Homework 3: Policy Gradient, Policy Evaluation, & Exploration

CS 6789: Foundations of Reinforcement Learning

1 Exploration in Absorbing MDPs (30 Points)

Let's first recall Problem 3 in HW2 where we constructed absorbing MDPs and have shown that the absorbing MDP can be used to provide optimism. We complete the rest exercise regarding absorbing MDPs in this section.

Specifically, we will see the Explore-or-Terminate phenomena, where the algorithm either has already found a near optimal policy—hence we terminate, or the algorithm can guarantee to explore new state-action pairs outside the current known set.

1.1 Exploration or Termination

Note that we know $\widehat{P}^\dagger$, as it uses our learned model $\widehat{P}$ and it has deterministic absorbing structure at unknown state-action pairs. So we can plan in this model using $r^\dagger$. Let's define $\widehat{\pi}^\star$ as the optimal policy of $\widehat{P}^\dagger$ and $r^\dagger$ (imagine we run PI under $\widehat{P}^\dagger$ and $r^\dagger$ until convergence).

We will consider two cases.

In case one, when we execute $\widehat{\pi}^\star$ under the real model P, $\widehat{\pi}^\star$ does not escape the known set $\mathcal{K}$ with big probability, i.e., $\sum_{s,a\in(\mathcal{S}\times\mathcal{A})\setminus\mathcal{K}} d^{\widehat{\pi}^\star}(s,a) \le \epsilon$ for some small $\epsilon \in \mathbb{R}^+$.

Q (a) (15 points): Let us prove the following inequality:

$$V^{\pi^\star} \le V^{\widehat{\pi}^\star} + \mathcal{O}\left(\frac{1}{(1-\gamma)^2}\epsilon + \frac{1}{(1-\gamma)^2}\sqrt{\frac{S\ln(SA/\delta)}{k}}\right).$$

This says that if k is big enough, and the escape probability is small, then we find a near-optimal policy $\widehat{\pi}^\star$.
(**Hint:** here we need to consider the difference between $\widehat{V}^{\dagger,\pi}$ and $V^{\dagger,\pi}$ using simulation lemma)

Q (b) (15 points): On the other hand, we encounter the case where $\widehat{\pi}^\star$ escapes, i.e., $\sum_{s,a\in(\mathcal{S}\times\mathcal{A})\setminus\mathcal{K}} d^{\widehat{\pi}^\star}(s,a) \ge \epsilon$. Let us assume that we have sampling oracle $\mathcal{O}(\pi)$ that samples an i.i.d state-action pair from d^π, i.e., $(s,a) \sim d^\pi$

Prove the following conclusion. *With probability at least* $1-\delta$*, after* $M = \Omega\left(\frac{kSA}{\epsilon} + \frac{\ln(1/\delta)}{\epsilon^2}\right)$ *many sampling oracle* $\mathcal{O}(\widehat{\pi}^\star)$ *calls, at least one state-action pair from the unknown set* $(\mathcal{S}\times\mathcal{A})\setminus\mathcal{K}$ *becomes known, i.e., it will be sampled more than* k *times*.
(**Hint**: First let's ask ourselves that after M many calls of $\mathcal{O}(\widehat{\pi}^\star)$, how many (s,a) samples out of M samples will fall into $(\mathcal{S}\times\mathcal{A})\setminus\mathcal{K}$? Now, if I have N many (s,a) samples falling into $(\mathcal{S}\times\mathcal{A})\setminus\mathcal{K}$, can I say anything useful here using the Pigeon hole principle here?)

1.2 Summary (no questions)

To this end, we have seen that using $\widehat{P}^\dagger$ and $r^\dagger$, we plan a policy $\widehat{\pi}^\star$ that either is near optimal if it stays inside the known set with big probability, or we guarantee to explore, i.e., making at least one previous unknown state-action pair to be known, by executing $\widehat{\pi}^\star$ in the real MDP for polynomially many times. With the new known set, we can repeat the above whole process again which will either terminate with a near optimal policy or will guarantee to identify another known state-action pair. Note that the total number of state-action pairs that can be made to the known set is at most SA. Thus, we can expect that the algorithm will find a near optimal policy in polynomial sample complexity.

2 Policy Gradient Theorem (30 Points)

We will focus on infinite horizon discounted MDP here.

2.1 Baslines

Q (15 points): For any function $f(s)$ (that is only a function of the states) show that:

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

2.2 Action-dependent Baselines for Factored Policies

We consider the following factored policy representation. We assume action $a\in\mathbb{R}^d$. We denote $a[i]$ as the i-th component of the d-dim vector a. Also we denote $a[-i]$ as all the components of a except the i-th component.

Following the most common open-source implementations in the RL community, we represent our parameterized policy as follows $\pi_\theta(a|s) = \prod_{i=1}^d \pi_{i;\theta}(a[i]|s)$, where each $\pi_{i;\theta}$ maps state s to a distribution in $\mathbb{R}$. Note that one of the most common examples of this in practice is in continuous control where $\pi_\theta(\cdot|s)$ is a Gaussian distribution with a diagonal covariance matrix (i.e., each coordinate of the action is independent).

Now let us consider d many baselines, $f_1,\ldots,f_d$, where each $f_i(s,a[-i])\in\mathbb{R}$. Namely, each f_i takes both state and action information, but f_i ignores the i-th component of the action vector a.

Q (15 points): Prove the following:

$$\nabla V^{\pi_\theta} = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\sum_{i=1}^{d}\nabla_\theta \ln \pi_{i;\theta}(a[i]|s)\left(Q^{\pi_\theta}(s,a) - f_i(s,a\,[-i])\right)\right].$$

Namely, we indeed can use action-dependent baselines if our policy is factorized over dimensions.

2.3 Best Baseline (Bonus)

Q Find the $f(\cdot)$ which leads to the minimum variance estimate of our gradient, assuming that our gradient estimate will be:

$$\widehat{\nabla V^{\pi_\theta}} = \frac{1}{1-\gamma}\Big(Q^{\pi_\theta}(s,a) - f(s)\Big)\nabla \log \pi_\theta(a|s)$$

where $s \sim d^{\pi_\theta}$, $a \sim \pi_\theta(\cdot|s)$.

3 Batch Policy Evaluation via Regression (40 Points)

Often in practice, instead of using an unbiased estimate of policy gradient, one would like to use a biased estimate, i.e., one would like to perform bias-variance tradeoff for policy gradient. In this homework problem, let us derive a specific algorithm that performs policy evaluation in a dynamic programming fashion.

Let's take a look at the setup below. We consider finite horizon setting $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, H, r, P, \rho\}$ with ρ being the initial state distribution. We consider offline setting where we have a H many datasets $\mathcal{D}_h = \{s_h^i, a_h^i, \overline{s}_h^i, r_h^i\}_{i=1}^N$, where $s_h^i, a_h^i \sim \nu_h$, and $r_h^i = r(s_h^i, a_h^i)$, and $\overline{s}_h^i \sim P(\cdot|s_h^i, a_h^i)$; We consider linear function approximation with feature mapping $\phi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^d$.

To make sure the problem is tractable, we make the following two key assumptions:

- Assumption 1: Realizability, i.e., for any $\pi : \mathcal{S} \to \mathcal{A}$, we have for all h, $Q_h^\pi(s,a) = \theta_h^\top \phi(s,a)$, for some w which has bounded norm $\|\theta_h\|_2 \le W \in \mathbb{R}^+$; In other words, Q_h^π is a linear function with respect to feature ϕ, for all h;
- Assumption 2: Coverage: i.e., for data distribution ν_h at any h, we have $\sigma_{\min}\left(\mathbb{E}_{s,a\sim\nu_h}\phi(s,a)\phi(s,a)^\top\right) \ge \kappa \in \mathbb{R}^+$. In other words, the data distribution ν_h's induced feature covariance matrix is full rank and has lower bounded eigenvalues. This assumption ensures that with reasonably enough data, our data matrix, i.e., $\Phi_h = [\phi(s_h^1, a_h^1), \ldots, \phi(s_h^N, a_h^N)]^\top \in \mathbb{R}^{N\times d}$ will be full rank.

We fix a policy $\pi : \mathcal{S} \to \mathcal{A}$, and we would like to estimate Q_0^π and its parameter θ_0 using the offline datasets $\mathcal{D}_h$ for $h = 0, \ldots, H-1$.

3.1 Least square regression at $h = H - 1$

From a dynamic programming perspective, it is obvious what we would like to do at $h = H - 1$, i.e., the last time step. We use $\mathcal{D}_{H-1}$ to perform regression:

$$\widehat{w}_{H-1} = \arg\min_w \sum_{i=1}^{N} \left(\phi(s^i_{H-1}, a^i_{H-1})^\top w - r^i_{H-1}\right)^2.$$

Let's write $\Phi_{H-1} \in \mathbb{R}^{N\times d}$, where the i-th row of Φ_{H-1} is $\phi(s^i_{H-1}, a^i_{H-1})^\top$, and let's write $r_{H-1} \in \mathbb{R}^N$ where the i-th element of r_{H-1} is r^i_{H-1}.

Assume Φ_{H-1} is full rank (i.e., assume $N > d$, and rank of Φ_{H-1} is d)

Q (10 points): Derive that:

$$\widehat{w}_{H-1} = \left(\Phi^\top_{H-1}\Phi_{H-1}\right)^{-1}\Phi^\top_{H-1} r_{H-1}.$$

3.2 Linear square regression at h

Now let's do dynamic programming. Assume that we have learned $\widehat{w}_{h+1}$ already. We use $\widehat{w}_{h+1}$ to learn $\widehat{w}_h$. Specifically, we consider the following least square regression using the dataset $\mathcal{D}_h$:

$$\widehat{w}_h = \arg\min_w \sum_{i=0}^{N} \left(\phi(s^i_h, a^i_h)^\top w - \left(r^i_h + \widehat{w}^\top_{h+1}\left[\phi(\overline{s}^i_h, \pi(\overline{s}^i_h))\right]\right)\right)^2.$$

Note that if we believed that $\widehat{w}_{h+1}$ is a good estimator, then we would believe that $r^i_h + \widehat{w}^\top_{h+1}\phi(\overline{s}^i_h, \pi(\overline{s}^i_h)) \approx Q^\pi_h(s^i_h, a^i_h)$, since $s^i_{h+1} \sim P(\cdot|s^i_h, a^i_h)$. From an induction perspective, we started at $H - 1$ where we should expect that $\widehat{w}_{H-1}$ is a good fit because the reward itself is a linear function with respect to ϕ, and if we were doing linear regression correctly, we should expect $\widehat{w}_{h+1}$ is a good estimator as well, as Q^π_{h+1} is a linear function with respect to ϕ.

Let's denote $\Phi_h \in \mathbb{R}^{N\times d}$ as a matrix where the i-th row is $\phi(s^i_h, a^i_h)^\top$, and $\overline{\Phi}_h \in \mathbb{R}^{N\times d}$ as a matrix where the i-th row is $\phi(\overline{s}^i_h, \pi(\overline{s}^i_h))$, and $r_h \in \mathbb{R}^N$ where the i-th element of r is r^i_h.

Q (10 points): Derive that:

$$\widehat{w}_h = \left(\Phi^\top_h\Phi_h\right)^{-1}\Phi^\top_h r_h + \left(\Phi^\top_h\Phi_h\right)^{-1}\Phi^\top_h\overline{\Phi}_h\widehat{w}_{h+1}.$$

3.3 Solution at $h = 0$

We can repeat the same process above for all $h = H - 1$ to $h = 0$.

Q (20 Points): Show that for $\widehat{w}_0$:

$$\widehat{w}_0 = \sum_{h=0}^{H-1}\left(\prod_{\tau=0}^{h-1}\left(\Phi^\top_\tau\Phi_\tau\right)^{-1}\Phi^\top_\tau\overline{\Phi}_\tau\right)\left(\Phi^\top_h\Phi_h\right)^{-1}\Phi^\top_h r_h$$

3.4 Quality of the Solution (Bonus)

We further simplify the setting by assuming that $r_h^i = 0$ for all $h = 0 \dots H-2$, i.e., we have nonzero reward at $H-1$ but zero reward in all previous time steps (i.e., we get reward 1 or 0 at the end of the game). In this case, the solution $\widehat{w}_0$ has a very simple expression:

$$\widehat{w}_0 = \left(\prod_{\tau=0}^{H-2} \left(\Phi_\tau^\top \Phi_\tau\right)^{-1} \Phi_\tau^\top \overline{\Phi}_\tau \right) \left(\Phi_{H-1}^\top \Phi_{H-1}\right)^{-1} \Phi_{H-1}^\top r_{H-1}$$

Given $\phi(s_0, \pi(s_0))$, we are interested in the prediction error $|Q_0^\pi(s_0, \pi(s_0)) - \widehat{w}_0 \phi(s_0, \pi(s_0))|$ at the initial state s_0.

Recall our assumption where for all $h = 0, \dots, H-1$, we have that for all s, a, $Q_h^\pi(s,a) = \theta_h \cdot \phi(s,a)$ for some θ_h.

Let us denote $\delta_h^i = \theta_{h+1} \cdot \phi(\overline{s}_h^i, \pi(\overline{s}_h^i)) - \mathbb{E}_{s' \sim P(\cdot|s_h^i, a_h^i)} \theta_{h+1}^\top \phi(s', \pi(s'))$. Since $\overline{s}_h^i \sim P(\cdot|s_h^i, a_h^i)$, we have that $\mathbb{E}[\delta_h^i | s_h^i, a_h^i] = 0$. Denote $\delta_h \in \mathbb{R}^N$ as a vector whose i-th element is δ_h^i.

Q Prove that:

$$\widehat{w}_0 - \theta_0 = \sum_{h=0}^{H-1} \left[\left(\prod_{\tau=0}^{h-1} (\Phi_\tau^\top \Phi_\tau)^{-1} \Phi_\tau^\top \overline{\Phi}_\tau \right) (\Phi_h^\top \Phi_h)^{-1} \Phi_h^\top \delta_h \right]$$

Remark (no question) As we can see above, $\widehat{w}_0$ is indeed an unbiased estimate of θ_0! This is because for any h, we have:

$$\mathbb{E}\left[\left(\prod_{\tau=0}^{h-1} (\Phi_\tau^\top \Phi_\tau)^{-1} \Phi_\tau^\top \overline{\Phi}_\tau \right) (\Phi_h^\top \Phi_h)^{-1} \Phi_h^\top \delta_h \Big| \mathcal{D}_0, \dots, \mathcal{D}_h \right] = 0,$$

as $\mathbb{E}[\delta_h^i | s_h^i, a_h^i] = 0$. However, the variance of our estimator $\widehat{w}_0$ could be exponential in H in worst case (notice the term $\prod_{\tau=0}^{h-1} (\Phi_\tau^\top \Phi_\tau)^{-1} \Phi_\tau^\top \overline{\Phi}_\tau$). For people who are interested in the potential exponentially large variance worst case, we encourage you to think about a lower bound construction.

4 Linear Quadratic Regulator (Bonus)

Recall the definition of infinite horizon LQR:

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t, \\ c(x,u) &= x^\top Q x + u^\top R u, \end{aligned}$$

where $x \in \mathbb{R}^d$ and $u \in \mathbb{R}^k$.

We consider a specific linear policy: $\pi(x) = Kx$ where $K \in \mathbb{R}^{k \times d}$ (assume this control is stable, i.e., the average expected cost under this control is finite). From the lectures, we know that

the value function of the linear policy is quadratic. Let us denote $P_K \in \mathbb{R}^{d \times d}$, and define the value function $V_K(x)$ as:

$$V_K(x) := \sum_{t=0}^{\infty} x_t^\top Q x_t + u_t^\top R u_t = x^\top P_K x,$$

where $x_0 = x, u_t = K x_t$, and Q function $Q_K(x, u)$ as:

$$Q_K(x, u) = x^\top Q x + u^\top R u + V_K(Ax + Bu).$$

4.1 Policy Iteration

Recall the Policy Iteration algorithm from our second lecture: Given a policy $\pi : \mathbb{R}^d \mapsto \mathbb{R}^k$, PI computes a new policy π:

$$\pi'(x) := \arg\min_{u \in \mathbb{R}^k} Q^\pi(x, u), \forall x \in \mathbb{R}^d.$$

Q: Now use the fact that we are considering an LQR system with $V_K(x) = x^\top P_K x$, derive the closed-form solution of π'
(**Hint:** first show that π' again will be a linear policy).