Paper Critique

Shuvrajeet Das, DA24D402

Course: DA7400, Fall 2024, IITM
Paper: [PROJECTION -BASED CONSTRAINED POLICY OPTIMIZATION]
Date: [18-09-24]

Make sure your critique Address the following points:
1. The problem the paper is trying to address
2. Key contributions of the paper
3. Proposed algorithm/framework
4. How the proposed algorithm addressed the described problem
Note: Be concise with your explanations. Unnecessary verbosity will be penalized. Please don't exceed 2 pages.

---

1 The problem the paper is trying to address

The paper addresses the challenge of learning control policies that optimize a reward function while adhering to safety, fairness, or cost constraints in real-world applications, such as autonomous vehicles and robotic systems. The problem is framed as a constrained Markov Decision Process (CMDP), where the goal is to maximize cumulative discounted rewards:

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

while ensuring that the cumulative discounted cost stays below a predefined threshold h:

$$J_C(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t C(s_t, a_t)\right] \leq h.$$

The paper proposes Projection-Based Constrained Policy Optimization (PCPO) to iteratively update the policy for reward maximization while projecting it back onto the constraint set to handle violations effectively.

2 Key contributions of the paper

Key Contributions:
The paper makes the following key contributions:

- **Projection-Based Constrained Policy Optimization (PCPO):** The authors introduce a novel algorithm, PCPO, designed to handle constrained policy optimization in reinforcement learning. The method performs policy updates in two stages:

  First stage: $\pi_{k+\frac{1}{2}} = \arg\max_{\pi} \mathbb{E}_{s\sim d_{\pi_k}, a\sim\pi}[A^R_{\pi_k}(s,a)]$ subject to $\mathbb{E}_{s\sim d_{\pi_k}}[D_{KL}(\pi||\pi_k)] \leq \delta$,

  Second stage: $\pi_{k+1} = \arg\min_{\pi} D(\pi, \pi_{k+\frac{1}{2}})$ subject to $J_C(\pi_k) + \mathbb{E}_{s\sim d_{\pi_k}, a\sim\pi}[A^C_{\pi_k}(s,a)] \leq h$.

  The algorithm ensures that the policy improves the reward in the first step and satisfies constraints in the second step through projection onto the constraint set.

- **Theoretical Performance Bound:** The paper provides theoretical guarantees for PCPO by deriving: - A lower bound on reward improvement:

$$J_R(\pi_{k+1}) - J_R(\pi_k) \geq -\frac{\sqrt{2\delta\gamma}\epsilon_R^{k+1}}{(1-\gamma)^2},$$

- An upper bound on constraint violation:

$$J_C(\pi_{k+1}) \leq h + \frac{\sqrt{2\delta\gamma}\epsilon_C^{k+1}}{(1-\gamma)^2}.$$

These bounds ensure that constraint violations and reward degradation remain tolerable.

- **Empirical Performance:** The authors demonstrate that PCPO achieves superior performance on several control tasks compared to state-of-the-art methods, with more than 3.5 times fewer constraint violations and approximately 15% higher reward. PCPO is tested on multiple tasks, including Mujoco safety environments and traffic management tasks, and it shows effectiveness in real-world scenarios.

3 Proposed algorithm/framework

Algorithm 1 Projection-Based Constrained Policy Optimization (PCPO)

1: Initialize policy $\pi^0 = \pi(\theta^0)$
2: **for** $k = 0, 1, 2, \cdots$ **do**
3: Run $\pi^k = \pi(\theta^k)$ and store trajectories in $\mathcal{D}$
4: Compute g, a, H, and b using $\mathcal{D}$
5: Obtain θ^{k+1}
6: Empty $\mathcal{D}$
7: **end for**

4 How the proposed algorithm addressed the problem

- **Two-Step Policy Update:** The algorithm separates reward optimization and constraint satisfaction into two steps. In the first step, the policy is updated to maximize the reward:

$$\pi_{k+\frac{1}{2}} = \arg\max_{\pi} \mathbb{E}_{s\sim d_{\pi_k}, a\sim\pi}[A^R_{\pi_k}(s,a)],$$

ensuring that the policy improves the reward function while staying within a trust region defined by a KL divergence constraint. This keeps the policy from deviating too far from the current policy, providing stability.

- **Projection for Constraint Satisfaction:** In the second step, the policy is projected back onto the set of feasible policies that satisfy the constraints. The projection solves:

$$\pi_{k+1} = \arg\min_{\pi} D(\pi, \pi_{k+\frac{1}{2}}) \quad \text{subject to} \quad J_C(\pi_k) + \mathbb{E}_{s\sim d_{\pi_k}, a\sim\pi}[A^C_{\pi_k}(s,a)] \leq h,$$

ensuring that any violations of the cost constraints are corrected in the projected policy.

- **Trust Region and Safe Exploration:** By limiting the step size in the policy update (through the KL divergence constraint), the algorithm ensures safe exploration during learning. This prevents the policy from taking large, risky updates that could violate the constraints.