Notes on Importance Sampling and Policy Gradient

Nan Jiang

November 22, 2023

1 Importance Sampling

1.1 Estimating expectation using samples from a different distribution

Consider the problem of estimating $\mathbb{E}_{x\sim p}[f(x)]$ for distribution $p \in \Delta(\mathcal{X})$ and function $f : \mathcal{X} \to \mathbb{R}$. If we can sample $x \sim p$, the standard Monte-Carlo estimate is $f(x)$, and averaging such estimates over multiple i.i.d. samples of x will give us an accurate estimate of $\mathbb{E}_{x\sim p}[f(x)]$. This is particularly useful if it is easy to sample from p but difficult to calculate the integral in $\mathbb{E}_{x\sim p}[f(x)]$.

Now what if we cannot sample from p, but have access to $x \sim q$ for some other distribution $q \in \Delta(\mathcal{X})$? It turns out that, if p is fully supported on q, that is, for all $x \in \mathcal{X}$ where $p(x) > 0$ we have $q(x) > 0$, then the following *importance weighted* estimator also gives an unbiased estimate of $\mathbb{E}_{x\sim p}[f(x)]$:

$$\frac{p(x)}{q(x)} f(x). \tag{1}$$

To verify unbiasedness:

$$\mathbb{E}_{x\sim q}\left[\frac{p(x)}{q(x)} f(x)\right] = \sum_{x\in\mathcal{X}} q(x)\frac{p(x)}{q(x)} f(x) = \sum_{x\in\mathcal{X}} p(x) f(x) = \mathbb{E}_{x\sim p}[f(x)].$$

$p(x)/q(x)$ has many names: importance weight, importance ratio, or inverse propensity score (IPS). A useful property of importance ratio to keep in mind is that

$$\mathbb{E}_{x\sim q}\left[\frac{p(x)}{q(x)}\right] = 1. \tag{2}$$

Bibliographical remarks Traditionally, the term "importance sampling" (IS) refers to the procedure of *designing* the distribution q to achieve *lower* variance than the standard MC estimate. The resulting estimator is called importance weighted estimator or IPS estimator. In RL, it is often the case that q is given and the IPS estimator has higher variance than on-policy MC, so IS is not a very appropriate term in this context, despite its prevalence in literalture.

1.2 Application to contextual bandits

Consider a contextual bandit problem with context space $\mathcal{X}$, discrete action space $\mathcal{A}$, and $R : \mathcal{X} \times \mathcal{A} \to \Delta([0,1])$ maps (x, a) to a distribution over rewards with bounded range $[0, 1]$. Let $d_0 \in \Delta(\mathcal{X})$ be the context distribution.

Suppose we have collected a dataset $\{(x, a, r)\}$ by sampling $x \sim d_0$, $a \sim \pi_b(x)$ where π_b is a stochastic *behavior* policy, and $r \sim R(x, a)$. Can we use this dataset to estimate $J(\pi) := \mathbb{E}[r|a \sim \pi]$, the value of a *target* policy π that is different from π_b?

It turns out that, if π is fully supported on π_b, then we can use importance sampling to form the following unbiased estimate: for a single sample (x, a, r), the estimate is

$$\rho\, r, \text{ where } \rho = \frac{\pi(a|x)}{\pi_b(a|x)}. \tag{3}$$

To verify unbiasedness, let p denote the joint distribution over (x, a, r) induced by π and q denote that induced by π_b. By importance sampling we have

$$\mathbb{E}[r|a \sim \pi] = \mathbb{E}_{(x,a,r)\sim p}[r] = \mathbb{E}_{(x,a,r)\sim q}\left[\frac{p(x, a, r)}{q(x, a, r)}\, r\right].$$

Now let's take a closer look at the importance ratio:

$$\frac{p(x, a, r)}{q(x, a, r)} = \frac{d_0(x)\pi(a|x)R(r|x, a)}{d_0(x)\pi_b(a|x)R(r|x, a)} = \frac{\pi(a|x)}{\pi_b(a|x)} = \rho.$$

The nice thing here is that $d_0(x)$ and $R(r|x, a)$ are inherent properties of the process (and unknown in most occasions) and do not change when we deploy different policies, so they cancel out in the importance ratio. Later we will see similar phenomenon in the multi-step case.

Variance analysis While the estimator is unbiased, the variance can be quite large when π and π_b are very different from each other. Below we analyze a typical setting where π_b is the uniformly random policy $a \sim U$ and π is a deterministic policy.

Let $K := |\mathcal{A}|$. With slight abuse of notation (we will treat π as a mapping from x to deterministic action below), the importance ratio can be written as $\rho = \frac{\mathbb{I}[a=\pi(x)]}{1/K}$.

While it is somewhat difficult to characterize the variance for a general reward function, it is instructive to consider a special case where r is a *deterministic constant*. Note that in this case the standard MC estimate has 0 variance. What is the variance of IS?

$$\begin{aligned}
&\mathbb{V}[\rho r|a \sim U] = r^2\mathbb{V}[\rho|a \sim U] \\
&= r^2(\mathbb{E}[\rho^2|a \sim U] - (\mathbb{E}[\rho|a \sim U])^2) \\
&= r^2(\mathbb{E}[\rho^2|a \sim U] - 1) && \text{(the mean of } \rho \text{ is always } 1) \\
&= r^2\left(\mathbb{E}\left[\frac{\mathbb{I}[a = \pi(x)]}{1/K^2}\,\Big|\, a \sim U\right] - 1\right) \\
&= r^2(K - 1).
\end{aligned}$$

So the variance of IS grows almost linearly with the number of actions K. One way to think about it is that IS is simply picking out all data points where a happens to be the action that π wants to take. In general only $1/K$ data points are "valid" so the effective sample size is K times smaller than what it appears to be.

For general reward distributions with bounded range $[0, 1]$, we can similarly upper bound the variance:

$$\mathbb{V}[\rho r|a \sim U] \le \mathbb{E}[\rho^2 r^2|a \sim U] \le \mathbb{E}[\rho^2|a \sim U] = K.$$

And the special case of deterministic constant r shows that this inequality is roughly tight.

Concentration While the variance of ρr is practically high, it should be considered as a "low-variance" random variable w.r.t. its range: If $r \in [0,1]$ almost surely, $\rho r \in [0, K]$, and we have shown above that $\mathbb{V}[\rho r] \le K$, so the range is roughly equal to the variance.

As a result, when we prove concentration of IS estimator, using Bernstein's will give significantly better results than Hoeffding's: With Hoeffding's, we get $K\sqrt{\frac{1}{2n}\ln\frac{2}{\delta}}$ as the high probability bound on deviation. With Bernstein's, we get

$$\sqrt{\frac{2K}{n}\ln\frac{2}{\delta}} + \frac{2K}{3n}\ln\frac{2}{\delta}.$$

The second term is a lower order term compared to the first term when n is large, so we have a deviation bound of $O(\sqrt{\frac{K}{n}\ln\frac{1}{\delta}})$. Compared to Hoeffding's, we are saving a factor of $\sqrt{K}$ by using Berstein's.

Variance reduction by weighted importance sampling (WIS) When π_b is uniform and π is deterministic, IS is basically picking out data points where a happens to match the action that π wants to take, and discarding everything else. However, IS is *taking average of r within this subsample*, how come that its variance is not 0 when r is constant, as we have seen above?

It turns out that IS is doing something slightly trickier. Let $(x_i, a_i, r_i)_{i=1}^n$ be the dataset. The final estimator given by IS is:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\mathbb{I}[a_i = \pi(x_i)]}{1/K} r_i = \frac{1}{n/K}\sum_{i: a_i = \pi(x_i)} r_i.$$

It's indeed adding up the rewards within that subsample. But when it comes to normalization, instead of normalizing using the subsample size $|\{i : a_i = \pi(x_i)\}|$, IS normalizes using the *expected size* n/K. The variance in IS for constant r essentially comes from the randomness of the subsample size.

So an obvious improvement would be to normalize by $|\{i : a_i = \pi(x_i)\}|$. Generalizing this idea beyond the specific setting, we get the weighted IS: let $\rho_i := \pi(a_i|x_i)/\pi_b(a_i|x_i)$, WIS forms the following estimate

$$\frac{1}{\sum_{i=1}^{n}\rho_i}\sum_{i=1}^{n}\rho_i\, r_i. \tag{4}$$

It is easy to verify that in the special case of uniform π_b and deterministic π, WIS indeed has 0 variance.

WIS is generally biased: It can even run into the issue of division by 0 when no a_i matches $\pi(x_i)$. On the other hand, its asymptotic behavior is similar to IS, as $\sum_{i=1}^{n}\rho_i \approx n/K$ when n is large, so WIS is also a consistent estimator.

Variance reduction by control variate Another way to fix the issue in the example of constant r is the following: Recall that it's possible to shift rewards around without making any actual changes. So we can always subtract a constant c from all rewards, perform IS, and add c back to the final estimate; Effectively we create a new CB problem where the reward is always lower than that in the current problem by a constant c, and we estimate the value of a policy in the new problem and infer its value in the current problem. It is easy to see that the variance of IS in the new problem is $(r-c)^2(K-1)$, so if we set $c = r$, the estimator has 0 variance!

In fact this idea can also be generalized: Suppose we are given $\hat{Q} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ such that $\hat{Q}(x,a) \approx \mathbb{E}[r|x,a]$, then the following estimator is also unbiased [1]: for every (x,a,r),

$$\mathbb{E}_{a' \sim \pi}[\hat{Q}(x,a')] + \rho \left(r - \hat{Q}(x,a) \right). \tag{5}$$

As long as $\hat{Q}$ is deterministic w.r.t. the data used for off-policy evaluation (which means, if $\hat{Q}$ is estimated from data, it has to use a separate dataset), the first term and $\rho\hat{Q}(s,a)$ will cancel each other in expectation, leaving alone ρr which is IS. In fact, IS can be viewed as a special case of DR with $\hat{Q} \equiv 0$.

The estimator is called a *doubly robust* (DR) estimator [1], for the following reason: sometimes π_b is unknown and ρ needs to be estimated from data, introducing bias to IS. In DR, however, if $\hat{Q}(x,a) = \mathbb{E}[r|x,a]$ then the estimator is unbiased with arbitrarily badly estimated ρ; on the other hand, if ρ is exact, then the estimator is also unbiased even with an arbitrarily bad $\hat{Q}$, hence "doubly" robust against potential biases.

1.3 Application to multi-step RL

Consider finite-horizon MDP $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma, d_0)$ where d_0 is the initial state distribution. For simplicity assume all trajectories terminate within H steps from the initial state. Given sample trajectories generated from π_b, we can estimate $J(\pi)$, the average return of a different policy π, using importance sampling, as long as π is supported on π_b.

Again, let p be the joint distribution over the entire trajectory $\tau := (s_1, a_1, r_1, s_2, \ldots, s_H, a_H, r_H)$ induced by π, and q be that induced by π_b. We have

$$\begin{aligned} J(\pi) &= \mathbb{E}\left[\sum_{h=1}^{H} \gamma^{h-1} r_h \;\middle|\; a_{1:H} \sim \pi\right] = \mathbb{E}_{\tau \sim p}\left[\sum_{h=1}^{H} \gamma^{h-1} r_h\right] = \mathbb{E}_{\tau \sim q}\left[\frac{p(\tau)}{q(\tau)} \sum_{h=1}^{H} \gamma^{h-1} r_h\right] \\ &= \mathbb{E}_{\tau \sim q}\left[\frac{d_0(s_1)\pi(a_1|s_1)R(r_1|s_1,a_1)P(s_2|s_1,a_1)\cdots\pi(a_H|s_H)R(r_H|s_H,a_H)}{d_0(s_1)\pi_b(a_1|s_1)R(r_1|s_1,a_1)P(s_2|s_1,a_1)\cdots\pi_b(a_H|s_H)R(r_H|s_H,a_H)} \sum_{h=1}^{H} \gamma^{h-1} r_h\right] \\ &= \mathbb{E}_{\tau \sim q}\left[\frac{\pi(a_1|s_1)\cdots\pi(a_H|s_H)}{\pi_b(a_1|s_1)\cdots\pi_b(a_H|s_H)} \sum_{h=1}^{H} \gamma^{h-1} r_h\right] = \mathbb{E}\left[\frac{\pi(a_1|s_1)\cdots\pi(a_H|s_H)}{\pi_b(a_1|s_1)\cdots\pi_b(a_H|s_H)} \sum_{h=1}^{H} \gamma^{h-1} r_h \;\middle|\; a_{1:H} \sim \pi_b\right]. \end{aligned}$$

So the expression in the bracket is an unbiased estimate of $J(\pi)$. Let $\rho_h := \pi(a_h|s_h)/\pi_b(a_h|s_h)$ and $\rho_{1:h}$ be a shorthand for $\prod_{h'=1}^{h} \rho_{h'}$, the **per-trajectory** IS estimator is [2, 3]:

$$\rho_{1:H} \sum_{h=1}^{H} \gamma^{h-1} r_h. \tag{6}$$

In the special case where π_b is uniformly random and π is deterministic, and reward is a non-zero constant that only occurs at the end of every trajectory, it is easy to verify that the estimator's variance is proportional to K^H, which is expoential in the problem horizon. (In fact we know that this is inevitable in the worst-case; see [4].)

Per-step IS An improved version of the estimator leverages the fact that the rewards are additive and can be treated separately. For r_h, the actions $a_{h+1:H}$ do not really matter any more so we only need to multiply it with the cumulative importance ratio up to step h. The **per-step** IS estimator is:

$$\sum_{h=1}^{H} \gamma^{h-1} \rho_{1:h}\, r_h. \tag{7}$$

The verification of its unbiasedness is left as an exercise.

Alternative interpretation of per-step IS, and DR for the multi-step setting [4] A re-expression of Eq.(7) reveals that per-step IS can be viewed as bandit IS recursively applied at each time step: Define $v_0 := 0$, and

$$v_{H-h+1} := \rho_h (r_h + \gamma v_{H-h}). \tag{8}$$

One can verify that v_H is exactly the same as Eq.(7). This recursive expression gives a new inductive proof of the unbiasedness of per-step IS: Assume that v_{H-h} is an unbiased estimate of $V^\pi(s_{h+1})$ for the s_{h+1} observed in data. (The base case trivially holds as there are no more steps when $h = H$ and $v_0 = 0$.) Then $r_h + \gamma v_{H-h}$ is an unbiased estimate of $Q^\pi(s_h, a_h)$ for (s_h, a_h) observed in data.

Recall that in the data a_h is chosen according to π_b. Then at step h we essentially have the following bandit problem: s_h is the context, a_h is the arm, and the random reward is $r_h + \gamma v_{H-h}$ with mean $Q^\pi(s_h, a_h)$. Therefore, $\rho_h(r_h + \gamma v_{H-h})$ is an unbiased bandit IS estimator for $Q^\pi(s_h, \pi) = V^\pi(s_h)$, so the induction holds.

This observation also makes it straightforward to apply the DR trick in the multi-step setting: Let $\hat{Q}^\pi$ be our estimated Q-value function for this problem. The following DR estimator [4]:

$$v_{H-h+1}^{DR} := \mathbb{E}_{a\sim\pi}[\hat{Q}^\pi(s_h, a)] + \rho_h \left(r_h + \gamma v_{H-h}^{DR} - \hat{Q}^\pi(s_h, a_h)\right) \tag{9}$$

is again an unbiased estimator for $J(\pi)$.

WIS The IS and the DR estimators in the multi-step setting can be similarly extended to their weighted versions; see [3, 5, 6].

Variance of per-step IS The variance of Eq.(7) also satisifies an interesting recursion, which has important implications outside off-policy evaluation. Let $\mathbb{V}_h[\cdot]$ and $\mathbb{E}_h[\cdot]$ denote conditional variance and expectation, respectively, conditioned on $s_1, a_1, r_1, \ldots, s_{h-1}, a_{h-1}, r_{h-1}$. For simplicity assume reward is a deterministic function of state and action, then

$$\begin{aligned}
&VV_h[v_{H-h+1}] = \mathbb{E}_h[v_{H-h+1}^2] - (\mathbb{E}_h[v_{H-h+1}])^2 \\
&= \mathbb{E}_h[v_{H-h+1}^2] - (\mathbb{E}_h[V^\pi(s_h)])^2 && (V^\pi(s_h) = \mathbb{E}_h[v_{H-h+1} \mid s_h]) \\
&= \mathbb{E}_h[(\rho_h Q^\pi(s_h, a_h) + \rho_h\,(r_h + \gamma v_{H-h} - Q^\pi(s_h, a_h)))^2] - (\mathbb{E}_h[V^\pi(s_h)])^2 \\
&= \mathbb{E}_h[(\rho_h Q^\pi(s_h, a_h))^2] + \mathbb{E}_h[\rho_h^2\,(r_h + \gamma v_{H-h} - Q^\pi(s_h, a_h)))^2] - (\mathbb{E}_h[V^\pi(s_h)])^2 \\
&= \mathbb{E}_h[(V^\pi(s_h) + \rho_h Q^\pi(s_h, a_h) - V^\pi(s_h))^2] + \gamma^2 \mathbb{E}_h[\rho_h^2\,(v_{H-h} - V^\pi(s_{h+1}))^2] - (\mathbb{E}_h[V^\pi(s_h)])^2 \\
&= \mathbb{E}_h[V^\pi(s_h))^2] + \mathbb{E}_h[\mathbb{V}_h[\rho_h Q^\pi(s_h, a_h) \mid s_h]] + \gamma^2 \mathbb{E}_h[\rho_h^2 \mathbb{V}_{h+1}[v_{H-h}]] - (\mathbb{E}_h[V^\pi(s_h)])^2 \\
&= \mathbb{V}_h[V^\pi(s_h)] + \mathbb{E}_h[\mathbb{V}_h[\rho_h Q^\pi(s_h, a_h) \mid s_h]] + \gamma^2 \mathbb{E}_h[\rho_h^2 \mathbb{V}_{h+1}[v_{H-h}]].
\end{aligned}$$

As a special case, if $\pi_b = \pi$ (on-policy) and the policy is deterministic, $\rho_h \equiv 1$ and the second term on the RHS is 0 (because a_h is not random conditioned on s_h). In this case, the above equation becomes

$$\mathbb{V}_h[v_{H-h+1}] = \mathbb{V}_h[V^\pi(s_h)] + \gamma^2 \mathbb{E}_h[\mathbb{V}_{h+1}[v_{H-h}]].$$

This is sometimes called the *Bellman equation for variance*, as it resembles the Bellman equation for policy evaulation except that the "reward" is replaced by the conditional variance of value function, and the discount factor is squared. Expanding this recursion yields

$$\mathbb{V}[v_H] = \sum_{h=1}^{H} \gamma^{2(h-1)} \mathbb{E}_{s_{h-1}\sim d_{h-1}^\pi, a_{h-1}\sim\pi} \left[\mathbb{V}_{s_h\sim P(s_{h-1},a_{h-1})}[V^\pi(s_h)]\right]. \tag{10}$$

The object on the RHS is the variance of V^π w.r.t. the transition dynamics, averaged on the distribution over states and actions induced by π.[1] Recall that this is the kind of object we are dealing with in the analysis of tabular methods, and using Hoeffding's inequality to derive concentration bounds is implicitly assuming maximum variance for every single transition.

However, while $\mathbb{V}_{s_h\sim P(s_{h-1},a_{h-1})}[V^\pi(s_h)]$ can possibly have $\Theta(V_{\max}^2)$ variance for an individual state-action pair, such worst-case variance cannot occur throughout the entire state space for the following reason: If every (s,a) has $\mathbb{V}_{s'\sim P(s,a)}[V^\pi(s_h)] = \Theta(V_{\max}^2)$, the RHS of Eq.(10) should be $\Theta(HV_{\max}^2)$ (ignoring γ for now); however, since v_H is the MC estimate of return, we have $\mathbb{V}[v_H] = O(V_{\max}^2)$! This implies that along the distribution induced by π, the conditional variance of V^π w.r.t. transition distributions sum up to *only* $V_{\max}^2$ across H steps and does not scale with H. In fact, state-of-the-art analyses of tabular RL often exploit this property and obtain tight bounds by Bernstein's inequality [see e.g., 7].

2 Policy gradient

2.1 Derivation

Consider the optimization of $J(\pi)$, the average value of π under initial state distribution. For simplicity assume that all trajectories terminate within H steps. Suppose we are given a parameterized class of stochastic policies $\Pi = \{\pi_\theta : \theta \in \Theta\}$, such that $\pi_\theta(a|s)$ is differentiable with respect to θ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$. By rolling out trajectories using π_θ, we can effectively estimate $\nabla_\theta J(\pi_\theta)$, with an accuracy *independent* of the size of the state space, and perform gradient descent to find a local optimum. The simplest method of this kind is called REINFORCE [8], which we derive below.

For simplicity we assume that reward is a deterministic function of state and action; the result extends to stochastic rewards trivially. Let $R(\tau)$ denote the discounted sum of rewards on a trajectory $\tau = s_1, a_1, r_1, \ldots, s_H, a_H, r_H$, i.e., $R(\tau) = \sum_{h=1}^{H} \gamma^{h-1} r_h$. Similarly let $P^\pi(\tau)$ be the probability of τ

[1] In fact, such decomposition has a very regular pattern: we go through each random variable s_h and (1) take expectation of the estimator conditioned on everything up to s_h, which gives us $V^\pi(s_h)$, (2) take the conditional variance of $V^\pi(s_h)$ (the result of (1)) w.r.t. the "local" randomness of s_h conditioned on everything before s_h, and (3) take the expectation of the conditional variance in (2) w.r.t. the variables before s_h. Note that in steps (1)-(3) we have integrated out all r.v.'s in this process, which has to be the case since the final variance is a deterministic quantity that does not depend on any realization of the r.v.'s.

under policy π. We will drop the θ in the subscript of ∇_θ and π_θ.

$$\begin{aligned}
\nabla J(\pi) &= \sum_\tau R(\tau)\nabla P^\pi(\tau) = \sum_\tau R(\tau)P^\pi(\tau)\nabla \log P^\pi(\tau) \\
&= \sum_\tau R(\tau)P^\pi(\tau)\nabla \log\left(d_0(s_1)\pi(a_1|s_1)P(s_2|s_1,a_1)\cdots\pi(a_H|s_H)\right) \\
&= \sum_\tau R(\tau)P^\pi(\tau)\nabla\left(\log d_0(s_1) + \sum_{h=1}^{H}\log\pi(a_h|s_h) + \sum_{h=1}^{H-1}\log P(s_{h+1}|s_h,a_h)\right) \\
&= \sum_\tau R(\tau)P^\pi(\tau)\nabla\left(\sum_{h=1}^{H}\log\pi(a_h|s_h)\right) = \mathbb{E}_{\tau\sim\pi}\left[R(\tau)\sum_{h=1}^{H}\nabla\log\pi(a_h|s_h)\right].
\end{aligned}$$

This gives a version of REINFORCE: we can compute a stochastic gradient of $\nabla J(\pi)$ by (1) generate a trajectory using π, and (2) compute $R(\tau)\sum_{h=1}^{H}\nabla\log\pi(a_h|s_h)$.

The similarity in the derivation between PG and IS suggests that, conceptually what PG does is essentially (1) use IS to evaluate the return of all policies in a small neighborhood around current π, and (2) compute gradient based on the (approximate) function evaluations.

It is possible to obtain a stochastic gradient with lower variance by decomposing the rewards over multiple steps. This is essentially the difference between per-trajectory & per-step importance sampling: let d^π be the normalized state occupancy of π from initial distribution d_0, and

$$\nabla J(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim d^\pi}\left[(\nabla\log\pi(a|s))Q^\pi(s,a)\right]. \tag{11}$$

Using this equation, we can obtain another version of REINFORCE as follows: (1) generate a trajectory using π, (2) pick a random time-step h with probability $\propto \gamma^h$, (3) compute $\nabla\log\pi(a_h|s_h)\sum_{t=h}^{H}\gamma^{t-h}r_t$ as an unbiased estimate of $\nabla J(\pi)$.

While Eq.(11) can also be derived in the "Monte-Carlo" style as above, below is a simpler proof that uses the recursive structure of Bellman equations [9]:

Proof. Let's start with the simple fact

$$V^\pi(s) = \sum_a \pi(a|s)Q^\pi(s,a).$$

Differentiate both sides:

$$\begin{aligned}
\nabla V^\pi(s) &= \sum_a \big((\nabla\pi(a|s))Q^\pi(s,a) + \pi(a|s)\nabla Q^\pi(s,a)\big) \\
&= \sum_a \big(\pi(a|s)(\nabla\log\pi(a|s))Q^\pi(s,a) + \pi(a|s)\nabla(R(s,a) + \gamma\mathbb{E}_{s'\sim P(s,a)}[V^\pi(s')])\big) \\
&= \sum_a \pi(a|s)\big((\nabla\log\pi(a|s))Q^\pi(s,a) + \gamma\mathbb{E}_{s'\sim P(s,a)}[\nabla V^\pi(s')]\big).
\end{aligned}$$

Now let d_t^π denote the distribution over s_t and a_t induced by policy π from the initial state distribution d_0; we will also write $s_t \sim d_t^\pi$ for its marginal on state. Take the expectation of the above equation w.r.t. $s \sim d_t^\pi$, we have

$$\begin{aligned}
\nabla(\mathbb{E}_{s\sim d_t^\pi}[V^\pi(s)]) &= \mathbb{E}_{s\sim d_t^\pi, a\sim\pi}[(\nabla \log \pi(a|s))Q^\pi(s,a)] + \gamma \mathbb{E}_{s\sim d_t^\pi, a\sim\pi, s'\sim P(s,a)}[\nabla V^\pi(s')] \\
&= \mathbb{E}_{(s,a)\sim d_t^\pi}[(\nabla \log \pi(a|s))Q^\pi(s,a)] + \gamma \mathbb{E}_{s'\sim d_{t+1}^\pi}[\nabla V^\pi(s')] \\
&= \mathbb{E}_{(s,a)\sim d_t^\pi}[(\nabla \log \pi(a|s))Q^\pi(s,a)] + \gamma \mathbb{E}_{(s',a')\sim d_{t+1}^\pi}[(\nabla \log \pi(a'|s'))Q^\pi(s',a')] + \gamma^2 \mathbb{E}_{s''\sim d_{t+2}^\pi}[\nabla V^\pi(s'')] \\
&= \ldots = \sum_{t'=t}^{\infty} \gamma^{t'-t} \mathbb{E}_{(s,a)\sim d_{t'}^\pi}[(\nabla \log \pi(a|s))Q^\pi(s,a)].
\end{aligned}$$

The result follows by noticing that when $t = 1$, the LHS is $\nabla \mathbb{E}_{s\sim d_t^\pi}[V^\pi(s)] = \nabla \mathbb{E}_{s\sim d_0}[V^\pi(s)] = \nabla J(\pi)$, and the RHS is the desired expression as the normalized discounted occupancy is precisely $d^\pi = (1-\gamma)\sum_{t'=1}^{\infty} \gamma^{t'-1} d_{t'}^\pi$. □

Variance reduction in policy gradient A useful property of $\nabla \log \pi(a|s)$ is the following: for any fixed s, if we draw actions $a \sim \pi(s)$, we would have

$$\mathbb{E}_{a\sim\pi(s)}[\nabla \log \pi(a|s)] = \sum_{a\in\mathcal{A}} \nabla \pi(a|s) = \nabla \sum_{a\in\mathcal{A}} \pi(a|s) = 0.$$

Therefore, we can add any function $f : \mathcal{S} \to \mathbb{R}$ to the policy gradient without affecting its unbiasedness as follows:

$$\nabla J(\pi) = \frac{1}{1-\gamma} \mathbb{E}_{(s,a)\sim d^\pi}\left[\nabla \log \pi(a|s)\left(Q^\pi(s,a) - f(s)\right)\right].$$

A popular choice of f is V^π, the value function of π. Of course, V^π is generally unknown and we can only obtain an estimate $\hat{V}^\pi$. One can use dynamic programming methods to estimate $\hat{V}^\pi$ and use it to reduce the variance in policy gradient. One can even go further to replace $Q^\pi(s,a)$ in Eq.(11) with an estimated $\hat{Q}^\pi(s,a)$ to further reduce variance at the cost of introducing bias to the gradient estimate. In general, incorporating estimated value functions into policy gradient methods is known as "actor-critic" [10]: the policy is an "actor" and the value function is a "critic" that assesses the policy's performance and offers guidance into how to improve the policy.

2.2 Analysis

We provide a sketch of typical analysis of PG here; refer to [11**?**] for further details. Roughly speaking, since PG is essentially (stochastic) gradient descent, it is guaranteed to find an approximate stationary point under mild conditions (i.e., even for non-convex problems), where the "size" of the gradient is small: for example, let's say the algorithm outputs $\hat{\pi} = \pi_{\hat{\theta}}$ with a small $\|\nabla J(\hat{\pi})\|$ for $\|\cdot\|$ being 2-norm.[2] The key question is, therefore, when does a small $\|\nabla J(\hat{\pi})\|$ translate to the global optimality of $\hat{\pi}$?

The standard analyses for answering this question is as follows:

[2] This definition applies when $\Theta = \mathbb{R}^d$. In general, especially when Θ is a restricted space, the definition should be $\nabla J(\hat{\pi})^\top(\theta' - \hat{\theta})$ being small for all $\theta' \in \Theta$.

$$\begin{aligned}
J(\pi^\star) - J(\hat{\pi}) &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^\star}}[Q^{\hat{\pi}}(s,\pi^\star) - V^{\hat{\pi}}(s)] && \text{(PD lemma)} \\
&\le \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^\star}}[Q^{\hat{\pi}}(s,\pi^+) - V^{\hat{\pi}}(s)] && (\pi^+ := \pi_{Q^{\hat{\pi}}}) \\
&\le \frac{\|d^{\pi^\star}/d^{\hat{\pi}}\|_\infty}{1-\gamma}\mathbb{E}_{s\sim d^{\hat{\pi}}}[Q^{\hat{\pi}}(s,\pi^+) - Q^{\hat{\pi}}(s,\hat{\pi})] && \text{(change of measure)} \\
&= \frac{\|d^{\pi^\star}/d^{\hat{\pi}}\|_\infty}{1-\gamma}\mathbb{E}_{s\sim d^{\hat{\pi}}}[\sum_a Q^{\hat{\pi}}(s,a)(\pi^+(a|s) - \hat{\pi}(a|s))].
\end{aligned}$$

The change of measure requires that the function inside expectation to be non-negative, which is why we relaxed π^* to π^+ earlier. Now part of the last line is somewhat similar to the PG expression: $\nabla J(\hat{\pi}) = \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim d^{\hat{\pi}}}\left[\nabla \log \hat{\pi}(a|s) Q^{\hat{\pi}}(s,a)\right] = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\hat{\pi}}}\left[\sum_a Q^{\hat{\pi}}(s,a)\nabla\hat{\pi}(a|s)\right]$. There are two differences: (1) $\pi^+ - \hat{\pi}$ vs. $\nabla\hat{\pi}$, and (2) the extra $\|d^{\pi^*}/d^{\hat{\pi}}\|_\infty$, which we discuss separately below.

$\pi^+ - \hat{\pi}$ vs. $\nabla\hat{\pi}$: We will see that this difference is mostly about the expressivity and parameterization of the policy class. Define $g^{\hat{\pi}}(\pi) = \mathbb{E}_{s\sim d^{\hat{\pi}}}[\sum_a Q^{\hat{\pi}}(s,a)\pi(a|s)]$; note that here $\hat{\pi}$ is fixed and considered a constant, and π as in $\pi(a|s)$ is the only variable. Then,

$$\mathbb{E}_{s\sim d^{\hat{\pi}}}[\sum_a Q^{\hat{\pi}}(s,a)(\pi^+(a|s) - \hat{\pi}(a|s))] = g^{\hat{\pi}}(\pi^+) - g^{\hat{\pi}}(\hat{\pi}).$$

Here $g^{\hat{\pi}}$ is a linear function of $[\pi(a|s)]_{s,a} \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times 1}$, $g^{\hat{\pi}} = \beta^\top \pi$ with $\beta(s,a) = Q^{\hat{\pi}}(s,a)d^{\hat{\pi}}(s)$. Likewise, the corresponding term in $\nabla J(\hat{\pi})$ can be written as $\mathbb{E}_{s\sim d^{\hat{\pi}}}\left[\sum_a Q^{\hat{\pi}}(s,a)\nabla\hat{\pi}(a|s)\right] = \beta^\top\nabla\pi$, where $\nabla\pi$ is treated as a $|\mathcal{S}\times\mathcal{A}| \times d$ matrix with d being the number of parameters in θ.

Recall that we want to know whether a small $\|\nabla J(\hat{\pi})\|$ (which means small $\|\beta^\top\nabla\pi\|$) implies a small $\beta^\top(\pi^+ - \hat{\pi})$. A clearly sufficient condition is that there exists $\alpha \in \mathbb{R}^d$ such that $(\nabla\pi)\times\alpha \approx (\pi^+ - \hat{\pi})$. Here $\nabla\pi$ is the $|\mathcal{S}\times\mathcal{A}|\times d$ Jacobian characterizing how the policy changes its action distributions in each state as the parameter changes. $(\nabla\pi)\times\alpha$ can be interpreted as the change of $\pi \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ when the parameter $\hat{\theta}\in\mathbb{R}^d$ is changed along the direction of $\alpha\in\mathbb{R}^d$, so $(\nabla\pi)\times\alpha \approx (\pi^+ - \hat{\pi})$ means that at $\theta = \hat{\theta}$, there exists a direction in the parameter space that changes the policy $\pi_{\hat{\theta}}$ towards the direction of π^+. This is guaranteed when we use the tabular [11] or certain linear representations (π^+ must be in the policy class in the latter case). In general, however, such a direction may not be necessarily available if the policy class does not have sufficient expressivity or is poorly parameterized around $\hat{\pi}$, which can lead to a poor stationary point policy.

$\|d^{\pi^*}/d^{\hat{\pi}}\|_\infty$: While the previous issue may be fixed by using a more expressive and proper parameterization, the distribution mismatch is a more fundamental problem. Since we have no control over the coverage of $d^{\hat{\pi}}$ due to the randomness of $\hat{\pi}$, one recommendation is to run PG on initial distribution ρ that is different from the d_0 we use to define $J(\pi)$, and ρ should be exploratory and cover π^*. The stationary point of PG then guarantees that $\left\|\mathbb{E}_{s\sim d_\rho^{\hat{\pi}}}[\sum_a Q^{\hat{\pi}}(s,a)\nabla\pi_{\beta_0}(a|s)]\right\|$, where d_ρ^π is the occupancy induced from ρ, so the density ratio we need to pay is $\|d^{\pi^*}/d_\rho^{\hat{\pi}}\|_\infty \le \|d^{\pi^*}/\rho\|_\infty/(1-\gamma)$. In fact, in the next part of the course where we focus on exploration, we will see simple counterexamples showing the failure of PG (and many other methods) with a non-exploratory ρ, even under the tabular representation.

2.3 Natural Policy Gradient (NPG)

As we see in the previous section, a problem with PG is that poor parameterization may lead to a bad optimization landscape for the $g^{\hat{\pi}}$ function. As an extreme case, suppose in the neighborhood of $\hat{\theta}$, regardless of the parameter value θ', the corresponding policy always has the same action distribution as $\hat{\theta}$, then clearly we will see $\nabla J(\pi) = 0$ which by no means implies global optimality. In fact, this is related to a more general problem with (stochastic) gradient descent, that it is not invariant to reparameterization. To address this problem, when the objective depends on the parameter (θ) through a probability distribution $p_\theta(\cdot)$,[3] we can choose a more "natural" way to measure the change in parameters (hence the name *natural gradient*): we can measure the difference between $\hat{\theta}$ and θ' using the difference (in e.g., KL divergence) between their induced probability distributions, which is invariant to reparameterization. We refer the readers to standard tutorials and [11] for further background on natural gradient, and will directly give the closed-form update rule for NPG when we use the most popular (tabular) softmax parameterization:[4] for iterations $k = 1, 2, \ldots$,

$$\pi^{(k+1)}(a|s) \propto \pi^{(k)}(a|s) \exp(\eta Q^{\pi^{(k)}}(s,a)), \quad \forall (s,a) \tag{12}$$

with the initial policy being uniformly random over actions, and η is some appropriate learning rate.

We make a few remarks about this update rule.

1. The NPG does not have a standalone policy class Π, as it is implicitly using a tabular softmax class. In the function approximation setting, the complexity of the policy class depends on the function class we use to approximate Q^π.
2. To present a policy at iteration t, we need to keep around the Q-functions $Q^{\pi^{(1)}}$, $Q^{\pi^{(2)}}$, $\ldots$, $Q^{\pi^{(k-1)}}$, which can be cumbersome and impractical.
3. Crucially, the update on action distribution is performed in per-state manner independently. In each state, the update moves the action distribution slightly towards $\arg\max_a Q^{\pi^{(k)}}$ in each round ($\exp(\eta Q^{\pi^{(k)}}(s,a))$ corresponds to softmax), making it a "soft" version of policy iteration.

The NPG update rule in Eq.(12) also has an alternative interpretation, which directly leads to its global optimality guarantees. That is, it can be viewed as performing an online learning algorithm called *mirror descent*.

Mirror descent interpretation Mirror descent is an online learning algorithm concerning the following problem: on a finite and discrete space $\mathcal{X}$, for round $k = 1, 2, \ldots, K$, the learner proposes distribution $p_k \in \Delta(\mathcal{X})$, and nature reveals an arbitrary (and possibly adversarially chosen) function $f_k : \mathcal{X} \to [0,1]$. The NPG update rule in Eq.(12) exactly corresponds to (up to rescaling of learning rate) running an independent mirror descent procedure on each state s, with $\mathcal{X} = \mathcal{A}$, $p_k(\cdot) = \pi^{(k)}(\cdot|s)$, and $f_k(\cdot) = Q^{\pi^{(k)}}(s, \cdot)/V_{\max}$. Under appropriate learning rate, mirror descent enjoys the guarantee: for any $p \in \Delta(\mathcal{X})$,

$$\sum_{k=1}^{K} \langle p - p_k, f_k \rangle = O(\sqrt{K \log |\mathcal{X}|}).$$

[3] In our case this corresponds to $\pi_\theta(\cdot|s)$, where θ determines the distribution over actions.

[4] As we see below, even with such a tabular class whose complexity scales with $|\mathcal{S}|$, NPG does not need to pay on $|\mathcal{S}|$. This is why it can avoid the representation issue associated with a restricted policy class $|\Pi|$.

Applying this guarantee to the NPG setup, we have: $\forall s, \forall \pi(\cdot|s)$,

$$\sum_{k=1}^{K} (Q^{\pi^{(k)}}(s,\pi) - Q^{\pi^{(k)}}(s,\pi^{(k)})) = \sum_{k=1}^{K} \langle \pi(\cdot|s) - \pi^{(k)}(\cdot|s), Q^{\pi^{(k)}} \rangle = O(V_{\max}\sqrt{K \log |\mathcal{A}|}) \tag{13}$$

Next we see how this guarantees a near-optimal policy found by NPG. The final policy output by NPG, $\hat{\pi}$, is the uniform mixture of $\pi_1, \ldots, \pi_K$, whose expected return is simply $J(\hat{\pi}) = \frac{1}{K}\sum_{k=1}^{K} J(\pi^{(k)})$. Then for any policy π we wish to compete with (such as the optimal policy $\pi^\star$), we have

$$\begin{aligned}
J(\pi) - J(\hat{\pi}) &= \frac{1}{K}\sum_{k=1}^{K} (J(\pi) - J(\pi^{(k)})) \\
&= \frac{1}{K}\sum_{k=1}^{K} \frac{1}{1-\gamma}\mathbb{E}_{d^\pi}[Q^{\pi^{(k)}}(s,\pi) - Q^{\pi^{(k)}}(s,\pi^{(k)})] && \text{(PD lemma)} \\
&= \frac{1}{K(1-\gamma)}\sum_{k=1}^{K}\sum_{s} d^\pi(s)(Q^{\pi^{(k)}}(s,\pi) - Q^{\pi^{(k)}}(s,\pi^{(k)})) \\
&= \frac{1}{K(1-\gamma)}\sum_{s} d^\pi(s)\sum_{k=1}^{K}(Q^{\pi^{(k)}}(s,\pi) - Q^{\pi^{(k)}}(s,\pi^{(k)})) && \text{(change the summation order)} \\
&\le \frac{1}{K(1-\gamma)}\sum_{s} d^\pi(s) O(V_{\max}\sqrt{K\log|\mathcal{A}|}) = O(\frac{V_{\max}}{1-\gamma}\frac{\sqrt{\log|\mathcal{A}|}}{\sqrt{K}}). && \text{(Eq.(13))}
\end{aligned}$$

Note that when we change the order of summation over s and k, it is crucial that for all k the expectation is under the same distribution, d^π, otherwise the summation over k cannot be pushed inside for each individual s.

Error in estimating $Q^{\pi^{(k)}}$ Strictly speaking the above is not a learning algorithm, as it does not require any data once the exact $Q^{\pi^{(k)}}$ is given. (Again, this is similar to policy iteration.) In learning settings, we will need to estimate the Q-functions from data, and here we briefly discuss how the estimation errors of Q-functions affect the optimality guarantee.

Suppose in each iteration of NPG, instead of the exact $Q^{\pi^{(k)}}$ we use an approximate version $f^{\pi^{(k)}} \approx Q^{\pi^{(k)}}$ in the update rule in Eq.(12). Then

$$\begin{aligned}
J(\pi) - J(\hat{\pi}) = &\frac{1}{K}\sum_{k=1}^{K}\frac{1}{1-\gamma}\mathbb{E}_{d^\pi}[f^{\pi^{(k)}}(s,\pi) - f^{\pi^{(k)}}(s,\pi^{(k)})] && \text{(I)} \\
&+ \frac{1}{K}\sum_{k=1}^{K}\frac{1}{1-\gamma}\mathbb{E}_{d^\pi}[f^{\pi^{(k)}}(s,\pi^{(k)}) - Q^{\pi^{(k)}}(s,\pi^{(k)})] && \text{(II)} \\
&- \frac{1}{K}\sum_{k=1}^{K}\frac{1}{1-\gamma}\mathbb{E}_{d^\pi}[f^{\pi^{(k)}}(s,\pi) - Q^{\pi^{(k)}}(s,\pi)]. && \text{(III)}
\end{aligned}$$

Since NPG is run on $f^{\pi^{(k)}}$, term (I) is controlled in exactly the same way as before, so we only need to handle (II) and (III). The way to handle them depends on the concrete learning setting.

1. On-policy estimation [11] Similar to PG, we can run NPG in an on-policy fashion, where we roll-out trajectories from $\pi^{(k)}$ and use Monte-Carlo return to fit the Q-function by regression. In this case, we can easily control (II) when the function class can realize $Q^{\pi^{(k)}}$. However, we cannot directly control (III), and can only hope that $d^{\pi^{(k)}}$ provides coverage over d^π, which results in a coverage coefficient similar to what we pay in the PG analysis in Section 2.2.

2. Off-policy estimation [12] We can also run NPG on an offline dataset drawn from $\mu \in \Delta(\mathcal{S} \times \mathcal{A})$, and use methods such as FQE (or its minimax variant) to estimate the Q-function. Under Bellman-completeness, we can expect $\|f^{\pi^{(k)}} - \mathcal{T}^{\pi^{(k)}} f^{\pi^{(k)}}\|_{2,\mu}$ to be small. The rest is to show that such an Bellman error (on μ) can control (II) and (III) under suitable coverage conditions.

To translate value function error into Bellman error, we use the familiar telescoping lemma: the t-th term in (II) is

$$\frac{1}{(1-\gamma)^2} \mathbb{E}_{d^{\pi^{(k)}}_{[d^\pi]_s}} [f^{\pi^{(k)}} - \mathcal{T}^{\pi^{(k)}} f^{\pi^{(k)}}],$$

where $d^{\pi^{(k)}}_{[d^\pi]_s} \in \Delta(\mathcal{S} \times \mathcal{A})$ is the discounted state-*action* occupancy of π when we use $[d^\pi]_s \in \Delta(\mathcal{S})$ as the initial distribution.[5] The t-th term in (II) can be decomposed in a similar fashion, but on the discounted state-action occupancy induced by the following process: (1) start from $[d^\pi]_s$ as the initial distribution, (2) take π as the first action, and (3) take $\pi^{(k)}$ in the remaining steps.

At this point, it is tempting to require coverage of μ over these cumbersome occupancy measures. However, note that the terms from (II) and (III) have the opposite signs in their Bellman errors. Therefore, the common components of the two occupancies can cancel each other. We leave the proof as an exercise to the reader; the simplified expression after cancellation is surprisingly clean:

$$\text{(II) + (III)} = \frac{1}{K(1-\gamma)} \left(\sum_{k=1}^{K} \mathbb{E}_{d^{\pi^{(k)}}} [f^{\pi^{(k)}} - \mathcal{T}^{\pi^{(k)}} f^{\pi^{(k)}}] - \sum_{k=1}^{K} \mathbb{E}_{d^\pi} [f^{\pi^{(k)}} - \mathcal{T}^{\pi^{(k)}} f^{\pi^{(k)}}] \right).$$

This implies that the offline data μ only needs to cover d^π and $d^{\pi^{(k)}}$. Furthermore,

$$\frac{1}{1-\gamma} \mathbb{E}_{d^{\pi^{(k)}}} [f^{\pi^{(k)}} - \mathcal{T}^{\pi^{(k)}} f^{\pi^{(k)}}] = \mathbb{E}_{d_0} [f^{\pi^{(k)}}(s, \pi^{(k)})] - J(\pi^{(k)}).$$

By using the pessimistic principle to construct the $f^{\pi^{(k)}}$ estimate [12], it is easy to guarantee that $\mathbb{E}_{d_0}[f^{\pi^{(k)}}(s, \pi^{(k)})] \le J(\pi^{(k)})$. This way, the $\mathbb{E}_{d^{\pi^{(k)}}}[\cdot]$ term vanishes from the bound, and we only need to pay coverage over the comparator policy π.

Remark Our derivations above reveal a result that holds more generally and connects to many concepts in this course: $\forall \pi, \hat{\pi}, f \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$,[6]

$$\begin{aligned} J(\pi) - J(\hat{\pi}) = {} & \frac{1}{1-\gamma} \mathbb{E}_{d^\pi} [f(s, \pi) - f(s, \hat{\pi})] \qquad \text{(policy optimization error)} \\ & + \frac{1}{1-\gamma} \mathbb{E}_{d^{\hat{\pi}}} [f - \mathcal{T}^{\hat{\pi}} f] - \frac{1}{1-\gamma} \mathbb{E}_{d^\pi} [f - \mathcal{T}^{\hat{\pi}} f]. \end{aligned}$$

The bound generalizes Lemma 4 in note3, which can be recovered by letting $\pi = \pi_f$ (the policy optimization error term becomes non-positive and $\mathcal{T}^{\hat{\pi}} f = \mathcal{T} f$). We briefly touched on how the Bellman error under $d^{\hat{\pi}}$ vanishes under *pessimism*; in upcoming lectures we will see that *optimism* removes the d^π term and leads to efficient exploration.

[5] All occupancies on state-actions by default, and we use $[\cdot]_s$ to refer to their state-marginals when there is ambiguity.

[6] This result came from discussion with Tengyang Xie; a model-based corollary can be found in [13, Lemma 3.1].

References

[1] Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1097–1104. Omnipress, 2011.

[2] Doina Precup, Richard S Sutton, and Satinder P Singh. Eligibility Traces for Off-Policy Policy Evaluation. In *Proceedings of the 17th International Conference on Machine Learning*, pages 759–766, 2000.

[3] Doina Precup. *Temporal abstraction in reinforcement learning*. PhD thesis, University of Massachusetts Amherst, 2000.

[4] Nan Jiang and Lihong Li. Doubly Robust Off-policy Value Evaluation for Reinforcement Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 652–661, 2016.

[5] Philip Thomas. *Safe Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst, 2015.

[6] Philip Thomas and Emma Brunskill. Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[7] Tor Lattimore and Marcus Hutter. PAC bounds for discounted MDPs. In *Algorithmic Learning Theory (ALT)*, 2012.

[8] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[9] 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.

[10] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[11] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, pages 64–66. PMLR, 2020.

[12] Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *arXiv preprint arXiv:2106.06926*, 2021.

[13] Anirudh Vemula, Yuda Song, Aarti Singh, Drew Bagnell, and Sanjiban Choudhury. The virtues of laziness in model-based rl: A unified objective and algorithms. In *International Conference on Machine Learning*, pages 34978–35005. PMLR, 2023.