COS 435/ECE 433 Week 4 Precept Notes

February 22, 2024

1 Dynamic Programming

Dynamic programming is widely applied to learn the *optimal policy* of a sequential decision-making process. Recall that we have the following identities for value function with respect to a Markov policy π, known as the Bellman equations:

$$Q^\pi(s,a) = r(s,a) + \gamma \cdot \mathbb{E}_{s' \sim p(\cdot|s,a)}[V^\pi(s')], \tag{1}$$
$$V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)}[Q^\pi(s,a)] \tag{2}$$

for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$, where

$$V^\pi(s) = \mathbb{E}_\pi\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \middle| s_0 = s\right],$$
$$Q^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \middle| s_0 = s, a_0 = a\right].$$

Policy Evaluation. First, we consider how to compute the state-value function V^π for an arbitrary policy π. This is called policy evaluation in the DP literature. For a finite state/action MDP, when the reward and transition is known, solving Eq. (1) is equivalent to solving a linear system with $O(|\mathcal{S}|)$ equations. Instead, we can estimate the value function by iterative dynamic programming:

$$V_k^\pi(s) = R(s) + \gamma \cdot \sum_{s'} P^\pi(s'|s) V_{k-1}^\pi(s'), \tag{3}$$

where $R(s) := \sum_a r(s,a)\pi(a|s)$, $P^\pi(s'|s) = \sum_a \pi(a|s)p(s'|s,a)$, and $V_0^\pi(s) = 0$ for all $s \in \mathcal{S}$. The following argument supports such an algorithm:

$$\mathbf{V}^\pi = \mathbf{R} + \gamma \mathbf{P}^\pi \cdot \mathbf{V}^\pi,$$

where $\mathbf{V}^\pi = (V^\pi(s))_{s \in \mathcal{S}}$, $\mathbf{P}^\pi_{\mathbf{s},\mathbf{s}'} = (P^\pi(s'|s))$. Note that the eigenvalue of $\mathbf{P}^\pi$ is always less than 1 (Gershgorin circle theorem), we have

$$\mathbf{V}^\pi = (\mathbf{I} - \gamma \mathbf{P}^\pi)^{-1}\mathbf{R} = \sum_{i \geq 0} (\gamma \mathbf{P}^\pi)^i \mathbf{R},$$

which exactly gives us Eq. (3).

Policy Learning. In reinforcement learning, we are interested in the following maximization:

$$\max_{\pi} \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h \cdot r(s_h, a_h) \middle| s_h \sim P(\cdot|s_{h-1}, a_{h-1}), a_h \sim \pi(\cdot|s_h)\right]$$

here $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ maps the current state s to a probability distribution on the action space $\mathcal{A}$. We denote the optimal policy with respect to the previous optimization problem by π^*. For an MDP with finite state/action space and known reward/transition, *the principle of dynamic programming* tells us the value function of the optimal policy π^* is characterized by the following condition:

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

and correspondingly $\pi^*(\cdot|s) = \mathrm{argmax}_{a \in \mathcal{A}} Q^*(s,a)$. We have the following iterative estimator for V^*:

$$V_k(s) = \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \cdot \sum_{s' \in \mathcal{S}} p(s'|s,a) V_{k-1}(s') \right\} := \mathbb{B}V_{k-1}(s),$$

here $\mathbb{B}V$ is called the *Bellman operator.* Such an estimation procedure is also called the *value iteration.* The validity of such an estimator can be proved by the *fixed point argument:*

- For the operator $\mathbb{B}$, we say V is a fixed point of $\mathbb{B}$ if $\mathbb{B}V = V$. By the Bellman equation, we have $\mathbb{B}V^* = V^*$.
- Since $V_k = \mathbb{B}V_{k-1}$, we have
$$\|V_k - V^*\|_\infty = \|\mathbb{B}V_{k-1} - \mathbb{B}V^*\|_\infty.$$
As long as $\|\mathbb{B}V_{k-1} - \mathbb{B}V^*\|_\infty \le c\|V_{k-1} - V^*\|_\infty$ for some $c < 1$, we can prove that $V_k \to V^*$ as $k \to \infty$. Actually we can prove that $\|V^k - V^*\|_\infty = O(c^k)$, i.e. an exponential convergence rate is guaranteed.

Now we prove that $\|\mathbb{B}V_{k-1} - \mathbb{B}V^*\|_\infty \le \gamma\|V_{k-1} - V^*\|_\infty$. In fact, utilizing $|\max_a f(s,a) - \max_a g(s,a)| \le \max_a |f(s,a) - g(s,a)|$, we have

$$\begin{aligned} \|\mathbb{B}V_{k-1} - \mathbb{B}V^*\|_\infty &= \max_s \left| \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) V_{k-1}(s') \right\} - \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) V^*(s') \right\} \right| \\ &\le \gamma \max_s \left| \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \left\{ p(s'|s,a)(V_{k-1}(s') - V^*(s') \right\} \right| \\ &\le \gamma \max_{s'} \max_{a'} \sum_{s' \in \mathcal{S}} p(s'|s,a) \|V_{k-1} - V^*\|_\infty \\ &= \gamma \|V_{k-1} - V^*\|_\infty, \end{aligned}$$

with the fixed point argument, we actually proved that $\|V_k - V^*\|_\infty \le O(\gamma^k)$. It is not hard to prove the same convergence rate for policy evaluation using almost the same proof. A natural question to ask is: how good is the greedy policy with respect to V^k, i.e.

$$\pi_k(\cdot|s) = \mathrm{argmax}_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \cdot \sum_{s' \in \mathcal{S}} p(s'|s,a) V_{k-1}(s') \right\}?$$

As we will see in Question 3 in HW4, we have

$$V^* - V^{\pi_k} \le \frac{2\|V^{\pi_k} - \mathbb{B}V^{\pi_k}\|_\infty}{1-\gamma} \le O\left(\frac{\|V^{\pi_k} - V^*\|_\infty + \|\mathbb{B}V^{\pi_k} - V^*\|_\infty}{1-\gamma} \right) = O\left(\frac{\|V^{\pi_k} - V^*\|_\infty + \|V^{\pi_{k+1}} - V^*\|_\infty}{1-\gamma} \right),$$

which gives us a bound of $O(\frac{\gamma^k}{1-\gamma})$.

2 Policy Gradient Method

In this section, we only discuss some simple PG methods, as more advanced methods with policy gradients will be discussed in future lectures. First, consider a finite horizon MDP. Note that we can always parameterize the agent's policy π with some parameter θ, and therefore we turn the policy learning problem into the following optimization problem:

$$\begin{aligned}&\max_{\theta} \mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h),s_{h+1}\sim p(\cdot|s_h,a_h)}\left[\sum_{h=0}^{H} r(s_h,a_h)\right] \\ &\quad= \int_{\mathcal{S}\times\mathcal{A}}\left\{\sum_{h=0}^{H} r(s_h,a_h)\right\} p_\theta(s_0,a_0,\ldots,s_h,a_h,\ldots,s_H,a_H)\mathrm{d}(s_0,a_0,s_1,\ldots,s_H,a_H),\end{aligned} \tag{4}$$

here $p_\theta(s_0,a_0,\ldots,s_h,a_h,\ldots,s_H,a_H)$ is the marginal probability of trajectory $(s_0,a_0,\ldots,s_h,a_h,\ldots,s_H,a_H)$ when the agent takes the policy π_θ. Inspired by first-order optimization methods, we would like to solve Eq. (4) with stochastic gradient descent. To do this, we need to obtain an unbiased estimate of the gradient. For simplicity in calculation, we assume that the MDP has a maximum horizon H. Note that

$$\begin{aligned}&\nabla_\theta \mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h),s_{h+1}\sim p(\cdot|s_h,a_h)}\left[\sum_{h=0}^{H} r(s_h,a_h)\right] \\ =&\int_{(\mathcal{S}\times\mathcal{A})^H}\left\{\sum_{h=0}^{H} r(s_h,a_h)\right\}\nabla_\theta p_\theta(s_0,a_0,\ldots,s_h,a_h,\ldots,s_H,a_H)\mathrm{d}s_0\mathrm{d}a_0\ldots\mathrm{d}s_H\mathrm{d}a_H \\ =&\int_{(\mathcal{S}\times\mathcal{A})^H}\left\{\sum_{h=0}^{H} r(s_h,a_h)\right\} p_\theta(s_h,a_h)p_\theta(s_0,a_0,\ldots,s_h,a_h,\ldots,s_H,a_H)\nabla_\theta\log p_\theta(s_0,a_0,\ldots,s_h,a_h,\ldots,s_H,a_H)\mathrm{d}s_0\mathrm{d}a_0\,. \\ =&\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h),s_{h+1}\sim p(\cdot|s_h,a_h)}\left[\left\{\sum_{h=0}^{H} r(s_h,a_h)\right\}\nabla_\theta\log p_\theta(s_0,a_0,\ldots,s_h,a_h,\ldots,s_H,a_H)\right],\end{aligned}$$

note that

$$p_\theta(s_0,a_0,\ldots,s_h,a_h,\ldots,s_H,a_H) = p(s_0)\pi_\theta(a_0|s_0)\prod_{i=1}^{h}\pi_\theta(a_i|s_i)p(s_i|s_{i-1},a_{i-1}),$$

we have $\nabla_\theta\log p_\theta(s_0,a_0,\ldots,s_h,a_h,\ldots,s_H,a_H) = \sum_{h=1}^{H}\nabla_\theta\log p_\theta(a_h|s_h)$, therefore a natural stochastic gradient for policy update is

$$g_\theta = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{h=0}^{H} r(s_h^i,a_h^i)\right)\left(\sum_{h=0}^{H}\nabla_\theta\log\pi_\theta(a_h^i|s_h^i)\right),$$

where $\{(s_0^i,a_0^i,\ldots,s_H^i,a_H^I)\}_{i\in[N]}$ are sampled by policy π_θ, and we can update θ by $\theta^{k+1} = \theta^k - \alpha g_{\theta^{k-1}}$.

- Gaussian policies (continuous action space): $a_h\sim\mathcal{N}(\mu_\theta(s_h),\Sigma)$. Then $\nabla_\theta\log\pi_\theta(a_h|s_h) = C\nabla_\theta\|a_h-\mu_\theta(s_h)\|^2_{\Sigma^{-1}} = C\Sigma^{-1}(\mu_\theta(s_h)-a_h)\nabla_\theta\mu_\theta(s_h)$.
- Linear policies (discrete action space): $p_\theta(a_h|s_h) = \frac{\exp(\phi(s_h,a_h)^\top\theta)}{\sum_a\exp(\phi(s_h,a)^\top\theta)}$, where ϕ is a given feature. Then $\nabla_\theta\log p(a_h|s_h) = \phi(s_h,a_h) - \frac{\sum_a\exp(\phi(s_h,a_h)^\top\theta)\phi(s_h,a)}{\sum_a\exp(\phi(s_h,a)^\top\theta)}$.

Drawbacks of Vanilla Policy Gradient methods. (1) The high variance of the policy gradient results in instability in the training process. (2) The on-policy nature of policy gradient: only utilizing $\{(s_h^k, a_h^k)\}_{h\geq 0}$ when estimating the k-th gradient, causing the waste of previous data.

2.1 Methods to reduce variance.

We aim to prove We first prove the following lemma.

Lemma 2.1 (Causality)**.** $\mathbb{E}_{\pi_\theta}[b_h(s_0, a_0, \ldots, a_{h-1}s_h)\nabla_\theta \log \pi_\theta(a_h|s_h)] = 0$ *for all* $h \in [H]$.

Proof.

$$\mathbb{E}_{\pi_\theta}[b_h(s_0, a_0, \ldots, a_{h-1}s_h)\nabla_\theta \log \pi_\theta(a_h|s_h)] = \mathbb{E}\bigg[b_h(s_0, a_0, \ldots, a_{h-1}s_h) \cdot \mathbb{E}\bigg[\nabla_\theta \log \pi_\theta(a_h|s_h)\bigg| s_0, a_0, \ldots, a_{h-1}, s_h\bigg]\bigg],$$

by Markovian property, we have

$$\begin{aligned}\mathbb{E}\bigg[\nabla_\theta \log \pi_\theta(a_h|s_h)\bigg| s_0, a_0, \ldots, a_{h-1}, s_h\bigg] &= \int_{\mathcal{S}\times\mathcal{A}} p_\theta(s_h)\pi_\theta(a_h|s_h)\nabla_\theta \log \pi_\theta(a_h|s_h)\mathrm{d}s_h\mathrm{d}a_h \\ &= \int_{\mathcal{S}\times\mathcal{A}} p_\theta(s_h)\nabla_\theta\pi_\theta(a_h|s_h)\mathrm{d}s_h\mathrm{d}a_h \\ &= 0,\end{aligned}$$

as $\int_{\mathcal{A}} \pi_\theta(a|s_h)\mathrm{d}a = 1$ always holds, and we conclude the proof. □

Based on the causality lemma, we can adjust the weight of $\nabla \log \pi(a_h|s_h)$ for every h as long as this adjustment only 'ignores' the information before h. Based on this observation, we immediately obtain the following variance-reduced estimates for policy gradient.

"Reward to go". From the causality lemma, we immediately acquire the following identity:

$$\begin{aligned}\nabla_\theta \mathbb{E}_{\pi_\theta}\bigg[\sum_{h=0}^{H} r(s_h, a_h)\bigg] &= \mathbb{E}_{\pi_\theta}\bigg[\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta(a_h|s_h)\bigg\{\sum_{t\geq h} r(s_t, a_t)\bigg\}\bigg] \\ &= \mathbb{E}_{\pi_\theta}\bigg[\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta(a_h|s_h) Q^{\pi_\theta}(s_h, a_h)\bigg],\end{aligned}$$

therefore another estimate for the policy gradient is

$$g_\theta = \frac{1}{N}\sum_{i=1}^{N}\bigg(\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta(a_h^i|s_h^i)\bigg\{\sum_{t\geq h} r(s_t^i, a_t^i)\bigg\}\bigg).$$

Intuitively, such a method reduces the bias by 'ignoring' the noise before the current step h.

Generalized Advantage function. Applying the causality Lemma one more time, we have

$$\nabla_\theta \mathbb{E}_{\pi_\theta}\bigg[\sum_{h=0}^{H} r(s_h, a_h)\bigg] = \mathbb{E}_{\pi_\theta}\bigg[\sum_{h=0}^{H} \nabla_\theta \log \pi_\theta(a_h|s_h)\big\{Q^{\pi_\theta}(s_h, a_h) - V^{\pi_\theta}(s_h)\big\}\bigg].$$

We define the *advantage function* $A^\pi(s_h, a_h) := Q^{\pi_\theta}(s_h, a_h) - V^{\pi_\theta}(s_h)$. When maintaining a estimate $\hat{V}^\theta$ for θ, we can estimate the advantage function by $\hat{A}^{\pi_\theta}(s_h, a_h) = r(s_h, a_h) + \hat{V}^\theta(s_{h+1}) - \hat{V}(s_h)$. Such a method further reduces variance in comparison to reward-to-go method, but would introduce bias, as the estimate $\hat{V}^\theta$ is biased. In future lectures, we will see how such an approach is closely related to other RL algorithms, such as Actor-critic.