Lecture 6: Policy improvement methods

Akshay Krishnamurthy
akshay@cs.umass.edu

February 28, 2022

1 Introduction and Recap

Last lecture we introduce the Markov decision process and the key definitions. We spent most of the lecture discussing the finite horizon episodic setting, since some of the main ideas are a bit easier to understand. At the end we discussed the infinite horizon discounted setting and saw the value iteration algorithm for planning. Today we will work primarily in the infinite horizon discounted setting and discuss policy optimization methods.

Recall that a discounted MDP is defined by $(\mathcal{S}, \mathcal{A}, R, P, \mu, \gamma)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, R is the reward function, P is the transition operator, μ is the initial state distribution, and γ is the discount factor. Recall the Bellman equations for policy evaluation and optimality (we will work mostly with Q-functions):

$$Q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a), a' \sim \pi(s')} Q^\pi(s',a') \tag{1}$$

$$Q^\star(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} \max_{a'} Q^\star(s',a') = (\mathcal{T}Q^\star)(s,a) \tag{2}$$

Today we will discuss methods for computing the optimal policy based on policy improvement. These methods are particularly nice because they involve directly optimizing the objective we care about, namely $J(\pi)$, rather than something more indirect, like finding a fixed point of the Bellman equations. They also have some limitations, which we will discuss. However, perhaps because of this "directness," they have been extremely successful in practice, and play a central role in some of the state-of-the-art Deep RL methods.

Let us start with policy iteration and then turn to more "soft" approaches like policy gradient methods. Two advantages of these methods are: (a) they can be applied when we only have sample access to the MDP and (b) they can accommodate function approximation. In the next lecture we will discuss these issues in detail.

2 Policy iteration.

Just like in the finite horizon setting, we can also consider a policy iteration procedure. Here, we start with an arbitrary policy $\pi^{(0)}$ and we repeat the iteration: (a) compute $Q^{(t)} = Q^{\pi^{(t)}}$ (called policy evaluation), (b) update $\pi^{(t+1)} = \pi_{Q^{(t)}}$ (called policy improvement). After T iterations, we simply output $\pi^{(T+1)}$. Note here that we always use Q-functions that correspond to actual policies, which is a departure from value-iteration or dynamic programming methods.

The key lemma for this algorithm also asserts that we make geometric progress toward $Q^\star$.

Lemma 1. *We have that $Q^{\pi^{(t+1)}} \geq \mathcal{T}Q^{\pi^{(t)}} \geq Q^{\pi^{(t)}}$ where the inequalities hold pointwise. Additionally $\|Q^{\pi^{(t+1)}} - Q^\star\|_\infty \leq \gamma \|Q^{\pi^{(t)}} - Q^\star\|_\infty$.*

Proof. First, by examining the Bellman equations, it is easy to see that $\mathcal{T}Q^{\pi^{(t)}} \geq Q^{\pi^{(t)}}$:

$$\mathcal{T}Q^{\pi^{(t)}}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} \left[\max_{a'} Q^{\pi^{(t)}}(s',a')\right] \geq r(s,a) + \gamma \mathbb{E}_{\substack{s' \sim P(s,a), \\ a' \sim \pi^{(t)}(s')}} \left[Q^{\pi^{(t)}}(s',a')\right] = Q^{\pi^{(t)}}(s,a).$$

Next we see that $Q^{\pi^{(t+1)}} \geq Q^{\pi^{(t)}}$ pointwise:

$$Q^{\pi^{(t)}}(s,a) = r(s,a) + \mathbb{E}_{\substack{s' \sim P(s,a), \\ a' \sim \pi^{(t)}(s')}} \left[Q^{\pi^{(t)}}(s',a')\right] \leq r(s,a) + \mathbb{E}_{\substack{s' \sim P(s,a), \\ a' \sim \pi^{(t+1)}(s')}} \left[Q^{\pi^{(t)}}(s',a')\right] \leq Q^{\pi^{(t+1)}}(s,a),$$

where the first inequality holds since $\pi^{(t+1)}$ is greedy with respect to $Q^{\pi^{(t)}}$ and the second inequality holds by repeated application of the first inequality inside the expectation. Now we can show that $Q^{\pi^{(t+1)}} \geq \mathcal{T}Q^{\pi^{(t)}}$:

$$Q^{\pi^{(t+1)}}(s,a) = r(s,a) + \mathbb{E}_{\substack{s' \sim P(s,a),\\ a' \sim \pi^{(t+1)}(s')}} \left[Q^{\pi^{(t+1)}}(s',a')\right] \geq r(s,a) + \mathbb{E}_{\substack{s' \sim P(s,a),\\ a' \sim \pi^{(t+1)}(s')}} \left[Q^{\pi^{(t)}}(s',a')\right] = \mathcal{T}Q^{\pi^{(t)}}$$

where the first inequality uses the fact that $Q^{\pi^{(t+1)}} \geq Q^{\pi^{(t)}}$ pointwise and the last inequality holds since $\pi^{(t+1)}$ is greedy with respect to $Q^{\pi^{(t)}}$. Finally

$$\|Q^\star - Q^{\pi^{(t+1)}}\|_\infty \leq \|Q^\star - \mathcal{T}Q^{\pi^{(t)}}\|_\infty = \|\mathcal{T}Q^\star - \mathcal{T}Q^{\pi^{(t)}}\|_\infty \leq \gamma\|Q^\star - Q^{\pi^{(t)}}\|_\infty \qquad \square$$

Recall the policy error lemma from the last lecture

Lemma 2 (Policy error lemma). *For any Q-function f, we have $J(\pi^\star) \leq J(\pi_f) + \frac{2}{1-\gamma}\|f - Q^\star\|_\infty$.*

Proof. Consider state s and let $a = \pi_f(s) = \mathrm{argmax}_{a'} f(s,a')$. Then

$$\begin{aligned} V^\star(s) - V^{\pi_f}(s) &= Q^\star(s,\pi^\star(s)) - Q^\star(s,a) + Q^\star(s,a) - Q^{\pi_f}(s,a) \\ &\leq Q^\star(s,\pi^\star(s)) - f(s,\pi^\star(s)) + f(s,a) - Q^\star(s,a) + Q^\star(s,a) - Q^{\pi_f}(s,a) \\ &\leq 2\|Q^\star - f\|_\infty + \gamma\mathbb{E}_{s' \sim P(s,a)}\left[V^\star(s') - V^{\pi_f}(s')\right] \\ &\leq 2\|Q^\star - f\|_\infty + \gamma\|V^\star - V^{\pi_f}\|_\infty \end{aligned} \tag{3}$$

Re-arranging this inequality actually proves a stronger statement, namely that $V^\star$ and V^{π_f} are close, which implies that $J(\pi^\star)$ and $J(\pi_f)$ are close. (If we examine the proof, note that we only care about errors on the distribution induced by π_f.) □

Since $\pi^{(T+1)}$ is greedy w.r.t. $Q^{(T)}$, this lemma and the "contraction lemma" immediately gives the same convergence guarantee as we obtained for value iteration, that is $O(\gamma^T/(1-\gamma))$ sub-optimality after T iterations.

Computational complexity. Observe that although the Bellman optimality equation is a non-linear fixed point, the Bellman evaluation equation (for a fixed π) is actually linear. Indeed, suppose we define an $SA \times SA$ matrix with entries, $P^\pi[(s,a),(s',a')] = P(s' \mid s,a)\pi(a' \mid s')$ then we can write

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

Thus, each iteration of the algorithm takes $\mathrm{poly}(S,A)$ time, roughly the cost of inverting an $SA \times SA$ matrix.

3 Policy gradient methods

One concern with policy iteration methods is that $\pi^{(t)}$ and $\pi^{(t+1)}$ can be very different, which can lead to some instabilities, especially when we introducing errors from sampling. In particular, if we look at (3) where we bound the policy error, we actually care about $Q^\star - f$ *on the data distribution induced by* π_f. In the context of *noiseless* policy iteration, $f = Q^{\pi^{(T)}}$ and $\pi_f = \pi^{(T+1)}$ is the greedy policy with respect to this Q function. If we are sampling, then $f = \widehat{Q}^{\pi^{(T)}}$ will be an estimate of $Q^{\pi^{(T)}}$, perhaps obtained by sampling trajectories according to $\pi^{(T)}$. But this function may be a poor estimate outside of the states that $\pi^{(T)}$ visits frequently. So if $\pi^{(T+1)}$ visits those states, our performance may be quite bad. Thus it is desirable to have algorithms that update less aggressively.

A natural approach is to parametrize the policies in some manner and directly apply standard continuous optimization methods like gradient descent to the objective $J(\pi)$.

Policy gradient methods. To apply gradient descent in the MDP setting, we first need to turn the problem from a discrete optimization into a continuous one (Currently the problem is discrete since we know that the optimal policy is deterministic). To do this, let us define a *parametrized policy* π_θ with some vector valued parameter θ so that $\pi_\theta(\cdot \mid s) \in \Delta(\mathcal{A})$ prescribes a distribution over actions for each state. Some examples to keep in mind are:

$$\begin{aligned}
\text{Tabular parametrization:} \quad & \pi_\theta(a \mid s) = \theta_{s,a}, \quad \theta \in \mathbb{R}^{S \times A} \\
\text{Softmax parametrization:} \quad & \pi_\theta(a \mid s) = \frac{\exp(\theta_{s,a})}{\sum_{a' \in \mathcal{A}} \exp(\theta_{s,a'})}, \quad \theta \in \mathbb{R}^{S \times A} \\
\text{Softmax-linear:} \quad & \pi_\theta(a \mid s) = \frac{\exp(\langle \theta, \phi(s,a) \rangle)}{\sum_{a' \in \mathcal{A}} \exp(\langle \theta, \phi(s,a) \rangle)}, \quad \theta \in \mathbb{R}^{d}.
\end{aligned}$$

In the last case, the feature map $\phi(s,a)$ is fixed and known.

Now we slightly change the optimization problem from $\max_\pi J(\pi)$ to $\max_\theta J(\pi_\theta)$, so we have a continuous parametrization. From here we can simply do gradient ascent with a learning rate η: Starting with $\theta^{(0)}$ initialized arbitrarily, on the t^{th} iteration we do the update

$$\theta^{(t)} \leftarrow \theta^{(t-1)} + \eta \nabla_\theta J(\pi_{\theta^{(t-1)}})$$

A basic issue when instantiating this algorithm is that computing the gradient $\nabla_\theta J(\pi_\theta)$ could be quite challenging, as we may have to differentiate through the dynamics of the MDP. Indeed $J(\pi_\theta)$ is a complicated expectation with θ influencing how we take every action along the trajectory. However, the next theorem, known as the *policy gradient* theorem, reveals a very simple structure for this gradient.

To state the theorem, let us define two additional quantities. For any policy π, let d^π denote the discounted state occupancy measure: $d^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s \mid s_0 \sim \mu, \pi)$ and let $A^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be the *advantage function* given by $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$.

Theorem 3 (Policy gradient theorem)**.** *We have the following expressions for $\nabla_\theta J(\pi_\theta)$:*

$$\textit{Reinforce: } \nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) \cdot \left(\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right) \mid \pi_\theta\right] \tag{4}$$

$$\textit{Q-version: } \nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_\theta}(s_t, a_t) \cdot \nabla_\theta \log \pi_\theta(a_t \mid s_t) \mid \pi_\theta\right] = \frac{1}{1-\gamma} \mathbb{E}_{\substack{s \sim d^{\pi_\theta} \\ a \sim \pi_\theta(\cdot \mid s)}} \left[Q^{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(a \mid s)\right] \tag{5}$$

$$\textit{A-version: } \nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \mathbb{E}_{\substack{s \sim d^{\pi_\theta} \\ a \sim \pi_\theta(\cdot \mid s)}} \left[A^{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(a \mid s)\right] \tag{6}$$

The expressions for $\nabla_\theta J(\pi_\theta)$ given by this theorem all are very convenient because they can be written as expectations under π_θ. This makes these gradients particularly suitable for estimation with samples, we can simply collect many trajectories from π_θ and obtain (nearly) unbiased estimates of the gradient. All that is required is that we can take derivatives of the policy parametrization itself.

The A-version, in terms of the advantage function, is somewhat nice for building intuition. Note that the advantage $A^{\pi_\theta}(s,a)$ is positive if increasing the probability of taking action a relative to the distribution $\pi_\theta(\cdot \mid s)$ (and keeping everything else fixed) leads to an improvement in value. Since $\nabla_\theta \log \pi_\theta(a \mid s)$ is always non-negative (since $\pi_\theta(\cdot \mid s)$ is a distribution), this expression shows that we increase the weight on actions that have positive advantage, which intuitively should lead to a better policy.

Proof. The key technical insight is that $\nabla_x \log(f(x)) = 1/f(x) \cdot \nabla_x f(x)$ and so:

$$\nabla_\theta \pi_\theta(a \mid s) = \pi_\theta(a \mid s) \cdot \nabla_\theta \log(\pi_\theta(a \mid s)),$$

which will allow us to preserve the distribution over which we are taking expectations.

Let us first prove (4). For a trajectory τ define $R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t$. Then:

$$\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \sum_\tau R(\tau) \nabla_\theta \mathbb{P}[\tau \mid \pi_\theta] \\
&= \sum_\tau R(\tau) \mathbb{P}[\tau \mid \pi_\theta] \cdot \nabla_\theta \log \left(\mathbb{P}[\tau \mid \pi_\theta] \right) \\
&= \sum_\tau R(\tau) \mathbb{P}[\tau \mid \pi_\theta] \cdot \nabla_\theta \log \left(\mu(s_0) \pi_\theta(a_0 \mid s_0) P(s_1 \mid s_0, a_0) \pi_\theta(a_1 \mid s_1) \ldots \right) \\
&= \sum_\tau R(\tau) \mathbb{P}[\tau \mid \pi_\theta] \cdot \left(\sum_{t=0}^{\infty} \nabla_\theta \log \left(\pi_\theta(a_t \mid s_t) \right) \right),
\end{aligned}$$

which, via the definition of $R(\tau)$, is precisely the REINFORCE expression for the policy gradient.

There are at least two ways to prove the Q-version in (5). One nice way is to start from the REINFORCE version and group the terms as follows

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \cdot \sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \mid \pi_\theta \right] = \sum_{t=0}^{\infty} \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \cdot \left(\underbrace{\sum_{t'=0}^{t-1} \gamma^{t'} r_{t'}}_{\text{past rewards}} + \underbrace{\sum_{t'=t}^{\infty} \gamma^{t'} r_{t'}}_{\text{future rewards}} \right) \mid \pi_\theta \right].$$

For the first term, we will use that the rewards are in the past to show that the gradient is actually zero. For the second term we will use that the rewards are in the future and the Markov property to give us the Q-function. Let us look at the second term first. Fix index t

$$\begin{aligned}
\mathbb{E}\left[\nabla_\theta \log(\pi_\theta(a_t \mid s_t)) \sum_{t'=t}^{\infty} \gamma^{t'} r_{t'} \mid \pi_\theta \right] &= \mathbb{E}\left[\nabla_\theta \log(\pi_\theta(a_t \mid s_t)) \cdot \mathbb{E}\left[\sum_{t'=t}^{\infty} \gamma^{t'} r_{t'} \mid s_t, a_t, \pi \right] \mid \pi_\theta \right] \\
&= \mathbb{E}\left[\gamma^t \nabla_\theta \log(\pi_\theta(a_t \mid s_t)) \cdot \mathbb{E}\left[\sum_{t'=0}^{\infty} \gamma^{t'} r_{t'} \mid s_0 = s_t, a_0 = a_t, \pi \right] \mid \pi_\theta \right] \\
&= \mathbb{E}\left[\gamma^t Q^{\pi_\theta}(s_t, a_t) \nabla_\theta \log(\pi_\theta(a_t \mid s_t)) \mid \pi_\theta \right].
\end{aligned}$$

Here the key point is that the sum of future rewards is independent of the gradient term, conditioned on s_t, a_t which allows us to obtain the Q function.

Similarly for the "past" term, first observe that for any state s_t at time t we have

$$\mathbb{E}_{a \sim \pi_\theta(\cdot \mid s_t)} \left[\nabla_\theta \log \pi_\theta(a \mid s_t) \mid s_t \right] = \sum_a \pi_\theta(a \mid s_t) \nabla_\theta \log \pi_\theta(a \mid s_t) = \sum_a \nabla_\theta \pi_\theta(a \mid s_t) = \nabla_\theta \sum_a \pi_\theta(a \mid s_t) = 0.$$

Here the last step holds because $\pi_\theta(\cdot \mid s_t)$ is a distribution. Now

$$\mathbb{E}\left[\nabla_\theta \log(\pi_\theta(a_t \mid s_t)) \sum_{t'=0}^{t-1} \gamma^{t'} r_{t'} \mid \pi_\theta \right] = \mathbb{E}\left[\sum_{t'=0}^{t-1} \gamma^{t'} r_{t'} \cdot \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \mid s_t, \pi_\theta \right] \mid \pi_\theta \right] = 0.$$

Combining the two steps establishes the second expression. (There is a different proof given in the RL theory monograph which I encourage you to study.) The version in terms of the occupancy measure is a simple re-writing.

The advantage version can be derived using an argument similar to how we handled the "past" term above. Indeed $V^{\pi_\theta}(s_t)$ is independent of a_t given s_t, so again we can see that the gradient term will be zero here. □

Estimation from samples. One advantage of PG methods is that they are fairly easy to implement even when we don't know the MDP and only have sample access. Let us discuss one sampling scheme to obtain unbiased estimates of $\nabla_\theta J(\pi_\theta)$ which will allow us to instead run stochastic gradient ascent. If we look a the Q-version in (5), we would like to obtain unbiased estimates of $Q^{\pi_\theta}(s, a)$ for some (s, a) pair and we immediately run into the issue that $Q^{\pi_\theta}(s, a)$ is an infinite sum of rewards. First we show how to avoid this with a truncation argument.

Fix (s, a) and some policy π. Collect a *truncated* trajectory τ starting from (s, a) and executing π, where at each time step t we terminate with probability $1 - \gamma$ (after seeing the reward r_t). This trajectory will be finitely long with probability 1 and the expected length is $O(\frac{1}{1-\gamma})$. Let $t^\star$ denote the time step that we terminate. We estimate $\widehat{Q^\pi}(s, a) = \sum_{t=0}^{t^\star} r_t$ as the *undiscounted* sum of rewards until the stopping time. Then

$$\begin{aligned} \mathbb{E}\left[\widehat{Q^\pi}(s,a)\right] &= \mathbb{E}\left[\sum_{t=0}^{t^\star} r_t \mid \pi, s_0 = s, a_0 = a\right] = \mathbb{E}\left[\sum_{T=0}^{\infty} \mathbf{1}\{t^\star = T\} \sum_{t=0}^{T} r_t \mid \pi, s_0 = s, a_0 = a\right] \\ &= \mathbb{E}\left[\sum_{t=0}^{\infty} r_t \mathbf{1}\{t^\star \geq t\} \mid \pi, s_0 = s, a_0 = a\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s, a_0 = a\right] \end{aligned}$$

Now that we can get an unbiased estimate of the Q-function for any (s, a) pair, we can estimate the Q-version of the policy gradient using the same trick. Roll-out a single trajectory starting from $s_0 \sim \mu$ and executing π_θ and truncating this trajectory at every time step with probability $(1 - \gamma)$. This gives us a sequence $(s_0, a_0, \ldots, s_{t^\star}, a_{t^\star})$ and by the same argument as above the undiscounted sum of Q-functions is unbiased for the policy gradient.

$$\mathbb{E}\left[\sum_{t=0}^{t^\star} Q^{\pi_\theta}(s_t, a_t) \cdot \nabla_\theta \log(\pi_\theta(a_t \mid s_t))\right] = \nabla_\theta J(\pi_\theta)$$

So all we have to do is estimate the Q-functions along this trajectory, which we can do using a couple more roll-outs. Actually using the same roll-out suffices, but it needs to be a bit longer so we can estimate $Q^{\pi_\theta}(s_{t^\star}, a_{t^\star})$.

4 Natural policy gradient

By considering gradient ascent approaches, we open up the possibility of usin the entire optimization toolbox to solve RL problems. One idea along this line is a related algorithm that takes a particularly clean form with the softmax parametrization. We will study the convergence properties of this algorithm in detail in the next lecture but for now let us explain the derivation.

A somewhat standard idea in optimization is what is called "preconditioning" where instead of performing the standard gradient update $\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta \nabla f(\theta^{(t)})$ on objective f, we multiply the gradient by some matrix to induce a better geometry, that is $\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta M_t \nabla f(\theta^{(t)})$. One perspective on this is that we allow different step sizes in different directions, which can be good if the curvature of the objective function is very different (these are the "adaptive gradient" methods). Also if we set M_t to be the inverse hessian $(\nabla^2 f(\theta^{(t)}))^{-1}$ then we obtain Newton's method.

One choice for M_t that is often used when optimizing over probability models is derived from the *Fisher information matrix* which results in the *natural gradient* method. For a general parametrized probability distribution $p_\theta(x)$ the Fisher information matrix is defined as $I(\theta) := \mathbb{E}_{x \sim p_\theta(\cdot)}\left[(\nabla_\theta \log p_\theta(x))(\nabla_\theta \log p_\theta(x))^\top\right]$. In the context of policy gradient methods, we define

$$F(\theta) := \mathbb{E}_{\substack{s \sim d^{\pi_\theta} \\ a \sim \pi_\theta(\cdot \mid s)}}\left[(\nabla_\theta \log \pi_\theta(a \mid s))(\nabla_\theta \log \pi_\theta(a \mid s))^\top\right]$$

Then, we perform the updates

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta F(\theta)^\dagger \nabla_\theta J(\pi_{\theta^{(t)}})$$

While this update looks quite confusing, it takes a very clean form when working with the softmax parametrization $\pi_\theta(\cdot \mid s) \propto \exp(\theta_{s,a})$. This is given in the following lemma.

Lemma 4. *For the softmax parametrization the NPG update takes the form:*

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + \frac{\eta}{1-\gamma} A^{(t)} + \eta v, \qquad \pi^{(t+1)}(a \mid s) \propto \pi^{(t)}(a \mid s) \cdot \exp(\eta A^{(t)}(s,a)/(1-\gamma))$$

Here v is a state-dependent offset ($v_{s,a} = c_s$ for all $a \in \mathcal{A}$) and $\propto$ denotes that $\pi^{(t+1)}(\cdot \mid s)$ is a distribution.

This resembles a soft policy iteration, since rather than taking the greedy action (max), we are taking a softmax. Note that the update does not depend on either the initial distribution μ or the distribution $d^{(t)}$ which is used in the definition of the pre-conditioner. Note also how the policy update looks strikingly similar to the Exponential Weights update we saw earlier in the course.

Proof. Let us fix an iteration t and drop the superscripts to simplify the notation. First, observe that, with the softmax parametrization

$$\nabla_\theta \log(\pi_\theta(a \mid s)) = e_{s,a} - \sum_{a'} e_{s,a'} \pi_\theta(a' \mid s)$$

Recall that we already saw that $\mathbb{E}_{a \sim \pi_\theta(\cdot|s)} [\nabla_\theta \log \pi_\theta(a \mid s)] = 0$ (which is also easy to see from the above expression).

The update uses the Moore-Penrose pseudoinverse, which is the minimum norm w that is also a solution of

$$\min_w \|\nabla J(\pi_\theta) - F(\theta) w\|_2^2.$$

Using the A-version of the policy gradient theorem we have

$$\nabla J(\pi_\theta) = \frac{1}{1-\gamma} \mathbb{E}_{\substack{s \sim d^{\pi_\theta} \\ a \sim \pi_\theta(\cdot|s)}} [A^{\pi_\theta}(s,a) \cdot \nabla_\theta \log \pi_\theta(a \mid s)] .$$

Let us expand the matrix-vector product involving the Fisher matrix:

$$\begin{aligned} F(\theta) w &= \mathbb{E}_{\substack{s \sim d^{\pi_\theta} \\ a \sim \pi_\theta(\cdot|s)}} \left[(\nabla_\theta \log \pi_\theta(a \mid s))(w^\top \nabla_\theta \log \pi_\theta(a \mid s)) \right] \\ &= \mathbb{E}_{\substack{s \sim d^{\pi_\theta} \\ a \sim \pi_\theta(\cdot|s)}} \left[(\nabla_\theta \log \pi_\theta(a \mid s))(w_{s,a} - \mathbb{E}_{a' \sim \pi_\theta(\cdot|s)}[w_{s,a'}]) \right] \\ &= \mathbb{E}_{\substack{s \sim d^{\pi_\theta} \\ a \sim \pi_\theta(\cdot|s)}} [w_{s,a} \nabla_\theta \log \pi_\theta(a \mid s)] , \end{aligned}$$

where in the first step we use the gradient computation above and at the end we use that the average gradient is zero. Intuitively, now we can see that a valid solution for w is $A^{\pi_\theta}/(1-\gamma)$, simply by comparing the two expression. More formally, using the form of $\nabla_\theta \log \pi_\theta(a \mid s)$ again, the $(s,a)^{\text{th}}$ component of both of these vectors is

$$\begin{aligned} [F(\theta) w]_{s,a} &= d^{\pi_\theta}(s) \pi_\theta(a \mid s) \left(w_{s,a} - \sum_{a'} w_{s,a'} \pi_\theta(a' \mid s) \right) \\ [\nabla_\theta J(\pi_\theta)]_{s,a} &= d^{\pi_\theta}(s) \pi_\theta(a \mid s) \left(A^{\pi_\theta}(s,a)/(1-\gamma) \right) , \end{aligned}$$

where in the second line we are using that the expected (over actions) advantage is 0, i.e., $\mathbb{E}_{a \sim \pi_\theta(\cdot|s)}[A^{\pi_\theta}(s,a)] = 0$. Examining these expressions we can see that $w_{s,a} = A^{\pi_\theta}(s,a)/(1-\gamma) + v_s$ where v_s is some state-dependent offset. This proves the first part and the second part is immediate since the state-dependent offset can be absorbed into the normalization term. □