Reinforcement Learning

Discounted Policy Gradient and Actor-Critic

Lecturer: *Matteo Papini* version June 13, 2026

Policy Gradient for Discounted MDPs

We are going to derive the policy gradient theorem for the discounted setting. Consider an infinite-horizon discounted MDP with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel $\{p(\cdot|s,a) : s \in \mathcal{S}, a \in \mathcal{A}\}$, reward function $r : \mathcal{S} \times \mathcal{A} \to [-1,1]$, starting-state distribution p_0, and discount factor $0 < \gamma < 1$. Fix a class of parametric stochastic Markovian policies $\Pi_\Theta = \{\pi_{\boldsymbol{\theta}} \in \Delta_{\mathcal{A}}^{\mathcal{S}} \,|\, \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d\}$. The expected discounted return of a policy can be written, as a function of policy parameters, as follows:

$$J(\boldsymbol{\theta}) = \frac{1}{1-\gamma}\mathbb{E}[r(S,A)], \tag{1}$$

where $A \sim \pi_{\boldsymbol{\theta}}(\cdot|S)$ and $S \sim d^{\pi_{\boldsymbol{\theta}}}$, the γ-discounted (normalized) **state-occupancy measure** induced by policy $\pi_{\boldsymbol{\theta}}$:

$$d^{\pi_{\boldsymbol{\theta}}}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}(S_t = s),$$

where $S_0 \sim p_0$, $A_t \sim \pi_{\boldsymbol{\theta}}(\cdot|S_t)$, and $S_{t+1} \sim p(\cdot|S_t, A_t)$ for $t \geq 0$. Note that $d^{\pi_{\boldsymbol{\theta}}}$ is a probability measure since:

$$\sum_{s\in\mathcal{S}} d^{\pi_{\boldsymbol{\theta}}}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \sum_{s\in\mathcal{S}}\mathbb{P}(S_t = s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t = 1,$$

by the convergence of the geometric series. Note that state occupancy depends on the starting-state distribution. Sampling from $d^{\pi_{\boldsymbol{\theta}}}$ is equivalent to the following iterative sampling procedure:

Algorithm 1 State Sampler (Random Stopping)

Input: Policy $\pi_{\boldsymbol{\theta}}$, discount factor $0 < \gamma < 1$.

$S_0 \sim p_0$
for $t = 0, 1, 2 \dots$ **do**
    With probability $1-\gamma$, **return** S_t
    $A_t \sim \pi_{\boldsymbol{\theta}}(\cdot|S_t)$
    $S_{t+1} \sim p(\cdot|S_t, A_t)$
end for

Another way to sample from $d^{\pi_{\boldsymbol{\theta}}}$ is to generate a trajectory $(S_0, A_0, \dots, S_T, A_T)$ of random length $T \sim \mathrm{Geom}(1-\gamma)$[1] and discard all but the final state-action pair:

[1] We denote by $X \sim \mathrm{Geom}(p)$ a geometrically distributed random variable with support $\{0, 1, 2, \dots\}$, so that $\mathbb{P}(X = n) = (1-p)^n p$.

Algorithm 2 State Sampler (Random Horizon)

Input: Policy $\pi_{\boldsymbol{\theta}}$, discount factor $0 < \gamma < 1$.

$S_0 \sim p_0$
$T \sim \text{Geom}(1-\gamma)$
for $t = 0, 1, 2 \ldots, T-1$ **do**
    $A_t \sim \pi_{\boldsymbol{\theta}}(\cdot|S_t)$
    $S_{t+1} \sim p(\cdot|S_t, A_t)$
end for
return S_T

Lemma 1 *Let T be either the stopping time of Algorithm 1 or the random horizon of Algorithm 2. Then, $\mathbb{P}(S_T = s) = d^{\pi_{\boldsymbol{\theta}}}(s)$.*

Proof. In both cases:

$$\begin{aligned}\mathbb{P}(S_T = s) &= \sum_{t=0}^{\infty} \mathbb{P}(S_T = s | T = t)\mathbb{P}(T = t) \\ &= \sum_{t=0}^{\infty} \mathbb{P}(S_t = s)\gamma^t(1-\gamma) \\ &= d^{\pi_{\boldsymbol{\theta}}}(s).\end{aligned}$$

□

In both cases, sampling one (independent) state-action pair from the occupancy measure requires generating an entire trajectory. We will later see that, in practice, state and actions observed during a continual interaction with the MDP are interpreted as samples from the occupancy measure, provided that rewards are discounted accordingly, often ignoring the fact that states along the trajectory are not independent. Let us prove that Equation (1) is indeed equivalent to the usual notion of expected discounted return:

Lemma 2 *Let $S \sim d^{\pi_{\boldsymbol{\theta}}}, A \sim \pi_{\boldsymbol{\theta}}(\cdot|S)$, and also $S_0 \sim p_0$, $A_t \sim \pi_{\boldsymbol{\theta}}(\cdot|S_t)$, $S_{t+1} \sim p(\cdot|S_t, A_t)$ for $t \geq 0$. Then:*

$$\frac{1}{1-\gamma}\mathbb{E}[r(S, A)] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t)\right].$$

Proof.

$$\begin{aligned}\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t)\right] &= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}[\mathbb{E}[r(S_t, A_t)|S_t]] \\ &= \sum_{t=0}^{\infty} \gamma^t \sum_{s \in \mathcal{S}} \mathbb{P}(S_t = s) \sum_{a \in \mathcal{A}} \pi_{\boldsymbol{\theta}}(a|s) r(s, a)\end{aligned}$$

$$
\begin{aligned}
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\left(\sum_{t=0}^{\infty}\gamma^t\mathbb{P}(S_t=s)\right)\pi_{\boldsymbol{\theta}}(a|s)r(s,a)\\
&= \frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d^{\pi_{\boldsymbol{\theta}}}(s)\pi_{\boldsymbol{\theta}}(a|s)r(s,a)\\
&= \frac{1}{1-\gamma}\mathbb{E}[r(S,A)].
\end{aligned}
$$

□

We are going to prove the policy gradient theorem from the **performance difference lemma**, stated here for parametric stochastic policies:

Lemma 3 (Performance Difference Lemma) *For any $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$,*

$$
J(\boldsymbol{\theta}) - J(\boldsymbol{\theta}') = \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_{\boldsymbol{\theta}}}(s)\\ a\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}}[\mathbb{A}^{\pi_{\boldsymbol{\theta}'}}(s,a)],
$$

where $\mathbb{A}^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$.

Proof. Let $S \sim d^{\pi_{\boldsymbol{\theta}}}$ and $A \sim \pi_{\boldsymbol{\theta}}(\cdot|S)$. By unrolling the Q function for one step:

$$
\begin{aligned}
\mathbb{E}[Q^{\pi_{\boldsymbol{\theta}'}}(S,A) - V^{\pi_{\boldsymbol{\theta}'}}(S)] &= \mathbb{E}[r(S,A)] + \gamma\mathbb{E}\left[\sum_{s'\in\mathcal{S}} p(s'|S,A)V^{\pi_{\boldsymbol{\theta}'}}(s')\right] - \mathbb{E}[V^{\pi_{\boldsymbol{\theta}'}}(S)]\\
&= (1-\gamma)J(\boldsymbol{\theta}) + \gamma\mathbb{E}\left[\sum_{s'\in\mathcal{S}} p(s'|S,A)V^{\pi_{\boldsymbol{\theta}'}}(s')\right] - \mathbb{E}[V^{\pi_{\boldsymbol{\theta}'}}(S)]. \qquad (2)
\end{aligned}
$$

Consider the second term, let $S_0 \sim p_0$, $A_t \sim \pi_{\boldsymbol{\theta}}(\cdot|S_t)$, and $S_{t+1} \sim p(\cdot|S_t, A_t)$:

$$
\begin{aligned}
\gamma\mathbb{E}\left[\sum_{s'\in\mathcal{S}} p(s'|S,A)V^{\pi_{\boldsymbol{\theta}'}}(s')\right] &= \gamma(1-\gamma)\sum_{s\in\mathcal{S}}\sum_{t=0}^{\infty}\gamma^t\mathbb{P}(S_t=s)\sum_{a\in\mathcal{A}}\pi_{\boldsymbol{\theta}}(a|s)\sum_{s'\in\mathcal{S}}p(s'|s,a)V^{\pi_{\boldsymbol{\theta}'}}(s')\\
&= (1-\gamma)\sum_{s'\in\mathcal{S}}\sum_{t=0}^{\infty}\gamma^{t+1}\left(\sum_{s\in\mathcal{S}}\mathbb{P}(S_t=s)\sum_{a\in\mathcal{A}}\pi_{\boldsymbol{\theta}}(a|s)p(s'|s,a)\right)V^{\pi_{\boldsymbol{\theta}'}}(s')\\
&= (1-\gamma)\sum_{s'\in\mathcal{S}}\sum_{t=0}^{\infty}\gamma^{t+1}\mathbb{P}(S_{t+1}=s')V^{\pi_{\boldsymbol{\theta}'}}(s')\\
&= (1-\gamma)\sum_{s'\in\mathcal{S}}\sum_{t=1}^{\infty}\gamma^{t}\mathbb{P}(S_{t}=s')V^{\pi_{\boldsymbol{\theta}'}}(s')\\
&= (1-\gamma)\sum_{s'\in\mathcal{S}}\left(\sum_{t=0}^{\infty}\gamma^t\mathbb{P}(S_t=s') - p_0(s')\right)V^{\pi_{\boldsymbol{\theta}'}}(s')\\
&= \mathbb{E}[V^{\pi_{\boldsymbol{\theta}'}}(S)] - (1-\gamma)\mathbb{E}[V^{\pi_{\boldsymbol{\theta}'}}(S_0)]
\end{aligned}
$$

$$= \mathbb{E}[V^{\pi_{\boldsymbol{\theta}'}}(S)] - (1-\gamma)J(\boldsymbol{\theta}').$$

Plugging this into Equation (2), the $\mathbb{E}[V^{\pi_{\boldsymbol{\theta}'}}]$ terms cancel. Dividing both sides by $(1-\gamma)$ completes the proof. □

The **advantage function** $\mathbb{A}^\pi$ is just the difference between the Q-value function and the V-function of policy π. The *advantage* $\mathbb{A}^\pi(s,a)$ is the extra reward that is obtained by deviating (just for one step) from policy π in state s and playing action a instead. This can be negative, in which case playing a is worse than just following the policy. Here are two important properties of the advantage function that can be easily verified:

1. $\mathbb{E}_{a\sim\pi(\cdot|s)}[\mathbb{A}^\pi(s,a)] = 0$ for all s, π.
2. If π^* is an optimal policy, then $\mathbb{A}^{\pi^*}(s,a) \leq 0$ for all s, a. This may not hold for a best-in-class policy when the policy class is a proper subset of the space of all policies.

The advantage function plays a fundamental role in the policy gradient theorem for the discounted setting.

Theorem 4 (Policy Gradient Theorem—Discounted Setting) *Let $\pi_{\boldsymbol{\theta}}$ be a differentiable stochastic policy. Then:*

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

Proof. Fix $\boldsymbol{\theta}'$ and differentiate both sides of the performance difference lemma w.r.t. $\boldsymbol{\theta}$. Using the chain rule and the log trick:

$$\nabla J(\boldsymbol{\theta}) = \frac{1}{1-\gamma}\mathbb{E}[\nabla \log d^{\pi_{\boldsymbol{\theta}}}(S)\mathbb{A}^{\pi_{\boldsymbol{\theta}'}}(S,A)] + \frac{1}{1-\gamma}\mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A|S)\mathbb{A}^{\pi_{\boldsymbol{\theta}'}}(S,A)],$$

where $S \sim d^{\pi_{\boldsymbol{\theta}}}$ and $A \sim \pi_{\boldsymbol{\theta}}(\cdot|S)$. This holds for any $\boldsymbol{\theta}' \in \Theta$, so let $\boldsymbol{\theta}' = \boldsymbol{\theta}$:

$$\nabla J(\boldsymbol{\theta}) = \frac{1}{1-\gamma}\mathbb{E}[\nabla \log d^{\pi_{\boldsymbol{\theta}}}(S)\mathbb{A}^{\pi_{\boldsymbol{\theta}}}(S,A)] + \frac{1}{1-\gamma}\mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A|S)\mathbb{A}^{\pi_{\boldsymbol{\theta}}}(S,A)].$$

By the tower rule and the first property of the advantage function, the first term is zero:

$$\mathbb{E}[\nabla \log d^{\pi_{\boldsymbol{\theta}}}(S)\mathbb{A}^{\pi_{\boldsymbol{\theta}}}(S,A)] = \mathbb{E}[\nabla \log d^{\pi_{\boldsymbol{\theta}}}(S)\mathbb{E}[\mathbb{A}^{\pi_{\boldsymbol{\theta}}}(S,A) \mid S]] = 0.$$

□

The following variant of the policy gradient theorem also holds:

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

This is equivalent to the original statement because the V-function does not contribute anything to the expectation. Recall the analysis of baselines from the previous lecture:

$$\mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A|S) V^{\pi_{\boldsymbol{\theta}}}(S)] = \mathbb{E}[V^{\pi_{\boldsymbol{\theta}}}(S) \mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A|S) \,|\, S]] = 0.$$

Indeed, the V-function can be seen as a state-dependent baseline for variance reduction. More in general:

$$\nabla J(\boldsymbol{\theta}) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi_{\boldsymbol{\theta}}}(s) \\ a \sim \pi_{\boldsymbol{\theta}}(\cdot|s)}} [\nabla \log \pi_{\boldsymbol{\theta}}(a|s) \left(Q^{\pi_{\boldsymbol{\theta}}}(s,a) - b(s)\right)],$$

for any baseline function b that does not depend on the action.

Similarly to REINFORCE, we can construct an unbiased model-free policy gradient estimator based on the policy gradient theorem:

Algorithm 3 Monte Carlo Policy Gradient Estimation (Discounted Setting)

Input: Policy parameters $\boldsymbol{\theta}$, baseline function b

$T \sim \text{Geom}(1-\gamma)$
$S_0 \sim p_0$
for $t = 0, 1, \ldots, T-1$ **do**
    Play $A_t \sim \pi_{\boldsymbol{\theta}}(\cdot|S_t)$, observe S_{t+1}
end for
$\Delta \sim \text{Geom}(1-\gamma)$
$G \leftarrow 0$
for $t = T, T+1, \ldots, T+\Delta$ **do**
    Play $A_t \sim \pi_{\boldsymbol{\theta}}(\cdot|S_t)$, observe $r(S_t, A_t)$ and S_{t+1}
    $G \leftarrow G + r(S_t, A_t)$
end for
return $\frac{1}{1-\gamma} \nabla \log \pi_{\boldsymbol{\theta}}(A_T|S_T)(G - b(S_T))$

Note that rewards are discounted *implicitly* by stopping at the random horizon, which is equivalent to stopping with probability $1-\gamma$ at each timestep.

Theorem 5 *The output of Algorithm 3 is an unbiased estimate of* $\nabla J(\boldsymbol{\theta})$.

Proof.

$$\begin{aligned}
\mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A_T|S_T)(G - b(S_T))] &= \mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A_T|S_T) \mathbb{E}[G \,|\, S_T, A_T]] \\
&\quad - \mathbb{E}[b(S_T) \mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A_T|S_T) \,|\, S_T]] \\
&= \mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A_T|S_T) \mathbb{E}[G \,|\, S_T, A_T]]
\end{aligned}$$

Let us focus on the inner term first (hiding the conditioning on S_T, A_T to reduce clutter):

$$\mathbb{E}[G \,|\, S_T, A_T] = \mathbb{E}\left[\sum_{t=T}^{T+\Delta} r(S_t, A_t)\right]$$

$$
\begin{aligned}
&= \mathbb{E}\left[\mathbb{E}\left[\sum_{t=T}^{T+\Delta} r(S_t, A_t) \,\middle|\, \Delta\right]\right] \\
&= \sum_{h=0}^{\infty} \gamma^h (1-\gamma) \mathbb{E}\left[\sum_{t=T}^{T+\Delta} r(S_t, A_t) \,\middle|\, \Delta = h\right] \\
&= (1-\gamma)\mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h \sum_{t=T}^{T+h} r(S_t, A_t)\right] \\
&= (1-\gamma)\mathbb{E}\left[\sum_{t=T}^{\infty} r(S_t, A_t) \sum_{h=t-T}^{\infty} \gamma^h\right] \\
&= \mathbb{E}\left[\sum_{t=T}^{\infty} \gamma^{t-T} r(S_t, A_t)\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{t} r(S_{T+t}, A_{T+t})\right] \\
&= Q^{\pi_{\boldsymbol{\theta}}}(S_T, A_T).
\end{aligned}
$$

Similarly:

$$
\begin{aligned}
\mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A_T|S_T) Q^{\pi_{\boldsymbol{\theta}}}(S_T, A_T)] &= \mathbb{E}[\mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A_T|S_T) Q^{\pi_{\boldsymbol{\theta}}}(S_T, A_T) \,|\, T]] && (3)\\
&= \sum_{t=0}^{\infty} \gamma^t (1-\gamma) \mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A_T|S_T) Q^{\pi_{\boldsymbol{\theta}}}(S_T, A_T) \,|\, T = t] && (4)\\
&= (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{E}[\nabla \log \pi_{\boldsymbol{\theta}}(A_t|S_t) Q^{\pi_{\boldsymbol{\theta}}}(S_t, A_t)] && (5)\\
&= (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \sum_{s \in \mathcal{S}} \mathbb{P}(S_t = s) \sum_{a \in \mathcal{A}} \pi_{\boldsymbol{\theta}}(a|s) \nabla \log \pi_{\boldsymbol{\theta}}(a|s) Q^{\pi_{\boldsymbol{\theta}}}(s, a) && (6)\\
&= \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi_{\boldsymbol{\theta}}} \\ a \sim \pi_{\boldsymbol{\theta}}(\cdot|s)}} [\nabla \log \pi_{\boldsymbol{\theta}}(a|s) Q^{\pi_{\boldsymbol{\theta}}}(s, a)] && (7)\\
&= (1-\gamma) \nabla J(\boldsymbol{\theta}), && (8)
\end{aligned}
$$

by the policy gradient theorem. Dividing both sides by $(1-\gamma)$ completes the proof. □

This gradient estimation strategy is impractical for several reasons: the high variance of Monte Carlo Q-function estimates, the fact that many data (the first T rewards of each trajectory) are discarded, and the episodic nature of the simulation, which may not be a good fit for some applications.

Actor-Critic

Actor-critic policy gradient algorithms employ **value function approximation** alongside policy approximation. They are still *policy-based* (opposed to value-based) methods since the parametric value function is only used to support policy gradient estimation.

Consider the Monte Carlo policy gradient estimate from Algorithm 3 (line 12): the Q-value of the state-action pair (S_T, A_T) is estimated by the Monte Carlo *return-to-go* estimate G_t. Actor-critic replaces this Monte Carlo value estimate with explicit value-function approximation:

$$\widehat{\nabla} J(\boldsymbol{\theta}) = \frac{1}{1-\gamma} \nabla \log \pi_{\boldsymbol{\theta}}(A_T|S_T)(q_{\mathbf{w}}(S_T, A_T) - b(S_T)),$$

where $q_{\mathbf{w}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is our **critic**, a parametric state-action value function approximator with parameters $\mathbf{w} \in \mathbb{R}^{d_w}$ (e.g., a neural network with states and actions as input and weights $\mathbf{w}$). The **actor** is, of course, the policy $\pi_{\boldsymbol{\theta}}$. A common choice of baseline is (an approximation of) the state-value function $V^{\pi_{\boldsymbol{\theta}}}(s)$. This can be an extra critic with its own parameters, or share some parameters with the Q-function approximator. A V-function approximator can also be used to estimate the advantage function $\mathbb{A}^{\pi_{\boldsymbol{\theta}}}$ directly as we will see in the following section.

Value function approximation can make the gradient estimator *biased*, but can significantly reduce the variance by avoiding long sums of rewards, similarly to how temporal difference algorithms trade off variance with bias in policy evaluation. The critic parameters $\mathbf{w}$ are optimized to track the value function of *the current policy*, which changes every time we update the policy parameters. Critic learning is thus a sequence of policy evaluation problems that must be carried out alongside policy improvement. This should remind you of (generalized) policy iteration from dynamic programming. There are several ways to interleave actor and critic updates, we will see a specific one in the following section on advantage estimation.

Advantage Estimation

It is possible, and is the basis of several deep RL algorithms, to design an actor-critic algorithm that uses a V-function approximator as its only critic. This technique relies on the following fact:

Lemma 6 *Fix a policy π and let $S_0 \sim p_0$, $A_t \sim \pi(\cdot|S_t)$, $S_{t+1} \sim p(\cdot|S_t, A_t)$. Let $\delta_t = r(S_t, A_t) + \gamma V^\pi(S_{t+1}) - V^\pi(S_t)$ be the* temporal difference error. *Then:*

$$\mathbb{E}[\delta_t \mid S_t, A_t] = \mathbb{A}^\pi(S_t, A_t). \tag{9}$$

Proof.

$$\begin{aligned}
\mathbb{E}[\delta_t \mid S_t, A_t] &= \mathbb{E}[r(S_t, A_t) + \gamma V^\pi(S_{t+1}) - V^\pi(S_t)] \\
&= r(S_t, A_t) + \gamma \mathbb{E}[V^\pi(S_{t+1}) \mid S_t, A_t] - V^\pi(S_t) \\
&= Q^\pi(S_t, A_t) - V^\pi(S_t) \\
&= \mathbb{A}^\pi(S_t, A_t).
\end{aligned}$$

□

This shows that the temporal difference error is an unbiased estimator of the advantage function if one has access to the state-value function of the policy. Instead, let $V_{\mathbf{w}} : \mathcal{S} \to \mathbb{R}$ be a state-value critic with parameters $\mathbf{w} \in \mathbb{R}^{d_w}$. If $V_{\mathbf{w}} \simeq V^{\pi_{\boldsymbol{\theta}}}$, the following is a good estimator of the advantage function:

$$\widetilde{\delta}_t = r(S_t, A_t) + \gamma v_{\mathbf{w}}(S_{t+1}) - v_{\mathbf{w}}(S_t),$$

although it might be slightly biased. The following pseudocode illustrates a possible actor-critic implementation based on advantage estimation and semi-gradient temporal difference:

Algorithm 4 Simplified Advantage Actor-Critic

Input: Actor parameters $\boldsymbol{\theta}_0$, critic parameters $\mathbf{w}_0$, actor learning rate α, critic learning rate β.

$S_0 \sim p_0$
for $t = 0, 1, \ldots$ **do**
  Play $A_t \sim \pi_{\boldsymbol{\theta}_t}(\cdot|S_t)$, observe $r(S_t, A_t)$ and S_{t+1}
  $\widetilde{\delta}_t = r(S_t, A_t) + \gamma v_{\mathbf{w}_t}(S_{t+1}) - v_{\mathbf{w}_t}(S_t)$
  $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \alpha \frac{1}{1-\gamma} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}_t}(A_t|S_t) \widetilde{\delta}_t$
  $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \beta \widetilde{\delta}_t \nabla_{\mathbf{w}} v_{\mathbf{w}_t}(S_t)$
end for

This is a simplified version of a popular deep RL algorithm called A2C (Advantage Actor Critic), covered in the next lecture. Note that the actor and the critic are updated simultaneously. Since the critic's job is to track the value function of the policy $\pi_{\boldsymbol{\theta}_t}$, which itself is continuously updated, it is a good practice to use a smaller learning rate for the actor compared to the critic ($\alpha \ll \beta$). This makes the policy parameter change more slowly than the critic parameter and the state-value function easier to track. As mentioned, the policy gradient estimates used in this and most practical implementations of actor-critic might be biased. One reason is the error introduced by the critic, although we will see in the next section that this can be avoided in principle. Moreover, the states observed along the infinite trajectory are distributed according to the state-occupancy measure, but are not independent.

Compatible Critic

It is actually possible to construct an unbiased actor-critic policy gradient estimator. We are dropping the baseline in the following for simplicity, as it does not contribute to the bias.

Theorem 7 (Compatible Function Approximation Theorem) *If the Q-value approximator satisfies the following properties:*

1. $\nabla_{\mathbf{w}} q_{\mathbf{w}}(s, a) = \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s)$,
2. $\mathop{\mathbb{E}}_{\substack{s \sim d^{\pi_{\boldsymbol{\theta}}} \\ a \sim \pi_{\boldsymbol{\theta}}(\cdot|s)}} \left[\left(Q^{\pi_{\boldsymbol{\theta}}}(s, a) - q_{\mathbf{w}}(s, a) \right) \nabla_{\mathbf{w}} q_{\mathbf{w}}(s, a) \right] = 0,$

then the actor-critic policy gradient estimator is unbiased:

$$\frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi_{\boldsymbol{\theta}}} \\ a \sim \pi_{\boldsymbol{\theta}}(\cdot|s)}} \left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s) q_{\mathbf{w}}(s, a) \right] = \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}).$$

PROOF. Let $S \sim d^{\pi_{\boldsymbol{\theta}}}$, $a \sim \pi_{\boldsymbol{\theta}}(\cdot|S)$:

$$\begin{aligned} 0 &= \mathbb{E}\left[\left(Q^{\pi_{\boldsymbol{\theta}}}(S,A) - q_{\mathbf{w}}(S,A)\right) \nabla_{\mathbf{w}} q_{\mathbf{w}}(S,A)\right] \\ &= \mathbb{E}\left[\left(Q^{\pi_{\boldsymbol{\theta}}}(S,A) - q_{\mathbf{w}}(S,A)\right) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A|S)\right] \\ &= (1-\gamma)\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) - \mathbb{E}[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A|S) q_{\mathbf{w}}(S,A)]. \end{aligned}$$

□

The first condition can be satisfied by using *linear function approximation* with the score function as features:

$$q_{\mathbf{w}}(s,a) = \mathbf{w}^\top \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s).$$

The second condition can be satisfied by minimizing the mean-squared error with respect to the critic parameters:

$$\mathbf{w} \in \arg\min_{\widetilde{\mathbf{w}} \in \mathbb{R}^{d_w}} \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi_{\boldsymbol{\theta}}} \\ a \sim \pi_{\boldsymbol{\theta}}(\cdot|s)}} \left[\left(Q^{\pi_{\boldsymbol{\theta}}}(s,a) - q_{\mathbf{w}}(s,a)\right)^2\right],$$

which can be solved approximately using on-policy data. The resulting Q-function approximator is the **compatible critic** of policy $\pi_{\boldsymbol{\theta}}$. In practice, more expressive critics are usually preferred (e.g., based on task-specific features or on deep neural networks), although they might introduce some bias in the policy gradient estimation.

Natural Actor-Critic

A surprising relation between the compatible critic and the natural policy gradient provides an implementation of the natural policy gradient update that does not require to estimate nor invert the Fisher information matrix. The Fisher information matrix for the discounted setting is defined as follows:

$$F(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi_{\boldsymbol{\theta}}} \\ a \sim \pi_{\boldsymbol{\theta}}(\cdot|s)}} \left[\nabla \log \pi_{\boldsymbol{\theta}}(a|s) \nabla \log \pi_{\boldsymbol{\theta}}(a|s)^\top\right].$$

Theorem 8 *Let $q_{\mathbf{w}}$ be a compatible critic for policy $\pi_{\boldsymbol{\theta}}$ satisfying the two conditions from Theorem 7. Then:*

$$F(\boldsymbol{\theta})^{-1} \nabla J(\boldsymbol{\theta}) = \frac{1}{1-\gamma} \mathbf{w}.$$

PROOF. By Theorem 7:

$$\begin{aligned} \nabla J(\boldsymbol{\theta}) &= \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi_{\boldsymbol{\theta}}} \\ a \sim \pi_{\boldsymbol{\theta}}(\cdot|s)}} \left[\nabla \log \pi_{\boldsymbol{\theta}}(a|s) q_{\mathbf{w}}(s,a)\right] \\ &= \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi_{\boldsymbol{\theta}}} \\ a \sim \pi_{\boldsymbol{\theta}}(\cdot|s)}} \left[\nabla \log \pi_{\boldsymbol{\theta}}(a|s) \nabla \log \pi_{\boldsymbol{\theta}}(a|s)^\top \mathbf{w}\right] \\ &= \frac{1}{1-\gamma} F(\boldsymbol{\theta}) \mathbf{w}. \end{aligned}$$

□

This provides a strikingly simple natural policy gradient update:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \frac{\alpha}{1-\gamma}\mathbf{w},$$

effectively reducing natural policy gradient to policy evaluation.

Deterministic Policy Gradient

Optimizing deterministic policies by policy gradient is also possible, provided the Q-function is differentiable with respect to (continuous) actions.

Theorem 9 (Deterministic Policy Gradient Theorem) *Let $\mu_{\boldsymbol{\theta}} : \mathcal{S} \to \mathcal{A}$ be a deterministic parametric policy. Then:*

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{s\sim d^{\mu_{\boldsymbol{\theta}}}} \left[\nabla_{\boldsymbol{\theta}} \mu_{\boldsymbol{\theta}}(s) \nabla_a Q^{\mu_{\boldsymbol{\theta}}}(s,a)|_{a=\mu_{\boldsymbol{\theta}}(s)}\right].$$

Proof. Consider again the performance difference lemma, written here for deterministic policies:

$$J(\boldsymbol{\theta}) - J(\boldsymbol{\theta}') = \frac{1}{1-\gamma}\mathbb{E}\left[Q^{\mu_{\boldsymbol{\theta}'}}(S, \mu_{\boldsymbol{\theta}}(S)) - V^{\mu_{\boldsymbol{\theta}'}}(S)\right],$$

where $S \sim d^{\mu_{\boldsymbol{\theta}}}$. Fix $\boldsymbol{\theta}'$ and differentiate w.r.t. $\boldsymbol{\theta}$:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) &= \frac{1}{1-\gamma}\mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \log d^{\mu_{\boldsymbol{\theta}}}(S)\left(Q^{\mu_{\boldsymbol{\theta}'}}(S, \mu_{\boldsymbol{\theta}}(S)) - V^{\mu_{\boldsymbol{\theta}'}}(S)\right)\right] + \frac{1}{1-\gamma}\mathbb{E}\left[\nabla_{\boldsymbol{\theta}} Q^{\mu_{\boldsymbol{\theta}'}}(S, \mu_{\boldsymbol{\theta}}(S))\right] \\
&= \frac{1}{1-\gamma}\mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \log d^{\mu_{\boldsymbol{\theta}}}(S)\left(Q^{\mu_{\boldsymbol{\theta}'}}(S, \mu_{\boldsymbol{\theta}}(S)) - V^{\mu_{\boldsymbol{\theta}'}}(S)\right)\right] \\
&\quad + \frac{1}{1-\gamma}\mathbb{E}\left[\nabla_a Q^{\mu_{\boldsymbol{\theta}'}}(S,a)|_{a=\mu_{\boldsymbol{\theta}}(S)} \nabla_{\boldsymbol{\theta}} \mu_{\boldsymbol{\theta}}(S)\right].
\end{aligned}$$

Since this holds for any $\boldsymbol{\theta}'$, let $\boldsymbol{\theta}' = \boldsymbol{\theta}$:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) &= \frac{1}{1-\gamma}\mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \log d^{\mu_{\boldsymbol{\theta}}}(S)\left(Q^{\mu_{\boldsymbol{\theta}}}(S, \mu_{\boldsymbol{\theta}}(S)) - V^{\mu_{\boldsymbol{\theta}}}(S)\right)\right] \\
&\quad + \frac{1}{1-\gamma}\mathbb{E}\left[\nabla_a Q^{\mu_{\boldsymbol{\theta}}}(S,a)|_{a=\mu_{\boldsymbol{\theta}}(S)} \nabla_{\boldsymbol{\theta}} \mu_{\boldsymbol{\theta}}(S)\right].
\end{aligned}$$

Since $Q^{\mu_{\boldsymbol{\theta}}}(S, \mu_{\boldsymbol{\theta}}(S)) = V^{\mu_{\boldsymbol{\theta}}}(S)$, the first term is zero. □

Deterministic policy gradient algorithms require a Q-function critic that is differentiable with respect parameters *and* actions, and an extrinsic exploration mechanism such as Gaussian action perturbation. For this reason, deterministic policy gradient algorithms are naturally actor-critic and off-policy. We will see a deep deterministic policy gradient algorithm (DDPG) in the next lecture.