CSE6243: Advanced Machine Learning **Fall 2023**

Lecture 23: Policy Gradient and Actor-Critic

Lecturer: Bo Dai *Scribes: Hiren Kumawat, Rishabh Goswami*

23.1 Recap

In prior lectures, we covered Markov Decision Processes and the Bellman equation. We encountered two types of problems:

- **Policy Evaluation:** attempt to find value of $V(\pi)$ or $Q(\pi)$ for a given policy π.
- **Policy Optimization:** estimate an optimal policy π.

Additionally, we discussed the differences in planning and learning. When planning, we have access to a model of the given environment. When learning, we do not have this model, so we are reliant on experience.

We also discussed categorization of these algorithms on various factors, like on-policy vs off-policy and model-based vs model-free. Of the model-free categorization, common methods include temporal-difference and policy gradient methods.

Today's lecture will focus primarily on policy gradient reinforcement learning.

23.2 New Content

23.2.1 Policy Gradient

We can express the policy gradient algorithm as follows:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, J(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$$

On a high level, this represents the expected value of the accumulation of rewards under a given policy. We can parameterize this policy gradient calculation with θ, allowing us to express the problem as: $\max_{\pi_\theta} J(\pi_\theta)$.

With this parameterization, we can describe the gradient ascent as follows:

1. Initialize θ_0.
2. For $i = 1, \dots$:

$$\theta_{t+1} = \theta_t + \eta \nabla_\theta J(\pi_\theta)$$

$$= \underset{\theta}{\operatorname{argmin}} \langle \theta, \nabla_{\theta_t} J(\pi_{\theta_t}) \rangle - \frac{1}{2\eta} ||\theta - \theta_t||^2$$

We can change this metric ($||\theta - \theta_t||^2$) in the last equation to generate the whole policy gradient family.

Going back to the generalized update, we can break up these terms using the chain rule as:

$$\nabla_\theta J(\pi_\theta) = \nabla_\pi J(\pi_\theta) \nabla_\theta \pi_\theta(a, s)$$

In this new equation, we can compute $\nabla_\pi J(\pi)$ to be:

$$= \nabla_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right]$$

For convenience, we will refer to the expectation term as $h(\tau)$ (i.e., $h(\tau) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$), where $\tau = s_0, a_0, r_0, \ldots s_n, a_n, r_n$ representing the state, action, reward triples over all timesteps t.

$$\nabla_\pi \mathbb{E}_\pi[h(\tau)] = \int \prod_{t=0}^{\infty} p(s_{t+1}|s_t, a_t) \nabla \pi(a_t|s_t) \mu(s_0) h(\tau) d\tau$$

However, this computation is susceptible to the curse of dimensionality – it is quite difficult to compute this integral. Instead, we can apply the following trick to ease the computation.

$$\nabla_p \mathbb{E}_{p(x)}[f(x)] = \int \nabla p(x) f(x) dx$$

$$= \int p(x) \frac{\nabla p(x)}{p(x)} f(x) dx$$

$$= \mathbb{E}_{p(x)} \left[\nabla \log p(x) - f(x) \right]$$

Going back to our previous equation, we can apply the above substitution:

$$\nabla_\pi \mathbb{E}_\pi[h(\tau)] = \int \prod_{t=0}^{\infty} p(s_{t+1}|s_t, a_t) \frac{\nabla \pi(a_t|s_t)}{\nabla \pi(a_t|s_t)} \pi(a_t|s_t) h(\tau) d\tau$$

$$= \int \prod_{t=0}^{\infty} p(s_{t+1}|s_t, a_t) \nabla \log \left(\prod_{t=0}^{\infty} \pi(a_t|s_t) \right) \pi(a_t|s_t) h(\tau) d\tau$$

$$= \mathbb{E}_\pi \left[h(\tau) \cdot \sum_{t=0}^{\infty} \nabla \log \pi(a_t|s_t) \right]$$

This equation can also be adapted to incorporate policy ratios, i.e.

$$\mathbb{E}_{\pi_b} \left[\prod_{t=0}^{\infty} \frac{\pi}{\pi_b} \cdots \right]$$

Note that this computation involves the policy ratios being multiplied for each step, which will go to 0 and result in updates that are not meaningful. This update approach is still good for one step.

23.2.2 Gradient Estimator

Unfortunately, when taking multiple steps, the computation of the gradient $\nabla_\pi J(\pi)$ in every step becomes computationally intensive. We address this by devising the gradient estimator below.

$$\nabla_\pi J(\pi) = \nabla_\pi \mathbb{E}_{\mu_0}[V^\pi(s)] = \mathbb{E}_{\mu_0}[\nabla_\pi V^\pi(s)] = \nabla_\pi V^\pi(s) =$$

$$\nabla_\pi \int \pi(a|s) Q^\pi(s,a) da =$$

$$\int [\nabla_\pi \pi(a|s) Q^\pi(s,a) + \pi(a|s) \nabla Q^\pi(s,a)] da =$$

We can now apply the Bellman equation on $Q^\pi(s,a)$:

$$\int \pi(a|s)[\nabla log(\pi(a|s) Q^\pi(s,a) + \nabla(R + \gamma \mathbb{E}_\pi[V^\pi(s_1)])] =$$

$$\int \pi(a|s)[\nabla log(\pi(a|s) Q^\pi(s,a) + \gamma \mathbb{E}_\pi[V^\pi(s_1)]]$$

$$\nabla_\pi V^\pi(s) = \mathbb{E}_\pi[\nabla log(\pi(a,s) Q^T(s,a)] + \gamma \mathbb{E}_{\pi_(s_1|s)}[V^\pi(s_1)] =$$

$$\mathbb{E}_\pi\left[\nabla \log \pi(a|s) Q^\pi(s,a)\right] + \gamma\, \mathbb{E}_{p^\pi(s\prime|s)}\left[\nabla \log \pi(a\prime|s\prime) Q^\pi(s\prime, a\prime)\right] + \gamma^k\, \mathbb{E}_{p^\pi(s_k|s)}\left[V^\pi(s_k)\right]$$

$$\mathbb{E}_{\mu_0}\left[\nabla_\pi V^\pi(s)\right] = \sum_{t=0}^{\infty} \gamma^t\, \mathbb{E}_\mu\, \mathbb{E}_{p^\pi(s_t|s)}\left[\nabla \log \pi(a_t|s_t) Q^\pi(s_t, a_t)\right]$$

We can introduce a notation change for the cumulative distribution, writing this expectation term as $d_t^\pi(s_t)$. Thus,

$$\sum_{t=0}^{\infty} \gamma^t d_t^\pi(s_t) = (1-\gamma) d^\pi(s)$$

We then have our gradient estimator expressed as

$$\nabla_\pi J(\pi) = \mathbb{E}_{d^\pi(s)\pi(a|s)}\left[\nabla \log \pi(a|s) Q^\pi(s,a)\right]$$

This new approach reduces variance and computational cost, and we can also use off-policy which makes it more data-efficient.

23.2.3 Q^π Estimator

With the gradient estimator equation, we have addressed concerns regarding variance domination and the computational cost of gradient calculation. The main difficulty now is calculating Q^π, which we can address by devising a Q^π estimator.

1. Initialize $\theta_0 \dots Q^0_\varphi$
2. for $i = 1 \dots n$, evaluate $\hat{Q}^{\pi_i}$:
   plug $\hat{Q}^{\pi_i}$ for $\widehat{\nabla}_{\pi_i} J(\pi_i)$
   update $\theta_{t+1} = \theta_t + \eta \widehat{\nabla}_\pi J(\pi)$

In this algorithm, θ_{t+1} is the actor that attempts to implement a policy, and $\hat{Q}^{\pi_i}$ is the critic that evaluates the current policy to provide feedback.

Now, we will use the policy gradient.

$$\nabla_\pi J(\pi) = \mathbb{E}_{d,\pi}\left[\nabla \log(\pi(a|s))(Q(s,a) - f(s))\right]$$

where $f(s)$ represents the baseline.

$$\mathbb{E}_d\left[\nabla \log(\pi(a|s))(f(s)\right] = 0$$
$$= \int_d \nabla \pi(a|s) f(s) da$$

$$\mathbb{E}_\pi\left[\nabla \log(\pi(a|s))(f(s)\right] = 0$$
$$= \int_d \pi(a|s) \frac{\nabla \pi(a|s)}{\pi(a|s)} f(s) da = f(s) \cdot \left(\nabla \int \pi(a|s) da\right) = f(s)$$

Thus, we have devised a valid Q^π estimator for any f(s).

$$f^* = \underset{f}{\operatorname{argmin}}(\operatorname{Var}[\nabla \log(\pi(a|s))(Q(s,a) - f(s))]$$

In practice, we don't use f^*. Instead, we use the control variate below to estimate it because it is easier to compute:

$$V^\pi(s) = \mathbb{E}[Q^\pi(s,a)], \quad \text{where } \pi_t(a^*|s) > 0 \quad \forall t$$