Ablation: Reinforcement Learning with Convex Constraints

Yao-Te Wang, Chun-Ting Kuo, Yu-Cheng Tsai
Department of Computer Science
National Chiao Tung University
{bpploabc.cs07, jack041286.eic09g}@nctu.edu.tw, snoopy.c@nycu.edu.tw

1 Problem Overview

In standard reinforcement learning (RL), a learning agent seeks to optimize the overall reward. However, many key aspects of a desired behavior are more naturally expressed as constraints. For instance, the designer may want to limit the use of unsafe actions, increase the diversity of trajectories to enable exploration, or approximate expert trajectories when rewards are sparse. In this paper, we propose an algorithmic scheme that can handle a wide class of constraints in RL tasks, specifically, any constraints that require expected values of some vector measurements (such as the use of an action) to lie in a convex set.

2 Background and The Algorithm

2.1 Define the feasibility problem

We begin with a description of our learning setting. A vector-valued Markov decision process is a tuple $M = (S, \mathcal{A}, \beta, P_s, P_z)$, where S is the set of states, $\mathcal{A}$ is the set of actions and β is the initial-state distribution. Each episode starts by drawing an initial state s_0 from the distribution. Then in each step i = 1, 2,... , the agent observes its current state s_i and takes action $a_i \in \mathcal{A}$ causing the environment to move to the next state $s_{i+1} \sim P_s(|s_i, a_i)$. However, in our setting, instead of receiving a scalar reward, the agent observes a d-dimensional measurement vector $z_i \in \mathbb{R}^d$, which, like s_{i+1}, is dependent on both the current state s_i and the action a_i, that is, $z_i \sim Pz(\cdot|s_i, a_i)$.

Our aim is to control the MDP so that measurements satisfy some constraints. For any policy π, we define the long-term measurement $z(\bar{\pi})$ as the expected sum of discounted measurements:

$$\bar{z}(\pi) = E\left[\sum_{i=0}^{\infty} \gamma^i z_i | \pi\right] \tag{1}$$

for some discount factor $\gamma \in [0, 1)$, and where expectation is over the random process described above. Our learning problem, called the feasibility problem, is specified by a convex target set $\mathcal{C}$. The goal is to find a mixed policy μ whose long-term measurements lie in the set $\mathcal{C}$:

$$\text{Feasibility Problem}: \quad \text{Find } \mu \in \Delta(\Pi) \quad \text{such that} \quad \bar{z}(\mu) \in \mathcal{C}. \tag{2}$$

For instance, if $\mathcal{S}$ and $\mathcal{A}$ are finite, then Π might consist of all deterministic policies.

2.2 Solving zero-sum games using online learning

Although feasibility is our main focus, we actually solve the stronger problem of finding a mixed policy μ that minimizes the Euclidean distance between $z(\mu)$ and $\mathcal{C}$, meaning the Euclidean distance between $z(\mu)$ and its closest point in $\mathcal{C}$. That is, we want to solve

$$\min_{\mu \in \Delta(\Pi)} dis(\bar{z}(\mu), \mathcal{C}) = \min_{\mu \in \Delta(\Pi)} \max_{\lambda \in \Lambda} \lambda \cdot z(\mu) \tag{3}$$

Preprint. Under review.

This min-max form immediately evokes interpretation as a two-person zero-sum game: the first player chooses a mixed policy μ, the second player responds with a vector λ, and $\lambda \cdot z(\mu)$ is the amount that the first player is then required to pay to the second player. Assuming this game satisfies certain conditions, the final payout under the optimal play, called the value of the game, is the same even when the order of the players is reversed:

$$\max_{\lambda \in \Lambda} \min_{\mu \in \Delta(\Pi)} \lambda \cdot \bar{z}(\mu) \tag{4}$$

In our case, we can use a no-regret algorithm for the λ-player, and best response for the μ-player. At time t = 1,...,T, the learner makes a decision $\lambda_t \in \Lambda$ the environment reveals a convex loss function $\ell_t : \Lambda \to R$ and the learner incurs loss $\ell_t(\lambda_t)$. The learner seeks to achieve small regret, the gap between its loss and the best in hindsight:

$$Regret_T = \left[\sum_{t=1}^{T} \ell_t(\lambda_t)\right] - \min_{\lambda \in \Lambda} \left[\sum_{t=1}^{T} \ell_t(\lambda)\right] \tag{5}$$

2.3 Algorithm and main result

We assume that our target set $\mathcal{C}$ is a convex cone closed under summation and also multiplication by non-negative scalars. With this assumption, we can apply the following lemma, in which distance to a convex cone $\mathcal{C} \subset R^d$ is written as a maximization over a dual convex cone $\mathcal{C}^o$ called the polar cone:

$$\mathcal{C}^o = \{\lambda : \lambda \cdot x \leq 0 \; for \; all \; x \in \mathcal{C}\} \tag{6}$$

Lemma 1 *For a convex cone $\mathcal{C} \subseteq \mathbb{R}^d$ and any point $x \in \mathbb{R}^d$*

$$dist(x, \mathcal{C}) = \max_{\lambda \in \mathcal{C}^o \cap \mathcal{B}} \lambda \cdot x, \tag{7}$$

where $\mathcal{B} = x : ||x|| \leq 1$ is the Euclidean ball of radius 1 at the origin.

Approachability-Based Policy Optimization (APPROPO)

Algorithm 1: APPROPO

input projection oracle T, for target set $\mathcal{C}$ which is a convex cone,
best-response oracle $\mathrm{BESTRESPONSE}(\cdot)$, estimation oracle $\mathrm{EST}(\cdot)$,
step size η , number of iterations T
define $\Lambda = \mathcal{C}^o \cap \mathcal{B}$, and its projection operator $\Gamma_\Lambda = (x - \Gamma_\mathcal{C}(x))/max\{1, ||x - \Gamma_\mathcal{C}(x)||\}$
initialize λ_1 arbitrarily in Λ
For $t = 1$ **to** T **do**
Compute an approximately optimal policy for standard RL with scalar reward $r = -\lambda_t \cdot z$:
$\pi_t \leftarrow \mathrm{BESTRESPONSE}(\lambda_t)$
Call the estimation oracle to approximate long-term measurement for π_t :
$\hat{z}_t \leftarrow \mathrm{EST}(\pi_t)$
Update λ_t using online gradient descent with the loss function $\ell_t(\lambda) = -\lambda \cdot \hat{z}_t$:
$\lambda_{t+1} \leftarrow \Gamma_\Lambda(\lambda_t + \eta \hat{z}_t)$
end for
return $\bar{\mu}$, a uniform mixture over $\pi_1, ..., \pi_T$

Best-response oracle: $\mathrm{BESTRESPONSE}(\lambda)$.
Given $\lambda \in \mathbb{R}^d$, return a policy $\pi \in \Pi$ that satisfies $R(\pi) \geq max_{\pi' \in \Pi} R(\pi') - \epsilon_0$, where $R(\pi)$ is the long-term reward of policy π with scalar reward defined as $r = -\lambda \cdot z$

Estimation oracle: $\mathrm{EST}(\pi)$.
Given policy π, return $\hat{z}$ satisfying $||\hat{z} - \bar{z}(\pi)|| \leq \epsilon_1$

Projection oracle: $\Gamma_\mathcal{C}(x) = argmin_{x' \in \mathcal{C}} \; ||x - x'||$.

Positive-response oracle: $\mathrm{POSRESPONSE}(\lambda)$.
Given $\lambda \in \mathbb{R}^d$, return a policy $\pi \in \Pi$ that satisfies $R(\pi) \geq -\epsilon_0$ if $max_{\pi' \in \Pi} R(\pi') \geq 0$, where $R(\pi)$ is the long-term reward of policy π with scalar reward defined as $r = -\lambda \cdot z$

For a λ-player, an effective no-regret learner, we use Best-response oracle to generate a policy π. We analyze using online gradient descent (OGD) of the loss function $\ell_t(\lambda) = -\lambda \cdot \bar{z}(\pi_t)$, where $\bar{z}(\pi_t)$ is the averaging observed measurements. We can use Estimation oracle to estimate $\bar{z}(\pi_t)$. Then we require projection to the set $\Lambda = \mathcal{C}^o \cap \mathcal{B}$. Consider an arbitrary x, and denote its projection onto polar cone $\mathcal{C}^o$ as $\Gamma_{\mathcal{C}^o}(x)$. First, we use Projection oracle to estimate $\Gamma_{\mathcal{C}}(x)$, and $\Gamma_{\mathcal{C}^o}(x) = x - \Gamma_{\mathcal{C}}(x)$. Given the projection $\Gamma_{\mathcal{C}^o}(x)$ and further projecting onto $\mathcal{B}$, we obtain $\Gamma_\Lambda(x) = (x - \Gamma_{\mathcal{C}}(x))/max1, ||x - \Gamma_{\mathcal{C}}(x)||$, because Dykstra's projection algorithm converges to this point after two steps. After estimating $\Gamma_\Lambda(x)$, we can update parameters $\lambda_{t+1} \leftarrow \Gamma_\Lambda(\lambda_t + \eta\hat{z}_t)$.

When the problem is feasible, it can be shown that there must exist $\pi \in \Pi$ with $R(\pi^{'}) \geq 0$, and furthermore, that $\ell_t(\lambda_t) \geq -(\epsilon_0 + \epsilon_1)$ is means, if the goal is feasibility, we can modify algorithm by replacing $\mathrm{BESTRESPONSE}(\lambda)$ with $\mathrm{POSRESPONSE}(\lambda)$, and adding a test at the end of each iteration to report infeasibility if $\ell_t(\lambda_t) < -(\epsilon_0 + \epsilon_1)$

2.4 Removing the cone assumption

Our results so far have assumed the target set $\mathcal{C}$ is a convex cone. If instead $\mathcal{C}$ is an arbitrary convex, compact set, we can apply our algorithm to a specific convex cone $\tilde{\mathcal{C}}$ constructed from $\mathcal{C}$ to obtain a solution with provable guarantees. Given $\mathcal{C}$, we have

$$\tilde{\mathcal{C}} = cone(\mathcal{C} \times \{\kappa\}), \quad \text{where } cone(\chi) = \{\alpha x | x \in \chi, \alpha \geq 0\} \tag{8}$$

Given our original vector-valued MDP $M = (S, \mathcal{A}, \beta, P_s, P_z)$, we define a new MDP $M' = (S, \mathcal{A}, \beta, P_s, P'_{z'})$ with (d + 1)-dimensional measurement $z' \in \mathbb{R}^{d+1}$, defined by

$$z'_i = z_i \oplus \langle (1-\gamma)\kappa \rangle, \qquad z_i \sim Pz(\cdot|s_i, a_i) \tag{9}$$

where $\oplus$ denotes vector concatenation. The main idea is to apply the algorithms described above to the modified MDP M' using the cone $\tilde{\mathcal{C}}$ as target set. For an appropriate choice of $\kappa > 0$, we show that the resulting mixed policy will approximately minimize distance to $\mathcal{C}$ for the original MDP M.

Lemma 2 *Consider a compact, convex set $\mathcal{C}$ in $\mathbb{R}^d$ and $x \in \mathbb{R}^d$. For any $\delta > 0$, let $\tilde{\mathcal{C}} = cone(\mathcal{C} \times \{\kappa\})$, where $\kappa = \frac{max_{x \in \mathcal{C}} ||x||}{\sqrt{2\delta}}$. Then $dist(x, \mathcal{C}) \leq (1 + \delta) dis(x \oplus \langle \kappa \rangle, \tilde{\mathcal{C}})$*

Assume that $\mathcal{C}$ is a convex, compact set and for all measurements we have $||z|| \leq B$. Then by putting $\eta = (\frac{B+\kappa}{1-\gamma} + \epsilon_1)^{-1} T^{-1/2})$ and running Algorithm for T rounds with M' as the MDP and $\tilde{\mathcal{C}}$ as the target set, the mixed policy $\bar{\mu}$ returned by the algorithm satisfies

$$\begin{aligned} dist(\bar{z}(\bar{\mu})), \mathcal{C}) &\leq (1+\delta) \left(\min_{\mu \in \Delta(\Pi)} dist(\bar{z}(\mu), \mathcal{C}) + (\frac{B+\kappa}{1-\gamma} + \epsilon_1) T^{-1/2} + \epsilon_0 + 2\epsilon_1 \right) & (10) \\ &\leq (1+\delta) \left((\frac{B+\kappa}{1-\gamma} + \epsilon_1) T^{-1/2} + \epsilon_0 + 2\epsilon_1) \right) & (11) \end{aligned}$$

where $\kappa = \frac{max_{x \in \mathcal{C}} ||x||}{\sqrt{2\delta}}$ for an arbitrary $\delta > 0$.

3 Detailed Implementation

Please explain your implementation in detail. You may do this with the help of pseudo code or a figure of system architecture. Please also highlight which parts of the algorithm lead to the most

difficulty in your implementation.

Algorithm 1: APPROPO

input projection oracle T, for target set $\mathcal{C}$ which is a convex cone,
best-response oracle $\mathrm{BESTRESPONSE}(\cdot)$, estimation oracle $\mathrm{EST}(\cdot)$,
step size η , number of iterations T
define $\Lambda = \mathcal{C}^o \cap \mathcal{B}$, and its projection operator $\Gamma_\Lambda = (x - \Gamma_\mathcal{C}(x))/max\{1, ||x - \Gamma_\mathcal{C}(x)||\}$
initialize λ_1 arbitrarily in Λ
For $t = 1$ **to** T **do**
  Compute an approximately optimal policy for standard RL with scalar reward $r = -\lambda_t \cdot z$:
    $\pi_t \leftarrow \mathrm{BESTRESPONSE}(\lambda_t)$
  Call the estimation oracle to approximate long-term measurement for π_t :
    $\hat{z_t} \leftarrow \mathrm{EST}(\pi_t)$
  Update λ_t using online gradient descent with the loss function $\ell_t(\lambda) = -\lambda \cdot \hat{z_t}$:
    $\lambda_{t+1} \leftarrow \Gamma_\Lambda(\lambda_t + \eta \hat{z_t})$
end for
return $\bar{\mu}$, a uniform mixture over $\pi_1, ..., \pi_T$

3.1 System architecture

we divided the following algorithm into 2 main part:

1. RL oracle

- Compute an approximately optimal policy π_t for standard RL with scalar reward
- Approximate long-term measurement for π_t

We implement a typical actor-critic network for learning reinforcement learning problem. Actor-critic methods are the natural extension of the idea of reinforcement comparison methods to TD learning. Typically, the critic is a state-value function. After each action selection, the critic evaluates the new state to determine whether things have gone better or worse than expected. Hence, we could compute an approximately optimal policy π_t with the network. After T episodes, we approximate long-term measurement(expected return) for π_t

2. Projection oracle

- Given expected return, we calculate loss
- Update λ by Online gradient descent(OGD)
- Do projection to feasible decision set

It requires projection to the set $\Lambda = \mathcal{C}^\circ \cap \mathcal{B}$. In fact, We simply project onto the target set $\mathcal{C}$, which is more natural, then it is possible to also project onto Λ. Consider an arbitrary x and denote its projection onto $\mathcal{C}$ as $\Gamma_\mathcal{C}(x)$. Then the projection of x onto the polar cone $\mathcal{C}^\circ$ is $\Gamma_{\mathcal{C}^\circ}(x) = x - \Gamma_\mathcal{C}(x)$. Given the projection $\Gamma_{\mathcal{C}^\circ}(x)$ and further projecting onto $\mathcal{B}$, we obtain $\Gamma_\Lambda(x) = x - \Gamma_\mathcal{C}(x)/\max\{1, \|x - \Gamma_\mathcal{C}(x)\|\}$ Therefore, it suffices to require access to a *projection oracle* for $\mathcal{C}$:

3.2 Difficulty

Due to the fact that we have implemented actor-critic network before, **RL oracle** we just need to modify the reward function and approximate expected return. However, the other part, **Projection oracle** is hard for me to understand, so I spent much time to realize the before and after of all equations. The goal of this part is to project the obtained λ from online gradient descent to the feasible set $\mathcal{C}$. We define the projection operator $\Gamma_\Lambda = (x - \Gamma_\mathcal{C}(x))/max\{1, ||x - \Gamma_\mathcal{C}(x)||\}$. Then, we could get the updated term: $\lambda_{t+1} \leftarrow \Gamma_\Lambda(\lambda_t + \eta \hat{z_t})$.

4 Empirical Evaluation

Please showcase your empirical results in this section. Please clearly specify which sets of experiments of the original paper are considered in your report. Please also report the corresponding hyperparameters of each experiment.

We compare APPROPO with the RCPO approach of Tessler et al. (2019), which adapts policy gradient, specifically, asynchronous actor-critic (A2C) (Mnih et al., 2016), to find a fixed point of the Lagrangian of the constrained policy optimization problem. Same as the paper, we ran our experiments on a small version of the Mars rover grid-world environment on both algorithms(RCPO and APPROPO). In the original paper, the three constraint are shown below:

- Probability of hitting a rock is at most a fixed threshold(0.2).
- An additional constraint requiring that the reward be at least 0.17.
- The uniform distribution over the upper-right triangle cells of the grid (excluding rocks) be at most 0.12.

However, in these constraints, Our implementation could not learn the policy in 3000 episode. So we adjust these constraints to make this game more easier. Our result base on these constraint:

- Probability of hitting a rock is at most a fixed threshold(0.5).
- An additional constraint requiring that the reward be at least 2.0.
- The uniform distribution over the upper-right triangle cells of the grid (excluding rocks) be at most 0.22.

This is our result. Base on the settings above, we compared the APPROPO with or without diversity version. The blue line is no diversity version, and the orange line is the diversity version. X axis represents the number of episodes. Showing average and standard deviation over 10 runs.

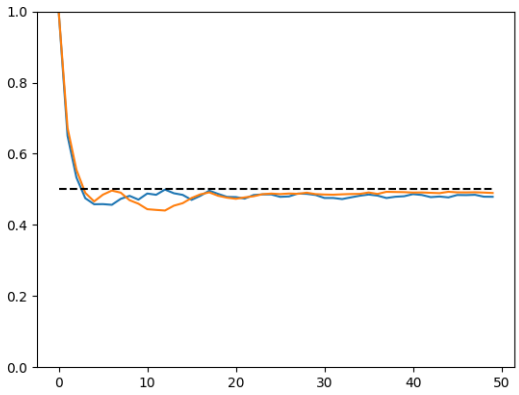

Figure 1 (Probability of failure)

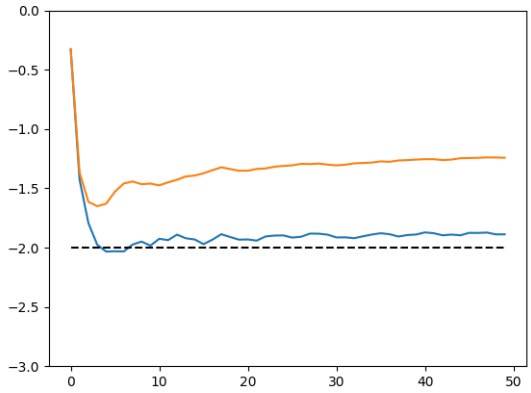

Figure 2 (Reward)

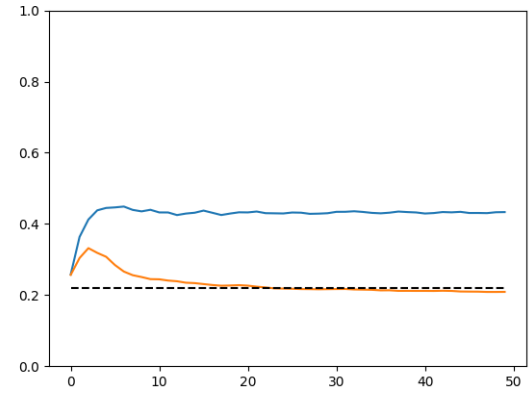

Figure 3 (Distance distribution)

In figure 1, which shows the result with constraint of probability of failure. We can find that both of them reach the setting goal because the constraint is an orthant constraint. This result can be find in Figure 2 with reward constraint. The main result is in Figure 3 We can find that APPROPO with diversity can reach the constrain of the third constrains, which is a convex constraint.

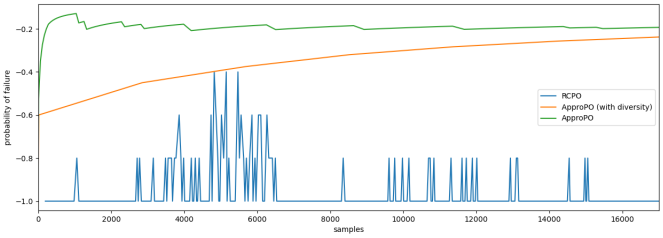

Figure 4

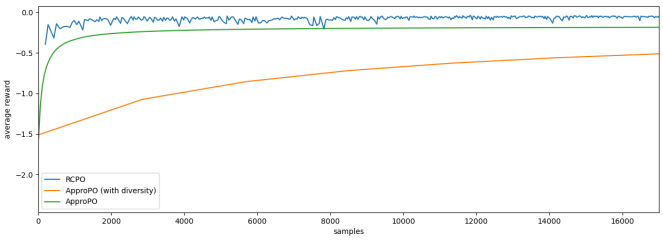

Figure 5

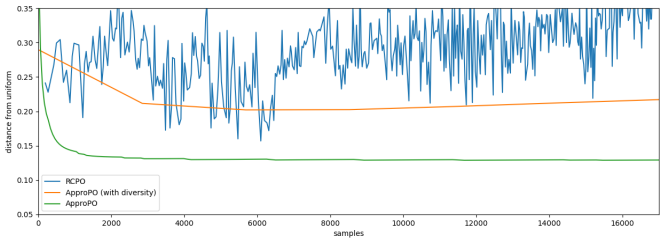

Figure 6

In figure 4, 5 and 6, we compare the result including RCPO. In these Figures, we can find that APPROPO achieves similar performance as RCPO while being able to satisfy additional types of constraints.

5 Conclusion

In this report, we discuss about the new algorithm, APPROPO, which can solve RL problem with convex constraint. This algorithm is a great solution for solving the approachability problem. But we still need to set up the constraints more carefully, or we would not get the expected result. In our experiment, we suffer from the difficulty because of the strict constraints. We think a proper measurements vector may be the key to optimize the algorithm since it determined the weight of the expected result and different constraints, and influenced the neural network parameters directly.To find an algorithm can afford more constraints or more strict constraint types with the same or more performance will be the future research directions.

References

Miryoosef et al., Reinforcement Learning with Convex Constraints, NIPS 2019