Reward Constrained Policy Optimization

Ting-Hsuan, Huang
Department of Information Management And Finance
National Yang Ming Chiao Tung University
gyxuan0527.mg07@nycu.edu.tw

1 Introduction

- The main research challenges tackled by the paper :
  Tessler et al. [2018] wanted to solve the issue of **failing to maximize the accumulated reward**.
- The high-level technical insights into the problem of interest :
  To solve a multi-objective problem, Tessler et al. [2018] introduced constraint and **incorporated the constraint as a penalty signal into the reward function**.
- The main contributions of the paper (compared to the prior works) : Tessler et al. [2018] proposed a novel multi-timescale approach for constrained optimization called **RCPO**. The novelty of this work lies in the ability to tackle :
  (1) General constraints, that is not only constraints which satisfy the recursive Bellman equation as in previous work.
  (2) Reward agnostic, that is invariant to scaling of the underlying reward signal.
  (3) Prior knowledge unnecessary.
- Your personal perspective on the proposed method :
  I think it is a powerful and classic actor-critic based algorithm which trains the actor-critic by an alternative , discounted penalty and can be proved to converge to a fixed feasible point.

2 Problem Formulation

- $\max_{\pi\in\Pi} J_R^\pi$, where $\quad J_R^\pi = \mathbb{E}_{s\sim\mu}^\pi [\sum_{t=0}^\infty \gamma^t r_t] = \sum_{s\in S} \mu(s) V_R^\pi(s)$ (1)

  The original maximization problem, to maximize the reward of following policy π: $\gamma \in [0, 1)$ is the discounted factor, r_t is reward at time t, $\mu(s)$ is initial state distribution, $V_R^\pi(s)$ is value of following policy π starting from state s.

- $J_C^\pi = \mathbb{E}_{s\sim\mu}^\pi[C(s)]$ (2)

  Define the constraint of following policy π: $C(s)$ is constraint of state s.

- $\max_{\pi\in\Pi} J_R^\pi \quad$, s.t. $\quad J_C^\pi \leq \alpha$ (3)

  To maximize the reward of following policy π with constraint(combine eq(2) into eq(2)): J_C^π is constraint of following policy π, $\alpha \in [0, 1]$ is threshold.

- $\min_{\lambda\geq 0} \max_\theta L(\lambda, \theta) = \min_{\lambda\geq 0} \max_\theta [J_R^{\pi_\theta} - \lambda \cdot (J_C^{\pi_\theta} - \alpha)]$ (4)

  The converted maximization problem, unconstrained optimization problem (doing Lagrange relaxation w.r.t eq(3)): $J_R^{\pi_\theta}$ is the reward of following policy π, $\lambda \geq 0$ is Lagrange

multiplier (penalty coefficient), $J_C^{\pi_\theta}$ is constraint of following policy π, $\alpha \in [0,1]$ is threshold.

- $\lambda_{k+1} = \Gamma_\lambda \left[\lambda_k - \eta_1(k) \nabla_\lambda L\left(\lambda_k, \theta_k\right)\right]$ (5)

  Soving the problem with penalty coefficient gradient update : $\Gamma_\lambda \in [0, \lambda_{max}]$ is projection operator, $\lambda_k \geq 0$ is penalty coefficient at time k, η_1 is step size, $\nabla_\lambda L\left(\lambda_k, \theta_k\right)$ is gradient(explain in (8)).

- $\theta_{k+1} = \Gamma_\theta \left[\theta_k + \eta_2(k) \nabla_\theta L\left(\lambda_k, \theta_k\right)\right]$ (6)

  Solving the problem with policy parameter gradient update : $\Gamma_\theta \in [0, \lambda_{max}]$ is projection operator, θ_k is policy parameter at time k, η_2 is step size, $\nabla_\theta L\left(\lambda_k, \theta_k\right)$ is gradient(explain in (7)).

- $\nabla_\theta L(\lambda, \theta) = \nabla_\theta \mathbb{E}_{s\sim\mu}^{\pi_\theta}[\log \pi(s, a; \theta)[R(s) - \lambda \cdot C(s)]]$ (7)

  The gradient of policy parameter: $\log \pi(s, a; \theta)$ is log probability at state s action a with parameter θ, $R(s)$ is reward, $\lambda \geq 0$ is penalty coefficient, $C(s)$ is constraint.

- $\nabla_\lambda L(\lambda, \theta) = -\left(\mathbb{E}_{s\sim\mu}^{\pi_\theta}[C(s)] - \alpha\right)$ (8)

  The gradient of penalty coefficient: $C(s)$ is constraint, $\alpha \in [0,1]$ is threshold.

- $V_{C_\gamma}^{\pi}(s) \triangleq \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t c\left(s_t, a_t\right) \mid s_0 = s\right]$ (9)

  Define the value of discounted penalty of following policy π starting from state s: $\gamma \in [0,1)$ is the discounted factor, $c\left(s_t, a_t\right)$ is constraint at state st action at.

- $\hat{r}(\lambda, s, a) \triangleq r(s, a) - \lambda c(s, a)$ (10)

  Define the value of penalized reward at state s action a with penalty coefficient λ: $r(s,a)$ is reward at state s action a, $\lambda \geq 0$ is penalty coefficient, $c(s,a)$ is constraint at state s action a.

- $$\begin{aligned} \hat{V}^{\pi}(\lambda, s) &\triangleq \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \hat{r}\left(\lambda, s_t, a_t\right) \mid s_0 = s\right] \\ &= \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r\left(s_t, a_t\right) - \lambda c\left(s_t, a_t\right)\right) \mid s_0 = s\right] = V_R^{\pi}(s) - \lambda V_{C_\gamma}^{\pi}(s) \end{aligned} \tag{11}$$

  Combine eq(9) and eq(10) into penalized reward function: $V_R^{\pi}(s)$ is value of following policy π starting from state s, $\lambda \geq 0$ is penalty coefficient, $V_{C_\gamma}^{\pi}(s)$ is value of discounted penalty of following policy π starting from state s.

- **Assumption 1.**: The value $V_R^{\pi}(s)$ is bounded for all policies $\pi \in \Pi$.
- **Assumption 2.**: Every local minima of $J_C^{\pi_\theta}$ is a feasible solution.
- **Assumption 3.**: $\sum_{k=0}^{\infty} \eta_1(k) = \sum_{k=0}^{\infty} \eta_2(k) = \infty$, $\sum_{k=0}^{\infty} \left(\eta_1(k)^2 + \eta_2(k)^2\right) < \infty$ and $\frac{\eta_1(k)}{\eta_2(k)} \to 0$
- **Theorem 1.**: Under Assumption 3, as well as the standard stability assumption for the iterates and bounded noise (Borkar et al., 2008), the iterates (θ_n, λ_n) converge to a fixed point (a local minima) almost surely.
- **Theorem 2.**: Denote by $\Theta = \{\theta : J_C^{\pi_\theta} \leq \alpha\}$ the set of feasible solutions and the set of local- minimas of $J_{C_\gamma}^{\pi_\theta}$ as Θ_γ . Assuming that $\Theta_\gamma \subseteq \Theta$ then the 'Reward Constrained Policy Optimization' (RCPO) algorithm converges almost surely to a fixed point $\left(\theta^*\left(\lambda^*, v^*\right), v^*\left(\lambda^*\right), \lambda^*\right)$ which is a feasible solution (e.g. $\theta^* \in \Theta$).

- **Lemma 1.**: Under assumptions 1 and 2, the fixed point of Theorem 1 is a feasible solution.

3 Theoretical Analysis

In order to solve the problem of failing to maximize total reward, Tessler et al. [2018] proposed a method called **RCPO**. Generally, we solve the multi-object problem by hyper-parameter tuning which is used to determine the reward signal coefficients. Tessler et al. [2018] **formulated the problem into a constrained optimization problem** (3) without the need for manually selection the penalty coefficients. CMDPs are often solved using the **Lagrange relaxation technique**. In Lagrange relaxation, the CMDP is converted into an equivalent unconstrained problem (4) and update the penalty coefficient and policy parameter using gradient update (5, 6). By assumption 3, theorem 1 and lemma 1, it can be proved to converge to a fixed feasible point.
To solve general constraint with **actor-critic based** method, Tessler et al. [2018] viewed the constraint as discounted penalty (9) and incorporated it as penalty signal into the reward function (11). The approaches to update the penalty coefficient and policy parameter are still gradient update (5, 6). By theorem 2, it can be proved to **converge to a fixed feasible point**.
RCPO uses a multi-timescale approach. On the fast timescale, objective is estimated using a **TD-critic**. On the intermediate timescale, it is solved with **policy gradient**. On the slow timescale, the penalty coefficient λ is learned **by ascending on the original constraint**.

$*$ **Error**:
p.1 1 Introduction
line8 : 3 dimensional -> 3-dimensional
line14: it's -> its

p.3 2.2 Constrained MDPs
line4 : throughout the paper we will... -> throughout the paper**,** we will...

p.3 2.3 Parametrized Policies
line1 : In this work we consider... -> In this work**,** we consider...

p.4 3.1 Estimating The Gradient
line7 derivied -> derived

p.5 4.2 Penalized Reward Functions
line14 The proof to Theorem 2... -> The proof **of** Theorem 2

p.8 Figure3.
X axis, Y axis -> X-axis, Y-axis

p.12 B.1 Mars Rover
line8 Initially... -> Initially**,**...

p.13 B.1 Mars Rover
line1 between the layers we apply... -> between the layers**,** we apply

p.13 B.2 Robotics
line1 For these experiments we used... -> For these experiments**,** we used...
line7 of the previous layers output -> of the previous layer**'**s output

p.13 C Proof Of Theorem 1
line10 earlier-> **the** earlier

p.14 E Proof Of Theorem 2
line1 in this case... -> in this case**,**...
line19 the set of stationary points of the process are limited ...-> the set of stationary points of the process **is** limited

p.15 E Proof Of Theorem 2
line10 however... -> however**,**...

4 Conclusion

- The potential future research directions :
  How to solve general CMDPs **more efficiently** and handle **wide class of constraints** in RL tasks.
- Any technical limitations :
  (1) Without regret and constraint violation analysis.
  (2) Fails to guide how to guarantee safety by an actor-critic structure design.
  (3) Only focus on orthant constraints and single-constraint case.(Miryoosefi et al. [2019])
  (4) Actor-critic is deep learning implementation and hence the computational resources required are high.
  (5) Solving the saddle-point optimization problem requires solving a sequence of MDPs with different reward functions. For a large scale problem, even solving a single MDP requires huge computational resources, making such an approach computationally infeasible.(Miryoosefi et al. [2019])
  (6) Existing theory only provides convergence to a stationary point where the gradient with respect to the policy parameter is zero. Moreover, the objective, as a bivariate function of the Lagrangian multiplier and the policy parameter is not convex-concave. Therefore, first-order iterative algorithms can be unstable.(Miryoosefi et al. [2019])
- Any latest results on the problem of interest :
  Wei et al. [2021] proposed a model-free algorithm for general CMDPs in the **infinite-horizon average reward setting** with provable guarantees. The design of the algorithm is based on the **primal-dual approach**. By using the Lyapunov drift analysis, it can be proved that their algorithm achieves sublinear regret and zero constraint violation. The algorithm is also computationally efficient from an algorithmic perspective because it is model-free, which means that it is potential to apply this method for **complex and challenging CMDPs** in practice.

References

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization, 2018.

Sobhan Miryoosefi, Kianté Brantley, Hal Daume III, Miro Dudik, and Robert E Schapire. Reinforcement learning with convex constraints. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/873be0705c80679f2c71fbf4d872df59-Paper.pdf`.

Honghao Wei, Xin Liu, and Lei Ying. A provably-efficient model-free algorithm for constrained markov decision processes, 2021. URL `https://arxiv.org/abs/2106.01577`.