Paper Critique

Anirud N, CE21B014

Course: DA7400, Fall 2024, IITM
Paper: Conservative Safety Critic for Exploration
Date: 20 September 2024
Make sure your critique addresses 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 Problem Statement

The aim of the paper is to develop safe exploration methods in RL, implement "conservative" behavior, be over-cautious, and limit the number of catastrophic failures. There have been several approaches to safe RL. However, most of the approaches required additional assumptions - they assumed access to the function that can be used to check if a state is safe or not or assumed access to a default safe controller. Few of them also focused on obtaining safe policies only after the training is complete, allowing "unsafe" actions during the training phase.

2 Key Contributions

- The paper proposes a general safe RL algorithm with bounds on the probability of failures during training
- Training a conservative critic that overestimates the probability of failures
- Designed an overall algorithm called Conservative Safety Critics that leans a conservative estimate of how safe the state is and uses this conservative estimate for safe explorations throughout the training process.
- Empirical evaluation in five different robotic environments to measure performance
- Theoretically categorize the tradeoff between safety and policy improvement

3 Proposed Algorithm/Framework

The safety constraint $C(s)$: $C(s) = 1$ when a state s is unsafe and $C(s) = 0$ when it is safe. So, we formulate the constraint as $V_C^\pi(\mu) = E_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} C(s_t)\right] \leq \chi$, where $\chi \in [0, 1)$ denotes *probability of failure*. We aim to reduce this probability of failure. The paper explains that being safe with respect to how safe the state is helps to overestimate the failure, and so it helps to ensure constrained exploration.
$Q_C(s, a)$ helps to identify how unsafe the action a at state s is. $Q_C(s, a)$ is used to ensure constrained exploration - it checks whether the chance of failure is less than ϵ, if it is not so then it re-samples action a from current policy $\pi(a|s)$

3.1 Learning Safety Critic Q_c

To train the safety critic Q_c we use code from CQL, to estimate Q_c, which gives an upper bound. Objective function :

$$\begin{aligned} \hat{Q}_C^{k+1} \leftarrow argmin_{Q_C} \quad & \alpha \cdot \Big(-\mathbb{E}_{s\sim\mathcal{D}_{env},a\sim\pi_\phi(a|s)}[Q_C(s,a)] + \mathbb{E}_{(s,a)\sim\mathcal{D}_{env}}[Q_C(s,a)]\Big) \\ & + \frac{1}{2}\mathbb{E}_{(s,a,s',c)\sim\mathcal{D}_{env}}\left[\Big(Q_C(s,a) - \hat{\mathcal{B}}^{\pi_\phi}\hat{Q}_C^k(s,a)\Big)^2\right] \end{aligned} \tag{1}$$

k denotes the iteration number, and B is the Belman operator. α is the weight term that gives importance to the first term.

3.2 Policy Learning

Constrained optimization problem - this lagrangian is constructed through a primal-dual gradient descent. The second term of the objective function is how satisfied safety is.

$$\max_{\pi_\phi}\min_{\lambda\geq 0}\mathbb{E}_{s\sim\rho_\phi,a\sim\pi_\phi}\left[A_R^{\pi_\phi}(s,a) - \lambda\left(Q_C(s,a) - \chi\right)\right]$$

3.3 Safe Exploration - rollouts

Instead of executing the learned policy directly, they sample actions $a \sim \pi_{\phi_{old}}(s)$ and only select actions where $Q_C(s,a) \leq \epsilon$, using rejection sampling. This process repeats for 100 iterations, and they choose the action with the minimum $Q_C(s,a)$ value. If no action meets the condition, select the action with the smallest $Q_C(s,a)$, even if it exceeds the threshold. ϵ varies across iterations - theoretically obtained.

3.4 Implementation

KL divergence is given to ensure that the successive policies are closer in order to help obtain bounds on the expected failures. A second-order approximation of this is used.

$$\begin{aligned} & \max_{\pi_\phi} \quad \mathbb{E}_{s\sim\rho_{\phi_{old}},a\sim\pi_\phi}\left[A_R^{\pi_{\phi_{old}}}(s,a)\right] \\ & \text{s.t.} \quad \mathbb{E}_{s\sim\rho_{\phi_{old}}}[D_{\mathrm{KL}}(\pi_{\phi_{old}}(\cdot|s)||\pi_\phi(\cdot|s))] \leq \delta \quad \text{and} \quad V_C^{\pi_\phi}(\mu) \leq \chi \end{aligned} \tag{2}$$

4 Conclusions and Results

The algorithm shows nearly better performance or equal lesser failures during the training. CPO and Q ensemble baselines also achieve near 0 average failures, but the advantage is CSC archives this much faster and earlier during training. CSC proves a better trade-off between performance returns and safety constraints. It was also noted that even when the safety threshold was set to 0, there were a few failures in the initial stages of training - this is due to the "high function approximation" error of the learned Q_c, which is a drawback- inability to account for failures due to approximations. The approach does not assume any access to the user-specified constraint function, which sometimes might lead to an issue in challenging environments - without additional assumptions.