Contribution Title

Minseok Seo[1][0000−1111−2222−3333] and Kyunghwan Choi[1][1111−2222−3333−4444]

Cho Chun Shik Graduate School of Mobility, KAIST
{seominseok,kh.choi}@kaist.ac.kr
https://kaist-mic-lab.github.io/

Abstract. Despite the remarkable success of deep reinforcement learning (deep RL) across various domains, its deployment in real-world remains limited due to safety concerns. To address this challenge, constrained reinforcement learning (constrained RL) has been proposed to learn safe policies while maintaining performance. However, since constrained RL enforces constraints in the form of cumulative costs, it cannot guarantee state-wise safety. In this paper, we extend the Lagrangian based approach, a representative method in constrained RL, by introducing state-dependent Lagrange multipliers so that the policy is trained to account for state-wise safety.

1 INTRODUCTION

Reinforcement learning (RL) learns policy that maximize rewards through trial and error. This way of learning the desired behaviors may seem simple and straightforward, but it is highly effective. Over the past few years, RL has demonstrated impressive achievements in diverse applications ... [12] [4] [8]. Nevertheless, deploying RL in physical real-world environments remains a major challenge. To deploy RL-trained agents in real-world environments, two requirements must be satisfied: First, the ability to successfully accomplish the given tasks, and second, the safety and reliability of the learned agent. In standard RL, such requirements are learned exclusively from reward signals, thereby necessitating careful reward engineering. However, even with carefully designed rewards and successful training in simulation, the learned policy may fail when tasks change or when transferring to real-world environments, due to issues such as reward hacking or lack of generalization [3]. To address these issues, constrained reinforcement learning (constrained RL) has recently been widely studied.

Constrained RL is a method that learns policy maximizing rewards while satisfying constraints, thus enabling agents to behave safely while successfully performing tasks. A common formulation of constrained RL specifies constraints at the trajectory level [5]. As a result, since the constraints are specified over entire trajectory, violations occurring at individual timesteps cannot be directly constrained/restricted. For example, even if a constraint is defined to limit the number of lane departures during the entire trip of an autonomous vehicle, a single deviation from the lane that results in a collision with another vehicle can lead to a catastrophic failure. This illustrates that trajectory-level constraints regulate only the outcome of a task but are limited in controlling the risky process itself. Therefore, in this paper, we propose an approach that considers state-wise constraints. Specifically, we extend the Lagrangian-based approach, a representative method in constrained RL, by introducing state-wise Lagrange multipliers so that the policy is encouraged to take safe actions at every timestep.

2 Related Work

In the context of constrained RL, Liu et al. [7] discuss various formulations in the CMDP setting and introduce corresponding model-free approaches. Among representative methods, constrained policy optimization (CPO) [1], surrogate functions are used to approximate both the objective and the constraints, and a projection step is employed to enforce constraint satisfaction. This procedure requires a backtracking line search, which makes the algorithm computationally expensive. Another line of work includes PPO Lagrangian and TRPO Lagrangian [9], which extend the PPO [11] and TRPO [10] algorithms to the constrained RL setting through Lagrangian relaxation, reformulating the constrained policy optimization problem as an unconstrained max-min optimization problem. By adaptively adjusting the Lagrange multiplier, these methods encourage policy updates toward satisfying the constraints. However, since constraint violations are necessary for learning a safe policy, the constraints are typically not satisfied during training. In interior-point policy optimization (IPO) [6], logarithmic barrier functions are added to the objective as penalty terms to account for the constraints. While this approach is easy to implement and can handle multiple constraints, it assumes feasible iterates, which can be problematic in situations such as random agent initialization where constraint violations may occur. Stooke et al. [13] demonstrate that in Lagrangian-based methods, the Lagrange multiplier is updated only through integral control, resulting in oscillation and overshoot issues, and proposea a method that incorporates proportional and derivate terms to stabilize the updates.

3 Methodology

3.1 Preliminaries

Constrained Markov Decision Processes In RL, problems are typically formulated as Markov decision process (MDP) [14], in contrast, constrained RL employs a constrained Markov decision process (CMDP) [2]. A CMDP extends an MDP by introducing a set of cost functions $C_1, \ldots, C_m$, which are separate from the reward function, together with corresponding thresholds $d_1, \ldots, d_m$. Formally, a CMDP is defined as a tuple $\langle S, A, P, R, C, d, \gamma \rangle$, where S and A denotes the state and action spaces, P is the transition probability, R is the reward function, C is the set of cost functions, and d denotes the corresponding cost thresholds, γ is the discount factor. In a CMDP, the set of feasible policies Π_C is defined as:

$$\Pi_C = \{\pi \in \Pi : \forall i, J_{C_i}(\pi) \leq d_i\}, \tag{1}$$

where $J_{C_i} = \mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{\infty}\gamma^t C(s_t, a_t)]$ is a cost-based constraint function, commonly defined in the same way as the expected return. Specifically, the expected return is defined as $J_R = \mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_t)]$, and d_i is a threshold chosen as a human-specified hyperparameter. Here, $\tau = (s_0, a_0, s_1, a_1, \ldots)$ denotes a trajectory generated by following policy π. In standard RL, the objective is to sove an optimization problem that maximizes the expected return $\pi^* = \arg\max J_R(\pi)$. Whereas constrained RL solves the following constrained optimization problem:

$$\pi^* = \arg\max_{\pi} J_R(\pi) \text{ s. t. } J_C(\pi) \leq d. \tag{2}$$

State-wise Constrained Markov Decision Process The CMDP framework can be extended to incorporate various types of cost-based constraints. One such extension is the state-wise constrained Markov decision process (SCMDP) [15], which introduces state-wise constraints to ensure that the expected cost at each state does not exceed a specified threshold. In a SCMDP, the set of feasible policies $\varPi_{SC}$ is defined as:

$$\begin{aligned} \varPi_{SC} = \{\pi \in \varPi : \forall (s_t, a_t, s_{t+1}) \sim \tau, \\ \forall i, C_i(s_t, a_t, s_{t+1}) \leq d_i\} \end{aligned} \tag{3}$$

where $C_i(s_t, a_t, s_{t+1})$ is the cost incurred at state s_t after taking action a_t and transitioning to state s_{t+1}, and d_i is the corresponding threshold. Similar to CMDPs, the optimization problem in SCMDPs can be formulated as

$$\pi^* = \arg\max_{\pi} J_R(\pi) \text{ s.t. } J_{SC}(\pi) \leq d, \tag{4}$$

where $J_{SC}(\pi)$ denotes the state-wise constraints.

Lagrangian Relaxation for Constrained Policy Optimization A common approach to solve constrained optimization problems is to use Lagrangian relaxation. In this approach, the constrained optimization problem (2) is reformulated as an unconstrained optimization problem by introducing Lagrange multiplier $\lambda \geq 0$ that penalizes constraint violations. The resulting Lagrangian can be written as:

$$L(\theta, \lambda) = J_R(\pi_\theta) - \lambda(J_C(\pi_\theta) - d), \tag{5}$$

where θ denotes the parameter of the policy π_θ. The objective is then to find a saddle point (θ^*, λ^*) that satisfies:

$$L(\theta^*, \lambda) \geq L(\theta^*, \lambda^*) \geq L(\theta, \lambda^*). \tag{6}$$

Since finding a global saddle point is often computationally intractable, in practice one aims to find a locally optimal solution using iterative updates of the policy parameters and the Lagrange multiplier. A common approach is to apply gradient-based updates of the form

$$\theta_{n+1} = \theta_n + \eta_\theta \nabla_\theta \Big(J_R(\pi_\theta) - \lambda_n J_C(\pi_\theta)\Big), \tag{7}$$

$$\lambda_{n+1} = \Big[\lambda_n + \eta_\lambda\big(J_C(\pi_\theta) - d\big)\Big]_+, \tag{8}$$

where $\eta_\theta, \eta_\lambda > 0$ are step sizes, and $[\cdot]_+$ denotes the projection onto the nonnegative orthant to ensure $\lambda \geq 0$.

3.2 Proposed Method

In this paper, we propose a method to extend constrained reinforcement learning algorithm to state-wise constrained reinforcement learning by estimating state-wise Lagrange multipliers. We build our method on Proximal Policy Optimization (PPO) [11], a widely used policy gradient algorithm, due to its simplicity and effectiveness. The objective function of PPO is defined as follows:

$$J^{\text{PPO}}(\theta) = \mathbb{E}_{\pi_{\theta_{\text{old}}}}\left[\min\left(r(\theta)A^{\pi_{\theta_{\text{old}}}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)A^{\pi_{\theta_{\text{old}}}}(s,a)\right)\right] \tag{9}$$

To solve Eq. (4), we apply Lagrangian relaxation and reformulate the objective as follows:

$$\begin{aligned} J^{\text{PPO-Lagnet}}(\theta) = \mathbb{E}_{\pi_{\theta_{\text{old}}}}\Big[& \min\left(r(\theta)A^{\pi_{\theta_{\text{old}}}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)A^{\pi_{\theta_{\text{old}}}}(s,a)\right) \\ & - \lambda_\xi(s) r(\theta) A_c^{\pi_{\theta_{\text{old}}}}\Big] \end{aligned} \tag{10}$$

In order to transform Eq. (4) into a Lagrangian function, a separate Lagrange multiplier is required for each state because the constraints are defined in (3). Therefore, we employ a neural network to estimate/approximate state-dependent Lagrange multipliers, enabling scalable and efficient policy updates. The state-wise Lagrange multiplier network is parameterized by ξ and takes the state s as input, outputting the corresponding Lagrange multiplier $\lambda_\xi(s)$. The parameters of the Lagrange multiplier network are updated according to

$$\lambda_\xi(s) \leftarrow \lambda_\xi(s) + \beta(\hat{J}_{SC} - d) \tag{11}$$

where $\hat{J}_{SC}$ is the empirical estimate of the state-wise constraint, and d is the threshold. In this update rule, the network's output increases when the constraint is violated ($\hat{J}_{SC} > d$) and decreases when the constraint is satisfied ($\hat{J}_{SC} < d$).

4 Experiments

5 CONCLUSIONS

Acknowledgments. A bold run-in heading in small font size at the end of the paper is used for general acknowledgments, for example: This study was funded by X (grant number Y).

Disclosure of Interests. It is now necessary to declare any competing interests or to specifically state that the authors have no competing interests. Please place the statement with a bold run-in heading in small font size beneath the (optional) acknowledgments[1], for example: The authors have no competing interests to declare that are relevant to the content of this article. Or: Author A has received research grants from Company W. Author B has received a speaker honorarium from Company X and owns stock in Company Y. Author C is a member of committee Z.

References

1. Achiam, J., Held, D., Tamar, A., Abbeel, P.: Constrained policy optimization. In: International conference on machine learning. pp. 22–31. PMLR (2017)
2. Altman, E.: Constrained Markov decision processes. Routledge (2021)
3. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in ai safety. arXiv preprint arXiv:1606.06565 (2016)
4. Andrychowicz, M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al.: Learning dexterous in-hand manipulation. The International Journal of Robotics Research **39**(1), 3–20 (2020)
5. Brunke, L., Greeff, M., Hall, A.W., Yuan, Z., Zhou, S., Panerati, J., Schoellig, A.P.: Safe learning in robotics: From learning-based control to safe reinforcement learning. Annual Review of Control, Robotics, and Autonomous Systems **5**(1), 411–444 (2022)
6. Liu, Y., Ding, J., Liu, X.: Ipo: Interior-point policy optimization under constraints. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 4940–4947 (2020)
7. Liu, Y., Halev, A., Liu, X.: Policy learning with constraints in model-free reinforcement learning: A survey. In: The 30th international joint conference on artificial intelligence (ijcai) (2021)
8. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)
9. Ray, A., Achiam, J., Amodei, D.: Benchmarking safe exploration in deep reinforcement learning. arXiv preprint arXiv:1910.01708 **7**(1), 2 (2019)
10. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. In: International conference on machine learning. pp. 1889–1897. PMLR (2015)
11. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
12. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of go without human knowledge. nature **550**(7676), 354–359 (2017)
13. Stooke, A., Achiam, J., Abbeel, P.: Responsive safety in reinforcement learning by pid lagrangian methods. In: International Conference on Machine Learning. pp. 9133–9143. PMLR (2020)
14. Sutton, R.S., Barto, A.G., et al.: Reinforcement learning: An introduction, vol. 1. MIT press Cambridge (1998)
15. Zhao, W., He, T., Chen, R., Wei, T., Liu, C.: State-wise safe reinforcement learning: A survey. arXiv preprint arXiv:2302.03122 (2023)

[1] If EquinOCS, our proceedings submission system, is used, then the disclaimer can be provided directly in the system.