Reinforcement Learning with Convex Constraints

Team members : 吳柏憲
呂明哲
江品萱

Outline

1. Problem Overview
2. Background
3. The Algorithm
4. Detail Implementation
5. Empirical Evaluation
6. Conclusion

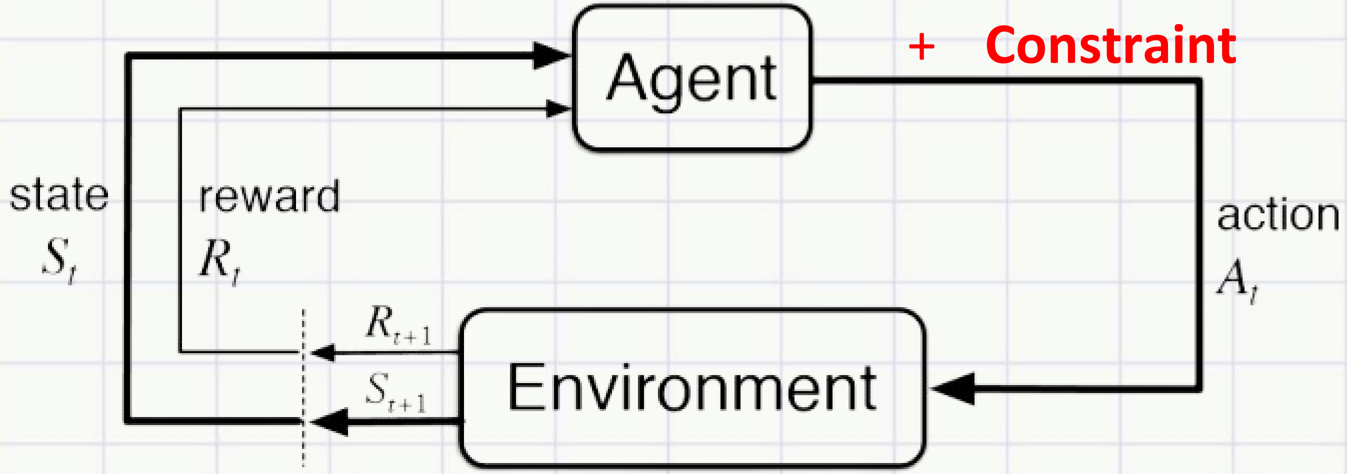
PART 01

1

Problem Overview

Problem Overview

In standard reinforcement learning, a learning agent seeks to optimize the overall reward.
However, many **key aspects of a desired behavior** are more naturally expressed as **constraints**.



Problem Overview

Moreover, a scalar reward might not be a natural formalism for stating certain learning objectives, such as **safety desires** or **exploration suggestions**.

We derive an algorithm, **approachability-based policy optimization ApproPO**, for solving such problems in terms of a **vector of measurements**, instead of scalars.

PART 02

2 Background

Background

Research Background

Given a **Markov decision process with vector-valued measurements**, and a **target constraint set**, we want our algorithm to learn a stochastic policy whose **expected measurements fall in that target set**

Setup and preliminaries

With a **standard MDP** (Markov Decision Process) setup, our aim is to **control the MDP so that measurements satisfy some constraints.**

For any policy ***π***, we define the long-term measurement ***z(π)*** as the expected sum of discounted measurements

$$\overline{\mathbf{Z}}(\pi) \triangleq \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \mathbf{z}_i \mid \pi\right]$$

Where $\mathbf{z}_i \sim P_z(\cdot \mid s_i, a_i)$ in MDP and discount factor $\gamma \in [0, 1)$

Setup and preliminaries

Consider **mixed policies *μ***, which are **distributions over finitely many stationary policies.**
The **space of all such mixed policies over** $\prod$ is denoted $\Delta(\Pi)$

The **long-term measurement of a mixed policy *z(μ)*** is defined accordingly:

$$\overline{\mathbf{z}}(\mu) \triangleq \mathbb{E}_{\pi\sim\mu}[\overline{\mathbf{z}}(\pi)] = \sum_{\pi} \mu(\pi)\overline{\mathbf{z}}(\pi)$$

The feasibility problem

Our learning problem, i.e. the feasibility problem, is specified by a convex target set *C*.

The goal is to find a mixed policy *μ* whose long-term measurements lie in the set *C*

Feasibility problem : Find $\mu \in \Delta(\Pi)$ such that $\overline{\mathbf{z}}(\mu) \in \mathcal{C}$

PART 03

3 The Algorithm

Analysis

The feasibility problem can be changed into a stronger problem, which is the **minimization** of the **distance** to the target convex set:

$$\min_{\mu \in \Delta(\Pi)} \quad \mathrm{dist}(\bar{\mathbf{z}}(\mu), \mathcal{C})$$

Where dist() is the distance between a point and a set. It will be shown later that dist() can be rewritten into:

$$\max_{\lambda \in \Lambda} \lambda \cdot \bar{z}(\mu)$$

For some convex set $\boldsymbol{\Lambda}$

Analysis

Then, the distance minimization can rewritten into:

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

This can be interpreted as a **zero-sum game**, where two players play λ and μ against each other,and by the Minimax theorem:

$$\min_{\mu \in \Delta(\pi)} \max_{\lambda \in \Lambda} \lambda \cdot \bar{z}(\mu) = \max_{\lambda \in \Lambda} \min_{\mu \in \Delta(\pi)} \lambda \cdot \bar{z}(\mu)$$

Solving zero-sum games

Zero-sum games can be solved by playing a **no-regret online learning** algorithm against a **best response** oracle.

Algorithm 1 Solving a game with repeated play

input concave-convex function $g : \Lambda \times \mathcal{U} \to \mathbb{R}$, online learning algorithm LEARNER
for $t = 1$ **to** T **do**
    LEARNER makes a decision $\boldsymbol{\lambda}_t \in \Lambda$
    $\mathbf{u}_t \leftarrow \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}} g(\boldsymbol{\lambda}_t, \mathbf{u})$
    LEARNER observes loss function $\ell_t(\boldsymbol{\lambda}) = -g(\boldsymbol{\lambda}, \mathbf{u}_t)$
end for
return $\overline{\boldsymbol{\lambda}} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{\lambda}_t$ and $\overline{\mathbf{u}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{u}_t$

Solving zero-sum games

Algorithm 1 Solving a game with repeated play

input concave-convex function $g : \Lambda \times \mathcal{U} \to \mathbb{R}$, online learning algorithm LEARNER
for $t = 1$ **to** T **do**
  LEARNER makes a decision $\boldsymbol{\lambda}_t \in \Lambda$
  $\mathbf{u}_t \leftarrow \mathrm{argmin}_{\mathbf{u} \in \mathcal{U}}\, g(\boldsymbol{\lambda}_t, \mathbf{u})$
  LEARNER observes loss function $\ell_t(\boldsymbol{\lambda}) = -g(\boldsymbol{\lambda}, \mathbf{u}_t)$
end for
return $\overline{\boldsymbol{\lambda}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\lambda}_t$ and $\overline{\mathbf{u}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{u}_t$

g() is the **payout** from the u-player to the λ-player

Main result

For now, we assume that the target set is a **convex cone**.

$$\mathcal{C}^\circ \triangleq \{\lambda : \lambda \cdot x \leq 0, \forall x \in \mathcal{C}\}$$

The distance between a point to a convex cone is:

$$\operatorname{dist}(x, \mathcal{C}) = \max_{\lambda \in \mathcal{C}^\circ \cap B} \lambda \cdot x$$

The aformentioned **distance minimization** can indeed be turned a **zero-sum game**, which can be solved with algorithm 1.

Best response oracle

Given λ, the best response oracle aims to minimize $\lambda \cdot \bar{\mathbf{z}}(\mu)$

Since $\bar{\mathbf{z}}(\mu)$ is a linear mixture of $\bar{\mathbf{z}}(\pi)$, it suffices to minimize $\lambda \cdot \bar{\mathbf{z}}(\pi)$.

For the Best response oracle, we can just use standard RL algorithm to maximize $r_i = \lambda \cdot \mathbf{z}_i$ for vector rewards $\mathbf{z}_i$ since:

$$R(\pi) \triangleq E[\sum_{i=0}^{\infty} \gamma^i r_i | \pi] = -\lambda \cdot E[\sum_{i=0}^{\infty} \gamma^i z_i | \pi] = -\lambda \cdot \bar{z}(\pi)$$

ApproPO

Then, we estimate the expected total vector reward of π, $\bar{\mathbf{z}}(\pi)$, which can be done by sampling trajectories using π.
Using this estimation of $\bar{\mathbf{z}}(\pi)$, the λ-player updates the choice of λ using online gradient descent.
finally, the ApproPO returns the uniform mixture of all of the selected π's

ApproPO

Algorithm 2 APPROPO

---

input projection oracle $\Gamma_{\mathcal{C}}(\cdot)$ for target set $\mathcal{C}$ which is a convex cone,
best-response oracle $\textsc{BestResponse}(\cdot)$, estimation oracle $\textsc{Est}(\cdot)$,
step size η, number of iterations T

define $\Lambda \triangleq \mathcal{C}^\circ \cap \mathcal{B}$, and its projection operator $\Gamma_\Lambda(\mathbf{x}) \triangleq (\mathbf{x} - \Gamma_{\mathcal{C}}(\mathbf{x}))/\max\{1, \|\mathbf{x} - \Gamma_{\mathcal{C}}(\mathbf{x})\|\}$

initialize $\boldsymbol{\lambda}_1$ arbitrarily in Λ

for $t = 1$ **to** T **do**

  Compute an approximately optimal policy for standard RL with scalar reward $r = -\boldsymbol{\lambda}_t \cdot \mathbf{z}$:
  $\pi_t \leftarrow \textsc{BestResponse}(\boldsymbol{\lambda}_t)$

  Call the estimation oracle to approximate long-term measurement for π_t:
  $\hat{\mathbf{z}}_t \leftarrow \textsc{Est}(\pi_t)$

  Update $\boldsymbol{\lambda}_t$ using online gradient descent with the loss function $\ell_t(\boldsymbol{\lambda}) = -\boldsymbol{\lambda} \cdot \hat{\mathbf{z}}_t$:
  $\boldsymbol{\lambda}_{t+1} \leftarrow \Gamma_\Lambda\big(\boldsymbol{\lambda}_t + \eta\hat{\mathbf{z}}_t\big)$

end for

return $\bar{\mu}$, a uniform mixture over $\pi_1, \ldots, \pi_T$

Removing cone assumption

Previously, we assumed that the target set is a convex cone.
To apply this algorithm to any convex set, we can construct a cone using the convex set:

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

We also need to change the form of the vector result:

$$z_i^{'} = z_i \oplus \langle (1-\gamma)\kappa \rangle), z_i \sim P_z(\cdot | s_i, a_i)$$

For an appropriate choice of κ>0, the resulting mixed policy will approximately minimize distance to convex set for the original MDP.

PART 04

4

Detail Implementation

Environments

We implemented the replication with **(mainly) python and pytorch**

Also implemented a game solver for **grid world game** as same as the paper for result replication and evaluation

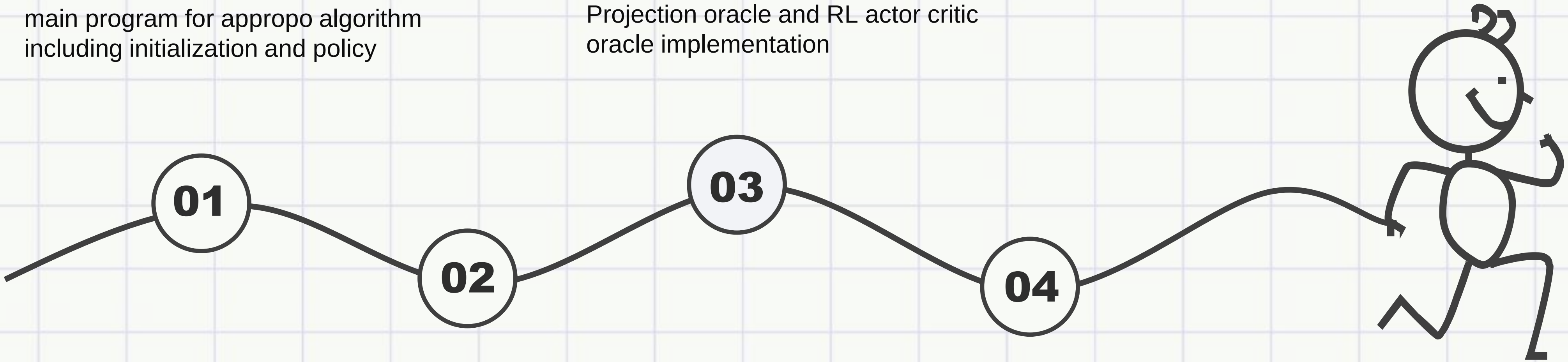
Implementations

appropo.py

main program for appropo algorithm including initialization and policy

oracle.py

Projection oracle and RL actor critic oracle implementation



rcpo.py

main program for rcpo algorithm including initialization and policy

game.py

Grid world game solver calling above 2 algorithms for results

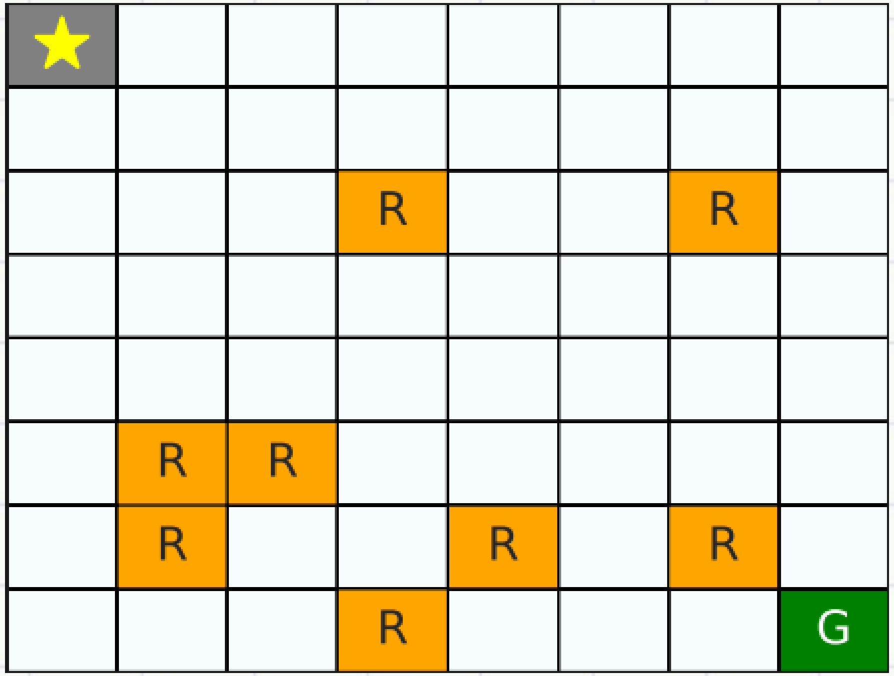
PART 05

5

Empirical Evaluation

Grid world game

The Grid world game is what we (and the paper) use for experiments.
The agent starts in top-left and needs to reach the goal in bottom-right while avoiding rocks.



Environments

For simplicity, we focus on the **feasibility version** and compare with **RCPO** approach of Tessler et al. (2019) **just as the paper's experiment.**

ApproPO uses A2C as a positive-response oracle, with the **same hyperparameters as used in RCPO**. Online learning in the outer loop of ApproPO was implemented via **online gradient descent with momentum.**

Experiment results

TBD

PART 06

6 Conclusion

Conclusion

Experimentally, we replicated and by experiment that **ApproPO can be applied with well-known RL algorithms for discrete domains, and achieves similar performance as RCPO** (Tessler et al., 2019), while being **able to satisfy additional types of constraints.**

In sum, this yields a theoretically justified, practical algorithm for solving the approachability problem in reinforcement learning

THANK YOU