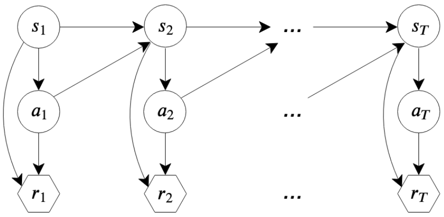
We adapt the **Adversarial Risk Analysis** framework to make a decision maker **more robust against an adversary** in **Reinforcement Learning** tasks

Markov Decision Processes Under Threats: a RL approach

Victor Gallego, Roi Naveiro, David Rios Insua, David Gómez-Ullate

1 Intro to RL

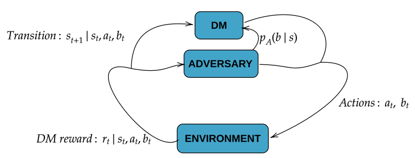
- RL is a computational approach to **Markov Decision Processes**.
- MDP models a single agent (DM) making decisions **sequentially**:



- Agent aims at finding the **policy** maximizing **long term discounted expected utility**: $E\left[\sum_{t=0}^{\infty} \gamma^t r(a_t, s_t)\right]$
- **Q-learning** is an efficient approach to this problem: agent sequentially estimates the expected cumulative reward (utility) through $Q(s,a) := (1-\alpha)Q(s,a) + \alpha\left(r(s,a) + \gamma \max_{a'} Q(s',a')\right)$
- Optimal policy $p(a|s)$: $\arg\max_a Q(s,a)$ with $1-\epsilon$ prob.

2 Threatened MDPs

- Q-learning **fails** if there is an adversary (now **reward distribution** is not stationary from DM's point of view)



- Our strategy: augment MDP to a TMDP
- Modified Q-learning rule: $Q(s,a,b) := (1-\alpha)Q(s,a,b) + \alpha(r(s,a,b) + \gamma \max_{a'} E_{p_A(b|s')}\left[Q(s',a',b)\right])$

3 Modelling opponents

- No common knowledge $\Rightarrow$ **uncertainty about adversary policy**, modelled through $p_A(b|s)$.
- **Non-strategic opponent**: $p_A(b|s) \sim$ Dirichlet. The DM would choose the action maximizing $\psi_s(a_i) = E_{p_A(b|S)}[Q(s,a_i,b)]$
- **Strategic opponent**: he may model us as non-strategic players (level-0), making himself a level-1 thinker...
  - We can define a hierarchy of nested TMDPs (up to a given level-k) and solve all of them simultaneously.
  - If the DM is level-k her policy is given by $\arg\max_{a_{i_k}} Q_k(s, a_{i_k}, b_{j_{k-1}})$ where $b_{j_{k-1}}$ is given by $\arg\max_{b_{j_{k-1}}} \hat{Q}_{k-1}(, a_{i_{k-2}}, b_{j_{k-1}})$

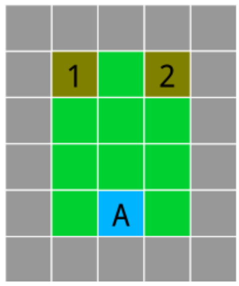
Algorithm 1 Level-2 thinking update rule

Require: Q_2, $\hat{Q}_1$, α_2, α_1 (DM and opponent Q-functions and learning rates, respectively).
Observe transition (s, a, b, r_A, r_B, s') from the TMDP environment
$\hat{Q}_1(s,b,a) := (1-\alpha_1)\hat{Q}_1(s,b,a) + \alpha_1(r_B + \gamma \max_{b'} \mathbb{E}_{p_B(a'|s')}\left[\hat{Q}_1(s',b',a')\right])$
Compute B's estimated $\epsilon-$greedy policy $p_A(b|s')$ from $\hat{Q}_1(s,b,a)$
$Q_2(s,a,b) := (1-\alpha_2)Q_2(s,a,b) + \alpha_2(r_A + \gamma \max_{a'} \mathbb{E}_{p_A(b'|s')}\left[Q_2(s',a',b')\right])$

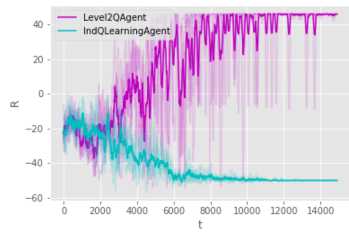
- **Opponent averaging**: we place a Dirichlet prior over the type/level of opponent, as in a Bayesian game. As iterations run, the DM's belief is updated.

Experiments

- *Friend or foe* RL security benchmark, from Deepmind 2017.
- The DM needs to travel a room and choose between two identical boxes, hiding **positive** and **negative** rewards, respectively.



- Reward assignment controlled by adaptive adversary (exponential smoother):
  - $p = (p_1, p_2)$ are the DM's probabilities (according to the adversary), of choosing 1 or 2. $p := \beta p + (1-\beta)a$,
  - Adversary places its reward at target $t = \arg\min_i(p)_i$.
- Whereas a naive Q-learner is exploited by their adversary, a level-2 Q-learner is able to account for her opponent:



- More experiments in the paper!!

Conclusions

- We have introduced TMDPs, a framework to provide one-sided prescriptive support to a RL agent who confront adversaries that interfere with the reward process.
- Suitable framework to use existing opponent modelling methods within Q-learning.
- Level-k reasoning scheme about opponents. We extend this approach to account for uncertainty about the opponent's model.
- Empirically, we see that the framework generalizes between different kinds of opponents!!

ICMAT Instituto de Ciencias Matemáticas

https://arxiv.org/abs/1809.01560
https://vicgalle.github.io