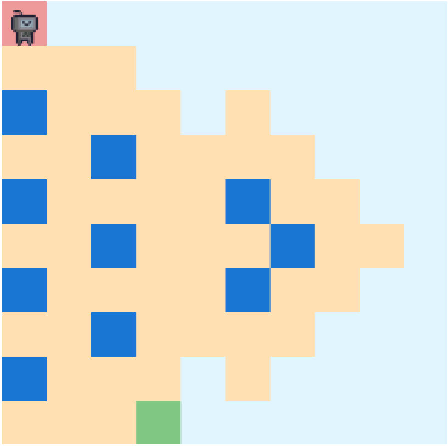
Safe Reinforcement Learning with Natural Language Constraints

Hongyou Zhou

TU Berlin

10.Nov.2022

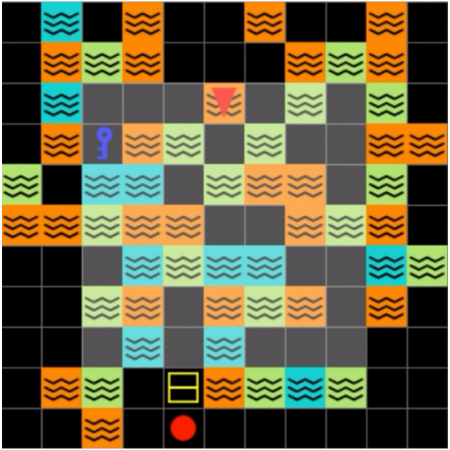
Motivation



We are all familiar with this example:

1. The robot wants to go to the green square.
2. The blue square is a broken hole on the ice surface.
3. There is uncertainty in the movement due to the ice surface.
4. The orange square is a restriction that we artificially added in order to ensure agent's success during exploration.

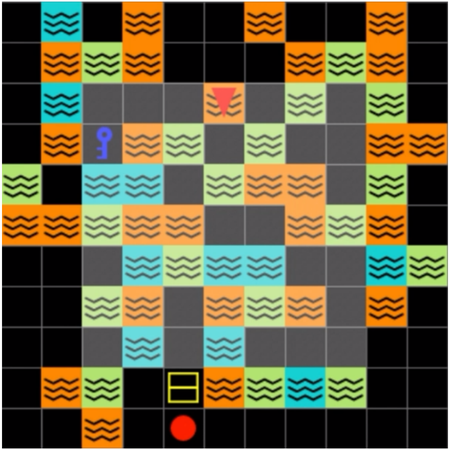
Motivation



Consider a more complex example:

1. The lava is really hot, so it will hurt you a lot. Only walk on it 3 times.
2. There should always be at least 3 squares between you and water
3. Make sure you don't walk on water after walking on grass

Motivation



- Such constraints demand domain expertise
- thus limiting the adoption of safe RL

Similar Works

- Without middle representation [1]
  - Jointly processes the observations and the constraints
  - Trained with an end-to-end approach
- Using trust region policy optimization [2]
  - Ignores all constraints and only optimizes the reward
  - Substantial constraint violations

Contributions

- Constraint interpreter that encodes textual constraints into spatial and temporal representations of forbidden states.
- Policy network that uses these representations to produce a policy achieving minimal constraint violations during training.

Problem formulation

$$\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, T, Z, \mathcal{X}, R, C \rangle$$

$$\mathcal{S} : \text{set of states} \tag{1a}$$

$$\mathcal{O} : \text{set of observations} \tag{1b}$$

$$\mathcal{A} : \text{set of actions} \tag{1c}$$

$$T : \text{conditional probability } T(s'|s,a) \tag{1d}$$

$$Z : \text{conditional probability } Z(o|s) \tag{1e}$$

$$\mathcal{X} : \text{set of textual constraint specifications} \tag{1f}$$

$$R : \text{reward function } \mathcal{S} \times \mathcal{A} \to \mathbb{R} \tag{1g}$$

$$C : \text{true underlying constraint function } \mathcal{S} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R} \tag{1h}$$

RL with constraints

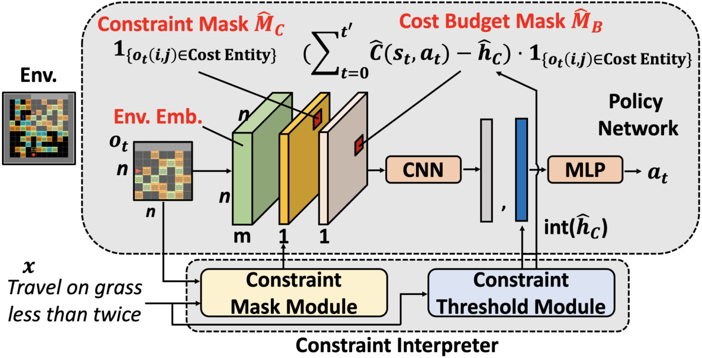
We seek a policy π that maximizes the cumulative discounted reward J_R

$$\max_{\pi} J_R(\pi) \doteq \mathbb{E}_{\tau \sim \pi}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)] \tag{2}$$

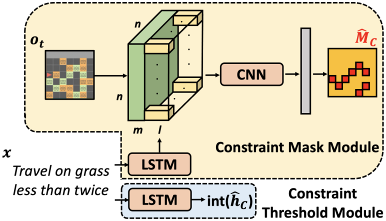
s.t.

$$J_C(\pi) \doteq \mathbb{E}_{\tau \sim \pi}[\sum_{t=0}^{\infty} \gamma^t C(s_t, a_t, x)] \leq h_C(x) \tag{3}$$

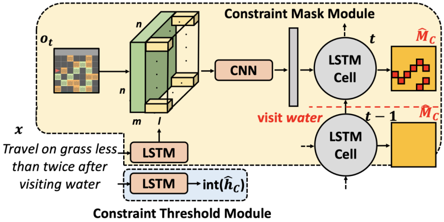
Model overview



Constraint Interpreter



(a) For budgetary and relational constraints



(b) For sequential constraints

Cost Function: Interpreter learning

For constraint mask module we minimize:

$$\mathcal{L}(\Theta_1) = -\mathbb{E}_{(o_t,x)\sim\mathcal{D}_{train}}\left[\frac{1}{|M_C|}\sum_{i,j=1}^{n} ylog\hat{y} + (1-y)log(1-\hat{y})\right] \tag{4}$$

where y is the target $M_C(i,j;o_t,x)$.

For constraint threshold module we minimize:

$$\mathcal{L}(\Theta_2) = -\mathbb{E}_{(o_t,x)\sim\mathcal{D}_{train}}\left[(h_C(x) - \hat{h}_C(x))^2\right] \tag{5}$$

Task setup

Safety training: for each episode

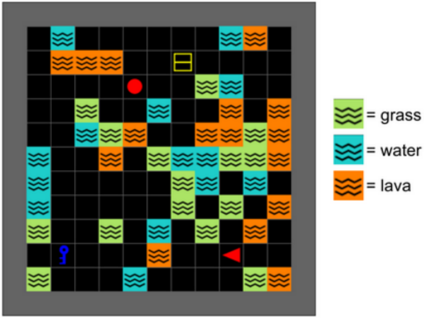
- fixed reward function
- randomly generate a constraint function C and h_C
- sample a constraint text x that describes C and h_C from training set of texts

Safety evaluation:

- Varies reward function
- Ability of an agent can learn new tasks while following previously learned textual constraints
- Applicability when using textual constraints unseen in training

Data Generation

1. The environment



- Randomized start location
- Randomly fills the environment with objects
- Randomly samples one constraint out of all possible constraints and assigns this constraint to the environment

Data Generation

2. Natural language constraints

One of these block types causes wear and tear, so your friend can only stand on them a limited number of times. For example, if **Number of Times = 3**, then your friend can only walk on the **Dangerous Block** three times, at most. Knowing this, tell the other person what she can or cannot do in this situation.

Dangerous Block: lava

Number of Times: 5

Type what you would say here...

(a) Budgetary prompt

One of these objects is dangerous, and your friend must maintain a minimum safe distance from that type of object. For example, if **Minimum Distance = 1**, then your friend must stay at least 1 step away from **Dangerous Blocks** at all times. Knowing this, tell the other person what she can or cannot do in this situation.

Dangerous Block: lava

Minimum Distance: 3

Type what you would say here...

(b) Relational prompt

One of the three block types is a trigger block. If you friend touches a trigger block, one of the block types becomes dangerous. Your job is to tell the other person what she can or cannot do in this situation. For example, if **Trigger Block = lava**, then your friend can't walk on **Dangerous Blocks** after walking on any **lava** blocks.

Trigger Block: lava

Dangerous Block: water

Type what you would say here...

(c) Sequential prompt

Training

1. Interpreter learning

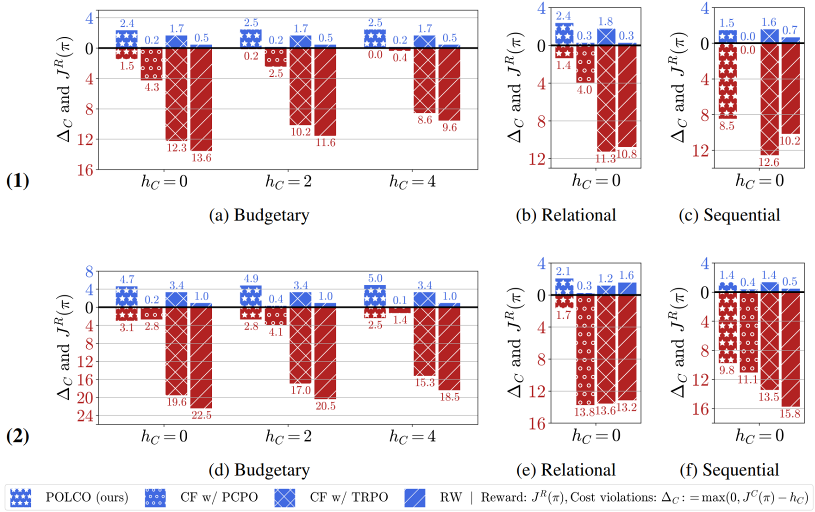
- Use a random policy to explore the environment
- Compares the constraint violations encountered in the trajectory and the cost specification C
- Compeletely separate from policy training

Data Generation

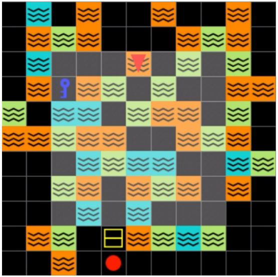
2. Policy learning

- Projection-based constrained policy optimization (PCPO) [3]
- Use $\hat{M}_C$ and $\hat{h}_C(x)$ from the trained constraint interpreter for computing $J_R(\pi)$ and $J_C(\pi)$

Result


(1)
(a) Budgetary
(b) Relational
(c) Sequential
(2)
(d) Budgetary
(e) Relational
(f) Sequential
Δ_C and $J^R(\pi)$
$h_C = 0$
$h_C = 2$
$h_C = 4$
POLCO (ours)
CF w/ PCPO
CF w/ TRPO
RW
Reward: $J^R(\pi)$, Cost violations: $\Delta_C := \max(0, J^C(\pi) - h_C)$

Summary



What they did:

- Deep learning model for interpreting constraints
- **P**olicy **O**ptimization with **L**anguage **CO**nstraints (POLCO)
- New benchmark called HazardWorld

Potential problems:

- Data generation
- Interpreter accuracy

References

[allowframebreaks]

[1] T.-Y. Yang, M. Y. Hu, Y. Chow, P. J. Ramadge, and K. Narasimhan, "Safe reinforcement learning with natural language constraints," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13 794–13 808, 2021.

[2] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[3] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge, "Projection-based constrained policy optimization," *arXiv preprint arXiv:2010.03152*, 2020.