Function Approximation and the NPG

Presented by :
Atmani Hanan

University mohammed vi polytechnic

July 12

Presentation plan

1. Introduction and motivation
2. Compatible function approximation and the NPG
3. Examples: NPG and Q-NPG
   - Log-linear Policy Classes and Soft Policy Iteration
   - Neural Policy Classes

Introduction

- In this chapiter we will analize the case of using parametric policy classes:

$$\Pi = \left\{ \pi_\theta \mid \theta \in \mathbb{R}^d \right\}$$

  - Π may not contain all stochastic policies (and it may not even contain an optimal policy)
  - Π are not fully expressive, $d \ll |\mathcal{S}||\mathcal{A}|$ (indeed $|\mathcal{S}|$ or $|\mathcal{A}|$ need not even be finite for the results in this section)
- Objective:
  - Establish a connection between the NPG (Natural Policy Gradient) algorithm and compatible function approximation.
  - Assess the effectiveness of NPG updates in the presence of errors due to statistical estimation (where we may not use exact gradients) and approximation

Compatible function

Definition

Compatible function approximation : *A compatible function is a function chosen to approximate a specific problem in such a way that it fits well with the characteristics of that problem.*

Lemma

Let $w^\star$ denote the following minimizer:

$$w^\star \in \operatorname{argmin}_w \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[\left(A^{\pi_\theta}(s,a) - w \cdot \nabla_\theta \log \pi_\theta(a \mid s) \right)^2 \right], \tag{1}$$

- The loss function mentioned above is referred to as the error of the compatible function approximation

Compatible function approximation and the NPG

- This optimization problem is a linear regression problem aiming to approximate the function $A^{\pi_\theta}(s,a)$ using the $\nabla_\theta \log \pi_\theta(\cdot \mid s)$ as features
- Denote the best linear predictor of $A^{\pi_\theta}(s,a)$ using $\nabla_\theta \log \pi_\theta(a \mid s)$ by $\widehat{A}^{\pi_\theta}(s,a)$, i.e.

$$\widehat{A}^{\pi_\theta}(s,a) := w^\star \cdot \nabla_\theta \log \pi_\theta(a \mid s).$$

Proposition

We have that:

$$\nabla_\theta V^{\pi_\theta}(\mu) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[\nabla_\theta \log \pi_\theta(a \mid s) \hat{A}^{\pi_\theta}(s,a) \right].$$

- **Proof:** Use the first-order optimality condition in (1), and utilize the advantage expression of $\nabla_\theta V^{\pi_\theta}(\mu)$

Compatible function approximation and the NPG

Lemma

We have that:

$$F_\rho(\theta)^\dagger \nabla_\theta V^\theta(\rho) = \frac{1}{1-\gamma} w^\star,$$

- This lemma shows that the weight vector above precisely corresponds to the ascent direction of NPG
- This lemma implies that we might write the NPG update rule as:

$$\theta \leftarrow \theta + \frac{\eta}{1-\gamma} w^\star. \tag{2}$$

Examples: NPG and Q-NPG

- In practice, the most common policy classes are of the form:

$$\Pi = \left\{ \pi_\theta(a \mid s) = \frac{\exp\left(f_\theta(s,a)\right)}{\sum_{a' \in \mathcal{A}} \exp\left(f_\theta\left(s,a'\right)\right)} \,\middle|\, \theta \in \mathbb{R}^d \right\},$$

  where f_θ is a differentiable function
- Π as the tabular softmax policy class if $f_\theta(s,a) = \theta_{s,a}$.
- Π as the Log-linear policies if $f_\theta(s,a) = \theta \cdot \phi_{s,a}$
- Π as the Neural softmax policies if $f_\theta(s,a)$ is a neural network parameterized by θ

Log-linear Policy Classes and Soft Policy Iteration

- For any state-action pair (s, a), suppose we have a feature mapping $\phi_{s,a} \in \mathbb{R}^d$. Each policy in the log-linear policy class is of the form Π where $f_\theta(s, a) = \theta \cdot \phi_{s,a}$
- Compatible function approximation for the log-linear policy class as:

$$\nabla_\theta \log \pi_\theta(a \mid s) = \bar{\phi}^\theta_{s,a}, \text{ where } \quad \bar{\phi}^\theta_{s,a} = \phi_{s,a} - \mathbb{E}_{a' \sim \pi_\theta(-|s)} \left[\phi_{s,a'}\right],$$

- $\bar{\phi}^\theta_{s,a}$ is the centered version of $\phi_{s,a}$.
- The NPG update using Log-linear Policy Classes

$$\text{NPG:} \quad \theta \leftarrow \theta + \eta w_\star$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s \sim d^\pi_\rho} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[\left(A^{\pi_\theta}(s, a) - w \cdot \bar{\phi}^\theta_{s,a} \right)^2 \right].$$

- We have rescaled the learning rate η in comparison to (2)

Log-linear Policy Classes and Soft Policy Iteration

- Here, the compatible function approximation error assesses how effectively our parameterization can capture the policy's advantage function using linear functions.
- The Q-NPG using Log-linear Policy Classes :

$$\text{Q-NPG:} \quad \theta \leftarrow \theta + \eta w_\star,$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s \sim d_\rho^\pi} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[\left(Q^{\pi_\theta}(s,a) - w \cdot \phi_{s,a} \right)^2 \right].$$

- We do not center the features for Q-NPG
- observe that $Q^\pi(s,a)$ is also not 0 in expectation under $\pi(\cdot \mid s)$, unlike the advantage function.

Log-linear Policy Classes and Soft Policy Iteration

- Using the last lemma from Chapter 2, we observe how both NPG and Q-NPG can be seen as an incremental (soft) version of policy iteration. We can write an equivalent update rule directly in terms of the (log-linear) policy π:

$$\text{NPG: } \pi(a \mid s) \leftarrow \pi(a \mid s) \exp\left(w_\star \cdot \phi_{s,a}\right) / Z_s,$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s \sim d_\rho^\pi} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[\left(A^\pi(s,a) - w \cdot \bar{\phi}^\pi_{s,a} \right)^2 \right],$$

where Z_s is normalization
- The normalization makes the update invariant to (constant) translations of the features.

- Similarly, an equivalent update for $Q - \mathrm{NPG}$, where we update π directly rather than θ, is:

$$\textit{Q}\text{-NPG: } \pi(a \mid s) \leftarrow \pi(a \mid s) \exp\left(w_\star \cdot \phi_{s,a}\right) / Z_s$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s \sim d_\rho^\pi} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[\left(Q^\pi(s,a) - w \cdot \phi_{s,a} \right)^2 \right].$$

- If the compatible function approximation error is 0 then the NPG and Q-NPG are equivalent algorithms

Neural Policy Classes

- Now, suppose that Now suppose $f_\theta(s, a)$ in Π is a neural network parameterized by θ
- Compatible function approximation in this case is:

$$\nabla_\theta \log \pi_\theta(a \mid s) = g_\theta(s, a)$$

where

$$g_\theta(s, a) = \nabla_\theta f_\theta(s, a) - \mathbb{E}_{a' \sim \pi_\theta(\cdot|s)} \left[\nabla_\theta f_\theta \left(s, a' \right) \right],$$

- the NPG update is:

$$\text{NPG: } \theta \leftarrow \theta + \eta w_\star,$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s \sim d_\rho^{\pi_\theta^*}, a \sim \pi_\theta(\cdot|s)} \left[\left(A^{\pi_\theta}(s, a) - w \cdot g_\theta(s, a) \right)^2 \right]$$

Neural Policy Classes

- The Q-NPG variant of this update rule is:

$$Q\text{-NPG:} \quad \theta \leftarrow \theta + \eta w_\star,$$

$$w_\star \in \operatorname{argmin}_w \mathbb{E}_{s\sim d_\rho^\pi} \mathbb{E}_{a\sim\pi(\cdot|s)} \left[\left(Q^{\pi_\theta}(s,a) - w \cdot \nabla_\theta f_\theta(s,a) \right)^2 \right].$$