Reinforcement Learning with Convex Constraints

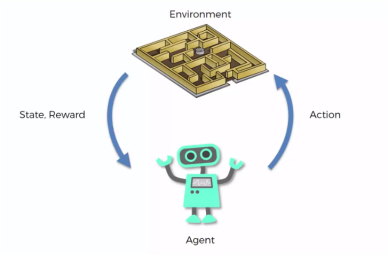
Sobhan Miryoosefi[1], **Kianté Brantley**[3], Hal Daumé III[2,3], Miro Dudík[2], Robert Schapire[2]

[1]Princeton University
[2]Microsoft Research
[3]University of Maryland

NeurIPS 2019

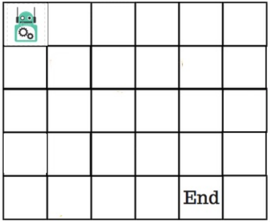
Reinforcement Learning (RL)

Agent interactively takes some action in the Environment and receive some **reward** for the action taken.



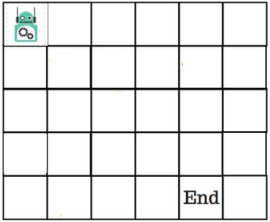
Agent's Goal: maximize long-term reward

Example



- **state**: position on the grid
- **actions**: $\{\leftarrow, \rightarrow, \uparrow, \downarrow\}$
- **desired behavior**: reaching the End

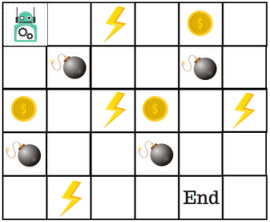
Example



- **state**: position on the grid
- **actions**: $\{\leftarrow, \rightarrow, \uparrow, \downarrow\}$
- **desired behavior**: reaching the End

Solving this task using **Standard RL**:
Let reward be $+1$ when agents reach the End and 0 otherwise
Let the agent maximize the reward

More Complex Desired Behavior



Desired Behavior

- Reach the End
- Don't step on the bombs
- Don't get electrocuted too much
- Collect many coins
- Finish fast

One Approach: Applying Standard RL

Desired Behavior

- reaching the End $\Rightarrow$ Big Positive Reward
- stepping on the bombs $\Rightarrow$ Big Negative Reward
- getting electrocuted $\Rightarrow$ Small Negative reward
- collecting coins $\Rightarrow$ Small Positive reward
- each time step $\Rightarrow$ Small Negative reward

One Approach: Applying Standard RL

Desired Behavior

- reaching the End $\Rightarrow$ Big Positive Reward
- stepping on the bombs $\Rightarrow$ Big Negative Reward
- getting electrocuted $\Rightarrow$ Small Negative reward
- collecting coins $\Rightarrow$ Small Positive reward
- each time step $\Rightarrow$ Small Negative reward

Let the agent learn to maximize the reward

One Approach: Applying Standard RL

Desired Behavior

- reaching the End $\Rightarrow$ Big Positive Reward
- stepping on the bombs $\Rightarrow$ Big Negative Reward
- getting electrocuted $\Rightarrow$ Small Negative reward
- collecting coins $\Rightarrow$ Small Positive reward
- each time step $\Rightarrow$ Small Negative reward

Let the agent learn to maximize the reward

Guarantee for satisfying our desired behavior?

This is difficult...

- Not a straightforward task
- Gets harder as desired behavior get more complex
- Agent might maximize the reward without satisfying our desired behavior
- Not possible (or at least clear) how to model some behaviors

Our Approach: Constraint-based RL

- Some behavior are easier to be expressed by constraints
- These constraints can be used to
  - enforce safety (e.g., not getting electrocuted)
  - mimic Expert's behavior (e.g., be close to an expert)
  - encourage diversity (e.g., visit more states)
  - ...

Example Constraint: Being Electrocuted

Let's model it as

- at time t we get electrocuted by electric current of $\mathrm{current}_t$ (it can be zero)
- threshold α
- constraint as $\mathbb{E}[\sum_{t=1}^{T} \mathrm{current}_t] \leq \alpha$

Setting

Standard RL setting:

For $t = 1, 2, \ldots, T$:

- arrive at state $s_t \in \mathcal{S}$
- take an action $a_t \sim \pi(s_t) \in \mathcal{A}$ $\quad \boxed{\text{policy } \pi : \mathcal{S} \to \Delta(\mathcal{A})}$
- receive reward $r_t \in \mathbb{R}$

Goal: find π that maximizes $R(\pi) = \mathbb{E}[\sum_{t=1}^{T} r_t]$
(expectation over randomness in both policy and environment)

Setting

Our Setting:

For $t = 1, 2, \ldots, T$:

- arrive at state $s_t \in \mathcal{S}$
- take an action $a_t \sim \pi(s_t)$
- ~~receive reward $r_t \in \mathbb{R}$~~
- receive measurement $\boldsymbol{z}_t \in \mathbb{R}^d$

$\boxed{\text{policy } \pi : \mathcal{S} \to \Delta(\mathcal{A})}$

~~**Goal**: find π that maximizes $R(\pi) = \mathbb{E}[\sum_{t=1}^{T} r_t]$~~

Goal: find π such that $Z(\pi) = \mathbb{E}[\sum_{t=1}^{T} \boldsymbol{z}_t] \in \mathcal{C}$ (target set)

General Constraints

we model our desired behavior as a target set $\mathcal{C}$ and we want the agent to behave in a way that long term measurement $Z(\pi)$ lie in the set.

Desired Behavior as $\mathcal{C}$

Examples:

- **Safety**: not getting electrocuted while collecting enough gold
  - $\boldsymbol{z}_t = (\mathrm{current}_t, \mathrm{gold}_t)$
  - $\mathcal{C} = \{\boldsymbol{z} = (z_1, z_2) \in \mathbb{R}^2 \mid z_1 \leq \alpha_1, z_2 \geq \alpha_2\}$
- **Diversity**: exploring the state space
  - $\boldsymbol{z}_t = (\boldsymbol{z}_t^1, \ldots, \boldsymbol{z}_t^{|\mathcal{S}|})$ where $\boldsymbol{z}_t^i = \mathbb{1}\{s_t = i\}$
  - $\mathcal{C}$ $\{$ z $|$ entropy of $\frac{\boldsymbol{z}}{T}$ is high $\}$
  - $= \{\boldsymbol{z} \in \mathbb{R}^{|S|} \mid \mathrm{H}(\frac{\boldsymbol{z}}{T}) \geq \alpha\}$

This talk

- Present an algorithm: solve this RL task with general convex constraint
- Make connection to online learning and game theory
- Guarantee on performance of the algorithm

Related Work

Constrained MDP (CMDP)

- maximizing reward subject to orthant constraints

$$\text{Find } \pi \text{ that maximizes } R(\pi) \text{ s.t. } Z(\pi) \in C$$

$$\mathcal{C} = \{\boldsymbol{z} = (z_1, z_2, \ldots, z_d) \mid z_i \leq \alpha_i \text{ for all } i\}$$

Related Work

Constrained MDP (CMDP)

- maximizing reward subject to orthant constraints
- Introduced by [?]
  - Lagrangian methods and solving the dual LP
  - Full knowledge of MDP

- Constrained Policy Optimization (CPO) [?]
  - safety constraints
  - guarantees for near-constraint satisfaction at each iteration

- Reward Constrained Policy Optimization (RCPO) [?]
  - asymptotic analysis for convergence

- Batch Policy Learning Under Constraints [?]
  - Iteration complexity
  - Generalization bounds

Other Related Work

- Provably Efficient Maximum Entropy Exploration [?]
  - maximize concave function over state distribution
- A game-theoretic approach to apprenticeship learning [?]
  - true reward as linear combination of features
  - mimic Expert's behavior
- Bandits with concave rewards and convex knapsacks [?]
  - maximize concave function over average measurement vector subject to average measurement vector lies in a convex set

Other Related Work

- Blackwell approachability and no-regret learning are equivalent [?]
  - show equivalence between no-regret learning and repeated game playing with vector payoff
  - some ideas and techniques used in this work has been inspired by this paper

Our Contribution

- Able to deal with general constraints
- Aim for more than one of these criteria. e.g., encourage diversity while satisfying some safety constraints.
- Theoretical guarantee

Recall Our Setting

For $t = 1, 2, \ldots, T$:

- arrive at state $s_t \in \mathcal{S}$
- take an action $a_t \sim \pi(s_t)$ $\boxed{\text{policy } \pi : \mathcal{S} \to \Delta(\mathcal{A})}$
- receive measurement $\boldsymbol{z}_t \in \mathbb{R}^d$

Goal: find π such that $Z(\pi) = \mathbb{E}[\sum_{t=1}^{T} \boldsymbol{z}_t] \in \mathcal{C}$ (target set)

Environment is an MDP

- Let $\beta \in \Delta(\mathcal{S})$ the initial distribution.
- **Markov Assumption**: next state and measurements according to some distribution which only depends on current state and action.
  - initial state $s_0 \sim \beta$
  - $s_{t+1} \sim \mathrm{P}_s(\cdot \mid s_t, a_t)$
  - actions $\boldsymbol{z}_t \sim \mathrm{P}_z(\cdot \mid s_t, a_t)$

Problem Statement

Problem (Feasibility)

$$\begin{aligned} \textit{find} \quad & \pi \in \Pi \\ \textit{s.t} \quad & Z(\pi) \in \mathcal{C} \end{aligned}$$

Π *is set of all* $\pi : \mathcal{S} \to \Delta(\mathcal{A})$*(set of stationary policies)*

It's actually a game. . .

$$\begin{aligned} \text{find} \quad & \pi \in \Pi \\ \text{s.t} \quad & Z(\pi) \in \mathcal{C} \end{aligned}$$

It's actually a game...

$$\begin{aligned} \text{find} \quad & \pi \in \Pi \\ \text{s.t} \quad & Z(\pi) \in \mathcal{C} \end{aligned}$$

Let's solve a stronger problem

$$\min_{\pi \in \Pi} \mathrm{dist}(Z(\pi), \mathcal{C})$$

It's actually a game...

$$\begin{aligned} \text{find} \quad & \pi \in \Pi \\ \text{s.t} \quad & Z(\pi) \in \mathcal{C} \end{aligned}$$

Let's solve a stronger problem

$$\min_{\pi \in \Pi} \operatorname{dist}(Z(\pi), \mathcal{C})$$

How to convert it into a game?

It's actually a game...

$$\begin{aligned} \text{find} \quad & \pi \in \Pi \\ \text{s.t} \quad & Z(\pi) \in \mathcal{C} \end{aligned}$$

Let's solve a stronger problem

$$\min_{\pi \in \Pi} \mathrm{dist}(Z(\pi), \mathcal{C})$$

How to convert it into a game?

Let's assume for now that we are able to write

$$\boxed{\mathrm{dist}(Z(\pi), \mathcal{C}) = \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\pi) \rangle}$$

for some convex set $\mathcal{K}$

It's actually a game...

We started with

$$\begin{aligned} \text{find} \quad & \pi \in \Pi \\ \text{s.t} \quad & Z(\pi) \in \mathcal{C} \end{aligned}$$

converted it into

$$\min_{\pi \in \Pi} \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

We have a game...

$$\min_{\pi \in \Pi} \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

We have a game...

$$\min_{\pi \in \Pi} \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

$$\text{payoff } g(\pi, \boldsymbol{\theta}) = \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

Min Player: (Plays First)

- pick some $\pi \in \Pi$
- wants to minimize $\max_{\boldsymbol{\theta} \in \mathcal{K}} g(\pi, \boldsymbol{\theta})$

Max Player: (Plays Second)

- observe π
- pick some $\boldsymbol{\theta} \in \mathcal{K}$
- wants to maximize $g(\pi, \boldsymbol{\theta})$

We have a game...

$$\min_{\pi \in \Pi} \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

$$\text{payoff } g(\pi, \boldsymbol{\theta}) = \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

Min Player: (Plays First)

- pick some $\pi \in \Pi$
- wants to minimize $\max_{\boldsymbol{\theta} \in \mathcal{K}} g(\pi, \boldsymbol{\theta})$

Max Player: (Plays Second)

- observe π
- pick some $\boldsymbol{\theta} \in \mathcal{K}$
- wants to maximize $g(\pi, \boldsymbol{\theta})$

Can we change the order of the play in this game?

Minimax Theorem

celebrated *minimax theorem* discovered by John von Neumann in 1920s.

Theorem

Assume:

- $\mathcal{X}, \mathcal{Y}$ *compact and convex*
- $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ *convex in first and concave in the second argument*

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} g(x, y) = \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} g(x, y)$$

Back to our game. . .

$$\min_{\pi \in \Pi} \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

Back to our game...

$$\min_{\pi \in \Pi} \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

Assume that conditions of minimax are satisfied
(needs small tweak)

$$\max_{\boldsymbol{\theta} \in \mathcal{K}} \min_{\pi \in \Pi} \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

Back to our game...

$$\min_{\pi \in \Pi} \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

Assume that conditions of minimax are satisfied
(needs small tweak)

$$\max_{\boldsymbol{\theta} \in \mathcal{K}} \min_{\pi \in \Pi} \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

now we can solve this game, because...

First

$$\max_{\boldsymbol{\theta}\in\mathcal{K}} \boxed{\min_{\pi\in\Pi}\langle\boldsymbol{\theta}, Z(\pi)\rangle}$$

Given $\boldsymbol{\theta}$, $\min_{\pi\in\Pi}\langle\boldsymbol{\theta}, Z(\pi)\rangle$ is equivalent to solving the standard RL setting with scalar reward of $r_t = -\langle\boldsymbol{\theta}, \boldsymbol{z_t}\rangle$

First

$$\max_{\boldsymbol{\theta}\in\mathcal{K}} \boxed{\min_{\pi\in\Pi}\langle\boldsymbol{\theta}, Z(\pi)\rangle}$$

Given $\boldsymbol{\theta}$, $\min_{\pi\in\Pi}\langle\boldsymbol{\theta}, Z(\pi)\rangle$ is equivalent to solving the standard RL setting with scalar reward of $r_t = -\langle\boldsymbol{\theta}, \boldsymbol{z_t}\rangle$

$$\begin{aligned}\langle\boldsymbol{\theta}, Z(\pi)\rangle &= \langle\boldsymbol{\theta}, \mathbb{E}[\sum_{t=1}^{T}\boldsymbol{z}_t]\rangle \\ &= \mathbb{E}[\sum_{t=1}^{T}\langle\boldsymbol{\theta}, \boldsymbol{z}_t\rangle] \\ &= -R(\pi)\end{aligned}$$

First

$$\max_{\boldsymbol{\theta} \in \mathcal{K}} \boxed{\min_{\pi \in \Pi} \langle \boldsymbol{\theta}, Z(\pi) \rangle}$$

Given $\boldsymbol{\theta}$, $\min_{\pi \in \Pi} \langle \boldsymbol{\theta}, Z(\pi) \rangle$ is equivalent to solving the standard RL setting with scalar reward of $r_t = -\langle \boldsymbol{\theta}, \boldsymbol{z_t} \rangle$

$$\begin{aligned} \langle \boldsymbol{\theta}, Z(\pi) \rangle &= \langle \boldsymbol{\theta}, \mathbb{E}[\sum_{t=1}^{T} \boldsymbol{z}_t] \rangle \\ &= \mathbb{E}[\sum_{t=1}^{T} \langle \boldsymbol{\theta}, \boldsymbol{z}_t \rangle] \\ &= -R(\pi) \end{aligned}$$

$$\boxed{\mathrm{argmin}_\pi \langle \boldsymbol{\theta}, Z(\pi) \rangle = \mathrm{argmax}_\pi R(\pi)}$$

Second

- Folklore result attributed to [?]

  $\boxed{\text{we can find optimal strategies for players of a game by pitting two online learning strategies against each other}}$

Second

- Folklore result attributed to [?]

  $\boxed{\text{we can find optimal strategies for players of a game by pitting two online learning strategies against each other}}$

- as a special case: a no-reget algorithm vs best response(✓)

Second

- Folklore result attributed to [?]

  $\boxed{\text{we can find optimal strategies for players of a game by pitting two online learning strategies against each other}}$

- as a special case: a no-reget algorithm vs best response(✓)

Before going further into details
Let's fill the gaps in our approach

Back to the game

Satisfying conditions minimax theorem:

$$\min_{\pi \in \Pi} \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\pi) \rangle$$

Mixed Policy

- a mixed policy μ is a distribution over countable number of policies.
- $\Pi_{\mathrm{mix}} = \{\mu : \Pi \to [0,1] \mid \sum_{\pi \in \Pi} \mu(\pi) = 1\}$
- $Z(\mu) = \mathbb{E}_{\pi \sim \mu}[Z(\pi)] \quad R(\mu) = \mathbb{E}_{\pi \sim \mu}[R(\pi)]$

Substituting Mixed Policy

Satisfying conditions minimax theorem:

$$\min_{\mu \in \Pi_{\text{mix}}} \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\mu) \rangle$$

- $\mathcal{K}$ is convex and $\langle \boldsymbol{\theta}, Z(\mu) \rangle$ is affine in $\boldsymbol{\theta}$ ✓

Substituting Mixed Policy

Satisfying conditions minimax theorem:

$$\min_{\mu \in \Pi_{\text{mix}}} \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\mu) \rangle$$

- $\mathcal{K}$ is convex and $\langle \boldsymbol{\theta}, Z(\mu) \rangle$ is affine in $\boldsymbol{\theta}$ ✓
- what about Π_{mix}?:

Substituting Mixed Policy

Satisfying conditions minimax theorem:

$$\min_{\mu \in \Pi_{\text{mix}}} \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\mu) \rangle$$

- $\mathcal{K}$ is convex and $\langle \boldsymbol{\theta}, Z(\mu) \rangle$ is affine in $\boldsymbol{\theta}$ ✓
- what about Π_{mix}?:
  - if we define $\mu = \alpha\mu_1 + (1-\alpha)\mu_2 \in \Pi_{\text{mix}}$ as

$$\mu(\pi) = \alpha\mu_1(\pi) + (1-\alpha)\mu_2(\pi)$$

- $Z(\mu) = \alpha Z(\mu_1) + (1-\alpha) Z(\mu_2)$ and consequently $\langle \boldsymbol{\theta}, Z(\mu) \rangle$ is affine in μ ✓

So far we showed...

we started with

$$\min_{\pi \in \Pi} \operatorname{dist}(Z(\pi), \mathcal{C})$$

So far we showed...

we started with

$$\min_{\pi \in \Pi} \operatorname{dist}(Z(\pi), \mathcal{C})$$

ended up with

$$\max_{\boldsymbol{\theta} \in \mathcal{K}} \min_{\mu \in \Pi_{\text{mix}}} \langle \boldsymbol{\theta}, Z(\mu) \rangle$$

So far we showed...

we started with

$$\min_{\pi \in \Pi} \operatorname{dist}(Z(\pi), \mathcal{C})$$

ended up with

$$\max_{\boldsymbol{\theta} \in \mathcal{K}} \min_{\mu \in \Pi_{\text{mix}}} \langle \boldsymbol{\theta}, Z(\mu) \rangle$$

missing part:

$$\boxed{\operatorname{dist}(Z(\pi), \mathcal{C}) = \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\pi) \rangle}$$

for some convex set $\mathcal{K}$

Conic Duality

Definition (Cone)

A set $C \subseteq \mathbb{R}^d$ is a cone if it is closed under multiplication by nonnegative scalars.

Definition (Conic Hull)

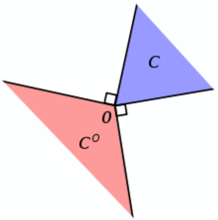
$K \subseteq \mathbb{R}$ *convex, define* $\operatorname{cone}(K) = \{\alpha \boldsymbol{x} : \alpha \in \mathbb{R}^+, \boldsymbol{x} \in K\}$

it is easy to check that $\operatorname{cone}(K)$ is also convex

Definition (Polar Cone)

Given any convex cone $C \subseteq \mathbb{R}^d$, we can define the polar cone of C as

$$C^0 := \{\boldsymbol{\theta} \in \mathbb{R}^d : \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle \leq 0 \text{ for all } \boldsymbol{x} \in C\}$$



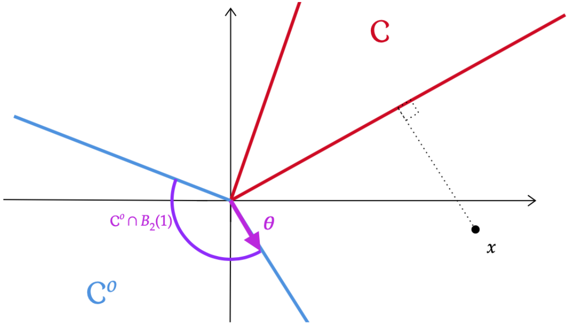
- C° is a convex cone
- $(C^\circ)^\circ = C$

Lemma ([?])

For every convex cone C in $\mathbb{R}^d$

$$\mathrm{dist}(\boldsymbol{x}, C) = \max_{\boldsymbol{\theta} \in C^0 \cap B_2(1)} \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle$$

where $B_2(r)$ is l_2 ball of radius r.



Cone Target Set

Assume: target set $\mathcal{C}$ is a cone

Cone Target Set

Assume: target set $\mathcal{C}$ is a cone

missing part:

$$\boxed{\mathrm{dist}(Z(\pi), \mathcal{C}) = \max_{\boldsymbol{\theta} \in \mathcal{K}} \langle \boldsymbol{\theta}, Z(\pi) \rangle}$$

$$\mathcal{K} = C^{\circ} \cap B_2(1)$$

How to solve this game

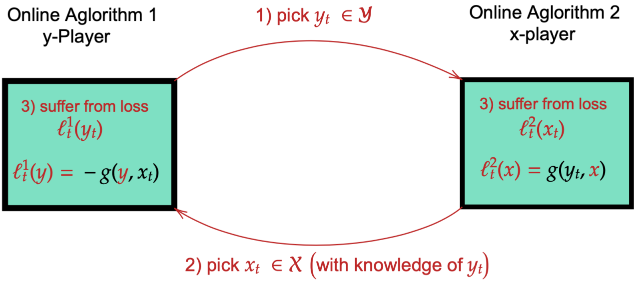
$$\boxed{\max_{\boldsymbol{\theta}\in\mathcal{K}} \min_{\mu\in\Pi_{\text{mix}}} \langle \boldsymbol{\theta}, Z(\mu)\rangle}$$

How to solve this game

$$\boxed{\max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} g(x, y)}$$

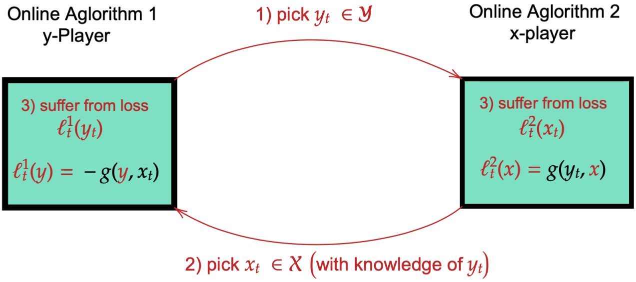
How to solve this game

$$\boxed{\max_{y\in\mathcal{Y}} \min_{x\in\mathcal{X}} g(x,y)}$$



How to solve this game

$\boxed{\text{y-Player strategy } \bar{y} = \frac{1}{T}\sum_{i=1}^{T} y_t}$ $\boxed{\text{x-Player strategy } \bar{x} = \frac{1}{T}\sum_{i=1}^{T} x_t}$



Online Learning

- Actor is given a convex decision set $\mathcal{K} \subseteq \mathbb{R}^d$
- At time $t = 1, 2, \ldots, T$
  - Actor takes an action $\boldsymbol{\theta}_t \in \mathcal{K}$
  - Receive a loss function $\ell_t : \mathcal{K} \to \mathbb{R}$
  - Incur loss of $\ell_t(\boldsymbol{\theta}_t)$

What's the Goal?

Learner wants to minimize regret
(i.e., Competing with best action in hindsight)

What's the Goal?

Learner wants to minimize regret
(i.e., Competing with best action in hindsight)

Definition (Regret)

$$\mathrm{Regret}_T = \sum_{t=1}^{T} \ell_t(\boldsymbol{\theta}_t) - \min_{\boldsymbol{\theta}\in\mathcal{K}} \sum_{t=1}^{T} \ell_t(\boldsymbol{\theta})$$

What's the Goal?

Learner wants to minimize regret
(i.e., Competing with best action in hindsight)

Definition (Regret)

$\mathrm{Regret}_T = \sum_{t=1}^{T} \ell_t(\boldsymbol{\theta}_t) - \min_{\boldsymbol{\theta}\in\mathcal{K}} \sum_{t=1}^{T} \ell_t(\boldsymbol{\theta})$

Cornerstone of online learning: no-regret learning
sublinear regret $\mathrm{Regret}_T \in o(T)$ (i.e., $\frac{\mathrm{Regret}_T}{T} \to 0$ as $T \to \infty$)

What's the Goal?

Learner wants to minimize regret
(i.e., Competing with best action in hindsight)

Definition (Regret)

$\mathrm{Regret}_T = \sum_{t=1}^{T} \ell_t(\boldsymbol{\theta}_t) - \min_{\boldsymbol{\theta} \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(\boldsymbol{\theta})$

Cornerstone of online learning: no-regret learning
sublinear regret $\mathrm{Regret}_T \in o(T)$ (i.e., $\frac{\mathrm{Regret}_T}{T} \to 0$ as $T \to \infty$)

When do we have such algorithm?

Convex Loss Function (OCO)

Case 1: all $\ell_t(\cdot)$ are convex
Online Convex Optimization (OCO)

Theorem

Assume: $\mathcal{K}$ *and* $\|\nabla \ell_t(\boldsymbol{\theta})\|$ *are bounded for every* t *and* $\boldsymbol{\theta} \in \mathcal{K}$.
Then, there exists an algorithm $\mathcal{O}_{\mathcal{K}}$ *with* $\mathrm{Regret}_T(\mathcal{O}_{\mathcal{K}}) \in o(T)$

We'll give such algorithm called Online Gradient Descent [**?**] in the next slide.

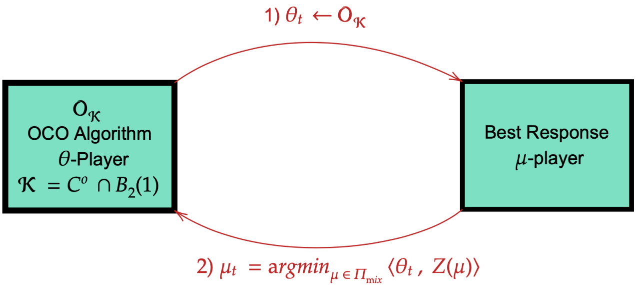
Online Gradient Descent

Algorithm Online Gradient Descent (OGD)

1: **input**: projection oracle $\Gamma_{\mathcal{K}}$ $\{\Gamma_{\mathcal{K}}(\boldsymbol{\theta}) = \mathrm{argmin}_{\boldsymbol{\theta}' \in \mathcal{K}} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2\}$
2: **init**: $\boldsymbol{\theta}_1$ arbitrarily
3: **parameters**: step size η_t
4: **for** $t = 1$ **to** T **do**
5: $\quad \boldsymbol{\theta}'_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla \ell_t(\boldsymbol{\theta}_t)$
6: $\quad \boldsymbol{\theta}_{t+1} = \Gamma_{\mathcal{K}}(\boldsymbol{\theta}'_{t+1})$
7: **end for**

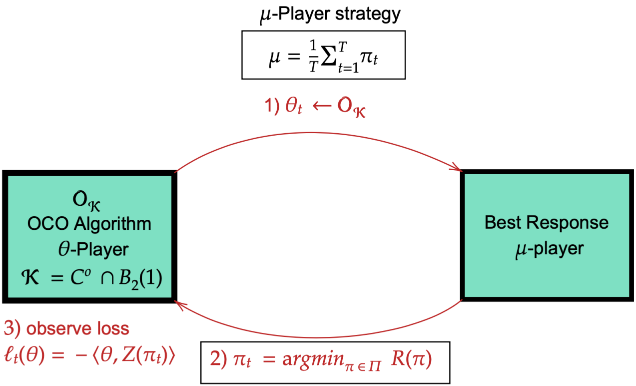
Plugging in. . .

$$\boxed{\max_{\boldsymbol{\theta}\in\mathcal{K}} \min_{\mu\in\Pi_{\text{mix}}} \langle \boldsymbol{\theta}, Z(\mu)\rangle}$$



Best Response can be simplified...

$$\boxed{\max_{\boldsymbol{\theta} \in \mathcal{K}} \min_{\mu \in \Pi_{\mathrm{mix}}} \langle \boldsymbol{\theta}, Z(\mu) \rangle}$$



scalar (standard) RL with reward $r = -\langle \theta_t, z \rangle$

The Algorithm

$$\boxed{\max_{\boldsymbol{\theta} \in \mathcal{K}} \min_{\mu \in \Pi_{\text{mix}}} \langle \boldsymbol{\theta}, Z(\mu) \rangle}$$

Algorithm Main Algorithm

input: convex cone $\mathcal{C}$, OCO Algorithm $\mathcal{O}$
set $\mathcal{K} := \mathcal{C}^\circ \cap B_2(1)$
for $t = 1$ **to** T **do**
  $\boldsymbol{\theta}_t \leftarrow \mathcal{O}_\mathcal{K}$ makes a decision $\mathcal{K}$
  $\pi_t \leftarrow \mathrm{argmax}_{\pi \in \Pi} R(\pi)$ {find best policy in scalar MDP with $r = -\langle \boldsymbol{\theta}_t, \boldsymbol{z} \rangle$ }
  $\mathcal{O}_\mathcal{K}$ observe loss $\ell_t(\boldsymbol{\theta}) = -\langle \boldsymbol{\theta}, Z(\pi_t) \rangle$
end for
$\mu = \frac{1}{T} \sum_{t=1}^{T} \pi_t$
return μ

Best-response Oracle

Best-response oracle: $\textsc{BestResponse}(\boldsymbol{\theta})$.

Given $\boldsymbol{\theta} \in \mathbb{R}^d$, 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 $r = -\boldsymbol{\theta} \cdot \boldsymbol{z}$.

OGD needs gradient and projection oracle

The gradient of loss function: $-Z(\pi_t)$ (can be simply estimated)

Estimation oracle: $\textsc{Est}(\pi)$.

Given policy π, return $\hat{\boldsymbol{z}}$ satisfying $\|\hat{\boldsymbol{z}} - Z(\pi)\| \leq \epsilon_1$.

If we can project into $\mathcal{C}$ we can project into $\mathcal{K} = \mathcal{C}^\circ \cap B_2(1)$

Projection oracle: $\Gamma_{\mathcal{C}}(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{x}' \in \mathcal{C}} \|\boldsymbol{x} - \boldsymbol{x}'\|$.

ApproPO

Algorithm ApproPO

input: $\textsc{BestResponse}(\cdot)$,$\textsc{Est}(\cdot)$, $\Gamma_{\mathcal{C}}$
set: $\mathcal{K} := \mathcal{C}^\circ \cap B_2(1)$
init: $\boldsymbol{\theta}_1$ arbitrarily in $\mathcal{K}$
for $t = 1$ **to** T **do**
  $\pi_t \leftarrow \textsc{BestResponse}(\boldsymbol{\theta}_t)$
  $\hat{\boldsymbol{z}}_t \leftarrow \textsc{Est}(\pi_t)$
  $\boldsymbol{\theta}_{t+1} = \Gamma_{\mathcal{K}}(\boldsymbol{\theta}_t + \eta \hat{\boldsymbol{z}}_t)$
end for
$\mu = \frac{1}{T} \sum_{t=1}^{T} \pi_t$
return μ

Main Theorem

Theorem (Main Theorem)

If we run ApproPO for T iteration and μ is the mixed policy returned by the algorithm, then we have

$$\mathrm{dist}(Z(\mu), \mathcal{C}) \leq \min_{\mu \in \Pi_{\mathrm{mix}}} \mathrm{dist}(Z(\mu), \mathcal{C}) + O(T^{-1/2}) + \epsilon_0 + 2\epsilon_1$$

ApproPO-feasibility

If we are only interested in **feasibility problem**, replacing best-response with a weaker oracle suffices

Positive-response oracle: $\textsc{PosResponse}(\boldsymbol{\theta})$.

Given $\boldsymbol{\theta} \in \mathbb{R}^d$, return $\pi \in \Pi$ that satisfies $R(\pi) \geq -\epsilon_0$ if $\max_{\pi' \in \Pi} R(\pi') \geq 0$ (and arbitrary π otherwise), where $R(\pi)$ is the long-term reward of π with scalar reward $r = -\boldsymbol{\theta} \cdot \boldsymbol{z}$.

Removing Cone Assumption

Lemma (extension of Lemma 14 [?])

Assume: compact and convex $\mathcal{C}$, *for any* $\delta > 0$, *let* $\kappa = \frac{\max_{\boldsymbol{c}\in\mathcal{C}} \|\boldsymbol{c}\|_2}{\sqrt{2\delta}}$
Then, for any $c \in \mathbb{R}^d$

$$\mathrm{dist}(c, \mathcal{C}) \leq (1+\delta)\mathrm{dist}(c \oplus \kappa, \tilde{\mathcal{C}})$$

where $\tilde{\mathcal{C}} = \mathrm{cone}(\mathcal{C} \times \{\kappa\})$

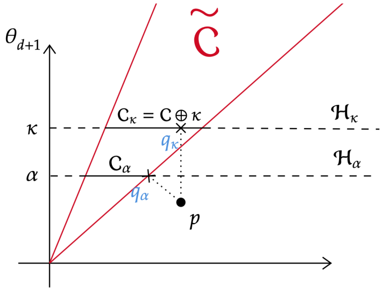
Projection into $\tilde{\mathcal{C}}$

Find $\Gamma_{\tilde{\mathcal{C}}}(p)$ given access to $\Gamma_{\mathcal{C}}(\cdot)$
Let $\mathcal{H}_\alpha = \{\boldsymbol{\theta} \in \mathbb{R}^{d+1} \mid \theta_{d+1} = \alpha\}$
Let $\mathcal{C}_\alpha = \mathcal{H}_\alpha \cap \tilde{\mathcal{C}} = \frac{\alpha}{\kappa} C \oplus \alpha$

$$q_\alpha = \Gamma_{\mathcal{C}_\alpha}(p) = \frac{\alpha}{\kappa}\Gamma_{\mathcal{C}}(\frac{\kappa}{\alpha}p^{1:d})$$

$$q = \Gamma_{\tilde{\mathcal{C}}}(p) = \operatorname{argmin}_{q_\alpha} \|q_\alpha - p\|_2$$

It's easy to check that $\|q_\alpha - p\|_2$ is convex in α. Therefore, we can find α^* which minimize this function, and the original projection will be on q_{α^*}



Practical Implementation:

Algorithm AppropPO

input: $\textsc{BestResponse}(\cdot)$,$\textsc{Est}(\cdot)$, $\Gamma_{\mathcal{C}}$
set: $\mathcal{K} := \mathcal{C}^{\circ} \cap B_2(1)$
init: $\boldsymbol{\theta}_1$ arbitrarily in $\mathcal{K}$
for $t = 1$ **to** T **do**
  $\pi_t \leftarrow \textsc{BestResponse}(\boldsymbol{\theta}_t)$
  $\hat{\boldsymbol{z}}_t \leftarrow \textsc{Est}(\pi_t)$
  $\boldsymbol{\theta}_{t+1} = \Gamma_{\mathcal{K}}(\boldsymbol{\theta}_t + \eta \hat{\boldsymbol{z}}_t)$
end for
$\mu = \frac{1}{T} \sum_{t=1}^{T} \pi_t$
return μ

Practical Implementation: Positive Response oracle

- Replaced Best Response oracle with Positive Response oracle

Algorithm AppproPO

input: $\textsc{PosResponse}(\cdot)$, $\textsc{Est}(\cdot)$, $\Gamma_{\mathcal{C}}$
set: $\mathcal{K} := \mathcal{C}^\circ \cap B_2(1)$
init: $\boldsymbol{\theta}_1$ arbitrarily in $\mathcal{K}$
for $t = 1$ **to** T **do**
  $\cancel{\pi_t \leftarrow \textsc{BestResponse}(\boldsymbol{\theta}_t)}$
  $\pi_t \leftarrow \textsc{PosResponse}(\boldsymbol{\theta}_t)$
  $\hat{\boldsymbol{z}}_t \leftarrow \textsc{Est}(\pi_t)$
  $\boldsymbol{\theta}_{t+1} = \Gamma_{\mathcal{K}}(\boldsymbol{\theta}_t + \eta \hat{\boldsymbol{z}}_t)$
end for
$\mu = \frac{1}{T} \sum_{t=1}^{T} \pi_t$
return μ

Practical Implementation: Estimation Oracle

- Replaced Best Response oracle with Positive Response oracle
- Average measurement vector $\hat{\boldsymbol{z}}_t$ collected from last n-trajectories from Positive Response oracle

Algorithm ApproPO

input: $\textsc{PosResponse}(\cdot)$, $\textsc{Est}(\cdot)$, $\Gamma_{\mathcal{C}}$, n
set: $\mathcal{K} := \mathcal{C}^\circ \cap B_2(1)$
init: $\boldsymbol{\theta}_1$ arbitrarily in $\mathcal{K}$
for $t = 1$ **to** T **do**
    $\cancel{\pi_t \leftarrow \textsc{BestResponse}(\boldsymbol{\theta}_t)}$
    $(\pi_t, \hat{\boldsymbol{z}}_t) \leftarrow \textsc{PosResponse}(\boldsymbol{\theta}_t, n)$
    $\cancel{\hat{\boldsymbol{z}}_t \leftarrow \textsc{Est}(\pi_t)}$
    $\boldsymbol{\theta}_{t+1} = \Gamma_{\mathcal{K}}(\boldsymbol{\theta}_t + \eta\hat{\boldsymbol{z}}_t)$
end for
$\mu = \frac{1}{T}\sum_{t=1}^{T} \pi_t$
return μ

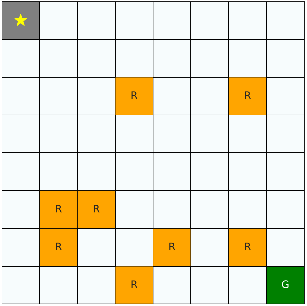
Practical Implementation: Cache

- Replaced Best Response oracle with Positive Response oracle
- Average measurement vector $\hat{\boldsymbol{z}}_t$ collected from last n-trajectories from Positive Response oracle
- Maintain cache of all $(\pi_t, \hat{\boldsymbol{z}}_t)$

Algorithm ApproPO

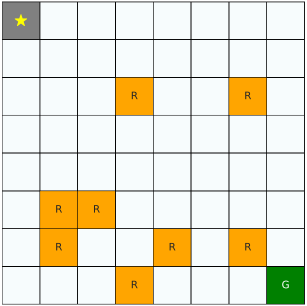
input: $\textsc{PosResponse}(\cdot)$,$\textsc{Est}(\cdot)$, $\Gamma_{\mathcal{C}}$, n
set: $\mathcal{K} := \mathcal{C}^\circ \cap B_2(1)$
init: $\boldsymbol{\theta}_1$ arbitrarily in $\mathcal{K}$
for $t = 1$ **to** T **do**
  $\cancel{\pi_t \leftarrow \textsc{BestResponse}(\boldsymbol{\theta}_t)}$
  $(\pi_t, \hat{\boldsymbol{z}}_t) \leftarrow \textsc{PosResponse}(\boldsymbol{\theta}_t, n)$
  $\cancel{\hat{\boldsymbol{z}}_t \leftarrow \textsc{Est}(\pi_t)}$
  $\boldsymbol{\theta}_{t+1} = \Gamma_{\mathcal{K}}(\boldsymbol{\theta}_t + \eta \hat{\boldsymbol{z}}_t)$
end for
$\mu = \frac{1}{T} \sum_{t=1}^{T} \pi_t$
return μ

Experiments: Mars Rover Gridworld [?]



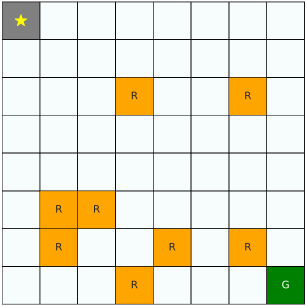
- **Star**: Start

Experiments: Mars Rover Gridworld [?]



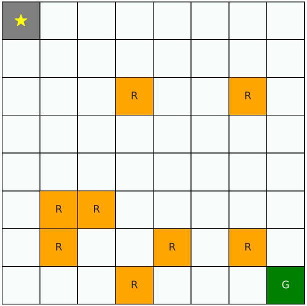
- **Star**: Start
- **G**: Goal

Experiments: Mars Rover Gridworld [?]



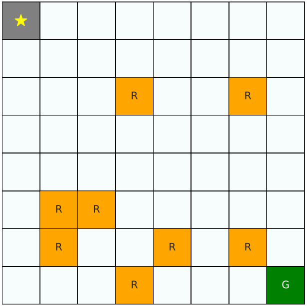
- **Star**: Start
- **G**: Goal
- **R**: Rocks

Experiments: Mars Rover Gridworld [?]



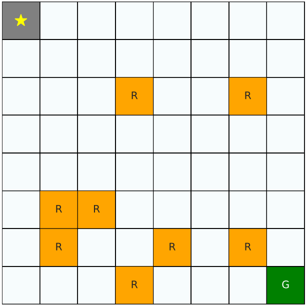
- **Star**: Start
- **G**: Goal
- **R**: Rocks
- **Episode**: terminates when a rock or goal is reached

Experiments: Mars Rover Gridworld [?]



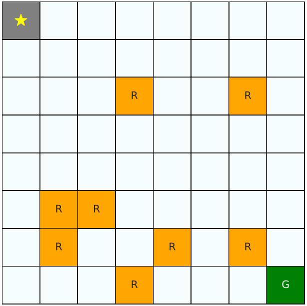
- **Star**: Start
- **G**: Goal
- **R**: Rocks
- **Episode**: terminates when a rock or goal is reached
- **Reward**: zero for terminating, and small negative reward each time step

Experiments: Mars Rover Gridworld [?]



- **Star**: Start
- **G**: Goal
- **R**: Rocks
- **Episode**: terminates when a rock or goal is reached
- **Reward**: zero for terminating, and small negative reward each time step
- **Constraint**: probability of hitting a rock below a threshold

Experiments: Mars Rover Gridworld [?]



- **Star**: Start
- **G**: Goal
- **R**: Rocks
- **Episode**: terminates when a rock or goal is reached
- **Reward**: zero for terminating, and small negative reward each time step
- **Constraint**: probability of hitting a rock below a threshold
- **Environment stochastic**: probability $\delta = 0.05$ agent takes random action

Algorithm RCPO

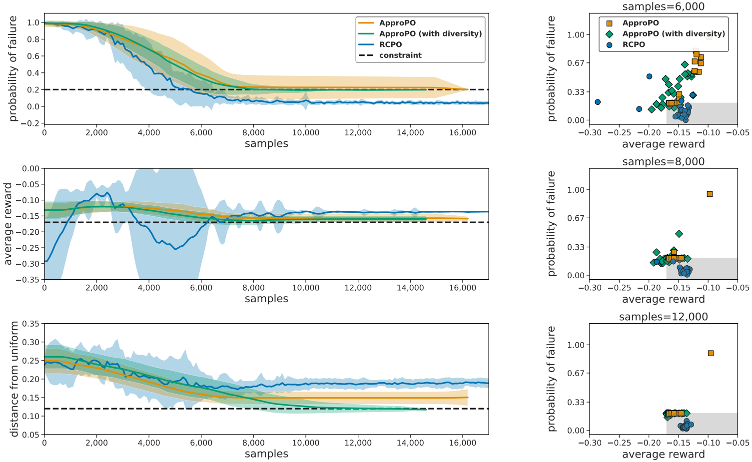
input: A2C$(\cdot)$, α
init: $\boldsymbol{\theta}_1$ arbitrarily
for $t = 1$ **to** T **do**
  $(\pi_t, \boldsymbol{c}_t) \leftarrow \text{A2C}(\boldsymbol{\theta}_t, 1)$
  $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta(\boldsymbol{c}_t - \alpha)$
end for
return π_t

[?]

Algorithm ApproPO

input:$\textsc{PosResponse}(\cdot)$,$\Gamma_{\mathcal{C}}$
set: $\mathcal{K} := \mathcal{C}^\circ \cap B_2(1)$
init: $\boldsymbol{\theta}_1$ arbitrarily in $\mathcal{K}$
for $t = 1$ **to** T **do**
  $(\pi_t, \hat{\boldsymbol{z}}_t) \leftarrow \textsc{PosResponse}(\boldsymbol{\theta}_t, n)$
  $\boldsymbol{\theta}_{t+1} = \Gamma_{\mathcal{K}}(\boldsymbol{\theta}_t + \eta \hat{\boldsymbol{z}}_t)$
end for
$\mu = \frac{1}{T}\sum_{t=1}^{T} \pi_t$
return μ

Results:



probability of failure
samples
ApproPO
ApproPO (with diversity)
RCPO
constraint
average reward
distance from uniform
samples=6,000
samples=8,000
samples=12,000

Any Questions?

References I