the set of all 2^H policies, then we must check every leaf, in the worst case (due that our algorithm is deterministic). This completes the proof. ■

In the previous proposition, our policy class was the complete class. Often, we are dealing with policies class which are far more restrictive. Even in this case, the following proposition strengthens this lower bound to be applicable to *arbitrary* policy classes, showing that even here (if we seek no dependence on $|\mathcal{S}|$), we must either have exponential dependence on the effective horizon or we must exhaustively try what is the effective size of all our policies.

Proposition 4.7. (Lower Bound for an Arbitrary Policy Class) Define $H = \lfloor \frac{\log(2)}{1-\gamma} \rfloor$. Suppose $|\mathcal{A}| = 2$ and let Π be an arbitrary policy class. There exists a family of MDPs such that if a deterministic algorithm $\mathcal{A}$ is guaranteed to find a policy $\widehat{\pi}$ such that:

$$\mathbb{E}\left[V^{\widehat{\pi}}(s_0)\right] \geq \arg\max_{\pi \in \Pi} V^{\pi}(s_0) - \epsilon.$$

(where the expectation is with respect to the trajectories the algorithm observes) then $\mathcal{A}$ must use an expected number of trajectories N where

$$N \geq c \frac{\min\{2^H, 2^{\text{VC}(\Pi)}\}}{\epsilon^2},$$

where c is a universal constant.

We can interpret $2^{\text{VC}(\Pi)}$ is the effective the number of policies in our policy class (by the definition of the VC dimension, it is number of different behaviors in our policy set). Thus, requiring $O(2^{\text{VC}(\Pi)})$ samples shows that, in the worst case, we are not able to effectively reuse data (as was the case in supervised learning), unless have an exponential dependence on the horizon.

Proof: We will only prove this result for $\epsilon = 1/4$, where we will see that we need

$$N \geq \min\{2^H, 2^{\text{VC}(\Pi)}\}$$

By definition of the VC dimension, our policy class can exhibit $2^{\text{VC}(\Pi)}$ distinct action sequences on $\text{VC}(\Pi)$ states. Suppose $\text{VC}(\Pi) \leq H$. Here, we can construct a binary tree where the set of distinct leaves visited by Π will be precisely equal to $2^{\text{VC}(\Pi)}$. By placing a unit reward at one of these leaves, the algorithm will be forced to explore all of the leaves. If $\text{VC}(\Pi) \leq H$, then exploring the full binary tree is necessary.

We leave the general case as an exercise for the reader. As a hint, consider two different types of leaf nodes: for all but one of the leaf nodes, we obtain unit reward with $1/2$ probability, and, if the remaining leaf node is reached, we obtain unit reward with $1/2 + \epsilon$ probability. ■

4.3 Interpretation: How should we study generalization in RL?

The above clearly shows that, 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 $1/(1-\gamma)$. Note that agnostic learning is not about being (unconditionally) optimal, but only being competitive among some restricted (hopefully lower complexity) set of models. Regardless, even with this weaker success criterion, avoiding the exponential dependence on the effective horizon is simply not possible.

This motivates the study of RL to consider either stronger assumptions or means in which the agent can obtain side information. Three examples of approaches that we will consider in this book are:

- Structural (and Modelling) Assumptions: By making stronger assumptions about the world, we can move away from agnostic learning and escape the curse of dimensionality. We will see examples of this in Part 2.
- Distribution Dependent Results (and Distribution Shift): When we move to policy gradient methods (in Part 3), we will consider results which depend on given distribution of how we obtain samples. Here, we will make connections to transfer learning.
- Imitation learning and behavior cloning: here will consider models where the agent has input from, effectively, a teacher, and we will see how this alleviates the problem of curse of dimensionality.

4.4 Approximation Limits with Linearity Assumptions

Given our previous lower bounds and discussion, it is natural to consider making assumptions. A common assumption is that the Q-function (or value function) is a (nearly) linear function of some given features (our representation); this is a natural assumption to begin our study of *function approximation*. In practice, suche features are either hand-crafted or a pre-trained neural network that transforms a state-action pair to a d-dimensional embedding [2].

We now see that, even when we make such linearity assumptions, there are hard thresholds, on the worst case approximation error of our representation, that have to be satisfied in order for our linearity assumption to be helpful.

We now provide a lower bound on the approximation limits for value-based learning, when we have a approximate linear representation. Formally, the agent is given a feature extractor $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, which can be hand-crafted or a pre-trained neural network that transforms a state-action pair to a d-dimensional embedding. The following assumption states that the given feature extractor can be used to predict the Q-function (of *any* policy) with approximation error at most ϵ_{approx} linear function.

In this section, *we assume we are in the finite horizon (undiscounted) setting.*

[2] The more challenging question is to *learn* the features

Assumption 4.8 (Linear Value Function Approximation)**.** There exists $\epsilon_{\text{approx}} > 0$, such that for any $h \in [H]$ and any policy π, there exists $\theta_h^\pi \in \mathbb{R}^d$ such that for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, $|Q_h^\pi(s,a) - \langle \theta_h, \phi(s,a) \rangle| \le \epsilon_{\text{approx}}$.

Here ϵ_{approx} is the approximation error, which indicates the quality of the representation. If $\epsilon_{\text{approx}} = 0$, then all Q-functions can be perfectly represented by a linear function of $\phi(\cdot,\cdot)$. In general, as we increase the dimension of ϕ we expect that ϵ_{approx} becomes smaller , since larger dimension usually has more expressive power.

Later on, we will see that if ϵ_{approx} is 0, then sample efficient learning is possible (with a polynomial dependence on H and d, but no dependence on $|\mathcal{S}|$ and $|\mathcal{A}|$). The following theorem shows that such assumptions, necessarily, need ϵ_{approx} close to 0, else sample efficient learning is not possible, which is consistent with our agnostic learning lower bounds in this chapter. In particular, The following theorem shows when $\epsilon_{\text{approx}} = \Omega\left(\sqrt{\frac{H}{d}}\right)$, the agent needs to sample exponential number of trajectories to find a near-optimal policy.

Theorem 4.9 (Exponential Lower Bound for Value-based Learning)**.** *There exists a family of MDPs with $|\mathcal{A}| = 2$ and a feature extractor ϕ that satisfy Assumption 4.8, such that any algorithm that returns a $1/2$-optimal policy with probability 0.9 needs to sample $\Omega\left(\min\{|\mathcal{S}|, 2^H, \exp(d\epsilon_{approx}^2/16)\}\right)$ trajectories.*

We state the theorem without proof. The lower bound is again based on a the deterministic binary tree hard instance, with only one rewarding node (i.e. state) at a leaf. With no further assumptions, as before to find a $1/2$-optimal policy for such MDPs, the agent must enumerate all possible states in level $H-1$ to find the state with reward $R = 1$. Doing so intrinsically induces a sample complexity of $\Omega(2^H)$.

Th key idea of the proof is that we can construct a set of features so that Assumption 4.8 holds, and, yet, these features reveal no additional information to the learner (and, so, the previous lower bound still applies). The main idea in the construction uses the following fact regarding the ϵ-approximate rank of the identity matrix of size 2^H: this (large) identity matrix can be approximated to ϵ- accuracy (in the spectral norm) with a matrix of rank only $O(H\epsilon^2)$ [3]. In our context, this fact can be used to construct a set of features ϕ, all of which live in an $O(H\epsilon^2)$ dimensional subspace, where these features well approximate all 2^H value function; the crucial property here is that the features can be constructed with *no knowledge* of the actual reward function.

4.5 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.

The approximation limits with linear function approximation are results from [Du et al., 2019].

[3] Such a result can be proven with the e Johnson-Lindenstrauss Lemma

Part 2

Strategic Exploration

Part 3

Policy Optimization

Part 4

Further Topics

Bibliography

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.

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, Sham Kakade, and Lin F. Yang. Model-based reinforcement learning with a generative model is minimax optimal. In *COLT*, volume 125, pages 67–83, 2020.

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.

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.

Richard Bellman. Dynamic programming and Lagrange multipliers. *Proceedings of the National Academy of Sciences*, 42(10):767–769, 1956.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, pages 355–366, 2008.

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.

Daniel Hsu, Sham Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78, 11 2008.

Sham Machandranath Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University of College London, 2003.

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.

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.

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.

Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

H. Robbins. Some aspects of the sequential design of experiments. In *Bulletin of the American Mathematical Society*, volume 55, 1952.

Satinder Singh and Richard Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.

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.

Appendix A

Concentration

Lemma A.1. *(Hoeffding's inequality) Suppose $X_1, X_2, \dots X_n$ are a sequence of independent, identically distributed (i.i.d.) random variables with mean μ. 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 $1 - \delta$:

$$\bar{X}_n - EX \leq (b_+ - b_-)\sqrt{\ln(1/\delta)/(2n)}\,.$$

Theorem A.2 (Hoeffding-Azuma Inequality)**.** *Suppose $X_1, \dots, X_T$ is a martingale difference sequence where each X_t is a σ sub-Gaussian, Then, for every $1 > \delta > 0$,*

$$\Pr(\sum X_i \geq \epsilon) \leq \exp\left(\frac{-\epsilon^2}{2N\sigma^2}\right).$$

Lemma A.3. *(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 $1 - \delta$,

$$\bar{X}_n - EX \leq \sqrt{2\text{Var}(X)\ln(1/\delta)/n} + \frac{2b\ln(1/\delta)}{3n}.$$

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.4. (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.5 (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 ε_i is $\mathcal{F}_i$ measurable, $\mathbb{E}[\varepsilon_i|\mathcal{F}_{i-1}] = 0$, and ε_i is conditionally σ-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 $t \geq 1$:*

$$\left\| \sum_{i=1}^{t} X_i \varepsilon_i \right\|_{\Sigma_t^{-1}}^2 \leq \sigma^2 \log\left(\frac{\det(\Sigma_t)\det(\Sigma)^{-1}}{\delta^2} \right).$$