

Reinforcement Learning: Theory and Algorithms

Alekh Agarwal Nan Jiang Sham M. Kakade

October 27, 2019

WORKING DRAFT: Text not yet at the level of publication.

Contents

0	Notation	7
1	MDP Preliminaries	9
1.1	Markov Decision Processes	10
1.1.1	Interaction protocol	10
1.1.2	The objective, policies, and values	10
1.1.3	Bellman consistency equations for stationary policies	11
1.1.4	Bellman optimality equations	12
1.2	Planning in MDPs	14
1.2.1	Q -Value Iteration	14
1.2.2	Policy Iteration	15
2	Sample Complexity with a Generative Model	17
2.1	The Generative Model Setting	18
2.2	Sample Complexity	18
2.2.1	A naive approach: accurate model estimation	18
2.2.2	A more refined approach: using a sparse model	19
2.2.3	Lower Bounds	20
2.2.4	What about the Value of the Policy $\hat{\pi}^*$?	20
2.3	Analysis	20
2.3.1	Completing the proof	23
2.4	Bibliographic Remarks	23
3	Strategic Exploration in RL	25

4	Policy Gradient Methods	33
4.1	The Policy Gradient Method	36
4.1.1	Optimization	37
4.2	The Softmax Policy and Relative Entropy Regularization	40
4.3	The Natural Policy Gradient	43
4.3.1	Global Convergence and the Softmax Policy Class	43
4.3.2	Function Approximation and a Connection to Transfer Learning	46
4.4	Related algorithms	51
4.4.1	Trust Region Policy Optimization (TRPO)	51
4.4.2	Proximal Policy Optimization (PPO)	51
4.4.3	Conservative Policy Iteration (CPI)	51
4.5	Bibliographic Remarks	51
5	Value Function Approximation	53
5.1	Approximate Policy Evaluation	54
5.2	Approximate Policy Improvement	59
5.2.1	Greedy policy improvement with ℓ_∞ approximation	59
5.2.2	Conservative Policy Iteration	61
6	Strategic Exploration in RL with rich observations	65
6.1	Problem setting	66
6.2	Value-function approximation	67
6.3	Bellman Rank	68
6.4	Sample-efficient learning for CDPs with a small Bellman rank	70
7	Behavioral Cloning and Apprenticeship Learning	73
7.1	Linear Programming Formulations	74
7.1.1	The Primal LP	74
7.1.2	The Dual LP	74
7.2	Behavioral Cloning	75
7.2.1	Behavioral Cloning via Supervised Learning	76
7.2.2	Behavioral Cloning via Distribution Matching	77
7.2.3	Sample Efficiency: comparing the approaches	78

7.3	Learning from Observation	78
7.3.1	Learning from Observations via Distribution Matching	78
7.4	Inverse Reinforcement Learning	79
A	Concentration	83

Chapter 0

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 say $v \leq v'$, if the inequality holds elementwise.
- For a vector v , we refer to the j -th component of this vector by either $v(j)$ or $[v]_j$
- Denote the **variance** of any real valued f under a distribution \mathcal{D} as:

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

- 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, $P_{(s,a),(s',a')}^\pi = P(s'|s, a)$ if $a' = \pi(s')$ and $P_{(s,a),(s',a')}^\pi = 0$ if $a' \neq \pi(s')$. With this notation,

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

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

表格方法（贪心策略求
最大值以及最优策略）

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

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

$$\mathcal{T}Q := (1 - \gamma)r + \gamma P V_Q. \tag{0.1}$$

Chapter 1

MDP Preliminaries

Markov Decision Processes

Aleksh Agarwal, Nan Jiang, Sham M. Kakade

Chapter 1

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) [Puterman, 1994], specified by:

- State space \mathcal{S} . In this course we only consider finite state spaces.
- Action space \mathcal{A} . In this course we only consider finite action spaces.
- Transition function $P : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$, 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 $P(\cdot | s, a)$.
- Reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$. $r(s, a)$ is the immediate reward associated with taking action a in state s .
- Discount factor $\gamma \in [0, 1)$, which defines a horizon for the problem.

1.1.1 Interaction protocol

In a given MDP $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, the agent interacts with the environment according to the following protocol: the agent starts at some state s_0 ; at each time step $t = 0, 1, 2, \dots$, 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

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

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

In some situations, it is necessary to specify how the initial state s_0 is generated. We consider s_0 sampled from an initial distribution $\mu \in \Delta(\mathcal{S})$. When μ is of importance to the discussion, we include it as part of the MDP definition, and write $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu)$.

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. $\pi : \mathcal{H} \rightarrow \mathcal{A}$ where \mathcal{H} is the set of all possibly trajectories. A deterministic, *stationary* policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ 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 $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$, and, overloading notation, we write $a_t \sim \pi(\cdot | s_t)$. A deterministic policy is its 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 $V_M^\pi : \mathcal{S} \rightarrow \mathbb{R}$ as the average, discounted

sum of future rewards

$$V_M^\pi(s) = (1 - \gamma) \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid \pi, s_0 = s \right].$$

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

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

$$Q_M^\pi(s, a) = (1 - \gamma) \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid \pi, s_0 = s, a_0 = a \right].$$

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:

$$\max_{\pi} V_M^\pi(s) \quad (1.1)$$

The dependence of on M may be dropped 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 $(1 - \gamma)\gamma^d$, where d is the number of steps required by the policy to reach the goal state.

Example 1.2 (Conversational agent). This is another fairly natural RL problem. The state of an agent can be the current transcript of the conversation so far, along with any additional information about the world, such as the context for the conversation, characteristics of the other agents or humans in the conversation etc. Actions depend on the domain. In the rawest 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 (Board games). This is perhaps the most popular category of RL applications, where RL has been successfully applied to solve Backgammon, Go and various forms of Poker. For board games, 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.

1.1.3 Bellman consistency equations for stationary policies

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

$$\begin{aligned} V^\pi(s) &= Q^\pi(s, \pi(s)). \\ Q^\pi(s, a) &= (1 - \gamma)r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} [V^\pi(s')] , \end{aligned} \quad (1.2)$$

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,

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

For a randomized stationary policy, we have $P_{(s,a),(s',a')}^\pi = P(s'|s, a)\pi(a'|s')$. With this notation, it is straightforward to verify:

$$Q^\pi = (1 - \gamma)r + \gamma P V^\pi \quad (1.3)$$

$$Q^\pi = (1 - \gamma)r + \gamma P^\pi Q^\pi. \quad (1.4)$$

The above implies that:

$$Q^\pi = (1 - \gamma)(I - \gamma P^\pi)^{-1}r \quad (1.5)$$

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

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

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

1.1.4 Bellman optimality equations

Due to the Markov structure, there exists a single stationary and deterministic policy that simultaneously maximizes $V^\pi(s)$ for all $s \in \mathcal{S}$ and maximizes $Q^\pi(s, a)$ for all $s \in \mathcal{S}, a \in \mathcal{A}$ [Puterman, 1994]; we denote this *optimal policy* as π_M^* (or π^*). We use V^* and Q^* as a shorthand for V^{π^*} and Q^{π^*} , respectively.

V^* and Q^* satisfy the following set of *Bellman optimality equations* [Bellman, 1956]: for all $s \in \mathcal{S}, a \in \mathcal{A}$,

$$\begin{aligned} V^*(s) &= \max_{a \in \mathcal{A}} Q^*(s, a). \\ Q^*(s, a) &= (1 - \gamma)r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s, a)} [V^*(s')]. \end{aligned} \quad (1.6)$$

Let us use shorthand π_Q to denote the greedy policy with respect to a vector $Q \in \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|}$, i.e

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

With this notation, the optimal policy π^* is obtained by choosing actions greedily (with arbitrary tie-breaking mechanisms) with respect to Q , i.e.

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

Let us also use the notation to greedily 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* $\mathcal{T}_M : \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|} \rightarrow \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|}$ is defined as follows: when applied to some vector $Q \in \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|}$,

$$\mathcal{T}Q := (1 - \gamma)r + \gamma PV_Q. \quad (1.7)$$

This allows us to rewrite Equation 1.6 in the concise form: $Q^* = \mathcal{T}Q^*$, i.e. Q^* is a fixed point of the operator \mathcal{T} . The classic result of [Bellman, 1956] shows that if Q satisfies $Q = \mathcal{T}Q$, then $Q = Q^*$. We state the result below formally.

Theorem 1.4. *Let $Q^*(s, a) = \max_{\pi \in \Pi} Q^\pi(s, a)$ where Π is the space of all (non-stationary and randomized) policies. We have that*

- *There exists a stationary and deterministic policy π such that $Q^\pi = Q^*$*
- *A vector $Q \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ is equal to Q^* if and only if it satisfies $Q = \mathcal{T}Q$.*

Proof: First observe that:

$$\begin{aligned} Q^*(s, a) &= (1 - \gamma)\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a\right] \\ &= (1 - \gamma)\mathbb{E}\left[r(s_0, a_0) + \dots r(s_{\tau-1}, a_{\tau-1}) + \gamma^\tau E\left[\sum_{t=\tau}^{\infty} \gamma^t r(s_{t+\tau}, a_{t+\tau}) \mid s_\tau = s, a_\tau = a\right] \mid s_0 = s, a_0 = a\right] \\ &= (1 - \gamma)\mathbb{E}\left[r(s_0, a_0) + \dots r(s_{\tau-1}, a_{\tau-1}) + \gamma^\tau \max_{\pi} \left(E\left[\sum_{t=0}^{\infty} \gamma^t r(s_{t+\tau}, a_{t+\tau}) \mid \pi, s_\tau = s, a_\tau = a\right]\right) \mid s_0 = s, a_0 = a\right]. \end{aligned}$$

where the inner max is over all policies which may also use the history of information before time τ . Note that s_τ and a_τ the future evolution at time τ does not depend on the $(s_0, a_0, \dots, s_{\tau-1}, a_{\tau-1})$, which implies that the max value can be achieved with a policy that, at time τ , chooses an action that only depends on s_τ . This proves the stationarity claim. Furthermore, by linearity of expectation, the choice of a_τ can be made deterministically.

For the second claim, we first show that Q^* satisfies $Q^* = \mathcal{T}Q^*$. We need only consider deterministic policies. We have:

$$\begin{aligned} Q^*(s, a) &= \max_{\pi} Q^\pi(s, a) = \max_{\pi} \{(1 - \gamma)r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)}[V^\pi(s')]\} \\ &= (1 - \gamma)r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)}[\max_{\pi} V^\pi(s')] \\ &= (1 - \gamma)r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)}[\max_{\pi} Q^\pi(s', \pi(s'))] \\ &= (1 - \gamma)r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)}[\max_{\pi, a'} Q^\pi(s', a')] \\ &= (1 - \gamma)r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)}[\max_{a'} Q^*(s', a')]. \end{aligned}$$

Thus Q^* satisfies the Bellman optimality equations.

For the converse, suppose $Q = \mathcal{T}Q$ for some Q . For $\pi = \pi_Q$, this implies that $Q = (1 - \gamma)r + \gamma P^{\pi_Q}Q$, and so $Q = Q^\pi$, i.e. Q is the action value of the policy π_Q . Now observe for any other policy π' :

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

The proof is completed by noting that $(P^{\pi'} - P^{\pi})Q^{\pi} \leq 0$. 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 Planning in MDPs

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

1.2.1 Q -Value Iteration

A simple algorithm is to iteratively applying the fixed point mapping: starting at some Q , we iteratively apply \mathcal{T} :

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

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

Lemma 1.5. (*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 , $|V_Q(s) - V_{Q'}(s)| \leq \max_{a \in \mathcal{A}} |Q(s, a) - Q'(s, a)|$. 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 suboptimality of the greedy policy itself, based on the error in Q -value function.

Lemma 1.6. [Singh and Yee [1994]] For any vector $Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$,

$$V^{\pi_Q} \geq V^* - \frac{2\|Q - Q^*\|_{\infty}}{1 - \gamma} \mathbf{1}.$$

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

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

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

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

Theorem 1.7. (*Q-value iteration convergence*). Set $Q^{(0)} = 0$. For $k = 0, 1, \dots$, suppose:

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

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

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

Proof: Since $\|Q^*\|_\infty \leq 1$, $Q^{(k)} = \mathcal{T}^k Q^{(0)}$ and $Q^* = \mathcal{T}Q^*$, Lemma 1.5 gives

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

The proof is completed with our choice of γ and using Lemma 1.6. ■

1.2.2 Policy Iteration

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

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 1.5, 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 1.8. *We have that:*

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

Proof: We start with the first part. 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,

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

Using this,

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

which proves the first claim.

For the second claim,

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

where we have used that $Q^* \geq Q^{\pi_{k+1}} \geq Q^{\pi_k}$ in second step and the contraction property of $\mathcal{T}(\cdot)$ (see Lemma 1.5 in the last step. \blacksquare)

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

Theorem 1.9. (*policy iteration convergence*). *Let π_0 be any initial policy. For $k \geq \frac{\log \frac{1}{\epsilon}}{1-\gamma}$, the k -th policy in policy iteration has the following performance bound:*

$$Q^{\pi^{(k)}} \geq Q^* - \epsilon.$$

Chapter 2

Sample Complexity with a Generative Model

Sample Complexity with a Generative Model

Alekh Agarwal, Nan Jiang, Sham M. Kakade

Chapter 2

2.1 The Generative Model Setting

We now characterize the optimal minimax sample complexity of estimating Q^* .

We 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 .

For this, we assume we have access to a *generative model*, which can provide us with a sample $s' \sim P(\cdot|s, a)$ upon input of any state action pair. Suppose we call our simulator N times at each state action pair. Let \hat{P} be our empirical model, defined as follows:

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

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

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

2.2 Sample Complexity

2.2.1 A naive approach: accurate model estimation

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 \hat{P} for planning.

Theorem 2.1. *Let $\epsilon \geq 0$. Suppose we obtain*

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

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

- The transition model has error bounded as:

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

- For all policies π ,

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

- The estimated \hat{Q}^* has error bounded as:

$$\|Q^* - \hat{Q}^*\|_\infty \leq \epsilon$$

2.2.2 A more refined approach: using a sparse model

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 \hat{Q}^* of Q^* , which we might hope would require less samples.

Let us start with a crude bound on the optimal action-values, which shows that an improvement is possible.

Lemma 2.2. (*Crude Value Bounds*) Let $\delta \geq 0$. With probability greater than $1 - \delta$,

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

where:

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

Proof: We have:

$$\begin{aligned}\|Q^* - \hat{Q}^{\pi^*}\|_\infty &= \gamma \|P^{\pi^*} Q^* - \hat{P}^{\pi^*} \hat{Q}^{\pi^*}\|_\infty \\ &\leq \gamma \|P^{\pi^*} Q^* - \hat{P}^{\pi^*} Q^*\|_\infty + \gamma \|\hat{P}^{\pi^*} Q^* - \hat{P}^{\pi^*} \hat{Q}^{\pi^*}\|_\infty \\ &= \gamma \|PV^* - \hat{P}V^*\|_\infty + \gamma \|\hat{P}^{\pi^*} (Q^* - \hat{Q}^{\pi^*})\|_\infty \\ &\leq \gamma \|(P - \hat{P})V^*\|_\infty + \gamma \|Q^* - \hat{Q}^{\pi^*}\|_\infty,\end{aligned}$$

and so we have shown that:

$$\|Q^* - \hat{Q}^{\pi^*}\|_\infty \leq \frac{\gamma}{1-\gamma} \|(P - \hat{P})V^*\|_\infty$$

By applying Hoeffding's inequality and the union bound,

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

which holds with probability greater than $1 - \delta$. This completes the proof of the first claim. The proof of the second claim is analogous. \blacksquare

The main result in this chapter will be to improve the bound on \hat{Q}^* to be optimal:

Theorem 2.3. For $\delta \geq 0$ and with probability greater than $1 - \delta$,

$$\|Q^* - \hat{Q}^*\|_\infty \leq \gamma \sqrt{\frac{c}{1-\gamma} \frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}} + \frac{c\gamma}{(1-\gamma)^2} \frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N},$$

where c is an absolute constant.

Corollary 2.4. Let $0 \leq \epsilon \leq \frac{1}{1-\gamma}$. Suppose we obtain

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

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

$$\|Q^* - \hat{Q}^*\|_\infty \leq \epsilon$$

2.2.3 Lower Bounds

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

Theorem 2.5. *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*

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

2.2.4 What about the Value of the Policy $\widehat{\pi}^*$?

Ultimately, we are interested in the value $V^{\widehat{\pi}^*}$ when we execute $\widehat{\pi}^*$, not just an estimate \widehat{Q}^* of Q^* . The following is an immediate corollary by Lemma 1.6.

Corollary 2.6. *For $\delta \geq 0$ and with probability greater than $1 - \delta$,*

$$V^{\widehat{\pi}^*} \geq V^* - \gamma \sqrt{\frac{c}{(1 - \gamma)^3} \frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}} - \frac{c\gamma}{(1 - \gamma)^3} \frac{\log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N},$$

where c is an absolute constant.

This bound is not sharp. With a more careful argument, the $(1 - \gamma)^3$ can be reduced to only $1 - \gamma$.

2.3 Analysis

Lemma 2.7. *(Component-wise Bounds) We have that:*

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

Proof: Due to the optimality of π^* in M ,

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

which proves the first claim. The second claim is left as an exercise to the reader. ■

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

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

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

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

Equivalently,

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

Lemma 2.8. *Let $\delta \geq 0$. With probability greater than $1 - \delta$,*

$$|(P - \hat{P})V^*| \leq \sqrt{\frac{2 \log(2|\mathcal{S}||\mathcal{A}|/\delta)}{N}} \sqrt{\text{Var}_P(V^*)} + \frac{2 \log(2|\mathcal{S}||\mathcal{A}|/\delta)}{3N} \mathbb{1}.$$

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 $\|(I - \gamma \hat{P}^{\pi^*})^{-1} \sqrt{\text{Var}_P(V^*)}\|_\infty$ and $\|(I - \gamma \hat{P}^{\pi^*})^{-1} \sqrt{\text{Var}_P(V^*)}\|_\infty$.

It is helpful to define Σ_M^π as the variance of the discounted reward, i.e.

$$\Sigma_M^\pi(s, a) := \mathbb{E} \left[\left((1 - \gamma) \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) - Q_M^\pi(s, a) \right)^2 \middle| s_0 = s, a_0 = a \right]$$

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

The following lemma shows that Σ_M^π satisfies a Bellman consistency condition.

Lemma 2.9. *(Bellman consistency of Σ) For any MDP M ,*

$$\Sigma_M^\pi = \gamma^2 \text{Var}_P(V_M^\pi) + \gamma^2 P^\pi \Sigma_M^\pi \quad (2.1)$$

where P is the transition model in MDP M .

The proof is left as an exercise to the reader.

Lemma 2.10. *For any policy π and MDP M ,*

$$\|(I - \gamma P^\pi)^{-1} \sqrt{\text{Var}_P(V_M^\pi)}\|_\infty \leq \sqrt{\frac{2}{1 - \gamma}}$$

where P is the transition model in 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 $(I - \gamma P^\pi)^{-1}v \leq 2(I - \gamma^2 P^\pi)^{-1}v$ for $v \geq 0$ (which we will prove shortly). The proof is completed as follows: by Equation 2.1, $\Sigma_M^\pi = \gamma^2(I - \gamma^2 P^\pi)^{-1}\text{Var}_P(V_M^\pi)$, so taking $v = \text{Var}_P(V_M^\pi) \geq 0$ and using that $\|\Sigma_M^\pi\|_\infty \leq \gamma^2$ completes the proof.

Finally, to see for $v \geq 0$ that $(I - \gamma P^\pi)^{-1}v \leq 2(I - \gamma^2 P^\pi)^{-1}v$, observe:

$$\begin{aligned} 2(I - \gamma^2 P^\pi)^{-1}v - (I - \gamma P^\pi)^{-1}v &\geq (1 + \gamma)(I - \gamma^2 P^\pi)^{-1}v - (I - \gamma P^\pi)^{-1}v \\ &= (I - \gamma^2 P^\pi)^{-1}((1 + \gamma)(I - \gamma P^\pi) - (I - \gamma^2 P^\pi))(I - \gamma P^\pi)^{-1}v \\ &= (I - \gamma^2 P^\pi)^{-1}(I - \gamma P^\pi)(I - \gamma P^\pi)^{-1}v \\ &= (I - \gamma^2 P^\pi)^{-1}v \geq 0 \end{aligned}$$

using that $v \geq 0$ and $(I - \gamma^2 P^\pi)^{-1}$ has positive entries. This proves the claim. \blacksquare

Lemma 2.11. *Let $\delta \geq 0$. With probability greater than $1 - \delta$, we have:*

$$\begin{aligned} \text{Var}_P(V^*) &\leq 2\text{Var}_{\hat{P}}(\hat{V}^{\pi^*}) + \Delta'_{\delta,N}\mathbb{1} \\ \text{Var}_P(V^*) &\leq 2\text{Var}_{\hat{P}}(\hat{V}^*) + \Delta'_{\delta,N}\mathbb{1} \end{aligned}$$

where

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

Proof: By definition,

$$\begin{aligned} \text{Var}_P(V^*) &= \text{Var}_P(V^*) - \text{Var}_{\hat{P}}(V^*) + \text{Var}_{\hat{P}}(V^*) \\ &= P(V^*)^2 - (PV^*)^2 - \hat{P}(V^*)^2 + (\hat{P}V^*)^2 + \text{Var}_{\hat{P}}(V^*) \\ &= (P - \hat{P})(V^*)^2 - ((PV^*)^2 - (\hat{P}V^*)^2) + \text{Var}_{\hat{P}}(V^*) \end{aligned}$$

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

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

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

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

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

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

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 above argument also shows $\text{Var}_{\hat{P}}(V^*) \leq 2\Delta_{\delta,N}^2 + 2\text{Var}_{\hat{P}}(\hat{V}^*)$ which proves the second claim. \blacksquare

Using Lemma 2.8 and 2.11, we have the following corollary.

Corollary 2.12. *Let $\delta \geq 0$. With probability greater than $1 - \delta$, we have:*

$$\begin{aligned} |(P - \hat{P})V^*| &\leq c\sqrt{\frac{\text{Var}_{\hat{P}}(\hat{V}^{\pi^*}) \log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}} + \Delta''_{\delta,N} \mathbb{1} \\ |(P - \hat{P})V^*| &\leq c\sqrt{\frac{\text{Var}_{\hat{P}}(\hat{V}^*) \log(c|\mathcal{S}||\mathcal{A}|/\delta)}{N}} + \Delta''_{\delta,N} \mathbb{1}, \end{aligned}$$

where

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

and where c is an absolute constant.

2.3.1 Completing the proof

Proof:(of Theorem 2.3) The proof consists of bounding the terms in Lemma 2.7. We have:

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

where the first step uses Corollary 2.12; the second uses Lemma 2.10; 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. \blacksquare

2.4 Bibliographic Remarks

The results in this section are originally due to the work in [Azar et al., 2013].

Chapter 3

Strategic Exploration in RL

Strategic Exploration in Reinforcement Learning

Aleksh Agarwal, Nan Jiang, Sham M. Kakade

Chapter 3

In this lecture we will see how an agent acting in an MDP can learn a near-optimal behavior policy over time. Compared with the setting of the previous lecture on a generative model, we no longer have easy access to transitions at each state, but only have the ability to execute trajectories in the MDP. The main complexity this adds to the learning process is that the agent has to engage in exploration, that is, plan to reach new states where enough samples have not been seen yet, so that optimal behavior in those states can be learned.

The content of this chapter will be based on Brafman and Tenenbholz [2003]. In particular, we will present a version of the R-MAX algorithm, but adapted to the discounted case which we will denote as R-MAX- γ . The pseudocode of the algorithm is given in Algorithm 1. It relies on the idea of *optimism in the face of uncertainty*, which is common to several exploration algorithms in reinforcement learning. In a nutshell, we presume that every unknown alternative will lead to a high reward, unless we learn otherwise.

Algorithm 1 R-MAX- γ algorithm for sample efficient reinforcement learning in discounted MDPs

Input: Parameter m to set known states. Accuracy parameter $\epsilon > 0$.

```

1: Initialize the set of known states  $K = \emptyset$ , counters  $n(s, a) = n(s, a, s') = 0$  for all  $s, a, s' \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$  and  $R(s, a) = 0$ .
2: Observe initial state  $s_0$  and let  $\pi_0$  be an arbitrary initial policy.
3: for all rounds  $t = 0, 1, 2, \dots$  do
4:   if a state has become known, i.e.  $n(s, a) \geq m$  for all  $a \in \mathcal{A}$  then
5:     update  $K = K \cup \{s\}$ .
6:     Let  $\widehat{M}$  have  $\widehat{P}(s'|s, a) = n(s, a, s')/n(s, a)$  and  $\widehat{r}(s, a) = R(s, a)/n(s, a)$ .
7:     Let  $\widehat{M}_K$  be the induced MDP (see Definition 3.1) and  $\pi_t = \pi^*(\widehat{M}_K)$  be the optimal policy in  $\widehat{M}_K$ .
8:   else
9:     If  $t \geq 1$ ,  $\pi_t = \pi_{t-1}$ .
10:  end if
11:  If  $s_t \in K$ , choose  $a_t = \pi_t(s_t)$ , else  $a_t = \arg \min_{a \in \mathcal{A}} n(s, a)$ .
12:  Receive reward  $r_t$  and observe next state  $s_{t+1}$ .
13:  if  $s_t \notin K$  then
14:    Update  $n(s_t, a_t) + 1$ ,  $R(s_t, a_t) + r_t$  and  $n(s_t, a_t, s_{t+1}) + 1$ .
15:  end if
16: end for
```

Concretely, the algorithm maintains an estimate of the transition probabilities $P(s'|s, a)$ for all the neighbors s' of a state s , given an action a . It also estimates the reward $r(s, a)$. Once the algorithm has visited s adequately often to ensure that these estimates are all accurate, it declares the state as *known*. Learning is complete when all the states are known. During a run of the algorithm, whenever it is in a known state, it already knows the optimal action to take and follows this action. However, when the algorithm is in an unknown state, it explores by picking the action chosen least often in the state so far.

While we will mostly focus on the statistical properties of the algorithm, the computational aspects are relatively straightforward. Within an episode, the main computational burden is in the computation of an optimal policy for the induced MDP M_K in line 7. This can be done, for example, using the value iteration algorithm from Chapter 1.2, as the MDP and reward function is fully known in this step. Since reasoning over the infinite horizon can be tricky computationally, a common trick is to restrict the step 7 in Algorithm 1 to computing a non-stationary H -step optimal policy instead, where $H = O\left(\frac{\log \frac{1}{\epsilon}}{1-\gamma}\right)$ is the *effective horizon*. That is, we find the policy which maximizes the

expected discounted reward over just H time steps. Such a policy can be easily computed via dynamic programming, but is required to be non-stationary. That is, it might choose different actions for the same state visited at different values of t . Our choice of H ensures that the infinite-horizon value functions and H -step value functions are at most ϵ apart, so that none of the subsequent theory is affected.

In order to more concretely discuss the algorithm, we need some important definitions. Given an MDP M and a set K of known states, we next define the notion of an *Induced MDP*.

Definition 3.1 (Induced MDP). Let M be an MDP parametrized by $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with $K \in \mathcal{S}$ being a subset of states. Based on this set, we define the *induced MDP* M_K parametrized by $(\mathcal{S}, \mathcal{A}, P_{M_K}, r_{M_K}, \gamma)$ in the following manner. For each $s \in K$, we define

$$P_{M_K}(s'|s, a) = P_M(s'|s, a) \quad \text{and} \quad r_{M_K}(s, a) = r_M(s, a).$$

For all the $s \notin K$, we define

$$P_{M_K}(s'|s, a) = \mathbf{1}(s' = s) \quad \text{and} \quad r_{M_K}(z|s, a) = \mathbf{1}(z = 1).$$

Thus, an induced MDP given a set K of known states is an optimistic process where we receive a reward of 1 (recall that the rewards $r_t \in [0, 1]$ so that 1 is the largest attainable reward) no matter which action we try in an unknown state. Furthermore, once we enter such an unknown state, we stay there and keep collecting this reward for the remainder of the episode. On the known states, naturally the transition and reward distributions follow their known behavior.

We will now analyze the R-MAX- γ algorithm, and provide a bound on the number of episodes before which it finds an ϵ -optimal policy. For the analysis, it will be useful to invoke the notation

$$H = \frac{\log(2/\epsilon(1-\gamma))}{1-\gamma}.$$

We will prove the following theorem.

Theorem 3.2. Let s_t be the state visited by the R-MAX- γ algorithm at round t and let the parameter m for deciding known states be set as $m = \mathcal{O}\left(\frac{SH^2}{\epsilon^2} \log \frac{S^2 A}{\delta}\right)$. For any $0 \leq \epsilon, \delta < 1$, with probability at least $1 - \delta$, $V_M^{\pi_t}(s_t) \geq V_M^*(s_t) - \epsilon$, for all but $\mathcal{O}\left(\frac{H^3 S^2 A}{\epsilon^3} \log \frac{S^2 A}{\delta}\right)$ rounds in the MDP.

In words, the algorithm finds policies such that those policies induce near optimal value functions for all but a bounded number of rounds. Note that this does not imply that the algorithm behaves sub-optimally for the first $\mathcal{O}\left(\frac{H^3 S^2 A}{\epsilon^3} \log \frac{S^2 A}{\delta}\right)$ rounds only. The dynamics of the MDP might be such that an unknown state is encountered with a small chance only, in which case the algorithm learns whenever it encounters these states. The guarantee also does not preclude settings where after some initial exploration, the algorithm encounters a state with a small value from which escape is not possible under the dynamics. In such a case, the guarantee of the algorithm trivially holds as any policy is optimal in that state.

An alternative optimality condition we might desire from the algorithm is that it finds a near-optimal policy, that is a policy whose expected reward is within ϵ of the optimal, when taking expectations over the start state as well. This is not the guarantee provided here, and in general requires some mixing conditions on the MDP which we do not consider here. A common way to ensure such mixing in practice is by assuming the ability to reset to the initial state distribution during the training of an agent. When such a reset ability is available, the guarantee provided here can be further strengthened into approximate optimality of the policy.

In order to prove the result, we will introduce a number of key concepts in understanding exploration in reinforcement learning. Throughout the analysis, we will abuse our notation r to also refer to the expected reward, given a (state, action) pair.

We start with a basic result which was first introduced in Kearns and Singh [2002] under the name of a simulation lemma.

Lemma 3.3 (Simulation lemma for MDPs). *Let M and M' be two MDPs with the same state and action spaces. If the transition and reward functions of these MDPs satisfy*

$$\sum_{s' \in \mathcal{S}} |P_M(s'|s, a) - P_{M'}(s'|s, a)| \leq \epsilon_1, \quad \forall s \in \mathcal{S} \text{ and } a \in \mathcal{A}, \quad |r_M(s, a) - r_{M'}(s, a)| \leq \epsilon_2 \quad \forall s \in \mathcal{S} \text{ and } a \in \mathcal{A}.$$

Then for every stationary policy π , the two MDPs satisfy $\|V_M^\pi - V_{M'}^\pi\|_\infty \leq \frac{\gamma}{1-\gamma}\epsilon_1 + \epsilon_2$.

The lemma is called a simulation lemma as it tells how much error we incur in evaluating policies if we build an approximate simulator M' for the true process M .

Proof: The lemma is proved using the conditions on the transition and reward distributions, along with the Bellman equations for value functions (1.2). For any state s , we have

$$\begin{aligned} |V_M^\pi(s) - V_{M'}^\pi(s)| &\leq (1-\gamma)\epsilon_2 + \gamma \left| \sum_{s' \in \mathcal{S}} (P_M(s'|s, \pi(s))V_M^\pi(s') - P_{M'}(s'|s, \pi(s))V_{M'}^\pi(s')) \right| \\ &\leq (1-\gamma)\epsilon_2 + \gamma \left| \sum_{s' \in \mathcal{S}} P_M(s'|s, a)(V_M^\pi(s') - V_{M'}^\pi(s')) \right| + \gamma \left| \sum_{s' \in \mathcal{S}} V_{M'}^\pi(s')(P_M(s'|s, a) - P_{M'}(s'|s, a)) \right| \\ &\leq (1-\gamma)\epsilon_2 + \gamma\|V_M^\pi - V_{M'}^\pi\|_\infty + \gamma\epsilon_1. \end{aligned}$$

Note that here we have used the normalization of value functions, that is $0 \leq V_M^\pi(s) \leq 1$. Since the inequality holds for any state, we can take the max on the LHS and rearrange terms to complete the proof. ■

The next lemma really formalizes our intuition that the optimal policy in the induced MDP encourages exploration of the currently unknown states. We will show that either the best policy π_i learned using the induced MDP at an episode i is already good, or it has a high chance of taking us to an unknown state. We will use the notation

$$\mathbb{P}_M^\pi[\text{escape from } K | s_0 = s] := \mathbf{1}(s \notin K) + \sum_{t=1}^{\infty} \gamma^t \mathbb{P}_M^\pi(s_t \notin K, s_0, \dots, s_{t-1} \in K).$$

That is, $\mathbb{P}_M^\pi[\text{escape from } K | s_0 = s]$ is the discounted probability of reaching an unknown state when executing π in the original MDP M , starting from the state s .

Lemma 3.4 (Induced inequalities). *Let M be an MDP with K being the set of known states. Let M_K be the induced MDP (Definition 3.1) with respect to K and M . For any stationary policy π and state $s \in \mathcal{S}$ we have*

$$V_{M_K}^\pi(s) \geq V_M^\pi(s) \quad \text{and} \quad V_M^\pi(s) \geq V_{M_K}^\pi(s) - \mathbb{P}_M^\pi[\text{escape from } K | s_0 = s].$$

The lemma has two implications. First it formalizes the notion that the induced MDP M_K is indeed an optimistic version of M , since it ascribes higher values to each state under *every* policy. At the same time, the optimism is not uncontrolled. The values ascribed by M_K to a policy π are higher only if the policy has a substantial probability of visiting an unknown state, and hence is useful for exploration.

Proof: The first inequality is a direct consequence of the definition of M_K . If $s \notin K$, it is immediate since we get the maximum reward of 1 at each time-step, while never leaving this state. If $s \in K$, then our immediate reward is identical to that in M . At the next step, we either stay in K , or leave. If we leave then we will obtain the largest reward for the remaining time steps. If we stay, we obtain the same reward as that in M . Thus we never obtain a smaller reward in M_K by definition.

For the second part, we have

$$\begin{aligned}
|V_M^\pi(s) - V_{M_K}^\pi(s)| &\leq \mathbf{1}(s \notin K) + \mathbf{1}(s \in K) \gamma \left| \sum_{s' \in \mathcal{S}} P_M(s'|s, \pi(s)) V_M^\pi(s') - P_{M_K}(s'|s, \pi(s)) V_{M_K}^\pi(s') \right| \\
&= \mathbf{1}(s \notin K) + \mathbf{1}(s \in K) \gamma \left| \sum_{s' \in \mathcal{S}} P_M(s'|s, \pi(s)) (V_M^\pi(s') - V_{M_K}^\pi(s')) \right| \\
&\leq \mathbf{1}(s \notin K) + \mathbf{1}(s \in K) \gamma P_M(s' \notin K | s, \pi(s)) + \mathbf{1}(s \in K) \gamma \left| \sum_{s' \in K} P_M(s'|s, \pi(s)) (V_M^\pi(s') - V_{M_K}^\pi(s')) \right|.
\end{aligned}$$

Here the first inequality holds since the two value functions can differ by at most the maximum value of 1 if the starting state is unknown. The following equality holds as the transition models under M and M_K are identical when $s \in K$. Now unrolling the summation in the last inequality further yields the statement of the lemma. ■

Given the lemma, we have a particularly useful corollary. It says that the policy π_i computed in each episode of Algorithm 1 is near optimal, with the error being the probability of leaving the known state set.

Corollary 3.5 (Implicit Explore-Exploit).

$$V_M^{\pi^*(M_K)}(s) \geq V_M^*(s) - \mathbb{P}_M^{\pi^*(M_K)}[\text{escape from } K | s_0 = s]$$

Proof: By the lemma, we have

$$\begin{aligned}
V_M^{\pi^*(M_K)}(s) &\geq V_{M_K}^{\pi^*(M_K)}(s) - \mathbb{P}_M^{\pi^*(M_K)}[\text{escape from } K | s_0 = s] \\
&\geq V_{M_K}^{\pi^*(M)}(s) - \mathbb{P}_M^{\pi^*(M_K)}[\text{escape from } K | s_0 = s] \\
&\geq V_M^{\pi^*(M)}(s) - \mathbb{P}_M^{\pi^*(M_K)}[\text{escape from } K | s_0 = s].
\end{aligned}$$

Here the first inequality follows from Lemma 3.4 applied with $\pi = \pi^*(M_K)$, second inequality uses that $\pi^*(M_K)$ is the optimal policy in M_K and hence obtains a higher reward than $\pi^*(M)$ and the third inequality follows from the optimism of M_K shown in Lemma 3.4. ■

We conclude with a final lemma which relates the probability of escape from K over an infinite trajectory to that of encountering an unknown state over H steps.

Lemma 3.6. *With probability at least $1 - \delta$, the number of rounds where $V_M^{\pi_t}(s_t) \leq V_M^*(s_t) - \epsilon$ is at most $\mathcal{O}\left(\frac{mHSA}{\epsilon} \ln \frac{1}{\delta}\right)$.*

Proof: In this lemma, we with sufficiently many visits to states with a large escape probability, all the states become known. Furthermore, with high probability, the number of rounds following such a visit where our policy's value function is significantly suboptimal is at most H .

The proof of the lemma has two parts. First we show that if at round t , the probability of escape from K starting from s_t is large, then we also have a large probability of escape in the next H steps. Using Bernoulli concentration, we then further show that the algorithm will indeed encounter an unknown state in the next H steps with a large probability. For the first part, let $s = s_t$ be the state encountered at some round t and let $\pi = \pi_t$ be the current policy. Suppose further that we know $\mathbb{P}_M^\pi[\text{escape from } K | s_0 = s] \geq \epsilon$. Let us define

$$p_H = \mathbf{1}(S \notin K) + \sum_{t=1}^H \mathbb{P}_M^\pi(s_t \notin K | s_0, \dots, s_{t-1} \in K).$$

Note that there is no discounting in the definition of p_H . We have

$$\begin{aligned} \epsilon &\leq \mathbf{1}(s \notin K) + \sum_{t=1}^{\infty} \gamma^t \mathbb{P}_M^{\pi}(s_t \notin K \mid s_0, \dots, s_{t-1} \in K) \\ &\leq p_H + \sum_{t=H+1}^{\infty} \gamma^t \mathbb{P}_M^{\pi}(s_t \notin K \mid s_0, \dots, s_{t-1} \in K) \\ &\leq p_H + \frac{\gamma^{H+1}}{1-\gamma}. \end{aligned}$$

Thus we see that $p_H \geq \epsilon - \gamma^{H+1}/(1-\gamma)$. It suffices to ensure that the tail term is at most $\epsilon/2$ to guarantee that $p_H \geq \epsilon/2$, which means that we want $H \geq \log \frac{2}{\epsilon(1-\gamma)} / \log \frac{1}{\gamma}$. Noting that $\log(1/\gamma) \geq 1-\gamma$ for $\gamma \in (0, 1]$ completes a lower bound of $\epsilon/2$ on p_H for the stated value of H .

It remains to bound the number of actions before we have sufficiently many visits to unknown states. For this, let t_1, t_2, \dots be the rounds such that $|t_i - t_{i+1}| > H$ and if π_i is the policy used at round t_i and K_i is the set of known states at t_i , then $\mathbb{P}^{\pi_i}(\text{escape from } K \mid s_0 = s_{t_i}) \geq \epsilon$. Let us define a random variable

$$X_i = \mathbf{1}(\exists s \{s_{t_i}, s_{t_i+1}, \dots, s_{t_i+H}\} : s \notin K_i).$$

By the definition of the rounds t_i , we know that $\mathbb{E}[X_i | s_{t_i}] \geq \epsilon$. Let us define a σ -field \mathcal{F}_i to consist of all the randomness in the MDP and the agent prior to t_i (this includes the randomness in the state transitions at round $t_{i+1} - 1$, so that the state $s_{t_{i+1}}$, the policy π_{i+1} and the known states K_{i+1} are known when conditioning on \mathcal{F}_i). It is easily checked that X_i is measurable with respect to \mathcal{F}_i , since the entire trajectory till round $t_{i+1} - 1$ is known under this conditioning, which in particular implies that we know the subtrajectory from rounds t_i to $t_i + H$ and hence can check the indicator in X_i . Also, we observe that $\mathbb{E}[X_i | \mathcal{F}_{i-1}] = \mathbb{P}^{\pi_i}(\text{escape from } K_i \text{ in } H \text{ steps} \mid s_{t_i}) \geq \epsilon/2$ using the earlier lower bound on p_H . Furthermore, since the X_i are binary valued, it is clear that

$$\mathbb{E}[(X_i - \mathbb{E}[X_i | \mathcal{F}_{i-1}])^2 \mid \mathcal{F}_{i-1}] \leq \mathbb{E}[X_i^2 \mid \mathcal{F}_{i-1}] \leq \mathbb{E}[X_i \mid \mathcal{F}_{i-1}].$$

Now applying the Freedman's inequality for martingales (Lemma A.3 in the appendix) to the sequence $Y_i = X_i - \mathbb{E}[X_i | \mathcal{F}_{i-1}]$, we see that with probability at least $1 - \delta$, for a fixed n we have

$$\begin{aligned} \sum_{i=1}^n (\mathbb{E}[X_i | \mathcal{F}_{i-1}] - X_i) &\leq 2 \sqrt{\ln \frac{1}{\delta} \sum_{i=1}^n \mathbb{E}[X_i | \mathcal{F}_{i-1}]} + \ln \frac{1}{\delta} \\ &\leq \frac{1}{2} \sum_{i=1}^n \mathbb{E}[X_i | \mathcal{F}_{i-1}] + 3 \ln \frac{1}{\delta}. \end{aligned}$$

This implies that

$$\sum_{i=1}^n X_i \geq \frac{1}{2} \sum_{i=1}^n \mathbb{E}[X_i | \mathcal{F}_{i-1}] - 3 \ln \frac{1}{\delta} \geq \frac{n\epsilon}{4} - 3 \ln \frac{1}{\delta}.$$

Since we desire $\sum_{i=1}^n X_i \geq mSA$, it suffices to pick $n \geq 4(mSA + 3 \ln(1/\delta))/\epsilon$.

Note that for the rounds $t \in [t_{i+H+1}, t_{i+1} - 1]$, we already know that the probability of escape is small, meaning that the value function of our policy is near optimal on those rounds. This concludes the proof. \blacksquare

Proof of Theorem 3.2

We now have most of the ingredients for the theorem. In order to prove the theorem, we need to ensure that m is large enough that when a state is declared known, then its transition and reward functions are reasonably accurate. For now, let us assume that m is chosen large enough so that the induced approximate MDP \widehat{M}_K is a good approximation to the true induced MDP M_K . Based on Lemma 3.3, we will assume that m is large enough so that the value functions of these two MDPs are at most $\epsilon/2$ different in any state. Then, applying this closeness twice, we see that

$$V_{M_K}^{\pi_t}(s) \geq V_{\widehat{M}_K}^{\pi_t}(s) - \frac{\epsilon}{2} \geq V_{\widehat{M}_K}^{\pi^*}(s) - \frac{\epsilon}{2} \geq V_{M_K}^{\pi^*}(s) - \epsilon.$$

Combining with Lemma 3.4, we see that for any starting state we have

$$\begin{aligned} V_M^{\pi_t}(s) &\geq V_{M_K}^{\pi^*}(s) - \epsilon - \mathbb{P}_M^{\pi_t}[\text{escape from } K | s_0 = s] \\ &\geq V_M^{\pi^*}(s) - \epsilon - \mathbb{P}_M^{\pi_t}[\text{escape from } K | s_0 = s], \end{aligned}$$

where the first inequality is combining Lemma 3.4 with the earlier bound, and the second inequality uses the optimism of the induced MDP M_K .

Thus, either the policy π_t is at most 2ϵ suboptimal, or it visits an unknown state with probability at least ϵ . Intuitively, this means that we visit an unknown state at least every $1/\epsilon$ steps, if $\widehat{\pi}_t$ is not already near optimal. Since the probabilities are discounted,

The total number of visits to unknown states are bounded by mSA . This is because for each unknown state s , we need to try every action a at least m times before s becomes known. Since we try the least frequently chosen action each time, it is ensured that each action is chosen *exactly* m times before s becomes known. Consequently, we need $O(mSA/\epsilon)$ episodes in order to ensure that every state is known and the algorithm can certifiably have a near optimal policy. In other words, with $O(mSAH/\epsilon)$ actions in the MDP, the agent is guaranteed to have marked all the states as known. This intuition is made precise in Lemma 3.6.

In order to obtain the theorem statement, we set $m = \mathcal{O}\left(\frac{SH^2}{\epsilon^2} \log \frac{S^2A}{\delta}\right)$. This number is based on satisfying the condition of Lemma 3.3 with $\gamma\epsilon_1/(1-\gamma) = \epsilon/2$, along with Lemma 8.5.6 of Kakade [2003]. Plugging this value of m in our bound on the number of samples as a function of m above completes the proof of the theorem.

Chapter 4

Policy Gradient Methods

Policy Gradient Methods

Aleksh Agarwal, Nan Jiang, Sham M. Kakade

Chapter 4

For a distribution ρ over states, define:

$$V_M^\pi(\rho) := \mathbb{E}_{s_0 \sim \rho}[V_M^\pi(s_0)],$$

where we slightly overload notation. Consider a class of parametric policies $\{\pi_\theta | \theta \in \Theta \subset \mathbb{R}^d\}$. The optimization problem of interest is:

$$\max_{\theta \in \Theta} V_M^{\pi_\theta}(\rho).$$

We drop the MDP subscript in this chapter.

In many settings, one of the most practically effective methods is gradient ascent. A “direct” approach often more easily handles large state and action spaces, and, in some cases, more readily handles settings where the model is not known. While we are interested in good performance under ρ , we will see how it is helpful to optimize under a different measure μ . Specifically, we consider optimizing $V^{\pi_\theta}(\mu)$ even though our ultimate goal is performance under $V^{\pi_\theta}(\rho)$.

One immediate issue is that if the policy class $\{\pi_\theta\}$ consists of deterministic policies then π_θ will, in general, not be differentiable. This motivates us to consider policy classes that are stochastic, which permit differentiability.

Example 4.1 (Softmax policies). It is instructive to explicitly consider a “tabular” policy representation, given by the *softmax policy*:

$$\pi_\theta(a|s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}, \quad (4.1)$$

where the parameter space is $\Theta = \mathbb{R}^{|S| \times |A|}$. Note that (the closure of) the set of softmax policies contains all stationary and deterministic policies.

Example 4.2 (Linear softmax policies). For any state, action pair s, a , suppose we have a feature mapping $\phi_{s,a} \in \mathbb{R}^d$. Let us consider the policy class

$$\pi_\theta(a|s) = \frac{\exp(\theta \cdot \phi_{s,a})}{\sum_{a' \in \mathcal{A}} \exp(\theta \cdot \phi_{s,a'})}$$

with $\theta \in \mathbb{R}^d$.

Example 4.3 (Neural softmax policies). Here we may be interested in working with the policy class

$$\pi_\theta(a|s) = \frac{\exp(f_\theta(s, a))}{\sum_{a' \in \mathcal{A}} \exp(f_\theta(s, a'))}$$

where the scalar function $f_\theta(s, a)$ may be parameterized by a neural network, with $\theta \in \mathbb{R}^d$.

Advantages and the state-action visitation distribution

Let us first introduce the concept of an advantage.

Definition 4.4. The advantage $A^\pi(s, a)$ of a policy π is defined as

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

Note that:

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

for all state-action pairs.

We now define the discounted state visitation distribution. Let τ denote a trajectory, whose unconditional distribution $\Pr_\mu^\pi(\tau)$ under π with starting distribution μ , is

$$\Pr_\mu^\pi(\tau) = \mu(s_0)\pi(a_0|s_0)P(s_1|s_0, a_0)\pi(a_1|s_1) \cdots \quad (4.2)$$

Our notation can be simplified through defining the discounted state visitation distribution $d_{s_0}^\pi$ as:

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

where $\Pr^\pi(s_t = s | s_0)$ is the state-action visitation probability, where π is used starting at state s_0 . We also write:

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

By construction, observe that for any function $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$,

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

The following lemma is a fundamental tool in the convergence analysis of directly policy search methods..

Lemma 4.5. (*The performance difference lemma*) For all policies π, π' and distributions μ ,

$$\begin{aligned} V^\pi(\mu) - V^{\pi'}(\mu) &= \mathbb{E}_{\tau \sim \Pr_\mu^\pi} \left[\sum_{t=0}^{\infty} \gamma^t A^{\pi'}(s_t, a_t) \right] \\ &= \frac{1}{1 - \gamma} \mathbb{E}_{s' \sim d_\mu^\pi} \mathbb{E}_{a' \sim \pi(\cdot | s')} [A^{\pi'}(s', a')] \end{aligned}$$

Proof: Fix a state s and let \Pr^π be the distribution over trajectories with $s_0 = s$. Using a telescoping argument, we have:

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

where (a) uses the tower property of conditional expectations. The proof is completed by linearity of expectation. ■

4.1 The Policy Gradient Method

It is convenient to define the discounted total reward of a trajectory as:

$$R(\tau) := (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$$

where s_t, a_t are the state-action pairs in τ . Observe that:

$$V^{\pi_\theta}(\mu) = \mathbb{E}_{\tau \sim \text{Pr}_\mu^{\pi_\theta}} [R(\tau)].$$

Theorem 4.6. (Policy gradients) The following are expressions for $\nabla_\theta V^{\pi_\theta}(\mu)$:

- *REINFORCE*:

$$\nabla V^{\pi_\theta}(\mu) = \mathbb{E}_{\tau \sim \text{Pr}_\mu^{\pi_\theta}} \left[R(\tau) \sum_{t=0}^{\infty} \nabla \log \pi_\theta(a_t | s_t) \right]$$

- *Action value expression*:

$$\begin{aligned} \nabla V^{\pi_\theta}(\mu) &= \mathbb{E}_{\tau \sim \text{Pr}_\mu^{\pi_\theta}} \left[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_\theta}(s_t, a_t) \nabla \log \pi_\theta(a_t | s_t) \right] \\ &= \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot | s)} \left[Q^{\pi_\theta}(s, a) \nabla \log \pi_\theta(a | s) \right] \end{aligned}$$

- *Advantage expression*:

$$\nabla V^{\pi_\theta}(\mu) = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot | s)} \left[A^{\pi_\theta}(s, a) \nabla \log \pi_\theta(a | s) \right]$$

The alternative expressions are more helpful to use when we turn to Monte Carlo estimation.

Proof: We have:

$$\begin{aligned} \nabla V^{\pi_\theta}(\mu) &= \nabla \sum_{\tau} R(\tau) \text{Pr}_\mu^{\pi_\theta}(\tau) \\ &= \sum_{\tau} R(\tau) \nabla \text{Pr}_\mu^{\pi_\theta}(\tau) \\ &= \nabla \sum_{\tau} R(\tau) \text{Pr}_\mu^{\pi_\theta}(\tau) \nabla \log \text{Pr}_\mu^{\pi_\theta}(\tau) \\ &= \sum_{\tau} R(\tau) \text{Pr}_\mu^{\pi_\theta}(\tau) \nabla \log (\mu(s_0) \pi_\theta(a_0 | s_0) P(s_1 | s_0, a_0) \pi_\theta(a_1 | s_1) \cdots) \\ &= \sum_{\tau} R(\tau) \text{Pr}_\mu^{\pi_\theta}(\tau) \left(\sum_{t=0}^{\infty} \nabla \log \pi_\theta(a_t | s_t) \right) \end{aligned}$$

which completes the proof of the first claim.

For the second claim, for any state s_0

$$\begin{aligned}
& \nabla V^{\pi_\theta}(s_0) \\
&= \nabla \sum_{a_0} \pi_\theta(a_0|s_0) Q^{\pi_\theta}(s_0, a_0) \\
&= \sum_{a_0} \left(\nabla \pi_\theta(a_0|s_0) \right) Q^{\pi_\theta}(s_0, a_0) + \sum_{a_0} \pi_\theta(a_0|s_0) \nabla Q^{\pi_\theta}(s_0, a_0) \\
&= \sum_{a_0} \pi_\theta(a_0|s_0) \left(\nabla \log \pi_\theta(a_0|s_0) \right) Q^{\pi_\theta}(s_0, a_0) \\
&\quad + \sum_{a_0} \pi_\theta(a_0|s_0) \nabla \left((1-\gamma)r(s_0, a_0) + \gamma \sum_{s_1} P(s_1|s_0, a_0) V^{\pi_\theta}(s_1) \right) \\
&= \sum_{a_0} \pi_\theta(a_0|s_0) \left(\nabla \log \pi_\theta(a_0|s_0) \right) Q^{\pi_\theta}(s_0, a_0) + \gamma \sum_{a_0, s_1} \pi_\theta(a_0|s_0) P(s_1|s_0, a_0) \nabla V^{\pi_\theta}(s_1) \\
&= \mathbb{E}_{\tau \sim \text{Pr}_{s_0}^{\pi_\theta}} [Q^{\pi_\theta}(s_0, a_0) \nabla \log \pi_\theta(a_0|s_0)] + \gamma \mathbb{E}_{\tau \sim \text{Pr}_{s_0}^{\pi_\theta}} [\nabla V^{\pi_\theta}(s_1)].
\end{aligned}$$

By linearity of expectation,

$$\begin{aligned}
& \nabla V^{\pi_\theta}(\mu) \\
&= \mathbb{E}_{\tau \sim \text{Pr}_\mu^{\pi_\theta}} [Q^{\pi_\theta}(s_0, a_0) \nabla \log \pi_\theta(a_0|s_0)] + \gamma \mathbb{E}_{\tau \sim \text{Pr}_\mu^{\pi_\theta}} [\nabla V^{\pi_\theta}(s_1)] \\
&= \mathbb{E}_{\tau \sim \text{Pr}_\mu^{\pi_\theta}} [Q^{\pi_\theta}(s_0, a_0) \nabla \log \pi_\theta(a_0|s_0)] + \gamma \mathbb{E}_{\tau \sim \text{Pr}_\mu^{\pi_\theta}} [Q^{\pi_\theta}(s_1, a_1) \nabla \log \pi_\theta(a_1|s_1)] + \dots
\end{aligned}$$

where the last step follows from recursion. This completes the proof of the second claim.

The proof of the final claim is left as an exercise to the reader. ■

4.1.1 Optimization

Gradient ascent and convergence to stationary points

Let us say a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is β -smooth if

$$\|\nabla f(w) - \nabla f(w')\| \leq \beta \|w - w'\|, \quad (4.3)$$

where the norm $\|\cdot\|$ is the Euclidean norm. In other words, the derivatives of f do not change too quickly.

Gradient ascent, with a fixed stepsize η , follows the update rule:

$$\theta_{t+1} = \theta_t + \eta \nabla V^{\pi_{\theta_t}}(\mu).$$

It is convenient to use the shorthand notation:

$$\pi^{(t)} := \pi_{\theta_t}, \quad V^{(t)} := V^{\pi_{\theta_t}}$$

The next lemma is standard in non-convex optimization.

Lemma 4.7. (Convergence to Stationary Points) Assume that for all $\theta \in \Theta$, V^{π_θ} is β -smooth and bounded below by V_* . Suppose we use the constant stepsize $\eta = 1/\beta$. For all T , we have that

$$\min_{t \leq T} \|\nabla V^{(t)}(\mu)\|^2 \leq \frac{2\beta(V^*(\mu) - V^{(0)}(\mu))}{T}.$$

Monte Carlo estimation and stochastic gradient ascent

One difficulty is that even if we know the MDP M , computing the gradient may be computationally intensive. It turns out that we can obtain unbiased estimates of π with only simulation based access to our model, i.e. assuming we can obtain sampled trajectories $\tau \sim \Pr_{\mu}^{\pi_{\theta}}$.

With respect to a trajectory τ , define:

$$\begin{aligned}\widehat{Q}^{\pi_{\theta}}(s_t, a_t) &:= (1 - \gamma) \sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'}) \\ \widehat{\nabla V}^{\pi_{\theta}}(\mu) &:= \sum_{t=0}^{\infty} \gamma^t \widehat{Q}^{\pi_{\theta}}(s_t, a_t) \nabla \log \pi_{\theta}(a_t | s_t)\end{aligned}$$

We now show this provides an unbiased estimated of the gradient:

Lemma 4.8. (*Unbiased gradient estimate*) We have :

$$\mathbb{E}_{\tau \sim \Pr_{\mu}^{\pi_{\theta}}} [\widehat{\nabla V}^{\pi_{\theta}}(\mu)] = \nabla V^{\pi_{\theta}}(\mu)$$

Proof: Observe:

$$\begin{aligned}\mathbb{E}[\widehat{\nabla V}^{\pi_{\theta}}(\mu)] &= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \widehat{Q}^{\pi_{\theta}}(s_t, a_t) \nabla \log \pi_{\theta}(a_t | s_t) \right] \\ &\stackrel{(a)}{=} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{E}[\widehat{Q}^{\pi_{\theta}}(s_t, a_t) | s_t, a_t] \nabla \log \pi_{\theta}(a_t | s_t) \right] \\ &\stackrel{(b)}{=} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_{\theta}}(s_t, a_t) \nabla \log \pi_{\theta}(a_t | s_t) \right]\end{aligned}$$

where (a) follows from the tower property of the conditional expectations and (b) follows from that the Markov property implies $\mathbb{E}[\widehat{Q}^{\pi_{\theta}}(s_t, a_t) | s_t, a_t] = Q^{\pi_{\theta}}(s_t, a_t)$. ■

Hence, the following procedure is a stochastic gradient ascent algorithm:

1. initialize θ_0 .
2. For $t = 0, 1, \dots$
 - (a) Sample $\tau \sim \Pr_{\mu}^{\pi_{\theta}}$.
 - (b) Update:

$$\theta_{t+1} = \theta_t + \eta_t \widehat{\nabla V}^{\pi_{\theta}}(\mu)$$

where η_t is the stepsize and $\widehat{\nabla V}^{\pi_{\theta}}(\mu)$ estimated with τ .

Note here that we are ignoring that τ is an infinite length sequence. It can be truncated appropriately so as to control the bias.

The following is standard result with regards to non-convex optimization. Again, with reasonably bounded variance, we will obtain a point θ_t with small gradient norm.

Lemma 4.9. (*Stochastic Convergence to Stationary Points*) Assume that for all $\theta \in \Theta$, V^{π_θ} is β -smooth and bounded below by V_* . Suppose the variance is bounded as follows:

$$\mathbb{E}[\|\widehat{\nabla V^{\pi_\theta}}(\mu) - \nabla V^{\pi_\theta}(\mu)\|^2] \leq \sigma^2$$

For $t \leq \beta(V^*(\mu) - V^{(0)}(\mu))/\sigma^2$, suppose we use a constant stepsize of $\eta_t = 1/\beta$, and thereafter, we use $\eta_t = \sqrt{2/(\beta T)}$. For all T , we have:

$$\min_{t \leq T} \mathbb{E}[\|\nabla V^{(t)}(\mu)\|^2] \leq \frac{2\beta(V^*(\mu) - V^{(0)}(\mu))}{T} + \sqrt{\frac{2\sigma^2}{T}}.$$

Baselines and stochastic gradient ascent

A significant practical issue is that the variance σ^2 is often large in practice. Here, a form of variance reduction is often critical in practice. A common method is as follows.

Let $f : \mathcal{S} \rightarrow \mathbb{R}$.

1. Construct f as an estimate of $V^{\pi_\theta}(\mu)$. This can be done using any previous data.
2. Sample a new trajectory τ , and define:

$$\begin{aligned} \widehat{Q}^{\pi_\theta}(s_t, a_t) &:= (1 - \gamma) \sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'}) \\ \widehat{\nabla V^{\pi_\theta}}(\mu) &:= \sum_{t=0}^{\infty} \gamma^t \left(\widehat{Q}^{\pi_\theta}(s_t, a_t) - f(s_t) \right) \nabla \log \pi_\theta(a_t | s_t) \end{aligned}$$

We often refer to $f(s)$ as a *baseline* at state s .

Lemma 4.10. (*Unbiased gradient estimate with Variance Reduction*) For any procedure used to construct to the baseline function $f : \mathcal{S} \rightarrow \mathbb{R}$, if the samples used to construct f are independent of the trajectory τ , where $\widehat{Q}^{\pi_\theta}(s_t, a_t)$ is constructed using τ , then:

$$\mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \left(\widehat{Q}^{\pi_\theta}(s_t, a_t) - f(s_t) \right) \nabla \log \pi_\theta(a_t | s_t) \right] = \nabla V^{\pi_\theta}(\mu)$$

where the expectation is with respect to both the random trajectory τ and the random function $f(\cdot)$.

Proof: For any function $g(s)$,

$$\mathbb{E}[\nabla \log \pi(a|s)g(s)] = \sum_a \nabla \pi(a|s)g(s) = g(s) \sum_a \nabla \pi(a|s) = g(s) \nabla \sum_a \pi(a|s) = g(s) \nabla 1 = 0$$

Using that $f(\cdot)$ is independent of τ , we have that for all t

$$\mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t f(s_t) \nabla \log \pi_\theta(a_t | s_t) \right] = 0$$

The result now follow froms Lemma 4.8. ■

4.2 The Softmax Policy and Relative Entropy Regularization

Let us now return to the softmax policy, from Equation 4.1.

Observe that:

$$\frac{\partial \log \pi_\theta(a|s)}{\partial \theta_{s',a'}} = \mathbb{1}[s = s'] \left(\mathbb{1}[a = a'] - \pi_\theta(a'|s) \right)$$

where $\mathbb{1}[\mathcal{E}]$ is the indicator of \mathcal{E} being true.

Lemma 4.11. *For the softmax policy class, we have:*

$$\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d^{\pi_\theta}(s) \pi_\theta(a|s) A^{\pi_\theta}(s, a)$$

Proof: We have:

$$\begin{aligned} \frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta_{s,a}} &= \mathbb{E}_{\tau \sim \text{Pr}_\mu^{\pi_\theta}} \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}[s_t = s] \left(\mathbb{1}[a_t = a] A^{\pi_\theta}(s, a) - \pi_\theta(a|s) A^{\pi_\theta}(s_t, a_t) \right) \right] \\ &= \mathbb{E}_{\tau \sim \text{Pr}_\mu^{\pi_\theta}} \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}[(s_t, a_t) = (s, a)] A^{\pi_\theta}(s, a) \right] + \pi_\theta(a|s) \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\tau \sim \text{Pr}_\mu^{\pi_\theta}} \left[\mathbb{1}[s_t = s] A^{\pi_\theta}(s_t, a_t) \right] \\ &= \frac{1}{1-\gamma} \mathbb{E}_{(s', a') \sim d^{\pi_\theta}} [\mathbb{1}[(s', a') = (s, a)] A^{\pi_\theta}(s, a)] + 0 \\ &= \frac{1}{1-\gamma} d^{\pi_\theta}(s, a) A^{\pi_\theta}(s, a), \end{aligned}$$

where the second step uses that $\pi(a|s)$ is a constant (a and s are not random variables) and the penultimate step uses that for any policy $\sum_a \pi(a|s) A^{\pi_\theta}(s, a) = 0$. \blacksquare

Global Convergence

Even for the case of the softmax policy class, our optimization problem is non-convex:

Lemma 4.12. *For the softmax policy class, there exists an MDP M such that $V^{\pi_\theta}(\mu)$ is not a convex function in θ .*

We leave the proof as an exercise to the reader.

A further difficulty when dealing with the softmax parameterization is that for any sequence of policies π^{θ_t} that becomes deterministic, then $\|\nabla V^{\pi^{\theta_t}}\| \rightarrow 0$; roughly speaking, any deterministic policy is a stationary point. We now see that, while our optimization problem is not convex, global convergence to the optimal solution is possible; furthermore, provided we add appropriate regularization, we can obtain a polynomial convergence rate. The difficulty is that the gradient, with respect to $\theta_{s,a}$, may be small when $\pi_\theta(a|s)$ is small, even if the advantage $A^{\pi_\theta}(s, a)$ is positive and large.

In order to prevent probabilities from becoming arbitrarily small, we consider an relative entropy regularization penalty. Recall that the relative-entropy for distributions p and q is defined as:

$$\text{KL}(p, q) := \mathbb{E}_{x \sim p} \left[-\log \frac{q(x)}{p(x)} \right].$$

Denote the uniform distribution over a set \mathcal{X} by $\text{Unif}_{\mathcal{X}}$, and define the following relative-entropy regularized objective

as:

$$\begin{aligned} L_\lambda(\theta) &:= V^{\pi_\theta}(\mu) - \lambda \mathbb{E}_{s \sim \text{Unif}_S} \left[\text{KL}(\text{Unif}_A, \pi_\theta(\cdot|s)) \right] \\ &= V^{\pi_\theta}(\mu) + \frac{\lambda}{|S||A|} \sum_{s,a} \log \pi_\theta(a|s) + \lambda \log |A|, \end{aligned} \quad (4.4)$$

where λ is a regularization parameter. The constant (i.e. the last term) is not relevant with regards to optimization. Note that this regularizer is different from using the entropy function as a regularizer.

The following theorem shows that approximate first-order stationary points of the entropy-regularized objective are approximately globally optimal, provided the regularization is sufficiently small.

Theorem 4.13. (*Relative entropy regularization*) Suppose θ is such that:

$$\|\nabla_\theta L_\lambda(\theta)\|_2 \leq \epsilon_{opt}$$

and $\epsilon_{opt} \leq \lambda/(2|S||A|)$. Then we have that for all starting state distributions ρ :

$$\begin{aligned} V^{\pi_\theta}(\rho) &\geq V^*(\rho) - 2\lambda \left\| \frac{d_\rho^{\pi^*}}{d_\mu^{\pi_\theta}} \right\|_\infty \\ &\geq V^*(\rho) - \frac{2\lambda}{1-\gamma} \left\| \frac{d_\rho^{\pi^*}}{\mu} \right\|_\infty. \end{aligned}$$

We refer to $\left\| \frac{d_\rho^{\pi^*}}{\mu} \right\|_\infty$ as the *distribution mismatch coefficient*. The above theorem shows the importance of having an appropriate measure $\mu(s)$ in order for the approximate first-order stationary points to be near optimal.

Proof: The proof consists of showing that $\max_a A^{\pi_\theta}(s, a) \leq 2(1-\gamma)\lambda/(d_\mu^{\pi_\theta}(s)|S|)$ for all states. To see that this is sufficient, observe that by the performance difference lemma (Lemma 4.5),

$$\begin{aligned} V^*(\rho) - V^{\pi_\theta}(\rho) &= \frac{1}{1-\gamma} \sum_{s,a} d_\rho^{\pi^*}(s) \pi^*(a|s) A^{\pi_\theta}(s, a) \\ &\leq \frac{1}{1-\gamma} \sum_s d_\rho^{\pi^*}(s) \max_{a \in A} A^{\pi_\theta}(s, a) \\ &\leq 2\lambda \sum_s d_\rho^{\pi^*}(s) / (d_\mu^{\pi_\theta}(s)|S|) \\ &\leq 2\lambda \max_s \left(\frac{d_\rho^{\pi^*}(s)}{d_\mu^{\pi_\theta}(s)} \right). \end{aligned}$$

which would then complete the proof of the first claim. The second claim in the proof follows from that $d_\mu^{\pi_\theta}(s) \geq (1-\gamma)\mu(s)$.

We now proceed to show that $\max_a A^{\pi_\theta}(s, a) \leq 2(1-\gamma)\lambda/(d_\mu^{\pi_\theta}(s)|S|)$. For this, it suffices to bound $A^{\pi_\theta}(s, a)$ for any state-action pair s, a where $A^{\pi_\theta}(s, a) \geq 0$ else the claim is trivially true. Consider an (s, a) pair such that $A^{\pi_\theta}(s, a) \geq 0$. Using the policy gradient expression for the softmax parameterization (see Lemma 4.11),

$$\frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d_\mu^{\pi_\theta}(s) \pi_\theta(a|s) A^{\pi_\theta}(s, a) + \frac{\lambda}{|S|} \left(\frac{1}{|A|} - \pi_\theta(a|s) \right). \quad (4.5)$$

The gradient norm assumption $\|\nabla_{\theta} L_{\lambda}(\theta)\|_2 \leq \epsilon_{\text{opt}}$ implies that:

$$\begin{aligned} \epsilon_{\text{opt}} &\geq \frac{\partial L_{\lambda}(\theta)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d_{\mu}^{\pi_{\theta}}(s) \pi_{\theta}(a|s) A^{\pi_{\theta}}(s, a) + \frac{\lambda}{|\mathcal{S}|} \left(\frac{1}{|\mathcal{A}|} - \pi_{\theta}(a|s) \right) \\ &\geq \frac{\lambda}{|\mathcal{S}|} \left(\frac{1}{|\mathcal{A}|} - \pi_{\theta}(a|s) \right), \end{aligned}$$

where we have used where $A^{\pi_{\theta}}(s, a) \geq 0$. Rearranging and using our assumption $\epsilon_{\text{opt}} \leq \lambda/(2|\mathcal{S}| |\mathcal{A}|)$,

$$\pi_{\theta}(a|s) \geq \frac{1}{|\mathcal{A}|} - \frac{\epsilon_{\text{opt}} |\mathcal{S}|}{\lambda} \geq \frac{1}{2|\mathcal{A}|}.$$

Solving for $A^{\pi_{\theta}}(s, a)$ in (4.5), we have:

$$\begin{aligned} A^{\pi_{\theta}}(s, a) &= \frac{1-\gamma}{d_{\mu}^{\pi_{\theta}}(s)} \left(\frac{1}{\pi_{\theta}(a|s)} \frac{\partial L_{\lambda}(\theta)}{\partial \theta_{s,a}} + \frac{\lambda}{|\mathcal{S}|} \left(1 - \frac{1}{\pi_{\theta}(a|s) |\mathcal{A}|} \right) \right) \\ &\leq \frac{1-\gamma}{d_{\mu}^{\pi_{\theta}}(s)} \left(2|\mathcal{A}| \epsilon_{\text{opt}} + \frac{\lambda}{|\mathcal{S}|} \right) \\ &\leq 2 \frac{1-\gamma}{d_{\mu}^{\pi_{\theta}}(s)} \frac{\lambda}{|\mathcal{S}|}, \end{aligned}$$

where the final step uses $\epsilon_{\text{opt}} \leq \lambda/(2|\mathcal{S}| |\mathcal{A}|)$. This completes the proof. \blacksquare

The policy gradient ascent updates for $L_{\lambda}(\theta)$ are given by:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_{\theta} L_{\lambda}(\theta^{(t)}). \quad (4.6)$$

By combining the above theorem with standard results on the convergence of gradient ascent, we obtain the following corollary.

Corollary 4.14. (Global convergence with the softmax policy class) Let $\beta_{\lambda} := \frac{8\gamma}{(1-\gamma)^2} + \frac{2\lambda}{|\mathcal{S}|}$. Starting from any initial $\theta^{(0)}$, consider the updates (4.6) with $\lambda = \frac{\epsilon(1-\gamma)}{2 \left\| \frac{d_{\rho}^{\pi^*}}{\mu} \right\|_{\infty}}$ and $\eta = 1/\beta_{\lambda}$. Then for all starting state distributions ρ , we have

$$\min_{t \leq T} \left\{ V^*(\rho) - V^{(t)}(\rho) \right\} \leq \epsilon \quad \text{whenever} \quad T \geq \frac{320|\mathcal{S}|^2 |\mathcal{A}|^2}{(1-\gamma)^4 \epsilon^2} \left\| \frac{d_{\rho}^{\pi^*}}{\mu} \right\|_{\infty}^2.$$

Proof:[of Corollary 4.14] Let β_{λ} be the smoothness of $L_{\lambda}(\theta)$. A valid upper bound on β_{λ} is:

$$\beta_{\lambda} \leq \frac{8\gamma}{(1-\gamma)^2} + \frac{2\lambda}{|\mathcal{S}|},$$

where we leave the proof as an exercise to the reader.

Using Theorem 4.13, the desired optimality gap ϵ will follow if we set $\lambda = \frac{\epsilon(1-\gamma)}{2 \left\| \frac{d_{\rho}^{\pi^*}}{\mu} \right\|_{\infty}}$ and if $\|\nabla_{\theta} L_{\lambda}(\theta)\|_2 \leq \lambda/(2|\mathcal{S}| |\mathcal{A}|)$. In order to complete the proof, we need to bound the iteration complexity of making the gradient sufficiently small.

By Lemma 4.7, after T iterations of gradient ascent with stepsize of $1/\beta_{\lambda}$, we have

$$\min_{t \leq T} \left\| \nabla_{\theta} V^{(t)}(\mu) \right\|_2^2 \leq \frac{2\beta_{\lambda}(V^*(\mu) - V^{(0)}(\mu))}{T} \leq \frac{2\beta_{\lambda}}{T},$$

where β_λ is an upper bound on the smoothness of $L_\lambda(\theta)$. We seek to ensure

$$\epsilon_{\text{opt}} \leq \sqrt{\frac{2\beta_\lambda}{T}} \leq \frac{\lambda}{2|\mathcal{S}||\mathcal{A}|}$$

Choosing $T \geq \frac{8\beta_\lambda |\mathcal{S}|^2 |\mathcal{A}|^2}{\lambda^2}$ satisfies the above inequality. Using our bound for β_λ ,

$$\frac{8\beta_\lambda |\mathcal{S}|^2 |\mathcal{A}|^2}{\lambda^2} \leq \frac{64 |\mathcal{S}|^2 |\mathcal{A}|^2}{(1-\gamma)^2 \lambda^2} + \frac{16 |\mathcal{S}| |\mathcal{A}|^2}{\lambda} \leq \frac{80 |\mathcal{S}|^2 |\mathcal{A}|^2}{(1-\gamma)^2 \lambda^2} = \frac{320 |\mathcal{S}|^2 |\mathcal{A}|^2}{(1-\gamma)^4 \epsilon^2} \left\| \frac{d_{\rho}^{\pi^*}}{\mu} \right\|_{\infty}^2$$

where we have used that $\lambda < 1$. This completes the proof. \blacksquare

4.3 The Natural Policy Gradient

Observe that a policy constitutes a family of probability distributions $\{\pi_\theta(\cdot|s) | s \in \mathcal{S}\}$. We now consider a pre-conditioned gradient descent method based on this family of distributions. Recall that the Fisher information matrix of a parameterized density $p_\theta(x)$ is defined as $\mathbb{E}_{x \sim p_\theta} [(\nabla \log p_\theta(x)) \nabla \log p_\theta(x)^\top]$. Now we let us define \mathcal{F}_μ^θ as an (average) Fisher information matrix on the family of distributions $\{\pi_\theta(\cdot|s) | s \in \mathcal{S}\}$ as follows:

$$\mathcal{F}_\mu^\theta := \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} [(\nabla \log \pi_\theta(a|s)) \nabla \log \pi_\theta(a|s)^\top].$$

Note that the average is under the state-action visitation frequencies.

The natural policy gradient method is defined as:

$$\theta \leftarrow \theta + \eta (\mathcal{F}_\mu^\theta)^\dagger \nabla V^{\pi_\theta}(\mu) \quad (4.7)$$

where M^\dagger denotes the Moore-Penrose pseudoinverse of the matrix M .

4.3.1 Global Convergence and the Softmax Policy Class

In this subsection, we restrict the optimization measure to be identical to the performance measure, i.e. $\mu = \rho$; there reason for this will become clear in our main theorem, where we see the NPG is effective when the algorithm is directly applied to the performance measure ρ . Furthermore, we restrict attention to states $s \in \mathcal{S}$ reachable from ρ , since, without loss of generality, we can exclude states that are not reachable under this start state distribution¹.

The functional form.

With the softmax parameterization, the NPG updates take a strikingly simple form.

Lemma 4.15. (NPG for the softmax parameterization) *For the softmax policy class, we have that:*

$$\left[(\mathcal{F}_\mu^\theta)^\dagger \nabla V^{\pi_\theta}(\mu) \right]_{s,a} = \frac{1}{1-\gamma} A^{\pi_\theta}(s, a) - c(s)$$

where c is a function only of the state. This implies that the NPG updates (4.7) take the form:

$$\pi^{(t+1)}(a|s) = \pi^{(t)}(a|s) \frac{\exp(\eta A^{(t)}(s, a)/(1-\gamma))}{Z_t(s)},$$

where $Z_t(s) = \sum_{a \in \mathcal{A}} \pi^{(t)}(a|s) \exp(\eta A^{(t)}(s, a)/(1-\gamma))$.

¹Specifically, we restrict the MDP to the set of states $\{s \in \mathcal{S} : \exists \pi \text{ such that } d_\rho^\pi(s) > 0\}$.

Proof: By definition of the Moore-Penrose pseudoinverse, we have that $(\mathcal{F}_\mu^\theta)^\dagger \nabla V^{\pi_\theta}(\mu) = w_*$ if and only if w_* is the minimum norm solution of:

$$\min_w \|\mathcal{F}_\mu^\theta w - \nabla V^{\pi_\theta}(\mu)\|^2.$$

For the softmax policy parameterization, we have by Lemma 4.11,

$$w^\top \nabla_\theta \log \pi_\theta(a|s) = w_{s,a} - \sum_{a' \in \mathcal{A}} w_{s,a'} \pi_\theta(a'|s) = w_{s,a} - c_s$$

where c_s is not a function of a . This implies that:

$$\begin{aligned} \mathcal{F}_\mu^\theta w &= \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[\nabla \log \pi_\theta(a|s) \left(w^\top \nabla_\theta \log \pi_\theta(a|s) \right) \right] \\ &= \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[\nabla \log \pi_\theta(a|s) \left(w_{s,a} - c_s \right) \right] \end{aligned}$$

Again using the functional form of derivative of the softmax policy parameterization,

$$\left[\mathcal{F}_\mu^\theta w \right]_{s',a'} = d^{\pi_\theta}(s') \pi_\theta(a'|s') \left(w_{s',a'} - c_{s'} \right)$$

which implies:

$$\|\mathcal{F}_\mu^\theta w - \nabla V^{\pi_\theta}(\mu)\|^2 = \sum_{s,a} \left(d^{\pi_\theta}(s) \pi_\theta(a|s) \left(w_{s,a} - A^{\pi_\theta}(s,a) - \sum_{a' \in \mathcal{A}} w_{s,a'} \pi_\theta(a'|s) \right) \right)^2.$$

By taking the minimum norm solution, the proof of the first claim is completed. The proof of the second claim follows by the definition of the NPG update rule. \blacksquare

A dimension free convergence rate.

We now provide a dimension free convergence rate of this algorithm.

Theorem 4.16 (Global convergence for Natural Policy Gradient Ascent). *For the softmax policy class, suppose we run the NPG updates (4.7) using $\mu \in \Delta(\mathcal{S})$ and with $\theta^{(0)} = 0$. Fix $\eta > 0$. For all $T > 0$, we have:*

$$V^{(T)}(\mu) \geq V^*(\mu) - \frac{\log |\mathcal{A}|}{\eta T} - \frac{1}{(1-\gamma)^2 T}.$$

Note in the above the theorem that the NPG algorithm is directly applied to the performance measure $V^\pi(\mu)$, and the guarantees are also with respect to μ . In particular, there is no distribution mismatch coefficient in the rate of convergence.

In particular, setting $\eta \geq (1-\gamma)^2 \log |\mathcal{A}|$, we see that NPG finds an ϵ -optimal policy in a number of iterations that is at most:

$$T \leq \frac{2}{(1-\gamma)^2 \epsilon},$$

which has no dependence on the number of states or actions, despite the non-concavity of the underlying optimization problem.

The proof strategy we take borrows ideas from the classical multiplicative weights algorithm.

First, the following improvement lemma is helpful:

Lemma 4.17 (Improvement lower bound for NPG). *For the iterates $\pi^{(t)}$ generated by the NPG updates (4.7), we have for all starting state distributions μ*

$$V^{(t+1)}(\mu) - V^{(t)}(\mu) \geq \frac{(1-\gamma)}{\eta} \mathbb{E}_{s \sim \mu} \log Z_t(s) \geq 0.$$

Proof: First, let us show that $\log Z_t(s) \geq 0$. To see this, observe:

$$\begin{aligned} \log Z_t(s) &= \log \sum_a \pi^{(t)}(a|s) \exp(\eta A^{(t)}(s, a)/(1-\gamma)) \\ &\geq \sum_a \pi^{(t)}(a|s) \log \exp(\eta A^{(t)}(s, a)/(1-\gamma)) = \frac{\eta}{1-\gamma} \sum_a \pi^{(t)}(a|s) A^{(t)}(s, a) = 0. \end{aligned}$$

where the inequality follows by Jensen's inequality on the concave function $\log x$. The performance difference lemma implies:

$$\begin{aligned} V^{(t+1)}(\mu) - V^{(t)}(\mu) &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\mu^{(t+1)}} \sum_a \pi^{(t+1)}(a|s) A^{(t)}(s, a) \\ &= \frac{1}{\eta} \mathbb{E}_{s \sim d_\mu^{(t+1)}} \sum_a \pi^{(t+1)}(a|s) \log \frac{\pi^{(t+1)}(a|s) Z_t(s)}{\pi^{(t)}(a|s)} \\ &= \frac{1}{\eta} \mathbb{E}_{s \sim d_\mu^{(t+1)}} \text{KL}(\pi_s^{(t+1)} || \pi_s^{(t)}) + \frac{1}{\eta} \mathbb{E}_{s \sim d_\mu^{(t+1)}} \log Z_t(s) \\ &\geq \frac{1}{\eta} \mathbb{E}_{s \sim d_\mu^{(t+1)}} \log Z_t(s) \geq \frac{1-\gamma}{\eta} \mathbb{E}_{s \sim \mu} \log Z_t(s). \end{aligned}$$

where the last step uses that $d_\mu^{(t+1)} \geq (1-\gamma)\mu$, componentwise, and that $\log Z_t(s) \geq 0$. ■

With this lemma, we now prove Theorem 4.16.

Proof:[of Theorem 4.16] Since ρ is fixed, we use d^* as shorthand for $d_\rho^{\pi^*}$; we also use π_s as shorthand for the vector of $\pi(\cdot|s)$. By the performance difference lemma (Lemma 4.5),

$$\begin{aligned} V^{\pi^*}(\rho) - V^{(t)}(\rho) &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d^*} \sum_a \pi^*(a|s) A^{(t)}(s, a) \\ &= \frac{1}{\eta} \mathbb{E}_{s \sim d^*} \sum_a \pi^*(a|s) \log \frac{\pi^{(t+1)}(a|s) Z_t(s)}{\pi^{(t)}(a|s)} \\ &= \frac{1}{\eta} \mathbb{E}_{s \sim d^*} \left(\text{KL}(\pi_s^* || \pi_s^{(t)}) - \text{KL}(\pi_s^* || \pi_s^{(t+1)}) + \sum_a \pi^*(a|s) \log Z_t(s) \right) \\ &= \frac{1}{\eta} \mathbb{E}_{s \sim d^*} \left(\text{KL}(\pi_s^* || \pi_s^{(t)}) - \text{KL}(\pi_s^* || \pi_s^{(t+1)}) + \log Z_t(s) \right), \end{aligned}$$

where we have used the closed form of our updates from Lemma 4.15 in the second step.

By applying Lemma 4.17 with d^* as the starting state distribution, we have:

$$\frac{1}{\eta} \mathbb{E}_{s \sim d^*} \log Z_t(s) \leq \frac{1}{1-\gamma} \left(V^{(t+1)}(d^*) - V^{(t)}(d^*) \right)$$

which gives us a bound on $\mathbb{E}_{s \sim d^*} \log Z_t(s)$.

Using the above equation and that $V^{(t+1)}(\rho) \geq V^{(t)}(\rho)$ (as $V^{(t+1)}(s) \geq V^{(t)}(s)$ for all states s by Lemma 4.17), we have:

$$\begin{aligned}
V^{\pi^*}(\rho) - V^{(T-1)}(\rho) &\leq \frac{1}{T} \sum_{t=0}^{T-1} (V^{\pi^*}(\rho) - V^{(t)}(\rho)) \\
&\leq \frac{1}{\eta T} \sum_{t=0}^{T-1} \mathbb{E}_{s \sim d^*} (\text{KL}(\pi_s^* || \pi_s^{(t)}) - \text{KL}(\pi_s^* || \pi_s^{(t+1)})) + \frac{1}{\eta T} \sum_{t=0}^{T-1} \mathbb{E}_{s \sim d^*} \log Z_t(s) \\
&\leq \frac{\mathbb{E}_{s \sim d^*} \text{KL}(\pi_s^* || \pi^{(0)})}{\eta T} + \frac{1}{(1-\gamma)T} \sum_{t=0}^{T-1} (V^{(t+1)}(d^*) - V^{(t)}(d^*)) \\
&= \frac{\mathbb{E}_{s \sim d^*} \text{KL}(\pi_s^* || \pi^{(0)})}{\eta T} + \frac{V^{(T)}(d^*) - V^{(0)}(d^*)}{(1-\gamma)T} \\
&\leq \frac{\log |\mathcal{A}|}{\eta T} + \frac{1}{(1-\gamma)^2 T}.
\end{aligned}$$

The proof is completed using that $V^{(T)}(\rho) \geq V^{(T-1)}(\rho)$. ■

4.3.2 Function Approximation and a Connection to Transfer Learning

This section studies policy classes parameterized by some parameter $\theta \in \mathbb{R}^d$. In contrast with the tabular results in the previous sections, the policies classes that we are often interested in are not fully expressive, e.g. $d \ll |\mathcal{S}||\mathcal{A}|$ (indeed $|\mathcal{S}|$ or $|\mathcal{A}|$ need not even be finite for the results in this section); in this sense, we are in the regime of function approximation.

We now consider the (unconstrained) policy optimization problem:

$$\max_{\theta \in \mathbb{R}^d} V^{\pi_\theta}(\rho)$$

for some state distribution ρ . As before, while we are interested in $V^{\pi_\theta}(\rho)$, we shall see that optimization with respect to a different distribution μ is also important in the approximate case.

Compatible function approximation

To analyze NPG for restricted policy classes, we leverage a close connection of the NPG update rule (4.7) to the notion of *compatible function approximation*. It is helpful to consider a distribution ν over state-action pairs, i.e. $\nu \in \Delta(\mathcal{S}) \times \Delta(\mathcal{A})$. For a weight vector $w \in \mathbb{R}^d$ and with respect to ν and θ , define the compatible function approximation error L_ν and the minimal compatible function approximation error L_ν^* as:

$$L_\nu(w; \theta) := \mathbb{E}_{s,a \sim \nu} \left[\left(A^{\pi_\theta}(s, a) - w \cdot \nabla_\theta \log \pi_\theta(a|s) \right)^2 \right], \quad L_\nu^*(\theta) := \min_w L_\nu(w; \theta). \quad (4.8)$$

The NPG update rule can be viewed as the minimizer of a certain function approximation problem: precisely, if we take $\nu(s, a) = d_\mu^{\pi_\theta}(s) \pi_\theta(a|s)$, then

$$F_\mu(\theta)^\dagger \nabla_\theta V^\theta(\mu) \in \operatorname{argmin}_w L_\nu(w; \theta). \quad (4.9)$$

This is a straightforward consequence of the first order optimality conditions. Consequently, when the minimizer is unique, then the NPG update direction is precisely identified by minimizing the regression problem implied by the

compatible function approximation error. When the solution is not unique, we allow the use of any minimizer of $L_\nu(w; \theta)$ and prove that this does not affect the convergence properties of the algorithm, beyond the norm of the solution we find.

This observation also opens a direct pathway to approximate updates where we can solve the regression problems with samples. Furthermore, if the minimal compatible function approximation error is 0, then Thm 4.16 already establishes that global convergence occurs for the NPG update rule. The more often encountered scenario is when the minimal compatible function error is nonzero, as is likely to happen when the parameterization using θ does not represent all possible policies.

Example 4.18 (Gradients and compatible function approximation). Consider the state-action visitation distribution under π starting from μ , i.e.

$$\nu(s, a) = d_\mu^{\pi_\theta}(s) \pi_\theta(a|s)$$

and the minimizer

$$w^*(\theta) \in \operatorname{argmin}_w L_\nu(w; \theta).$$

Let us denote the best linear predictor of $A^{\pi_\theta}(s, a)$ using $\nabla_\theta \log \pi_\theta(a|s)$ by $\hat{A}^{\pi_\theta}(s, a)$, i.e.

$$\hat{A}^{\pi_\theta}(s, a) := w^*(\theta) \cdot \nabla_\theta \log \pi_\theta(a|s).$$

The first order optimality conditions for $\hat{A}^{\pi_\theta}(s, a)$ being the best linear predictor directly imply that:

$$\begin{aligned} \nabla_\theta V^{\pi_\theta}(\mu) &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} [\nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(s, a)] \\ &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} [\nabla_\theta \log \pi_\theta(a|s) \hat{A}^{\pi_\theta}(s, a)]. \end{aligned}$$

The proof is left as an exercise to the reader. In other words, computing the gradient with the best linear approximation, $\hat{A}^{\pi_\theta}(s, a)$, instead of $A^{\pi_\theta}(s, a)$ results in identical outcomes.

Example 4.19 (Linear policies and approximation). To gain further intuition about the compatible function approximation error, we can specialize it to the linear setting of Example 4.2. Here,

$$\nabla_\theta \log \pi_\theta(a|s) = \tilde{\phi}_{s,a}, \quad \text{where } \tilde{\phi}_{s,a} := \phi_{s,a} - \mathbb{E}_{a' \sim \pi_\theta(\cdot|s)} [\phi_{s,a'}]$$

is the centered version of $\phi_{s,a}$. Thus, we get

$$L_\nu(w; \theta) = \mathbb{E}_{s,a \sim \nu} \left[\left(A^{\pi_\theta}(s, a) - w \cdot \tilde{\phi}_{s,a} \right)^2 \right].$$

That is, the compatible function approximation error measures the expressivity of our parameterization in how well linear functions of the parameterization can capture the policy's advantage function. More generally, the error provides a particularly useful characterization of the policy class expressivity for NPG updates as characterized in the main result of this section (Theorem 4.22).

Example 4.20 (Neural policies and approximation). Let us now specialize to the more general case of neural policies, as in Example 4.3. Here,

$$\nabla_\theta \log \pi_\theta(a|s) = g_\theta(s, a), \quad \text{where } g_\theta(s, a) := \nabla f_\theta(s, a) - \mathbb{E}_{a' \sim \pi_\theta(\cdot|s)} [\nabla f_\theta(s, a')].$$

Thus, we get

$$L_\nu(w; \theta) = \mathbb{E}_{s,a \sim \nu} \left[\left(A^{\pi_\theta}(s, a) - w \cdot g_\theta(s, a) \right)^2 \right].$$

A Helper Lemma

We now assume that the policy class is smooth in the sense that $\log \pi_\theta$ is a smooth function of θ .

Assumption 4.21. (Policy Smoothness) Assume for all $s \in \mathcal{S}$, $a \in \mathcal{A}$ that $\log \pi_\theta(a|s)$ is a β -smooth function of θ (recall (4.3)).

This smoothness assumption is a strict generalization of our previous analysis for the softmax parameterization in the tabular setting. For the case of linear policies, as in Example 4.2, $\log \pi_\theta(a|s)$ is a β -smooth function, provided we have the uniform bound $\|\phi_{s,a}\|_2^2 \leq \beta$. For neural policies, as in Example 4.3, the smoothness of $\log \pi_\theta$ will depend on both the Lipschitz and smoothness constants of $f_\theta(s, a)$.

The main technical result of this section quantifies the effect of error in the compatible function approximation. The theorem is stated in an abstract manner, so that it can be subsequently utilized in concrete settings to deal with optimization and statistical error. The theorem relates the compatible function approximation errors on an arbitrary sequence $\{w^{(t)}, \theta^{(t)}\}$ to the sub-optimality of the resulting policy.

Let us more abstractly consider an update rule of the form

$$\theta^{(t+1)} = \theta^{(t)} + \eta w^{(t)}. \quad (4.10)$$

where $w^{(t)}$ is either the exact NPG update rule or a (sample based) approximation to it.

Theorem 4.22. (NPG approximation) Fix a comparison policy π and a state distribution ρ . Define ν as the induced state-action measure under π , i.e.

$$\nu(s, a) = d_\rho^\pi(s) \pi(a|s).$$

Suppose that the update rule (4.10) starts with $\theta^{(0)} = 0$ and uses the (arbitrary) sequence of weights $w^{(0)}, \dots, w^{(T)}$; that Assumption 4.21 holds; and that for all $t < T$,

$$L_\nu(w^{(t)}; \theta^{(t)}) \leq \epsilon_{approx}, \quad \|w^{(t)}\|_2 \leq W.$$

We have that:

$$\min_{t < T} \left\{ V^\pi(\rho) - V^{(t)}(\rho) \right\} \leq \frac{1}{1-\gamma} \left(\sqrt{\epsilon_{approx}} + \frac{\log |\mathcal{A}|}{\eta T} + \frac{\eta \beta W^2}{2} \right).$$

Proof: First, by smoothness (see Equation (4.3)),

$$\begin{aligned} \log \frac{\pi^{(t+1)}(a|s)}{\pi^{(t)}(a|s)} &\geq \nabla_\theta \log \pi^{(t)}(a|s) \cdot (\theta^{(t+1)} - \theta^{(t)}) - \frac{\beta}{2} \|\theta^{(t+1)} - \theta^{(t)}\|_2^2 \\ &= \eta \nabla_\theta \log \pi^{(t)}(a|s) \cdot w^{(t)} - \eta^2 \frac{\beta}{2} \|w^{(t)}\|_2^2. \end{aligned}$$

We use d as shorthand for d_ρ^π (note ρ and π are fixed); we also use π_s as shorthand for the vector $\pi(\cdot|s)$. By the performance difference lemma (Lemma 4.5),

$$\begin{aligned}
& \mathbb{E}_{s \sim d} \left(\text{KL}(\pi_s \| \pi_s^{(t)}) - \text{KL}(\pi_s \| \pi_s^{(t+1)}) \right) \\
&= \mathbb{E}_{s \sim d} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[\log \frac{\pi^{(t+1)}(a|s)}{\pi^{(t)}(a|s)} \right] \\
&\geq \eta \mathbb{E}_{s \sim d} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[\nabla_{\theta} \log \pi^{(t)}(a|s) \cdot w^{(t)} \right] - \eta^2 \frac{\beta}{2} \|w^{(t)}\|_2^2 && \text{(using previous display)} \\
&= \eta \mathbb{E}_{s \sim d} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[A^{(t)}(s, a) \right] - \eta^2 \frac{\beta}{2} \|w^{(t)}\|_2^2 \\
&\quad + \eta \mathbb{E}_{s \sim d} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[\nabla_{\theta} \log \pi^{(t)}(a|s) \cdot w^{(t)} - A^{(t)}(s, a) \right] \\
&\geq (1 - \gamma) \eta \left(V^{\pi}(\rho) - V^{(t)}(\rho) \right) - \eta^2 \frac{\beta}{2} \|w^{(t)}\|_2^2 \\
&\quad - \eta \sqrt{\mathbb{E}_{s \sim d} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[\left(\nabla_{\theta} \log \pi^{(t)}(a|s) \cdot w^{(t)} - A^{(t)}(s, a) \right)^2 \right]} && \text{(Lemma 4.5 and Jensen)} \\
&\geq (1 - \gamma) \eta \left(V^{\pi}(\rho) - V^{(t)}(\rho) \right) - \eta^2 \frac{\beta}{2} W^2 - \eta \sqrt{\epsilon_{\text{approx}}}.
\end{aligned}$$

where we have used our assumption on ϵ_{approx} .

Rearranging, we have:

$$\begin{aligned}
& V^{\pi}(\rho) - V^{(t)}(\rho) \\
&\leq \frac{1}{1 - \gamma} \left(\frac{1}{\eta} \mathbb{E}_{s \sim d} \left(\text{KL}(\pi_s \| \pi_s^{(t)}) - \text{KL}(\pi_s \| \pi_s^{(t+1)}) \right) + \frac{\eta \beta}{2} W^2 + \sqrt{\epsilon_{\text{approx}}} \right)
\end{aligned}$$

Proceeding,

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} (V^{\pi}(\rho) - V^{(t)}(\rho)) &\leq \frac{1}{\eta T (1 - \gamma)} \sum_{t=0}^{T-1} \mathbb{E}_{s \sim d} (\text{KL}(\pi_s \| \pi_s^{(t)}) - \text{KL}(\pi_s \| \pi_s^{(t+1)})) \\
&\quad + \frac{1}{T (1 - \gamma)} \sum_{t=0}^{T-1} \left(\frac{\eta \beta W^2}{2} + \sqrt{\epsilon_{\text{approx}}} \right) \\
&\leq \frac{\mathbb{E}_{s \sim d} \text{KL}(\pi_s \| \pi^{(0)})}{\eta T (1 - \gamma)} + \frac{\eta \beta W^2}{2(1 - \gamma)} + \frac{\sqrt{\epsilon_{\text{approx}}}}{1 - \gamma} \\
&\leq \frac{\log |\mathcal{A}|}{\eta T (1 - \gamma)} + \frac{\eta \beta W^2}{2(1 - \gamma)} + \frac{\sqrt{\epsilon_{\text{approx}}}}{1 - \gamma}
\end{aligned}$$

which completes the proof. ■

Convergence, Approximation, and Transfer Learning

In order to leverage Theorem 4.22 for obtaining concrete convergence rates, we now consider the setting where we have access to exact natural policy gradients, or equivalently (due to Equation 4.9), we use the exact minimizer of $L_{\nu^{(t)}}(w; \theta^{(t)})$ at each iteration t .

The difficulty with applying Theorem 4.22 is that, ideally, we seek to ensure that $L_{\nu}(w; \theta^{(t)})$ is small, where ν is the unknown state-action measure of a comparator policy π . Like previous results, we use an exploratory initial distribution over states and actions to partially address this issue as described below.

Formally, we consider a (slightly) more general version of the natural policy gradient algorithm, and we provide an agnostic learning result for this algorithm (where we compare to the best policy in the policy class). Instead of using a starting state distribution μ over states, the generalized version uses a starting distribution ν_0 over state-action pairs. Analogous to d_μ^π , this starting state-action distribution induces a state-action visitation distribution $\nu_{\nu_0}^\pi$, defined as:

$$\nu_{\nu_0}^\pi(s, a) := (1 - \gamma) \mathbb{E}_{s_0, a_0 \sim \nu_0} \sum_{t=0}^{\infty} \gamma^t \Pr^\pi(s_t = s, a_t = a | s_0, a_0) \quad (4.11)$$

where $\Pr^\pi(s_t = s, a_t = a | s_0, a_0)$ is the state-action visitation probability that $s_t = s$ and $a_t = a$, after starting at state s_0 , taking action a_0 , and following π thereafter. As per our convention, $\nu^{(t)}$ is shorthand for $\nu_{\nu_0}^{\pi^{(t)}}$.

We now consider the update rule:

$$w^{(t)} \in \operatorname{argmin}_w L_{\nu^{(t)}}(w; \theta^{(t)}). \quad (4.12)$$

which exactly computes a minimizer (of a least squares objective function); the following subsection provides a finite sample version. This form is a more general version than that of (4.9); the generality is helpful in the following corollary, which provides a (worst-case) upper bound in terms of ν_0 .

Corollary 4.23. (*Agnostic Learning with NPG*) Suppose that we follow the update rule in (4.12) starting with $\theta^{(0)} = 0$. Fix a state distribution ρ and a state-action distribution ν_0 ; let $\pi^* = \pi_{\theta^*}$ the best policy in Π for ρ , i.e. $\theta^* \in \operatorname{argmax}_{\theta \in \Theta} V^{\pi_\theta}(\rho)$. Define ν^* as the induced state-action measure under π^* , i.e.

$$\nu^*(s, a) = d_\rho^{\pi^*}(s) \pi^*(a | s).$$

Suppose $\eta = \sqrt{2 \log |\mathcal{A}| / (\beta W^2 T)}$; Assumption 4.21 holds; and that for all $t < T$,

$$L_{\nu^*}(w^{(t)}; \theta^{(t)}) \leq \epsilon_{\text{transfer}}, \quad L_{\nu^{(t)}}^*(\theta^{(t)}) \leq \epsilon_{\text{approx}}, \quad \|w^{(t)}\|_2 \leq W.$$

We have that:

$$\begin{aligned} \min_{t < T} \left\{ V^{\pi^*}(\rho) - V^{(t)}(\rho) \right\} &\leq \left(\frac{W \sqrt{2 \beta \log |\mathcal{A}|}}{(1 - \gamma)} \right) \cdot \frac{1}{\sqrt{T}} + \sqrt{\frac{1}{(1 - \gamma)^2} \epsilon_{\text{transfer}}} \\ &\leq \left(\frac{W \sqrt{2 \beta \log |\mathcal{A}|}}{(1 - \gamma)} \right) \cdot \frac{1}{\sqrt{T}} + \sqrt{\frac{1}{(1 - \gamma)^3} \left\| \frac{\nu^*}{\nu_0} \right\|_\infty \epsilon_{\text{approx}}}. \end{aligned}$$

The connection to transfer learning is due to that $L_{\nu^*}(\cdot; \theta^{(t)})$, the error under ν^* , is what ultimately governs our approximation error, yet our algorithm minimizes the error under $L_{\nu^{(t)}}(\cdot; \theta^{(t)})$. This is due to that ν^* is not known.

Proof: The first claim follows from Theorem 4.22 along with our choice of η and the definition of $\epsilon_{\text{transfer}}$. For the second claim,

$$L_{\nu^*}(w^{(t)}; \theta^{(t)}) \leq \left\| \frac{\nu^*}{\nu^{(t)}} \right\|_\infty \cdot L_{\nu^{(t)}}(w^{(t)}; \theta^{(t)}) \leq \left\| \frac{\nu^*}{\nu^{(t)}} \right\|_\infty \cdot \epsilon_{\text{approx}} \leq \frac{1}{(1 - \gamma)} \left\| \frac{\nu^*}{\nu_0} \right\|_\infty \cdot \epsilon_{\text{approx}}.$$

where we have used $\nu^{(t)}(s, a) \geq (1 - \gamma) \nu_0(s, a)$. ■

When $\epsilon_{\text{approx}} = 0$, as in the tabular setting, the term depending on distribution mismatch coefficient disappears, consistent with Theorem 4.16. However, the convergence rate is only $O(\sqrt{1/T})$ instead of the $O(1/T)$ bound there. This is because obtaining the fast rate in the function approximation regime appears to require even stronger conditions on the distributions (including $\nu^{(t+1)}$) under which the approximation errors are controlled at each round t .

4.4 Related algorithms

TODO

4.4.1 Trust Region Policy Optimization (TRPO)

TODO: explain how TRPO and NPG are closely related.

4.4.2 Proximal Policy Optimization (PPO)

4.4.3 Conservative Policy Iteration (CPI)

TODO: give CPI guarantee, which is similar to NPG guarantee

4.5 Bibliographic Remarks

The convergence rates in this chapter are largely derived from [Agarwal et al., 2019]. The proof strategy has origins in the online regret framework in changing MDPs [Even-Dar et al., 2009].

The performance difference lemma was widely realized as an important analysis tool due to [Kakade and Langford, 2002, Kakade, 2003]; its origins have been implicit in a number of earlier works.

For standard optimization results in non-convex optimization (e.g. Lemma 4.7 and 4.9), we refer the reader to [Beck, 2017] (also, see [Ghadimi and Lan, 2013]).

The REINFORCE algorithm is due to [Williams, 1992]. The natural policy gradient method was originally presented in [Kakade, 2001]; arguments for this method have been provided based on information geometry [Kakade, 2001, Bagnell and Schneider, 2003].

The notion of *compatible function approximation* was due to [Sutton et al., 1999], which also proved the claim in Example 4.18. The close connection of the NPG update rule to compatible function approximation was noted in [Kakade, 2001].

Chapter 5

Value Function Approximation

Value Function Approximation

Aleksh Agarwal, Nan Jiang, Sham M. Kakade

Chapter 5

The previous lecture showed how policy iteration like methods can be generalized to large state spaces (think discrete, but exponentially large in the desired accuracy parameters) and for policy classes that are not necessarily tabular. This chapter will consider a similar question, but for the class of value iteration like methods. We will attack the question in two parts. First we will study how the value function of a given policy can be evaluated in large state spaces, before moving to the question of policy improvement. The result and proofs in this chapter have been obtained in collaboration with Nan Jiang.

5.1 Approximate Policy Evaluation

First we need some notational preliminaries. Throughout this lecture, we will consider the problem of approximating Q -value functions. We will assume that we are given a class of candidate value functions \mathcal{F} , where each $f \in \mathcal{F}$ is a mapping from $\mathcal{S} \times \mathcal{A}$ to $[0, 1]$ so that it satisfies the desired semantics of a Q -value function. Given a policy π , one question of interest is to find a function $f \in \mathcal{F}$ which approximates Q^π . In order to define the notion of approximation, we introduce additional notation. Note that in our previous lectures, we typically considered approximating value functions uniformly, that is under an ℓ_∞ norm over states and actions. While this is feasible in the tabular case, once the number of states grows large, we can no longer hope to find such a function f which will approximate Q^π uniformly for all states and actions from a reasonable amount of data, and with a function class \mathcal{F} of reasonable statistical complexity (these concepts will be made precise in the sequel). In order to sidestep this issue, taking a cue from supervised learning, we will instead switch to measuring our errors in expectation over some state distribution.

Concretely, given a function $g : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, and a distribution μ over $\mathcal{S} \times \mathcal{A}$, we define:

$$\|g\|_\mu := \sqrt{\mathbb{E}_{(s,a) \sim \mu} g^2(x)}. \quad (5.1)$$

For instance, if $g(s, a) = f(s, a) - Q^\pi(s, a)$, then $\|g\|_\mu$ measures the expected squared loss of f in approximating Q^π under the distribution μ over \mathcal{S} and \mathcal{A} . For evaluating the quality of f in approximating Q^π , a natural measure to consider is

$$d^\pi(s)\pi(a | s) = d_{d_0}^\pi(s)\pi(a | s) = \mathbb{E}_{s_0 \sim d_0} [(1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}^\pi(s_t = s | s_0)] \pi(a | s),$$

where the notation $d_{d_0}^\pi(s)$ is consistent with our definition from the previous lecture. As before, we will drop the subscript d_0 as it will be fixed throughout our treatment. We will further overload the $\|g\|_\mu$ notation, and also allow the use of unnormalized and signed measures μ , interpreting them as integrals with respect to an appropriate base measure. We quickly visit a few useful properties of this generalization, the proofs of which are left to the reader.

Fact 5.1. The definition (5.1) of $\|g\|_\mu$, once extended to signed and unnormalized measures implies that:

1. $\|g\|_{c\mu} = \sqrt{c} \|g\|_\mu$.
2. $\|g\|_{\mu_1 - \mu_2} \leq \|g\|_{\mu_1}$, whenever $\|g\|_{\mu_1 - \mu_2}^2 \geq 0$.

We further define the Bellman operator \mathcal{T}^π . For a function $g : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, we have

$$\mathcal{T}^\pi[g](s, a) := (1 - \gamma)r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} g(s', \pi(s')). \quad (5.2)$$

We now define a simple procedure for estimating the function Q^π . Suppose we are given a set of N samples (s, a, s') drawn according to $d^\pi(s) \nu(a | s) \times P(s' | s, a)$. Here $\nu(a | s)$ is an arbitrary distribution used for generating actions. While π might be a natural choice of ν as mentioned earlier, we allow more general choices as the question of policy improvement which we consider next will require our Q -value estimates to be precise not just for the actions chosen by π , as we will see in the sequel. A reasonable mental model for generating such samples is to generate N trajectories, where we take a random number of steps according to π , starting from the initial state drawn according to d_0 , and then take one more action as per ν . It is easily seen that this sampling model induces the desired probability distribution over the (s, a, s') triples. We will use the shorthand $d^{\pi, \nu}$ to denote the joint distribution over (s, a) pairs which draws $s \sim d_{d_0}^\pi$ and $a \sim \nu(\cdot | s)$.

Let (s_i, a_i, s'_i) refer to the i_{th} triple for $i = 1, 2, \dots, N$. Let $s_{t,i}$ denote the state encountered at time t in trajectory i and similarly for the actions. Let us define:

$$\begin{aligned} \mathcal{L}(f; f') &= \mathbb{E}_{(s,a) \sim d^{\pi, \nu}, s' \sim P(\cdot | s, a)} [(f(s, a) - (1 - \gamma)r(s, a) - \gamma f'(s', \pi(s')))^2]. \\ \hat{\mathcal{L}}_N(f; f') &= \frac{1}{N} \sum_{i=1}^N (f(s_i, a_i) - (1 - \gamma)r(s_i, a_i) - \gamma f'(s'_i, \pi(s'_i)))^2. \\ \hat{\mathcal{T}}_F^\pi f' &= \operatorname{argmin}_{f \in \mathcal{F}} \hat{\mathcal{L}}_N(f; f'). \end{aligned}$$

Intuitively, $\hat{\mathcal{T}}_F^\pi(f')$ maps a function f' to an f which minimizes the Bellman error on π 's trajectories, when the future values are predicted under f' . For the population loss $\mathcal{L}(f; f')$, we expect it to be minimized when $f = f' = Q^\pi$ due to the Bellman equations for value functions.

The procedure we consider for obtaining an estimate of Q^π from samples is the following sample-based value iteration. Initialize $f_0 \in \mathcal{F}$ arbitrarily and iterate:

$$f_k = \hat{\mathcal{T}}_F^\pi f_{k-1} \quad \text{for } k > 0. \quad (5.3)$$

For this procedure, we will establish the following convergence guarantee.

Theorem 5.2. *For the procedure defined in Equation 5.3, for all $k = 0, 1, 2, \dots$, with probability at least $1 - \delta$,*

$$\|f_k - Q^\pi\|_{d^{\pi, \nu}} \leq \sqrt{\|\rho\|_\infty \|1/\rho\|_\infty} \gamma^{k/2} \|f_0 - Q^\pi\|_{d^{\pi, \nu}} + \frac{2\sqrt{\|\rho\|_\infty \|1/\rho\|_\infty}}{1 - \gamma} \left(\sqrt{3} \max_{f \in \mathcal{F}} \|\mathcal{T}_F^\pi f - \mathcal{T}^\pi f\|_{d^{\pi, \nu}} + \sqrt{\frac{19}{N} \log \frac{|\mathcal{F}|}{\delta}} \right).$$

While the second part of the error bound which depends on N goes to zero as we increase the number of samples, the term $\max_{f \in \mathcal{F}} \|\mathcal{T}^\pi f - \mathcal{T}_F^\pi f\|_{d^{\pi, \nu}}$ does not asymptotically vanish in general. It is easily seen to be 0 in the tabular function class, but can be arbitrarily large for other classes. This quantity has been referred to as the *inherent one-step Bellman error* in Antos et al. [2008] and is well-known to control the quality of function approximation guarantees.

We will analyze the properties of this procedure in two parts. First we will obtain an error bound on the quality of f_k in approximating Q^π , assuming all the sample-based estimates are close to their expectations. We will then quantify the size of the statistical deviations.

The error measure we will use to study the convergence of f_k to Q^π is $\|f_k - Q\|_{d^\pi}$. We start with a simple lemma.

Lemma 5.3 (Error decomposition). *For any sequence $f_k : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, we have the bound*

$$\|f_k - Q^\pi\|_{d^{\pi,\nu}} \leq \gamma^{k/2} \sqrt{\|\rho\|_\infty \|1/\rho\|_\infty} \|f_0 - Q^\pi\|_{d^{\pi,\nu}} + \sum_{i=1}^k \gamma^{(k-i)/2} \sqrt{\|\rho\|_\infty \|1/\rho\|_\infty} \|f_i - \mathcal{T}^\pi f_{i-1}\|_{d^{\pi,\nu}}$$

Proof: We will prove the lemma in two steps. First we establish a bound on $\|f_k - Q^\pi\|_{d^{\pi,\pi}}$ in terms of $\|f_i - \mathcal{T}^\pi f_{i-1}\|_{d^{\pi,\pi}}$. We then apply importance weighting to correct for the measure mismatch on both sides of the inequality to obtain the final bound.

For any scaled probability measure μ , d^μ is always a norm so that it satisfies triangle inequality. This implies

$$\begin{aligned} \|f_k - Q^\pi\|_{d^{\pi,\pi}} &\leq \|f_k - \mathcal{T}^\pi f_{k-1}\|_{d^{\pi,\pi}} + \|\mathcal{T}^\pi f_{k-1} - Q^\pi\|_{d^{\pi,\pi}} \\ &= \|f_k - \mathcal{T}^\pi f_{k-1}\|_{d^{\pi,\pi}} + \|\mathcal{T}^\pi f_{k-1} - \mathcal{T}^\pi Q^\pi\|_{d^{\pi,\pi}} && \text{(Bellman equation for } Q^\pi\text{)} \\ &= \|f_k - \mathcal{T}^\pi f_{k-1}\|_{d^{\pi,\pi}} + \gamma \|P^\pi f_{k-1} - P^\pi Q^\pi\|_{d^{\pi,\pi}} \\ &\leq \|f_k - \mathcal{T}^\pi f_{k-1}\|_{d^{\pi,\pi}} + \gamma \|f_{k-1} - Q^\pi\|_{d^{\pi,\pi} P^\pi}, \end{aligned}$$

where the final equality treats $d^{\pi,\pi}$ as a row vector with cardinality $|\mathcal{S}| \cdot |\mathcal{A}|$. This equality follows from the definition of \mathcal{T}^π and since the reward function is assumed to be known. The last inequality is a consequence of Jensen's inequality which gives that $\mathbb{E}[X]^2 \leq \mathbb{E}[X^2]$. Note that by the first part of Fact 5.1, we further obtain

$$\|f_k - Q^\pi\|_{d^{\pi,\pi}} \leq \|f_k - \mathcal{T}^\pi f_{k-1}\|_{d^{\pi,\pi}} + \sqrt{\gamma} \|f_{k-1} - Q^\pi\|_{\gamma d^{\pi,\pi} P^\pi}.$$

In order to simplify further, we define $\mu_0(s, a) = d_0(s)\pi(a | s)$ to be the initial measure extended to (s, a) pairs and observe that $d^{\pi,\pi}$ can be written as

$$d^{\pi,\pi} = (1 - \gamma)\mu_0(I - \gamma P^\pi)^{-1} \quad \text{so that} \quad \gamma P^\pi d^{\pi,\pi} = d^{\pi,\pi} - (1 - \gamma)\mu_0,$$

where the inequalities hold elementwise. Substituting this in our earlier bound, we further obtain

$$\begin{aligned} \|f_k - Q^\pi\|_{d^{\pi,\pi}} &\leq \|f_k - \mathcal{T}^\pi f_{k-1}\|_{d^{\pi,\pi}} + \sqrt{\gamma} \|f_{k-1} - Q^\pi\|_{d^{\pi,\pi} - (1-\gamma)\mu_0} \\ &\leq \|f_k - \mathcal{T}^\pi f_{k-1}\|_{d^{\pi,\pi}} + \sqrt{\gamma} \|f_{k-1} - Q^\pi\|_{d^{\pi,\pi}} && \text{(Part 2 of Fact 5.1)} \\ &\leq \gamma^{k/2} \|f_0 - Q^\pi\|_{d^{\pi,\pi}} + \sum_{i=1}^k \gamma^{(k-i)/2} \|f_i - \mathcal{T}^\pi f_{i-1}\|_{d^{\pi,\pi}}, \end{aligned}$$

where we obtain the final inequality by iterating the same set of steps to bound $\|f_i - Q^\pi\|_{d^{\pi,\pi}}$ for each $i = 1, 2, \dots, k$.

In order to complete the proof, let us recall the notation $\rho(s, a) = \nu(s, a)/\pi(s, a)$. Note that for any two functions $f_1, f_2 \in \{\mathcal{S} \times \mathcal{A} \rightarrow [0, 1]\}$, $\|f_1\|_{d^{\pi,\pi}} \leq \|f_2\|_{d^{\pi,\pi}}$ implies that

$$\|f_1\|_{d^{\pi,\nu}} \leq \|f_1\|_{\|\rho\|_\infty d^{\pi,\pi}} \leq \|f_2\|_{\|\rho\|_\infty d^{\pi,\pi}} \leq \|f_1\|_{\|\rho\|_\infty \|1/\rho\|_\infty d^{\pi,\nu}}.$$

Combining this with our earlier bound completes the proof of the lemma. ■

The bound above suggests that if we had good control over the errors $\|f_i - \mathcal{T}^\pi f_{i-1}\|$ at each iteration, then we get successively good approximations to Q^π . Our next lemma takes a step in this direction by relating the loss function \mathcal{L} whose empirical counterpart is minimized in our iteration (5.3) and the Bellman errors which show up in the previous bound.

Lemma 5.4. For any $f' \in \mathcal{F}$, we have

$$\|f - \mathcal{T}^\pi f'\|_{d^{\pi,\nu}}^2 = \mathcal{L}(f; f') - \mathcal{L}(\mathcal{T}^\pi f'; f').$$

Proof: The proof essentially follows using standard properties of least squares regression. Note that

$$\begin{aligned} \mathcal{L}(f; f') &= \mathbb{E}_{(s,a) \sim d^{\pi,\nu}, s' \sim P(\cdot|s,a)} [(f(s,a) - (1-\gamma)r(s,a) - \gamma f'(s', \pi(s'))))^2] \\ &= \mathbb{E}_{(s,a) \sim d^{\pi,\nu}, s' \sim P(\cdot|s,a)} [(f(s,a) - \mathcal{T}^\pi f'(s,a) + \mathcal{T}^\pi f'(s,a) - (1-\gamma)r(s,a) - \gamma f'(s', \pi(s'))))^2] \\ &= \|f - \mathcal{T}^\pi f'\|_{d^\pi}^2 + \mathcal{L}(\mathcal{T}^\pi f'; f') \\ &\quad + 2\mathbb{E}_{(s,a) \sim d^{\pi,\nu}, s' \sim P(\cdot|s,a)} (f(s,a) - \mathcal{T}^\pi f'(s,a))(\mathcal{T}^\pi f'(s,a) - (1-\gamma)r(s,a) - \gamma f'(s', \pi(s'))) \\ &= \|f - \mathcal{T}^\pi f'\|_{d^{\pi,\nu}}^2 + \mathcal{L}(\mathcal{T}^\pi f'; f') + 2\mathbb{E}[(f(s,a) - \mathcal{T}^\pi f'(s,a))\mathbb{E}[\mathcal{T}^\pi f'(s,a) - (1-\gamma)r(s,a) - \gamma f'(s', \pi(s')) \mid s, a]] \\ &= \|f - \mathcal{T}^\pi f'\|_{d^{\pi,\nu}}^2 + \mathcal{L}(\mathcal{T}^\pi f'; f'). \quad (\text{since } \mathbb{E}[(1-\gamma)r(s,a) - \gamma f'(s', \pi(s')) \mid s, a] = \mathcal{T}^\pi f'(s, a)) \end{aligned}$$

■

This lemma has a particularly useful corollary for approximate minimizers of the loss. For stating the result, let us define the notation $\mathcal{T}_\mathcal{F}^\pi f' = \operatorname{argmin}_{f \in \mathcal{F}} \mathcal{L}(f; f')$ and the corresponding empirical counterparts as well. Then we have the following result.

Corollary 5.5. If f is an ϵ suboptimal minimizer of \mathcal{L} over \mathcal{F} , that is, $\mathcal{L}(f; f') \leq \mathcal{L}(\mathcal{T}_\mathcal{F}^\pi f'; f) + \epsilon$, then $\|f - \mathcal{T}^\pi f'\|_{d^{\pi,\nu}}^2 \leq \|\mathcal{T}_\mathcal{F}^\pi f' - \mathcal{T}^\pi f'\|_{d^{\pi,\nu}}^2 + \epsilon$.

Proof: The ϵ -optimality of f implies that

$$\begin{aligned} \|f - \mathcal{T}^\pi f'\|_{d^{\pi,\nu}}^2 &= \mathcal{L}(f; f') - \mathcal{L}(\mathcal{T}^\pi f'; f') \\ &\leq \mathcal{L}(\mathcal{T}_\mathcal{F}^\pi f'; f') + \epsilon - \mathcal{L}(\mathcal{T}^\pi f'; f') \\ &= \|\mathcal{T}_\mathcal{F}^\pi f' - \mathcal{T}^\pi f'\|_{d^{\pi,\nu}}^2 + \epsilon, \end{aligned}$$

where the first inequality uses the approximate optimality of f and $\mathcal{T}_\mathcal{F}^\pi f' \in \mathcal{F}$ while the second uses Lemma 5.4 with $f = \mathcal{T}_\mathcal{F}^\pi f'$. ■

The corollary has a rather intuitive interpretation. If f approximately minimizes \mathcal{L} , then it is also close to $\mathcal{T}^\pi f'$ up to an amount which depends on the suboptimality as well as the distance of $\mathcal{T}_\mathcal{F}^\pi f'$ to \mathcal{F} . The second term can be thought of as an approximation error, and is always zero, for example, when the function class is fully tabular.

Since f_k minimizes the empirical loss $\widehat{\mathcal{L}}(f; f_{k-1})$, we expect it to approximately minimize the population version $\mathcal{L}(f; f_{k-1})$ by invoking concentration arguments. If we can use this intuition to quantify ϵ , then we can plug the result of Corollary 5.5 into that of Lemma 5.3 to quantify the convergence of f_i to Q^π . We give a lemma that gives a bound on ϵ next. We give the simplest version of this argument assuming that the N samples we use to construct $\widehat{\mathcal{L}}$ are drawn i.i.d., but similar arguments continue to hold if the samples are drawn according to a suitably regular Markov chain, as is the case when they are consecutive triples from a trajectory. For the ease of presentation, we will also restrict \mathcal{F} to a large, but finite class so that we can control statistical deviations uniformly over all functions in \mathcal{F} using a union bound.

Lemma 5.6. For the sequence f_k , $k = 0, 1, \dots, \infty$ as defined in Equation 5.3, with probability at least $1 - \delta$, we have

$$\mathcal{L}(f_k; f_{k-1}) - \mathcal{L}(\mathcal{T}_\mathcal{F}^\pi f_{k-1}; f_{k-1}) \leq 2\|\mathcal{T}_\mathcal{F}^\pi f_{k-1} - \mathcal{T}^\pi f_{k-1}\|_{d^{\pi,\nu}}^2 + \frac{19}{N} \log \frac{|\mathcal{F}|}{\delta}.$$

Proof: For $i = 1, 2, \dots, N$, and given two functions f, f' , let us define

$$X_i = (f(s_i, a_i) - (1 - \gamma)r(s_i, a_i) - \gamma f'(s'_i, \pi(s'_i)))^2 - (\mathcal{T}_{\mathcal{F}} f'(s_i, a_i) - (1 - \gamma)r(s_i, a_i) - \gamma f'(s'_i, \pi(s'_i)))^2.$$

It is easily checked that

$$\widehat{\mathcal{L}}(f; f') - \widehat{\mathcal{L}}(\mathcal{T}_{\mathcal{F}} f'; f') = \frac{1}{N} \sum_{i=1}^N X_i.$$

Furthermore, the random variables X_i are i.i.d. by assumption, bounded by 1 in absolute value and the variance is bounded by

$$\begin{aligned} \mathbb{E}[X_i^2] &= \mathbb{E}[(f(s_i, a_i) - \mathcal{T}_{\mathcal{F}}^{\pi} f'(s_i, a_i))^2 (f(s_i, a_i) + \mathcal{T}_{\mathcal{F}}^{\pi} f'(s_i, a_i) - 2(1 - \gamma)r(s_i, a_i) - 2\gamma f'(s'_i, \pi(s'_i)))^2] \\ &\leq 4\mathbb{E}[(f(s_i, a_i) - \mathcal{T}_{\mathcal{F}}^{\pi} f'(s_i, a_i))^2] = 4\|f - \mathcal{T}_{\mathcal{F}}^{\pi} f'\|_{d^{\pi, \nu}}^2 \\ &\leq 8(\|f - \mathcal{T}^{\pi} f'\|_{d^{\pi, \nu}}^2 + \|\mathcal{T}_{\mathcal{F}}^{\pi} f' - \mathcal{T}^{\pi} f'\|_{d^{\pi, \nu}}^2) \quad (\text{Cauchy-Schwarz inequality}) \\ &= 8(\mathcal{L}(f; f') - \mathcal{L}(\mathcal{T}^{\pi} f'; f') + \|\mathcal{T}_{\mathcal{F}}^{\pi} f' - \mathcal{T}^{\pi} f'\|_{d^{\pi, \nu}}^2) \quad (\text{Lemma 5.4}) \\ &= 8(\mathcal{L}(f; f') - \mathcal{L}(\mathcal{T}_{\mathcal{F}}^{\pi} f'; f') + 2\|\mathcal{T}_{\mathcal{F}}^{\pi} f' - \mathcal{T}^{\pi} f'\|_{d^{\pi, \nu}}^2). \quad (\text{Lemma 5.4 with } f = \mathcal{T}_{\mathcal{F}}^{\pi} f') \end{aligned}$$

Since $\mathbb{E}[X_i] = \mathcal{L}(f; f') - \mathcal{L}(\mathcal{T}_{\mathcal{F}}^{\pi} f'; f') \geq 0$, invoking Bernstein's inequality yields that with probability $1 - \delta$, simultaneously for all pairs $f, f' \in \mathcal{F}$

$$\begin{aligned} &\mathcal{L}(f; f') - \mathcal{L}(\mathcal{T}_{\mathcal{F}}^{\pi} f'; f') - \widehat{\mathcal{L}}(f; f') + \widehat{\mathcal{L}}(\mathcal{T}_{\mathcal{F}}^{\pi} f'; f') \\ &\leq \sqrt{\frac{16}{N}(\mathcal{L}(f; f') - \mathcal{L}(\mathcal{T}_{\mathcal{F}}^{\pi} f'; f') + 2\|\mathcal{T}_{\mathcal{F}}^{\pi} f' - \mathcal{T}^{\pi} f'\|_{d^{\pi, \nu}}^2) \log \frac{|\mathcal{F}|^2}{\delta}} + \frac{2}{3N} \log \frac{|\mathcal{F}|^2}{\delta} \\ &\leq \frac{1}{2}(\mathcal{L}(f; f') - \mathcal{L}(\mathcal{T}_{\mathcal{F}}^{\pi} f'; f') + 2\|\mathcal{T}_{\mathcal{F}}^{\pi} f' - \mathcal{T}^{\pi} f'\|_{d^{\pi, \nu}}^2) + \frac{8}{N} \log \frac{|\mathcal{F}|}{\delta} + \frac{4}{3N} \log \frac{|\mathcal{F}|}{\delta}, \end{aligned}$$

where the second inequality uses $\sqrt{ab} \leq (a + b)/2$ from Cauchy-Schwarz inequality. Rearranging terms, we see that with probability at least $1 - \delta$

$$\mathcal{L}(f; f') - \mathcal{L}(\mathcal{T}_{\mathcal{F}}^{\pi} f'; f') \leq 2(\widehat{\mathcal{L}}(f; f') - \widehat{\mathcal{L}}(\mathcal{T}_{\mathcal{F}}^{\pi} f'; f')) + 2\|\mathcal{T}_{\mathcal{F}}^{\pi} f' - \mathcal{T}^{\pi} f'\|_{d^{\pi, \nu}}^2 + \frac{19}{N} \log \frac{|\mathcal{F}|}{\delta}.$$

Choosing $f' = f_{k-1}$ and $f = f_k$ so that f_k minimizes $\widehat{\mathcal{L}}(f; f_{k-1})$ in the inequality above, we see that with probability at least $1 - \delta$,

$$\mathcal{L}(f_k; f_{k-1}) - \mathcal{L}(\mathcal{T}_{\mathcal{F}}^{\pi} f_{k-1}; f_{k-1}) \leq 2\|\mathcal{T}_{\mathcal{F}}^{\pi} f_{k-1} - \mathcal{T}^{\pi} f_{k-1}\|_{d^{\pi, \nu}}^2 + \frac{19}{N} \log \frac{|\mathcal{F}|}{\delta}.$$

In order to obtain a uniform bound across all iterations, we note that the bound here holds for all pairs f, f' so that across all the iterations, the failure probability is at most δ . \blacksquare

Having obtained a bound on the statistical errors, we are now ready to prove the theorem.

Proof:[Proof of Theorem 5.2] The proof largely combines the results of our lemmas so far. We condition on the high probability event in Lemma 5.6 since it holds with probability at least $1 - \delta$ throughout the run of the algorithm. Under this event, combining Lemma 5.6 along with Corollary 5.5 yields that

$$\|f_k - \mathcal{T}^{\pi} f_{k-1}\|_{d^{\pi, \nu}}^2 \leq 3\|\mathcal{T}_{\mathcal{F}}^{\pi} f_{k-1} - \mathcal{T}^{\pi} f_{k-1}\|_{d^{\pi, \nu}}^2 + \frac{19}{N} \log \frac{|\mathcal{F}|}{\delta}$$

Using $\sqrt{a + b} \leq \sqrt{a} + \sqrt{b}$, we see that for all $i = 1, 2, \dots, k$

$$\|f_i - \mathcal{T}^{\pi} f_{i-1}\|_{d^{\pi, \nu}} \leq \sqrt{3} \max_{f \in \mathcal{F}} \|\mathcal{T}_{\mathcal{F}}^{\pi} f - \mathcal{T}^{\pi} f\|_{d^{\pi, \nu}} + \sqrt{\frac{19}{N} \log \frac{|\mathcal{F}|}{\delta}} := \phi.$$

Plugging this result into the conclusion of Lemma 5.3, we obtain

$$\begin{aligned} \|f_k - Q^\pi\|_{d^{\pi,\nu}} &\leq \sqrt{\|\rho\|_\infty \|1/\rho\|_\infty} \gamma^{k/2} \|f_0 - Q^\pi\|_{d^{\pi,\nu}} + \sqrt{\|\rho\|_\infty \|1/\rho\|_\infty} \sum_{i=1}^k \gamma^{(k-i)/2} \phi \\ &\leq \sqrt{\|\rho\|_\infty \|1/\rho\|_\infty} \left(\gamma^{k/2} \|f_0 - Q^\pi\|_{d^{\pi,\nu}} + \frac{\phi}{1 - \sqrt{\gamma}} \right) \leq \sqrt{\|\rho\|_\infty \|1/\rho\|_\infty} \left(\gamma^{k/2} \|f_0 - Q^\pi\|_{d^{\pi,\nu}} + \frac{2\phi}{1 - \gamma} \right). \end{aligned}$$

Recalling the value of ϕ completes the proof of the theorem. \blacksquare

5.2 Approximate Policy Improvement

Evaluating the function Q^π for a given policy π is often an intermediate step in the larger algorithm, and we now visit the question of how we might leverage the policy evaluation result of the previous section. We start with a basic result, which resembles classical bounds for approximate policy iteration, but has some differences due to our use of Q functions rather than V functions. These bounds are apt for tabular function classes and small state spaces, as they rely on a uniformly good approximation to Q^π at all states and actions. After seeing the underlying reasons that give rise to such limitations, we will consider a better policy improvement operator described in the Conservative Policy Iteration [Kakade and Langford, 2002], and discuss using the results of our previous section as a plug-in to that algorithm.

5.2.1 Greedy policy improvement with ℓ_∞ approximation

The setting of this section closely resembles that of the Policy Iteration algorithm in Chapter 1.2. We consider the following iterative procedure, which starts with an initial policy π_0 and updates for $i = 0, 1, 2, \dots$:

$$\text{Obtain } \hat{Q}_i \text{ such that } \|\hat{Q}_i - Q^{\pi_i}\|_\infty \leq \epsilon. \text{ Set } \pi_{i+1}(s) = \operatorname{argmax}_{a \in \mathcal{A}} \hat{Q}_i(s, a) \text{ for all } s \in \mathcal{S}. \quad (5.4)$$

We do not discuss how to obtain such an approximation \hat{Q} for Q^{π_i} here, but instead try to understand the reason why need such a uniformly good approximation, and how we might relax the condition. Intuitively, the greedy policy improvement in Equation 5.4 can induce a policy π_{i+1} which is dramatically different from π_i in its state visitation distribution. If we only had approximation guarantees between \hat{Q} and Q^π on d^{π_i} , we can already guess that it would not be sufficient as the approximation might be arbitrarily worse on $d^{\pi_{i+1}}$, and if this happens, then π_{i+1} is being defined rather arbitrarily on the states which are not visited under d^{π_i} . That is, we are effectively encountering another instance of the distribution mismatch which we also saw in the policy gradient analysis earlier.

In order to quantify these intuitions, we will establish the following result regarding the iteration (5.4).

Theorem 5.7. *The approximate policy iteration algorithm described in Algorithm 5.4 satisfies for all $k = 0, 1, 2, \dots$*

$$\|Q^* - Q^{\pi_k}\|_\infty \leq \gamma^k \|Q^* - Q^{\pi_0}\|_\infty + \frac{2\epsilon\gamma}{(1 - \gamma)^2}.$$

Proof: As in the case of Theorem 1.9, the main step is to show that the distance between Q^* and Q^{π_i} contracts when we do one step of policy improvement. However, we now need to account for the approximation in \hat{Q}_i in order to show that the contraction is not offset by the errors in our estimates. We start with some simple definitions and properties. For a policy π , let \hat{Q} be a function satisfying $\|\hat{Q} - Q^\pi\|_\infty \leq \epsilon$, and let

$$\pi^+(s) = \operatorname{argmax}_{a \in \mathcal{A}} Q^\pi(s, a) \quad \text{and} \quad \tilde{\pi}(s) = \operatorname{argmax}_{a \in \mathcal{A}} \hat{Q}(s, a).$$

Based on the definitions of \hat{Q} and $\tilde{\pi}$, it is easily verified that for all states s , $A^\pi(s, \tilde{\pi}(s)) \geq -2\epsilon$. That is, $\tilde{\pi}(s)$ is a 2ϵ -approximate greedy action with respect to $Q^\pi(s, a)$. Based on this, the performance difference lemma (Lemma 4.5) gives that

$$\begin{aligned} Q^{\tilde{\pi}}(s, a) - Q^\pi(s, a) &= \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) (V^{\tilde{\pi}}(s') - V^\pi(s')) \\ &= \frac{\gamma}{1 - \gamma} \sum_{s' \in \mathcal{S}} P(s'|s, a) \sum_{s'' \in \mathcal{S}} d_{s'}^{\tilde{\pi}} A^\pi(s'', \tilde{\pi}(s'')) \\ &\geq -\frac{2\epsilon\gamma}{1 - \gamma}. \end{aligned}$$

In the middle equality, we recall the notation d_s^π for the distribution over states induced under π starting from the state s . Now we can lower bound the one-step policy improvement. For this, recall the notation \mathcal{T} for the Bellman operator which maps $\mathcal{T}Q = (1 - \gamma)r + \gamma PV_Q$, where $V_Q(s) = \max_{a \in \mathcal{A}} Q(s, a)$

$$\begin{aligned} Q^{\tilde{\pi}}(s, a) - \mathcal{T}Q^\pi(s, a) &= \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) (V^{\tilde{\pi}}(s') - \max_{a' \in \mathcal{A}} Q^\pi(s', a')) \\ &\geq \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) (V^{\tilde{\pi}}(s') - Q^\pi(s', \tilde{\pi}(s'))) - 2\epsilon \\ &= -2\epsilon\gamma + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) (Q^{\tilde{\pi}}(s', \tilde{\pi}(s')) - Q^\pi(s', \tilde{\pi}(s'))) \\ &\geq -2\epsilon\gamma - \frac{2\epsilon\gamma^2}{1 - \gamma} = -\frac{2\epsilon\gamma}{1 - \gamma}, \end{aligned}$$

where the first inequality uses the observation that $A^\pi(s', \tilde{\pi}(s')) \geq -2\epsilon$, while second inequality from our earlier bound on the difference between $Q^{\tilde{\pi}}$ and Q^π .

Now following the same logic as in the convergence analysis of policy iteration, we see that

$$\begin{aligned} \|Q^* - Q^{\tilde{\pi}}\|_\infty &\leq \|Q^* - \mathcal{T}Q^\pi\|_\infty + \frac{2\epsilon\gamma}{1 - \gamma} \\ &= \|\mathcal{T}Q^* - \mathcal{T}Q^\pi\|_\infty + \frac{2\epsilon\gamma}{1 - \gamma} \\ &\leq \gamma \|Q^* - Q^\pi\|_\infty + \frac{2\epsilon\gamma}{1 - \gamma}. \end{aligned}$$

Applying this inequality between every pair of successive iterates π_i and π_{i+1} and summing the resulting geometric series yields the theorem. \blacksquare

While similar results have been shown before for approximate policy iteration where V -function estimates are used instead of the Q -function estimates (see e.g. Chapter 6.2 in [Bertsekas and Tsitsiklis, 1996]), we find this version more natural as greedy maximization using V functions requires knowledge of the MDP dynamics.

Inspecting the proof closely also show the challenges with relaxing the ℓ_∞ assumption on the closeness between \hat{Q} and Q^π . Note that the performance difference lemma invoked in the first part of the proof requires that the estimates \hat{Q} be accurate on the state distribution induced by $\tilde{\pi}$. This is not feasible using techniques from the previous section, as $\tilde{\pi}$ is a function of \hat{Q} itself, and could pick out a pathological state distribution. One natural intuition to fix this difficulty is by algorithmically controlling the distribution mismatch between successive policies, which we will see in our next approach.

5.2.2 Conservative Policy Iteration

As the name suggests, we will now describe a more conservative version of the policy iteration algorithm, which shifts the next policy away from the current policy with a small step size to prevent drastic shifts in successive state distributions. The algorithm and analysis is adapted from the original result of Kakade and Langford [2002].

As before, the algorithm will iteratively generate a sequence of policies π_i , and we will assume that each time we can find a Q -value estimate for π_i , denoted as \hat{Q}_i which satisfies

$$\left\| \hat{Q}_i - Q^{\pi_i} \right\|_{d^{\pi, U}} \leq \epsilon, \quad (5.5)$$

where U denotes the uniform distribution over actions. Given such a guarantee, it is easily seen that for any randomized policy ν we have

$$\left\| \hat{Q}_i - Q^{\pi_i} \right\|_{d^{\pi, \nu}} \leq \sqrt{|\mathcal{A}|} \epsilon. \quad (5.6)$$

Given access to such estimates of Q -value functions, we now describe the Conservative Policy Iteration (CPI) algorithm. For the algorithm, we use $\pi_\alpha = (1 - \alpha)\pi + \alpha\pi'$ to refer to a randomized policy which chooses an action according to π with probability $1 - \alpha$ and according to π' with probability α .

Algorithm 2 Conservative Policy Iteration (CPI)

Input: Initial policy π_0 , accuracy parameter ϵ .

- 1: **for** $i = 0, 1, 2 \dots$ **do**
 - 2: Obtain a Q value approximation \hat{Q}_i satisfying (5.5) with parameter $\epsilon/2\sqrt{|\mathcal{A}|}$.
 - 3: Define $\tilde{\pi}_i(s) = \operatorname{argmax}_{a \in \mathcal{A}} \hat{Q}_i(s, a)$, and let $\hat{A}_i = \mathbb{E}_{s \in d^{\pi_i}} \hat{Q}_i(s, \tilde{\pi}_i(s)) - \hat{Q}_i(s, \pi_i(s))$.
 - 4: If $\hat{A}_i \leq 2\epsilon$, return π_i .
 - 5: Update $\pi_{i+1}(a | s) = (1 - \alpha_i)\pi_i(a | s) + \alpha_i\tilde{\pi}_i(a | s)$, for $\alpha_i = (\hat{A}_i - \epsilon)(1 - \gamma)/4$.
 - 6: **end for**
-

The main intuition behind the algorithm is that the stepsize α controls the difference between state distributions of π_i and π_{i+1} . In particular, with high probability, the trajectories generated from π_i and π_{i+1} differ in only one action. The following result formalizes this iteration at one step of the algorithm.

Theorem 5.8 (One step improvement in CPI). *Let $\pi = \pi_i$, $\tilde{\pi} = \tilde{\pi}_i$ and $\nu = \pi_{i+1}$ be the successive policies at one iteration of Algorithm 2 and $A = \mathbb{E}_{s \sim d^\pi} A^\pi(s, \tilde{\pi}(s))$. Then*

$$V^\nu - V^\pi \geq \frac{\alpha}{1 - \gamma} \left(A - \frac{2\alpha\gamma}{1 - \gamma(1 - \alpha)} \right).$$

Proof: By performance difference lemma, we know that

$$V^\nu - V^\pi = \sum_{t=0}^{\infty} \gamma^t \sum_{s \in \mathcal{S}} \mathbb{P}^\nu(s_t = s) \sum_a \nu(a | s) A^\pi(s, a).$$

Now on a time t , with probability $(1 - \alpha)^t$, all the actions have been chosen as per π on the previous time steps. If any actions were chosen by ν with the remaining probability $1 - (1 - \alpha)^t$, we can simply lower bound the advantage function by its largest value on that event. Doing so, we see that

$$\begin{aligned} V^\nu - V^\pi &\geq \sum_{t=0}^{\infty} \gamma^t \left((1 - \alpha)^t \sum_{s \in \mathcal{S}} \mathbb{P}^\pi(s_t = s) \sum_a \nu(a | s) A^\pi(s, a) - (1 - (1 - \alpha)^t) \max_s \sum_a \nu(a | s) A^\pi(s, a) \right) \\ &\geq \sum_{t=0}^{\infty} \gamma^t \left(\sum_{s \in \mathcal{S}} \mathbb{P}^\pi(s_t = s) \sum_a \nu(a | s) A^\pi(s, a) - 2(1 - (1 - \alpha)^t) \max_s \sum_a \nu(a | s) A^\pi(s, a) \right). \end{aligned}$$

By definition of ν , for any state s , we observe that $\sum_a \nu(a|s) A^\pi(s, a) = \alpha A^\pi(s, \tilde{\pi}(s)) \leq \alpha$. Summing the geometric series, we see that

$$V^\nu - V^\pi \geq \frac{\alpha}{1-\gamma} \mathbb{E}_{s \sim d^\pi} A^\pi(s, \tilde{\pi}(s)) - 2\alpha \left(\frac{1}{1-\gamma} - \frac{1}{1-\gamma(1-\alpha)} \right).$$

Simplifying terms yields the result. \blacksquare

A crucial aspect of this theorem is that it always requires a good estimate of the advantage function of π only under d^π , which is the main upshot of the conservative updates. Our next result bounds the gap between the quantities \hat{A}_i and the true advantages of $\tilde{\pi}_i$ over π_i at each iteration.

Lemma 5.9. *Using the notation of Theorem 5.8, under the conditions of Equation 5.6, we have $\mathbb{E}_{s \sim d^\pi} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) \leq \hat{A}_i + 2\epsilon\sqrt{|\mathcal{A}|}$.*

Proof: Note that by Jensen's inequality, we have $\mathbb{E}_{s \sim d^{\pi, \nu}} |\hat{Q}(s, a) - Q^\pi(s, a)| \leq \|\hat{Q} - Q^\pi\|_{d^{\pi, \nu}}$, where we have dropped the subscripts involving i as we focus on a specific iteration. Denote $\pi^+(s) = \operatorname{argmax}_{a \in \mathcal{A}} Q^\pi(s, a)$. Now we have

$$\begin{aligned} \mathbb{E}_{s, a \sim d^\pi} A^\pi(s, \pi^+(s)) &= \mathbb{E}_{s \sim d^\pi} \left[Q^\pi(s, \pi^+(s)) - \hat{Q}(s, \pi^+(s)) + \hat{Q}(s, \pi^+(s)) - \hat{Q}(s, \tilde{\pi}(s)) + \hat{Q}(s, \tilde{\pi}(s)) \right. \\ &\quad \left. - \hat{Q}(s, \pi(s)) + \hat{Q}(s, \pi(s)) - Q^\pi(s, \pi(s)) \right] \\ &\leq \|\hat{Q} - Q^\pi\|_{d^{\pi, \pi^+}} + \|\hat{Q} - Q^\pi\|_{d^{\pi, \pi}} + \mathbb{E}_{s \sim d^\pi} [\hat{Q}(s, \tilde{\pi}(s)) - \hat{Q}(s, \pi(s))], \end{aligned}$$

where we have used the inequality $\hat{Q}(s, \pi^+(s)) \leq \hat{Q}(s, \tilde{\pi}(s))$ by the definition of $\tilde{\pi}$. \blacksquare

Putting these two results together, we obtain the following useful corollary which gives a lower bound on the per-step improvement of the CPI algorithm for our choice of α_i .

Corollary 5.10. *At each iteration of Algorithm 2, we have $V^{\pi_{i+1}} - V^{\pi_i} \geq (\hat{A}_i - \epsilon)^2/8$.*

Proof: From Theorem 5.8, we have

$$\begin{aligned} V^{\pi_{i+1}} - V^{\pi_i} &\geq \frac{\alpha}{1-\gamma} \mathbb{E}_{s \sim d^\pi} A^{\pi_i}(s, \tilde{\pi}_i(s)) - \frac{2\alpha^2}{(1-\gamma)(1-\gamma(1-\alpha))} \\ &\geq \frac{\alpha}{1-\gamma} (\hat{A}_i - \epsilon) - \frac{2\alpha^2}{(1-\gamma)(1-\gamma(1-\alpha))} \quad (\text{Lemma 5.9 and Line 2 in Algorithm 2}) \\ &\geq \frac{\alpha}{1-\gamma} \left(\hat{A}_i - \epsilon - \frac{2\alpha}{1-\gamma} \right) \\ &\geq \frac{(\hat{A}_i - \epsilon)^2}{8} \quad (\text{Using } \alpha_i = (\hat{A}_i - \epsilon)(1-\gamma)/4) \end{aligned}$$

Putting these results together, we obtain the following overall convergence guarantee for the CPI algorithm.

Theorem 5.11 (Local optimality of CPI). *Algorithm 2 terminates in at most $8/\epsilon^2$ steps and outputs a policy π satisfying $\mathbb{E}_{s \sim d^\pi} \max_a A^\pi(s, a) \leq 3\epsilon$.*

Proof: We can now prove the theorem. Note that each at each iteration, CPI either terminates or ensures that $\hat{A}_i \geq 2\epsilon$. By Corollary 5.10, this implies that if the algorithm has performed k iterations, we must have

$$V^{\pi_k} - V^{\pi_0} \geq \frac{k\epsilon^2}{8}.$$

Since all policies take values in $[0, 1]$, this means that the algorithm must terminate in at most $8\epsilon^2$ iterations. Furthermore, at termination, we have $\hat{A}_k \leq 2\epsilon$. Combined with Lemma 5.9 and the accuracy of \hat{Q}_i in Line 2 in Algorithm 2 completes the proof. ■

Theorem 5.11 can be viewed as a local optimality guarantee in a sense. It shows that when CPI terminates, no improvements are possible to the returned policy π by updating it in the direction of one-step advantages. However, this does not necessarily imply that the value of π is close to V^* . Indeed if the MDP has a chain like structure which we saw in the policy gradient lecture, and π advances towards the goal state with a small probability only, then it is easy to convince ourselves that no local improvements might be possible even when the policy is significantly suboptimal. However, similar to the policy gradient analysis, we can turn this local guarantee into a global one when the resulting policy's state distribution is sufficiently exploratory. We formalize this intuition next.

Theorem 5.12 (Global optimality of CPI). *Given any policy π satisfying $\mathbb{E}_{s \sim d^\pi} \max_{a \in \mathcal{A}} A^\pi(s, a) \leq \epsilon$, we have*

$$V^* - V^\pi \leq \frac{\epsilon}{1 - \gamma} \left\| \frac{d^{\pi^*}}{d^\pi} \right\|_\infty \leq \frac{\epsilon}{(1 - \gamma)^2} \left\| \frac{d^{\pi^*}}{d_0} \right\|_\infty,$$

where the ratio d^{π^*}/d^π represents elementwise division.

Proof: The proof is essentially contained in that of Theorem ?? . By the performance difference lemma,

$$\begin{aligned} V^* - V^\pi &= \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^{\pi^*}} A^\pi(s, \pi^*(s)) \\ &\leq \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^{\pi^*}} \max_{a \in \mathcal{A}} A^\pi(s, a) \\ &\leq \frac{1}{1 - \gamma} \left\| \frac{d^{\pi^*}}{d^\pi} \right\|_\infty \mathbb{E}_{s \sim d^\pi} \max_{a \in \mathcal{A}} A^\pi(s, a). \end{aligned}$$

Consequently, if the initial distribution, or the visitation distribution of the final policy output by CPI is sufficiently exploratory relative to π^* , then we converge to a globally optimal policy.

It is informative to contrast CPI and policy gradient algorithms due to the similarity of their guarantees. Both provide local optimality guarantees. For CPI, the local optimality always holds, while for policy gradients it requires a smooth value function as a function of the policy parameters. If the distribution mismatch between an optimal policy and the output of the algorithm is not too large, then both algorithms further yield a near optimal policy. The similarities are not so surprising. Both algorithms operate by making local improvements to the current policy at each iteration, by inspecting its advantage function. The changes made to the policy are controlled using a stepsize parameter in both the approaches. It is the actual mechanism of the improvement which differs in the two cases. Policy gradients assume that the policy's reward is a differentiable function of the parameters, and hence make local improvements through gradient ascent. The differentiability is certainly an assumption and does not necessarily hold for all policy classes. An easy example is when the policy itself is not an easily differentiable function of its parameters. For instance, if the policy is parametrized by regression trees, then performing gradient updates can be challenging.

In CPI, on the other hand, the basic computational primitive required on the policy class is the ability to maximize the advantage function relative to the current policy. Notice that Algorithm 2 does not necessarily restrict to a policy class, such as a set of parametrized policies as in policy gradients. But in practice, we typically seek to find a policy from a restricted class which yields the largest improvement on the current policy. In this case, we can no longer set $\tilde{\pi}_i(s) = \max_{a \in \mathcal{A}} \hat{Q}_i(s, a)$, but instead seek $\tilde{\pi}_i(s) = \operatorname{argmax}_{\pi \in \Pi} \mathbb{E}_{s \sim d^{\pi_i}} \hat{Q}_i(s, \pi(s))$. The latter can be viewed as a weighted classification problem where the policies π are seen as probabilistic classifiers mapping states to actions, and the cost of prediction an action a on a state s is $-\hat{Q}_i(s, a)$. If we collect a dataset of (s, a) samples with $s \sim d^{\pi_i}$ and uniformly random actions, then it is easy to show that the best improvement policy can be obtained by easy

modifications of existing classification or regression algorithms. This property makes CPI extremely attractive. Any policy class over which efficient supervised learning algorithms exist can be adapted to reinforcement learning with performance guarantees.

A second important difference between CPI and policy gradients is in the notion of locality. Policy gradient updates are local in the parameter space, and we hope that this makes small enough changes to the state distribution that the new policy is indeed an improvement on the older one (for instance, when we invoke the performance difference lemma between successive iterates). While this is always true in expectation for correctly chosen stepsizes based on properties of stochastic gradient ascent on smooth functions, the variance of the algorithm and lack of robustness to suboptimal stepsizes can make the algorithm somewhat finicky. Indeed, there are a host of techniques in the literature to both lower the variance (through control variates) and explicitly control the state distribution mismatch between successive iterates of policy gradients (through trust region techniques). On the other hand, CPI explicitly controls the amount of perturbation to the state distribution by carefully mixing policies in a manner which does not drastically alter the trajectories with high probability. Indeed, this insight is central to the proof of CPI, and has been instrumental in several follow-ups, both in the direct policy improvement as well as policy gradient literature.

Chapter 6

Strategic Exploration in RL with rich observations

Strategic Exploration in RL with rich observations

Aleksh Agarwal, Nan Jiang, Sham M. Kakade

Chapter 6

Our previous lectures on exploration in RL focused on the RMAX algorithm designed for the tabular representation, in that we expect each state to be visited sufficiently often that we can maintain separate counters for the number of visits to each state, action pair, and learn the optimal behavior separately in each state using the approximate dynamics model derived from these counters. However, in most natural examples of RL (such as the three examples from Lecture 1), the number of states can be rather large and infinite, and it is impractical to expect sufficiently many visits to each state for building an approximate dynamics model, as in the RMAX algorithm. Furthermore, it is often conceptually flawed in such problems to think of each observation received by the agent about its current environment as a distinct physical state. For instance, several distinct utterances in the conversational agent example reveal the same underlying intent. Similarly, in navigation tasks, the agent's precise observation about its current location through a camera or LIDAR sensor might differ even when the positions are equivalent in terms of the task at hand. To better make this distinction between the semantics of a state in tabular RL, and the nature of observations in typical applications, we will reuse the term context to refer to the object which an agent uses to form its policies and value functions in this lecture (we will formalize this shortly), due to the semantic similarity with the usage of this term in contextual bandits.

6.1 Problem setting

For this chapter, we consider finite-horizon episodic Contextual Decision Processes (CDPs), following the setup of Jiang et al. [2017]. Let \mathcal{X} denote a context space. Then a CDP is described as a tuple $(\mathcal{X}, \mathcal{A}, P, d_0, r, H)$, where $P : \mathcal{X} \times \mathcal{A} \rightarrow \Delta(\mathcal{X})$ is a transition operator in the context space, and $d_0 \in \Delta(\mathcal{X})$ is the distribution over the initial contexts. H denotes the horizon of the problem. We consider finite-horizon settings, where each trajectory consists of precisely H steps, following which, the agent is reset according to the initial context distribution. Formally, a trajectory $\tau = (x_0, a_0, r_0, x_1, \dots, x_{H-1}, a_{H-1}, r_{H-1}, x_H)$, with $x_0 \sim d_0$ and $x_h \sim P(\cdot | x_{h-1}, a_{h-1})$ for $h \in \{1, 2, \dots, H\}$.¹

A (randomized) policy is now described as a mapping from contexts to (distributions over) actions. In general, the optimal policy in a finite horizon CDP is non-stationary, that is, it can take different actions in the same context, depending on the step along the trajectory at which it is encountered. To avoid indexing all our policies and value functions by h , we instead assume that each context contains as a part of it, the level at which it is encountered. Formally, this means that there is a partition $\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1 \cup \dots \cup \mathcal{X}_H$ such that $x_h \in \mathcal{X}_h$ for any trajectory τ . This assumption (often known as the acyclic or layered structure) allows us to focus on stationary policies and value functions.

The goal of an agent is to maximize the cumulative expected reward it obtains over H steps. We make the following boundedness assumption on the rewards.

Assumption 6.1. Almost surely, for any tranjectory τ and step h , $0 \leq r_h \leq 1$. Additionally, $0 \leq \sum_{h=0}^{H-1} r_h \leq 1$ almost surely for any trajectory τ .

While the first part of the assumption is the standard boundedness assumption we have made throughout, the second assumes that the trajectory level rewards are also bounded by 1, instead of H , which is helpful for capturing sparse-reward goal-directed problems with rewards only at one point in a successful trajectory. While normalization of the

¹This definition considers the contexts to be Markovian in that the next context and rewards only depend on the current context and action, independent of the history. The original definition in Jiang et al. [2017] allows non-Markovian contexts as well, but we omit this generalization for ease of presentation.

trajectory level reward also keeps the net reward bounded, this makes the total reward only scale as $1/H$ if the rewards are sparse along the trajectory.

6.2 Value-function approximation

Now that we have set up the reinforcement learning problem as a CDP, we need a solution concept which will lead to generalization across similar contexts, since finding the best possible policy as a function of the context has a prohibitive sample complexity. Stated another way, we know from Chapter 2 that finding a near-optimal policy requires $\Omega(|\mathcal{X}||\mathcal{A}|)$ samples when we have $|\mathcal{X}|$ unique states. So without limiting the solution concept, our sample complexity will scale with $|\mathcal{X}|$ which we seek to avoid. Taking a cue from value-function approximation in Chapter 5, we can consider access to a class of functions $\mathcal{F} \subseteq \{\mathcal{X} \times \mathcal{A} \rightarrow [0, 1]\}$, where each function $f \in \mathcal{F}$ maps a context, action pair to $[0, 1]$. We also assume that $f(x_H, a) = 0$ for all $a \in \mathcal{A}$ and $f \in \mathcal{F}$, since there are no future rewards after the last step.

Since we want to learn a near-optimal behavior, we seek to approximate the Q -value function of the optimal policy, namely Q^* using $f \in \mathcal{F}$. To this end, we start with a simplifying assumption that Q^* lies in \mathcal{F} . In practice, this can be weakened to having a good approximation for Q^* in \mathcal{F} , but we focus on exact containment for the cleanest setting. Formally, we make the following *realizability* assumption.

Assumption 6.2 (Value-function realizability). The function class \mathcal{F} satisfies $Q^* \in \mathcal{F}$.

Armed with this assumption, we may ask whether we can find Q^* using a number of samples which does not scale as $|\mathcal{X}|$, trading it off for a statistical complexity measure for \mathcal{F} such as $\ln |\mathcal{F}|$, as we saw in our previous chapters on value function approximation for a fixed policy as well as for contextual bandits. The next result, adapted from Krishnamurthy et al. [2016] shows that this is not possible.

Theorem 6.3. Fix $H, K \in \mathbb{N}$ with $K \geq 2$ and $\epsilon \in (0, \sqrt{1/8}]$. For any algorithm, there exists a CDP with a horizon of H and K actions, a class of predictors \mathcal{F} with $|\mathcal{F}| = K^H$ and $Q^* \in \mathcal{F}$ and a constant $c > 0$ such that the probability that the algorithm outputs a policy $\hat{\pi}$ with $V(\hat{\pi}) \geq V^* - \epsilon$ after collecting T trajectories from the CDP is at most $2/3$ for all $T \leq cK^H/\epsilon^2$.

In words, the theorem says that for any algorithm, there exists a CDP where it cannot find a good policy in fewer than an exponential number of samples in the planning horizon, even when $Q^* \in \mathcal{F}$. Furthermore, the size of the class \mathcal{F} required for this result is K^H , so that a logarithmic dependence on $|\mathcal{F}|$ will not explain the lower bound.

We give an informal sketch of the proof now. A formal proof will be added later. Informally, we will define a class of CDPs, and show that one of the CDPs in this class will witness the lower bound for any possible algorithm. The CDPs in our family all share identical transition dynamics and context space. The context space $\mathcal{X} = \cup_{h=1}^H [K]^h$, so that each context will correspond to the sequence of actions executed that lead to it in the CDP. The transition dynamics are deterministic with $P(x_h \circ a_h \mid x_h, a_h) = 1$, where $x_h \circ a_h$ refers to concatenating the action a_h to the action sequence denoted by x_h . Each CDP in the family corresponds to a path p of length H , and the reward function is non-zero only in the state x_H . In particular, we set $r_p(p) = \text{Bernoulli}(1/2 + \epsilon)$ and $r_p(x_H) = \text{Bernoulli}(1/2)$ for any other terminal state, in the CDP parameterized by the path p . We define the class \mathcal{F} to similarly consist of functions f_p where

$$f_p(x, a) = \frac{1}{2} + \epsilon \mathbf{1}(x \circ a \text{ is a prefix of } p).$$

Clearly f_p describes Q^* in the CDP parameterized by p . As we have a total of K^H paths with K actions over a horizon of H , we see that $|\mathcal{F}| = K^H$. Furthermore, we have $V^* = 1/2 + \epsilon$ for each CDP in the family by following the path p in the CDP corresponding to p . It is also clear that finding a policy $\hat{\pi}$ such that $V(\hat{\pi}) \geq V^* - \epsilon$ is equivalent to identifying the path p . Thus we can prove the theorem by establishing that no algorithm can successfully identify the

path parameter correctly for each CDP in our family. To do this, we relate our construction to identifying the best arm in a multi-armed bandit problem with K^H arms. This is done by viewing each path as an arm. The reward distribution at the terminal state of that path gives the reward distribution for the corresponding arm. Now we note that the setup of this multi-armed bandit problem is exactly identical to the lower bound instance of multi-armed bandits [Auer et al., 2002]. Since their lower bound scales linearly in the number of actions, we get the stated lower bound.

6.3 Bellman Rank

Having concluded that we cannot find a near optimal policy using a reasonable number of samples with just the realizability assumption, it is clear that additional structural assumptions on the problem are required in order to make progress. We now give one example of such a structure, named Bellman rank, which was introduced by Jiang et al. [2017]. In order to motivate and define this quantity, we need some additional notation. For a function $f \in \mathcal{F}$, let us define $\pi_f(x) = \operatorname{argmax}_{a \in \mathcal{A}} f(x, a)$. For a function f , we define $V_f = \mathbb{E}_{x \sim d_0} f(x, \pi_f(x))$ to be the value predicted by f for its greedy policy in the initial state distribution. For a policy π , function $f \in \mathcal{F}$ and $h \in [H]$, let us also define the *average Bellman error*:

$$\mathcal{E}(f, \pi, h) = \mathbb{E}[f(x_h, a_h) - r_h - f(x_{h+1}, a_{h+1}) \mid a_{0:h-1} \sim \pi, a_{h:h+1} \sim \pi_f]. \quad (6.1)$$

This is called the average Bellman error as it is not the error on an individual context x , but an expected error under the distribution over contexts induced after taking $h - 1$ actions according to π . The error checks the self-consistency of f with its own predicted future, but only for actions chosen by its greedy policy at steps h and $h + 1$. To see why the definition might be natural, we note that the following property of Q^* from the Bellman optimality equations.

Fact 6.4. $\mathcal{E}(Q^*, \pi, h) = 0$, for all policies π and levels h .

The fact holds since Q^* satisfies $Q^*(x, a) = r(x, a) + E[Q^*(x', \pi^*(x'))] \mid x, a$ for each context x , and hence also for any distribution over x induced by a policy. More generally, these errors are extremely useful due to the following lemma.

Lemma 6.5 (Policy loss decomposition). *For any $f : \mathcal{X} \times \mathcal{A} \rightarrow [0, 1]$, we have*

$$V_f - V^{\pi_f} = \sum_{h=0}^{H-1} \mathcal{E}(f, \pi_f, h).$$

The lemma effectively says that finding a policy π_f is equivalent to finding a function f with large V_f , as long as the Bellman errors for f under the context distributions induced by π_f are 0 at each level. Note that the lemma is an equality.

Proof: Expanding the RHS of the lemma, we see that

$$\begin{aligned} \sum_{h=0}^{H-1} \mathcal{E}(f, \pi_f, h) &= \sum_{h=0}^{H-1} \mathbb{E}[f(x_h, a_h) - r_h - f(x_{h+1}, a_{h+1}) \mid a_{0:h-1} \sim \pi_f, a_{h:h+1} \sim \pi_f] \\ &= \sum_{h=0}^{H-1} \mathbb{E}[f(x_h, a_h) - r_h - f(x_{h+1}, a_{h+1}) \mid a_{0:H-1} \sim \pi_f] \\ &= \mathbb{E}[f(x_0, \pi_f(x_0))] - \mathbb{E}[f(x_H, a_H) \mid a_{0:H-1} \sim \pi_f] - \mathbb{E}\left[\sum_{h=0}^{H-1} r_h \mid a_{0:H-1} \sim \pi_f\right] \\ &= \mathbb{E}[f(x_0, \pi_f(x_0))] - V^{\pi_f} = V_f - V^{\pi_f}. \end{aligned}$$

Here the second equality follows since all the actions are being chosen according to π_f in each summand. This completes the proof. \blacksquare

This lemma is powerful, since the initial state distribution is fixed and can be sampled from. The lemma suggests the following optimization problem to find Q^* .

$$\max_{f \in \mathcal{F}} V_f \quad \text{subject to} \quad \mathcal{E}(f, \pi_f, h) = 0 \quad \text{for all } h \in [H - 1]. \quad (6.2)$$

The objective function in this problem is easy to optimize, since V_f for any function f can be easily approximated with enough samples from d_0 . However, the constraints are significantly harder to enforce, since they require evaluating the Bellman error of f on the context distributions induced by π_f . Since each f 's Bellman error is being evaluated on a different policy, it is not clear how we might check these constraints without effectively collecting data with each policy π_f for $f \in \mathcal{F}$, in which case we will have a sample complexity scaling with $O(|\mathcal{F}|)$ instead of $O(\ln |\mathcal{F}|)$.

In order to get around this fact, we make an observation and an assumption. We observe that the set of equations enforced in the optimization problem (6.2) are only a subset of those satisfied by Q^* as per Fact 6.4. In order to better leverage the structure of Q^* , we strengthen the constraints in (6.2) to instead assert:

$$\mathcal{E}(f, \pi_g, h) = 0 \quad \text{for all } g \in \mathcal{F} \text{ and } h \in [H - 1]. \quad (6.3)$$

That is, we demand no Bellman error under f for any greedy policy induced by functions in \mathcal{F} . Note that we have restricted attention to all greedy policies as per \mathcal{F} instead of all policies in Fact 6.4, since this will suffice for our purposes as will shortly see.

This strengthening of constraints looks promising on the one hand as we can use data collected with one policy to simultaneously rule out many candidate f 's. At the same time, it might not be sufficient in isolation, since all policies $\{\pi_f : f \in \mathcal{F}\}$ might induce very different distributions over trajectories and we might still end up needing $O(|\mathcal{F}|)$ samples to even test the feasibility of a given f . This hardness is indeed fundamental and the core driving force behind the lower bound, where detecting the suboptimality of a wrong function f requires executing either the path it prefers, or the optimal path p in the CDP parameterized by p .

To circumvent the above challenge, we now make a structural assumption on average Bellman errors, which allows us to reason about the Bellman errors induced by all policies π_f in a sample-efficient manner. For any $h \in [H - 1]$, let us define the *Bellman error matrix* $\mathcal{E}_h \in \mathbb{R}^{|\mathcal{F}| \times |\mathcal{F}|}$ as

$$[\mathcal{E}_h]_{f,g} = \mathcal{E}(f, \pi_g, h). \quad (6.4)$$

That is, each entry in the matrix captures the Bellman error of the function indexed by the row under the greedy policy induced by the column at step h . With this notation, we define the Bellman rank of a CDP and a function class \mathcal{F} below.

Definition 6.6 (Bellman Rank). The Bellman rank of a CDP and a function class \mathcal{F} is the smallest integer M such that $\text{rank}(\mathcal{E}_h) \leq M$ for all $h \in [H - 1]$.

Intuitively, if the Bellman rank is small, then for any level h , the number of linearly independent columns is small. That is, the average Bellman error for any function under most policies can be expressed as linear combination of the Bellman errors of that function on a small set of policies corresponding to the linearly independent column. Note that the definition presented here is a simplification of the original definition from Jiang et al. [2017]. There are several known examples of problem structures with a small Bellman rank described in that paper. It is easily seen that the Bellman rank of a CDP is never larger than the number of unique contexts, so that it legitimately generalizes the concept of states in a tabular setting. More interestingly, it can be shown that Bellman rank can be further upper bounded in terms of latent quantities such as the rank of the transition matrix, or the number of latent states if the

CDP has an equivalent formulation as an MDP with a small number of latent states. We refer the reader to Jiang et al. [2017] for detailed examples, as well as connections of Bellman rank with other rank type notions in the RL literature to measure problem complexity.

6.4 Sample-efficient learning for CDPs with a small Bellman rank

Having defined our main structural assumption, we now describe an algorithm whose sample complexity depends on the Bellman rank, with no explicit dependence on $|\mathcal{X}|$ and only logarithmic scaling with $|\mathcal{F}|$. For ease of presentation, we will assume that all the expectations can be measured exactly with no errors, which serves to illustrate the key ideas. For a more careful analysis with finite samples, we refer the reader to Jiang et al. [2017]. The algorithm, named OLIVE for Optimism Led Iterative Value-function Elimination is an iterative algorithm which successively prunes value functions that violate one of the constraints described in (6.3). It then uses the principle of optimism in the face of uncertainty to select its next policy. The algorithm is described in Algorithm 3.

Algorithm 3 The OLIVE algorithm for CDPs with low Bellman rank

Input: Function class \mathcal{F} .

```

1: Initialize  $\mathcal{F}_0 = \mathcal{F}$ .
2: for  $t = 1, 2, \dots$ , do
3:   Define  $f_t = \operatorname{argmax}_{f \in \mathcal{F}_{t-1}} V_f$  and  $\pi_t = \pi_{f_t}$ .
4:   if  $V_{f_t} = V^{\pi_t}$  then return  $\pi_t$ .
5:   else
6:     Update  $\mathcal{F}_t = \{f \in \mathcal{F}_{t-1} : \mathcal{E}(f, \pi_t, h) = 0, \text{ for all } h \in [H - 1]\}$ .
7:   end if
8: end for
```

The update of \mathcal{F}_t in Line 6 requires some care as naïvely it requires executing a different policy π_f for each function f at steps h and $h + 1$. However, these expectations are easy to simultaneously evaluate in a sample efficient manner, since

$$\mathcal{E}(f, \pi, h) = \mathbb{E}[K\mathbf{1}(a_h = \pi_f(x_h))(f(x_h, a_h) - r_h - f(x_{h+1}, \pi_f(x_{h+1}))) \mid a_{0:h-1} \sim \pi, a_h \sim \mathcal{A} \text{ u.a.r.}].$$

This observation allows us to compute all the Bellman errors in Line 6 using expectation under a single distribution. By standard concentration arguments, all these expectations are also easy to approximate with finitely many samples, with only a logarithmic dependence on $|\mathcal{F}|$ coming from a union bound.

Since we assume that all the expectations are available exactly, the main complexity analysis in OLIVE concerns the number of iterations before it terminates. When we estimate expectations using samples, this iteration complexity is critical as it also scales the sample complexity of the algorithm. We will state and prove the following theorem regarding the iteration complexity of OLIVE.

Theorem 6.7. *For any CDP and \mathcal{F} with Bellman rank M , OLIVE terminates in at most MH iterations and outputs π^* .*

Proof: Consider an iteration t of OLIVE. Due to Assumption 6.2 and Fact 6.4, we know that $Q^* \in \mathcal{F}_{t-1}$. Suppose OLIVE terminates at this iteration and returns π_t . Then we have

$$V^{\pi_t} = V_{f_t} = \max_{f \in \mathcal{F}_{t-1}} V_f \geq V_{Q^*} = V^*,$$

since $Q^* \in \mathcal{F}_{t-1}$. So the algorithm correctly outputs an optimal policy when it terminates.

On the other hand, if it does not terminate then $V^{\pi_t} \neq V_{f_t}$ and Lemma 6.5 implies that $\mathcal{E}(f_t, \pi_t, h) > 0$ for some step $h \in [H - 1]$. This certainly ensures that $f_t \notin \mathcal{F}_t$, but has significantly stronger implications. Note that $f_t \in \mathcal{F}_{t-1}$ implies that $\mathcal{E}(f_t, \pi_s, h) = 0$ for all $s < t$ and $h \in [H - 1]$. Since we just concluded that $\mathcal{E}(f_t, \pi_t, h) > 0$ for some h , it must be the case that the column corresponding to π_t is linearly independent of those corresponding to π_1, \dots, π_{t-1} in the matrix \mathcal{E}_h . Consequently, at each non-final iteration, OLIVE increases the rank of at least one matrix \mathcal{E}_h by 1. Since the rank of each matrix is bounded by M , after a total of MH iterations, the algorithm must terminate, which gives the statement of the theorem. ■

The proof of the theorem makes it precise that the factorization underlying Bellman rank really plays the role of an efficient basis for exploration in this complex CDP. Extending these ideas to noisy estimates of expectations requires some care since algebraic notions like rank are not robust to noise. Instead Jiang et al. [2017] use a more general volumetric argument to analyze the noisy case, as well as describe robustness to requirements of exact low-rank factorization and realizability.

Unfortunately, the OLIVE algorithm is not computationally efficient, and a computational hardness result was discovered by Dann et al. [2018]. Developing both statistically and computationally efficient exploration algorithms for RL with rich observations is an area of active research.

Chapter 7

Behavioral Cloning and Apprenticeship Learning

Apprenticeship Learning, Imitation Learning, and Behavioral Cloning

Alekh Agarwal, Nan Jiang, Sham M. Kakade

Chapter 7

Learning from demonstrations is the problem of learning a policy from expert demonstrations. In contrast to reinforcement learning, such procedures do not require carefully designed reward functions.

Algorithms for learning from demonstrations may be classified according to the interaction model they operate in. The two popular approaches

1. *Behavioral Cloning* (a.k.a. *Imitation Learning*): the learner attempts to directly learn a state-to-action map from the expert demonstrations, where we observe the state-action pairs on expert's trajectories. [Ross and Bagnell, 2010].
2. *Inverse Reinforcement Learning*: The learner chooses the best policy to optimise a reward function that is inferred from expert demonstrations [Ng et al., 2000, Abbeel and Ng, 2004, Syed and Schapire, 2008].
3. *Learning from observations*: the learner attempts to directly learn a state-to-action map from the expert state trajectories, where we only observe the states on the expert's trajectories.

7.1 Linear Programming Formulations

Before we look at solution concepts, it is helpful to understand a different formulation of finding a optimal policy for a known MDP.

7.1.1 The Primal LP

Consider the optimization problem over $V \in \mathbb{R}^{|S|}$:

$$\begin{aligned} \max \quad & \sum_s d_0(s) V(s) \\ \text{subject to} \quad & V(s) \geq (1 - \gamma)r(s, a) + \gamma \sum_{s'} P(s'|s, a) V(s') \quad \forall a \in \mathcal{A}, s \in \mathcal{S} \end{aligned}$$

The optimal value function $V^*(d_0)$ is the value of this linear program, and the policy derived from the solution vector V achieves the optimal value $V^*(d_0)$.

7.1.2 The Dual LP

For a fixed (possibly stochastic) policy π , let us define the state-action visitation distribution $\mu_{d_0}^\pi$ as:

$$\mu_{d_0}^\pi(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr^\pi(s_t = s, a_t = a)$$

where $\Pr^\pi(s_t = s, a_t = a)$ is the state-action visitation probability, where we use π in M starting at state $s_0 \sim d_0$. We drop the d_0 dependence when clear from context.

It is possible to verify that μ satisfies, for all states $s \in \mathcal{S}$:

$$\sum_a \mu^\pi(s, a) = (1 - \gamma)d_0(s) + \gamma \sum_{s', a'} P(s|s', a') \mu^\pi(s', a')$$

Now let us define the state-action polytope as follows:

$$\mathcal{K} := \{\mu \mid \mu \geq 0 \text{ and } \sum_a \mu(s, a) = (1 - \gamma)d_0(s) + \gamma \sum_{s', a'} P(s|s', a') \mu(s', a')\}$$

Note that \mathcal{K} We now see that this set precisely characterizes all state-action visitation distributions.

Lemma 7.1. [Puterman, 1994] *We have that \mathcal{K} is equal to the set of all feasible state-action distributions, i.e. $\mu \in \mathcal{K}$ if and only if there exists a (stationary) policy π such that $\mu^\pi = \mu$.*

For $\mu \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$, the dual LP formulation is as follows:

$$\begin{aligned} \max \quad & \sum_{s, a} \mu(s, a) r(s, a) \\ \text{subject to} \quad & \mu \in \mathcal{K} \end{aligned}$$

If μ^* is the solution to this LP, then we have that:

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

An alternative optimal policy is $\arg\max_a \mu^*(s, a)$ (and these policies are identical if the optimal policy is unique).

7.2 Behavioral Cloning

Let us now suppose that we observe some expert behavior π_e , where we hope that π_e has value near to that of an optimal policy.

In the simplest setting, let us assume when we query the expert, we can get an independent sample:

$$(s, a) \sim \mu^{\pi_e}.$$

Note that if we were observing independent expert trajectories of length $\frac{c}{1-\gamma}$ (for a constant c), then each trajectory gives us $\frac{c}{1-\gamma}$ correlated samples.

For analysis, it is more natural to abstract away this issue for now and assume the samples are independent. One can address this dependence issue in a algorithm dependent manner.

Assume we obtain m samples from the expert. Let us say that $\hat{\mu}^e$ is the empirical estimate of $|\mu^{\pi_e}|$. It will also be natural to consider the effectiveness of various approaches when specialized do the tabular for setting. Here, it is helpful to observe that, with a standard concentration argument implies, we have with probability greater than $1 - \delta$,

$$\|\mu^{\pi_e} - \hat{\mu}^e\|_1 \leq 2\sqrt{\frac{|\mathcal{S}||\mathcal{A}|\log(1/\delta)}{m}} \quad (7.1)$$

7.2.1 Behavioral Cloning via Supervised Learning

The supervised learning approach is to learn a policy that matches the behavioral policy. In the tabular setting, the most straightforward approach is to use the empirical samples $\hat{\mu}^e$ to learn a deterministic multi-class classifier; note that this classifier is actually our policy as it is predicting our actions. Let us suppose that our *classification error* is less than ϵ ; precisely, suppose

$$\mathbb{E}_{s,a \sim \mu^{\pi_e}} [\mathbb{1}[\pi_{\text{SL}}(s) \neq a]] \leq \epsilon. \quad (7.2)$$

where $\pi_{\text{SL}} : \mathcal{S} \rightarrow \mathcal{A}$ is our deterministic policy.

Theorem 7.2. *Suppose the classification error of π_{SL} is less than ϵ , as per Equation 7.2, then we have:*

$$|V^{\pi_{\text{SL}}}(d_0) - V^{\pi_e}(d_0)| \leq \frac{\epsilon}{1-\gamma}$$

Proof: Due to that $A^{\pi_{\text{SL}}}(s, \pi_{\text{SL}}(s)) = 0$, the performance difference lemma (Lemma 4.5) implies:

$$\begin{aligned} |V^{\pi_{\text{SL}}}(d_0) - V^{\pi_e}(d_0)| &= \frac{1}{1-\gamma} |\mathbb{E}_{s,a \sim \mu^{\pi_e}} [A^{\pi_{\text{SL}}}(s, a)]| \\ &= \frac{1}{1-\gamma} \left| \mathbb{E}_{s,a \sim \mu^{\pi_e}} \left[A^{\pi_{\text{SL}}}(s, a) \mathbb{1}[\pi_{\text{SL}}(s) = a] \right] + \mathbb{E}_{s,a \sim \mu^{\pi_e}} \left[A^{\pi_{\text{SL}}}(s, a) \mathbb{1}[\pi_{\text{SL}}(s) \neq a] \right] \right| \\ &= \frac{1}{1-\gamma} \left| 0 + \mathbb{E}_{s,a \sim \mu^{\pi_e}} \left[A^{\pi_{\text{SL}}}(s, a) \mathbb{1}[\pi_{\text{SL}}(s) \neq a] \right] \right| \\ &\leq \frac{1}{1-\gamma} \|A^{\pi_{\text{SL}}}\|_{\infty} \cdot \mathbb{E}_{s,a \sim \mu^{\pi_e}} [\mathbb{1}[\pi_{\text{SL}}(s) \neq a]] \\ &\leq \frac{\epsilon}{1-\gamma}, \end{aligned}$$

which completes the proof. ■

The tabular case: In the tabular setting, the most straightforward approach is to use the empirical estimate $\hat{\mu}^e$. Specifically, for s s.t. $\sum_{a'} \hat{\mu}^e(s, a') > 0$, then we can use:

$$\pi_{\text{SL}}(a|s) = \frac{\hat{\mu}^e(s, a)}{\sum_{a'} \hat{\mu}^e(s, a')}.$$

else we can use any action at an unobserved state.

Corollary 7.3. *Suppose the expert policy is deterministic. In the tabular setting, with m samples, we have that with probability greater than $1 - \delta$,*

$$|V^{\pi_{\text{SL}}}(d_0) - V^{\pi_e}(d_0)| \leq \frac{2}{1-\gamma} \sqrt{\frac{|\mathcal{S}||\mathcal{A}| \log(1/\delta)}{m}}$$

Proof:

$$\begin{aligned} \mathbb{E}_{s,a \sim \mu^{\pi_e}} [\mathbb{1}[\pi_{\text{SL}}(s) \neq a]] &= \sum_{s,a} \mu^{\pi_e}(s, a) [\mathbb{1}[\pi_{\text{SL}}(s) \neq a]] \\ &= \sum_{s,a} \hat{\mu}^e(s, a) [\mathbb{1}[\pi_{\text{SL}}(s) \neq a]] + \sum_{s,a} \left(\mu^{\pi_e}(s, a) - \hat{\mu}^e(s, a) \right) [\mathbb{1}[\pi_{\text{SL}}(s) \neq a]] \\ &= 0 + \|\mu^{\pi_e} - \hat{\mu}^e\|_1, \end{aligned}$$

which completes the proof using Equation 7.1. ■

Lower bounds: To be added...

7.2.2 Behavioral Cloning via Distribution Matching

Note the above algorithm does not need any further interaction with the MDP. We may hope that we can improve upon this algorithm if we allow for interaction with the world. In many cases, expert demonstrations are costly to obtain while interactions with the environment are far less costly.

Let us consider now the case where the model dynamics P are known. We now present an alternative algorithm, which provides a substantial improvement in the tabular case.

Let us suppose we use a density estimation algorithm which has ϵ error in the following sense:

$$\|\mu^{\pi_e} - \hat{\mu}^e\|_1 \leq \epsilon \quad (7.3)$$

One natural algorithm is as follows:

$$\begin{aligned} \min \quad & \sum_{s,a} |\mu(s,a) - \hat{\mu}^e(s,a)| \\ \text{subject to} \quad & \mu \in \mathcal{K} \end{aligned} \quad (7.4)$$

Note that the cost function is convex subject to convex constraints. In fact, for the particular case of an ℓ_1 cost function, we can actually formulate the optimization program as a linear program.

If μ_{DM} is a solution to this LP, then let us define the policy to be:

$$\pi_{\text{DM}}(a|s) = \frac{\mu_{\text{DM}}(s,a)}{\sum_{a'} \mu_{\text{DM}}(s,a')}.$$

Theorem 7.4. Suppose the density estimation error of $\hat{\mu}^e$ is less than ϵ , as per Equation 7.3, then we have:

$$|V^{\pi_{\text{DM}}}(d_0) - V^{\pi_e}(d_0)| \leq 2\epsilon$$

Proof: Note that μ^{π_e} is a feasible point in the linear program in Equation 7.4, which has an objective value of ϵ . Let μ^* be the optimal solution, since μ^* has a lower objective value, we have that:

$$\|\mu_{\text{DM}} - \hat{\mu}^e\|_1 \leq \epsilon$$

By construction, $\mu_{\text{DM}} = \mu^{\pi_{\text{DM}}}$, so we have that:

$$\begin{aligned} |V^{\pi_{\text{DM}}}(d_0) - V^{\pi_e}(d_0)| &= \left| \sum_{s,a} \mu_{\text{DM}}(s,a)r(s,a) - \sum_{s,a} \mu^{\pi_e}(s,a)r(s,a) \right| \\ &= \left| \sum_{s,a} \left(\mu_{\text{DM}}(s,a) - \mu^{\pi_e}(s,a) \right) r(s,a) \right| \\ &\leq \|\mu_{\text{DM}} - \mu^{\pi_e}\|_1 \\ &\leq \|\mu_{\text{DM}} - \hat{\mu}^e\|_1 + \|\hat{\mu}^e - \mu^{\pi_e}\|_1 \\ &\leq 2\epsilon, \end{aligned}$$

which completes the proof. ■

The tabular case: In the tabular setting, the most straightforward approach is to use the empirical plugin estimate, $\hat{\mu}^e$, of μ^{π^e} . By Equation 7.1), we have the immediate Corollary, which is an improvement by a factor of $1/(1 - \gamma)$.

Corollary 7.5. *Suppose the expert policy is deterministic. In the tabular setting, with m samples, we have that with probability greater than $1 - \delta$,*

$$|V^{\pi_{SL}}(d_0) - V^{\pi^e}(d_0)| \leq 4\sqrt{\frac{|\mathcal{S}||\mathcal{A}|\log(1/\delta)}{m}}$$

Lower bounds: To be added... (basically, with knowledge of P , this approach is sample optimal.)

7.2.3 Sample Efficiency: comparing the approaches

While density estimation is often considered more challenging than supervised learning (i.e. regression or classification), it is important to note that in the tabular setting, the distributional matching approach (when P is known or when we have simulation access) is more sample efficient with regards to expert demonstrations. This suggest that the relying on supervised learning in setting with function approximation may be suboptimal and that alternative approaches may be more sample efficient.

7.3 Learning from Observation

Let us now suppose that we only observe the trajectories of states from an expert, as opposed to the state-action pairs. As before, we only assume sampling access to states via μ^{π^e} . In particular, let us define the marginal distribution over states as:

$$d^{\pi^e}(s) = \sum_a \mu^{\pi^e}(s, a).$$

Now, when we query the expert, assume we get an independent sample:

$$s \sim d^{\pi^e}.$$

Again, if we were observing independent expert trajectories of length $\frac{c}{1-\gamma}$ (for a constant c), then each trajectory gives us $\frac{c}{1-\gamma}$ correlated samples. For analysis, it is more natural to abstract away this issue for now and assume the samples are independent.

7.3.1 Learning from Observations via Distribution Matching

Again, assume the model dynamics P are known. Let us suppose we use a density estimation algorithm which has ϵ error in the following sense:

$$\|d^{\pi^e} - \hat{d}^e\|_1 \leq \epsilon \tag{7.5}$$

One natural algorithm is as follows:

$$\begin{aligned} \min \quad & \sum_s |\hat{d}^e(s) - \sum_a \mu(s, a)| \\ \text{subject to} \quad & \mu \in \mathcal{K} \end{aligned} \tag{7.6}$$

Note that the cost function is convex subject to convex constraints, and, for this cost function, we can actually formulate the optimization program as a linear program.

If μ_{DM} is a solution to this LP, then let us define the policy to be:

$$\pi_{\text{DM}}(a|s) = \frac{\mu_{\text{DM}}(s, a)}{\sum_{a'} \mu_{\text{DM}}(s, a')}.$$

Theorem 7.6. Assume the reward function of the expert is only state dependent, i.e. $r^e(s, a) = r^e(s)$. Suppose the density estimation error of \hat{d}^e is less than ϵ , as per Equation 7.5, then we have:

$$|V^{\pi_{\text{DM}}}(d_0) - V^{\pi^e}(d_0)| \leq 2\epsilon$$

Proof: To be added... ■

7.4 Inverse Reinforcement Learning

In inverse reinforcement learning, let us say that the expert has an unknown reward function $r^e(s, a)$. In many settings, writing down an effective reward function by hand is difficult. Here, we may hope that by using expert demonstrations, we may seek to extract a reward function from the expert, such that if we planned according to the our learned reward function, our value would be as good as the expert (on the experts unknown reward function).

Here let us assume we know some basis of reward functions ϕ_1, \dots, ϕ_d , where each $\phi_i : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, and that r lies in this basis. Specifically, suppose for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ that

$$r^e(s, a) = \sum_{i=1}^d w_i \phi_i(s, a), \quad (7.7)$$

where w are the unknown coefficients.

Again, suppose that the model P is known. Also, note that experts value function with respect to basis function ϕ_i (instead of r^e) is:

$$V_i^{\mu^{\pi^e}}(d_0) := \sum_{s,a} \mu^{\pi^e}(s, a) \phi_i(s, a).$$

We will use V_i^e as shorthand for $V_i^{\mu^{\pi^e}}(d_0)$. Note that with samples we can estimate V_i^e as follows:

$$\hat{V}_i^e := \sum_{s,a} \hat{\mu}(s, a) \phi_i(s, a).$$

where we can take $\hat{\mu}^e$ to be the plug-in estimator.

For now, let us suppose that \hat{V}_i^e is known exactly in order to have a more transparent algorithm. Incorporating sampling error is possible. Consider the following feasibility problem for $\mu \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$:

$$\begin{aligned} & \text{find} && \mu \\ & \text{subject to} && \mu \in \mathcal{K} \\ & && V_i^e = \sum_{s,a} \mu(s, a) \phi_i(s, a) \quad \forall i \in \{1, \dots, d\} \end{aligned}$$

Suppose μ_{IRL} is a feasible point in this LP. Let us define the implied policy to be:

$$\pi_{\text{IRL}}(a|s) = \frac{\mu_{\text{IRL}}(s, a)}{\sum_{a'} \mu_{\text{IRL}}(s, a')}.$$

Theorem 7.7. *The above feasibility problem is non-empty. Furthermore, the implied policy π_{IRL} (derived from the above LP) satisfies $V_M^{\pi_{\text{IRL}}}(d_0) = V_M^{\mu^{\pi_e}}(d_0)$, where M is the MDP with experts reward function r^e .*

Proof: By construction, μ^{π_e} is a feasible point. Now suppose μ_{IRL} is a feasible point. By construction, $V_i^{\pi_{\text{IRL}}} = \hat{V}_i^e$ for all i , due to that π_{IRL} has state-action visitation frequency μ_{IRL} . The proof is completed using the span condition in Equation 7.7. ■

Bibliography

- Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.
- Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes, 2019.
- András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.
- Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.
- Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J Kappen. Minimax pac bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91(3):325–349, 2013.
- J. Andrew Bagnell and Jeff Schneider. Covariant policy search. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI’03*, pages 1019–1024, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc. URL <http://dl.acm.org/citation.cfm?id=1630659.1630805>.
- A. Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi: 10.1137/1.9781611974997.
- Richard Bellman. Dynamic programming and Lagrange multipliers. *Proceedings of the National Academy of Sciences*, 42(10):767–769, 1956.
- Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.
- Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandit algorithms with supervised learning guarantees. In *AISTATS*, 2011.
- Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231, 2003.
- Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E. Schapire. On oracle-efficient PAC reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems 31*, 2018.
- Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. Online Markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.
- Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E. Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, 2017.

- S. Kakade. A natural policy gradient. In *NIPS*, 2001.
- Sham Kakade and John Langford. Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the 19th International Conference on Machine Learning*, volume 2, pages 267–274, 2002.
- Sham Machandranath Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University of College London, 2003.
- Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 2002.
- Akshay Krishnamurthy, Alekh Agarwal, and John Langford. PAC reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, pages 1840–1848, 2016.
- Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.
- Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.
- Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668, 2010.
- Satinder Singh and Richard Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.
- Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 99, pages 1057–1063, 1999.
- Umar Syed and Robert E Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in neural information processing systems*, pages 1449–1456, 2008.
- Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

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 $1 - \delta$:

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

Lemma A.2. (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 $\text{Var}(X_i)$ denote the variance of X_i . If $X_i - EX_i \leq b$ for all i , then

$$P(\bar{X}_n \geq \mu + \epsilon) \leq \exp \left[-\frac{n^2 \epsilon^2}{2 \sum_{i=1}^n \text{Var}(X_i) + 2nb\epsilon/3} \right].$$

If all the variances are equal, the Bernstein inequality implies that, with probability at least $1 - \delta$,

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

Lemma A.3 (Version of Freedman's inequality from Beygelzimer et al. [2011]). Let X_1, X_2, \dots, X_T be a sequence of real-valued random variables adapted to the filtration \mathcal{F}_i . That is, X_i is measurable with respect to \mathcal{F}_i and further assume that $\mathbb{E}[X_i | \mathcal{F}_{i-1}] = 0$. Define $S = \sum_{t=1}^T X_t$, $V = \sum_{t=1}^T \mathbb{E}[X_t^2 | \mathcal{F}_{t-1}]$ and let $X_t \leq R$ almost surely for all t . Then for any $\delta \in (0, 1)$ and $\lambda \in [0, 1/R]$, with probability at least $1 - \delta$,

$$S \leq (e - 2)\lambda V + \frac{\ln(1/\delta)}{\lambda}.$$

In particular, choosing $\lambda = \min \left\{ \frac{1}{R}, \sqrt{\ln(1/\delta)/V} \right\}$, we get the Bernstein-style bound

$$S \leq 2\sqrt{V \ln \frac{1}{\delta}} + R \ln \frac{1}{\delta}.$$