Markov Decision Processes under Threats

Víctor Gallego, Roi Naveiro, David Gómez-Ullate, David Ríos Insúa

victor.gallego@icmat.es, roi.naveiro@icmat.es

Institute of Mathematical Sciences (ICMAT-CSIC)

ADA, June 2019 Milano

RL success story

Disclaimer: focus on Reinforcement Learning (RL).

RL success story

Disclaimer: focus on Reinforcement Learning (RL).

The problem

- Reinforcement Learning (RL) is more than playing Go...
- Applications of RL are continuously growing.
- Some applications in settings where **security issues** are crucial (autonomous driving)...

The problem

- Reinforcement Learning (RL) is more than playing Go...
- Applications of RL are continuously growing.
- Some applications in settings where **security issues** are crucial (autonomous driving)...
- ...where there could be adversaries that interfere the reward generating process.

The problem

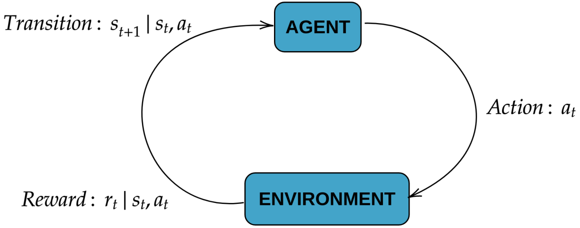
- Reinforcement Learning (RL) is more than playing Go...
- Applications of RL are continuously growing.
- Some applications in settings where **security issues** are crucial (autonomous driving)...
- ...where there could be adversaries that interfere the reward generating process.

Traditional single-agent RL fails...

...as it does not take into account the presence of other agents.

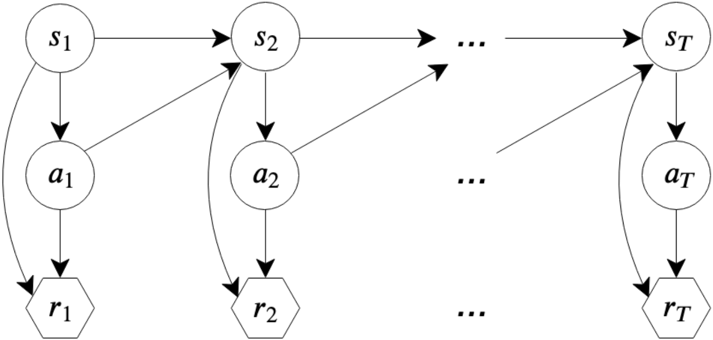
Quick review of RL

- RL is a computational approach to **Markov Decision Processes** (MDP).
- MDP models a single agent (decision maker, DM) making decisions **sequentially** while interacting with an environment.



Quick review of RL

- RL is a computational approach to **Markov Decision Processes** (MDP).
- MDP models a single agent (decision maker, DM) making decisions **sequentially** while interacting with an environment.



Quick review of RL

- Agent aims at finding the **policy** maximizing **long term discounted expected utility**.

$$\mathbb{E}_{\tau}\left[\sum_{t=0}^{\infty}\gamma^{t} r(a_t, s_t)\right]$$

Quick review of RL

- Agent aims at finding the **policy** maximizing **long term discounted expected utility**.

$$\mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma^t r(a_t, s_t) \right]$$

- **Q-learning** is an efficient approach to this problem: agent sequentially estimates the expected cumulative reward (utility) through

$$Q(s,a) := (1-\alpha) Q(s,a) + \alpha \left(r(s,a) + \gamma \max_{a'} Q(s', a') \right)$$

- If environment is **stationary**, this converges to the optimal policy, under some conditions, Sutton & Barto (2018).
- Optimal policy $p(a|s)$: $\arg\max_a Q(s,a)$ with $1-\epsilon$ prob.

Our objective

- If there are adversaries interfering with the reward process, Q-learning fails.
- We need to reason about and **forecast the adversaries' behaviour**.

Our objective

- If there are adversaries interfering with the reward process, Q-learning fails.
- We need to reason about and **forecast the adversaries' behaviour**.
- Previous work has studied how to model the whole multi-agent system through **Markov Games**, with strong **common knowledge assumptions**, **or too restrictive (i.e., minimax Q-learning)**.

Our objective

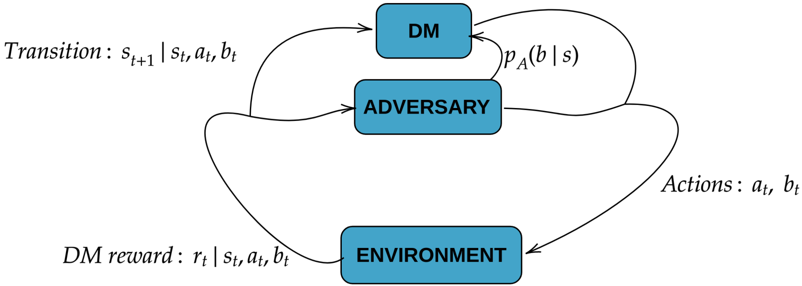
- If there are adversaries interfering with the reward process, Q-learning fails.
- We need to reason about and **forecast the adversaries' behaviour**.
- Previous work has studied how to model the whole multi-agent system through **Markov Games**, with strong **common knowledge assumptions**, **or too restrictive (i.e., minimax Q-learning)**.
- We focus on the problem of prescribing decisions to a **single agent** in adversarial, non-stationary RL settings, accounting for the **lack of information**. **That is, we adapt the Adversarial Risk Analysis framework to RL**.

From MDPs to TMDPs

- Our strategy: augment MDPs to account for adversaries whose actions modify state and reward dynamics.
- TMDP: **Threatened Markov Decision Processes**

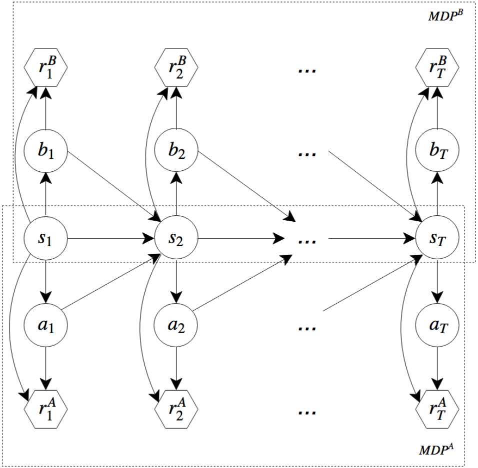
From MDPs to TMDPs

- Our strategy: augment MDPs to account for adversaries whose actions modify state and reward dynamics.
- TMDP: **Threatened Markov Decision Processes**
- We restrict to the **single-adversary** case.
- **Key element:** $p_A(b|s)$.



From MDPs to TMDPs

- Our strategy: augment MDPs to account for adversaries whose actions modify state and reward dynamics.
- TMDP: **Threatened Markov Decision Processes**



Extending Q-learning to TMDPs

- Modified Q-learning rule:

$$
\begin{aligned}
Q(s,a,b) := {} & (1-\alpha)Q(s,a,b) \\
& + \alpha\left(r(s,a,b) + \gamma \max_{a'} \mathbb{E}_{p_A(b|s')}\left[Q(s',a',b)\right]\right)
\end{aligned}
$$

Extending Q-learning to TMDPs

- Modified Q-learning rule:

$$Q(s,a,b) := (1-\alpha)Q(s,a,b) + \alpha\left(r(s,a,b) + \gamma \max_{a'} \mathbb{E}_{p_A(b|s')}\left[Q(s',a',b)\right]\right)$$

- To choose actions, we compute:

$$Q(s,a) := \mathbb{E}_{p_A(b|s)}\left[Q(s,a,b)\right].$$

  and choose $a^* = \arg\max_a Q(s,a)$ with probability $1-\epsilon$ or an action uniformily at random with probability ϵ.
- The DM will learn both $Q(s,a,b)$ and $p_A(b|s)$.

Modelling the adversary

- No common knowledge $\Rightarrow$ uncertainty about adversary policy, modelled through $p_A(b|s)$.
- How to learn $p_A(b|s)$?

Non strategic opponent

- Let's call $p_j|s$ the probability of the adversary taking action b_j in state s.
- Place a Dirichlet prior $(p_1|s, \ldots, p_n|s) \sim \mathcal{D}(\alpha_1(s), \ldots, \alpha_n(s))$.
- The posterior is $\mathcal{D}(\alpha_1(s) + h_1(s), \ldots, \alpha_n(s) + h_n(s))$, where $h_i(s)$ counts how many times did the adversary took action i in state s.

Non strategic opponent

- Let's call $p_j|s$ the probability of the adversary taking action b_j in state s.
- Place a Dirichlet prior $(p_1|s, \ldots, p_n|s) \sim \mathcal{D}(\alpha_1(s), \ldots, \alpha_n(s))$.
- The posterior is $\mathcal{D}(\alpha_1(s) + h_1(s), \ldots, \alpha_n(s) + h_n(s))$, where $h_i(s)$ counts how many times did the adversary took action i in state s.
- The DM would choose the action maximizing

$$\psi_s(a_i) \quad = \mathbb{E}_{\imath_{\mathbb{A}}(|\mathbb{S})}[Q(s, a_i, b)] \propto \sum_{b_j \in \mathcal{B}} Q(s, a_i, b_j)(\alpha_j(s) + h_j)$$

Level-k thinking

- If the opponent is strategic, he may model us as non-strategic players (level-0), making himself a level-1 thinker...
- How to model a level-k thinker?

Level-k thinking

- If the opponent is strategic, he may model us as non-strategic players (level-0), making himself a level-1 thinker...
- How to model a level-k thinker?
- Let's call $TMDP_i^k$ the TMDP agent i needs to optimize if considering his rival a level-$(k-1)$ thinker.

Level-k thinking

- To optimize $TMDP_A^k$, the DM keeps an estimate $\hat{Q}_{k-1}$ of her opponent's Q-function.
- This could be computed optimizing $TMDP_B^{k-1}$, and so on until $k = 1$.
- $k = 1$ could be solved the non-strategic opponent model.

Level-k thinking

- To optimize $TMDP_A^k$, the DM keeps an estimate $\hat{Q}_{k-1}$ of her opponent's Q-function.
- This could be computed optimizing $TMDP_B^{k-1}$, and so on until $k = 1$.
- $k = 1$ could be solved the non-strategic opponent model.
- The top level DM's 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

$$\arg\max_{b_{j_{k-1}}} \hat{Q}_{k-1}(s, a_{i_{k-2}}, b_{j_{k-1}})$$

Combining opponents

- In several situations, we do not have information about the actual opponent model.
- We could place a Dirichlet prior $p(M_i)$ on the opponent model.

Opponent average updating

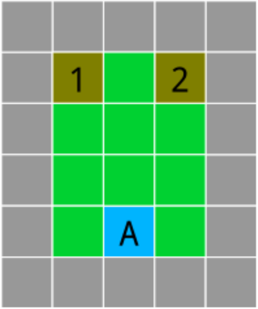
Require: $p(M|H) \propto (n_1, n_2, \dots, n_m)$, where H is the sequence $(b_0, b_1, \dots, b_{t-1})$ of past opponent actions.

1. Observe transition $(s_t, a_t, b_t, r_{A,t}, r_{B,t}, s_{t+1})$.
2. For each M_i, sample $b^i \sim p_{M_i}(b|s)$.
3. If $b^i = b_t$ then update posterior:

$$p(M|(H||b_t)) \propto (n_1, \dots, n_i + 1, \dots, n_m)$$

Experiments

- *Friend or foe* RL security benchmark.
- The DM needs to travel a room and choose between two identical boxes, hiding **positive** and **negative**, respectively.
- Reward assignment controlled by adaptive adversary.



Experiments - Stateless Variant

- No state in this case.
- The adaptive opponent estimates the DM's actions using an 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 $t = \arg\min_i (p)_i$.

Experiments - Stateless Variant

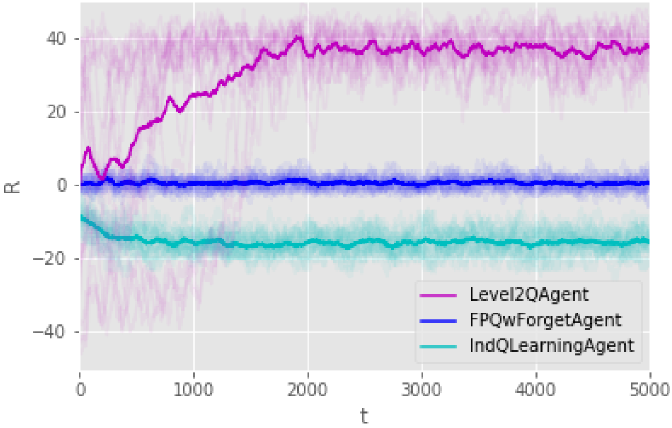


Figure 1: Level 2 and Level 1 vs Exponential Smoother

Experiments - More powerful adversaries

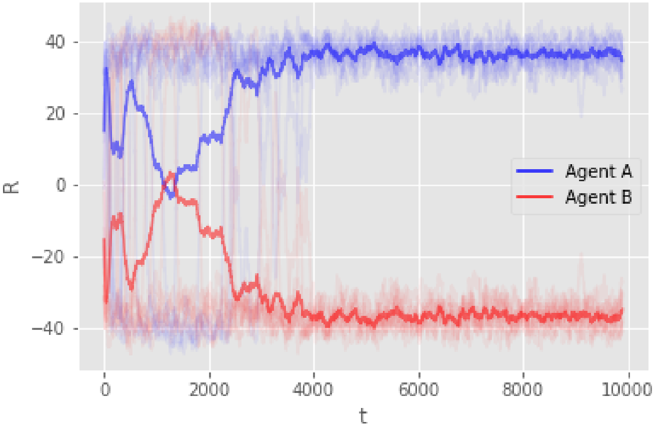


Figure 2: Level 3 with opponent averaging vs Level 1

Experiments - More powerful adversaries

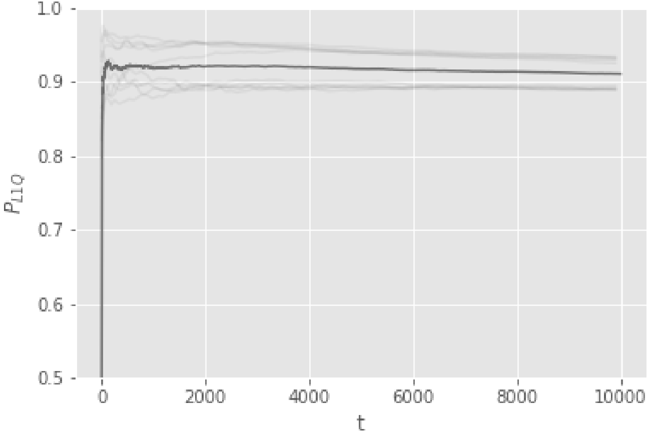
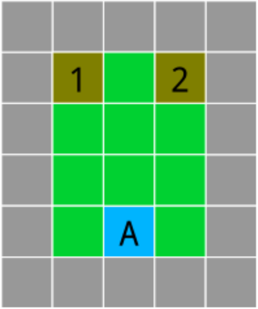


Figure 3: DM's beliefs that her opponent is level-1

Experiments - Spatial Variant

- ± 50 reward depending on chosen target.
- Each step taken, penalized with reward -1.



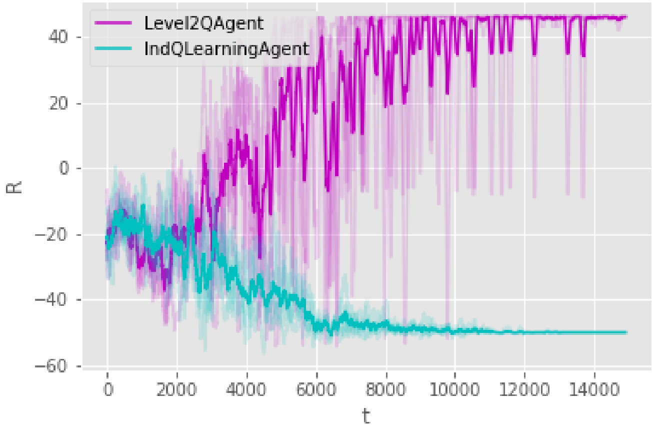


Figure 4: Level 2 and Independent Q learner vs Exponential Smoother

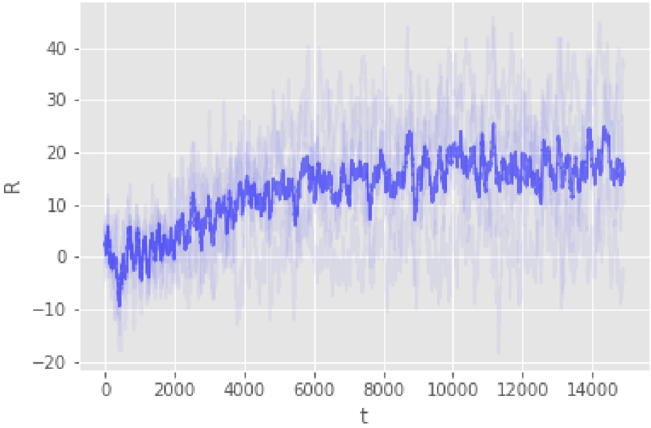


Figure 5: DM with opponent models for a Level 1 and a Level 2 vs Exponential Smoother

Conclusions and future work

- 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!!

Conclusions and future work

- 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!!
- More than one adversaries!
- Deep Q-networks instead of tabular Q-learning

Thank you!

victor.gallego@icmat.es

roi.naveiro@icmat.es

Experiments - More powerful adversaries

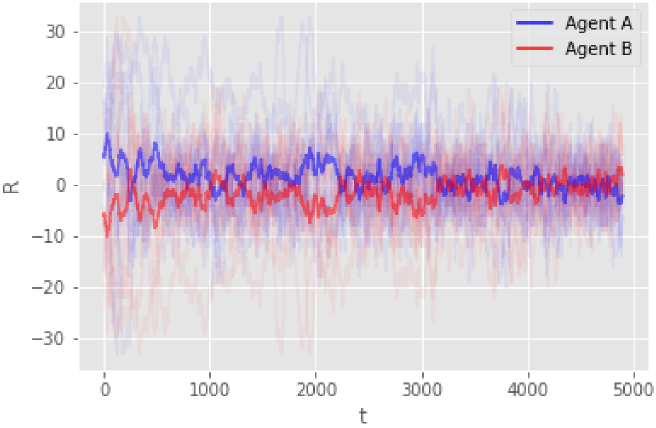


Figure 6: Level 2 vs Level 2

Experiments - More powerful adversaries

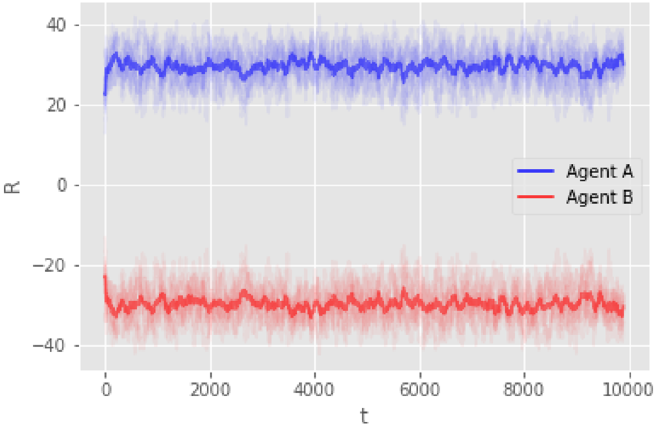


Figure 7: Level 3 vs Level 2

Experiments - More powerful adversaries

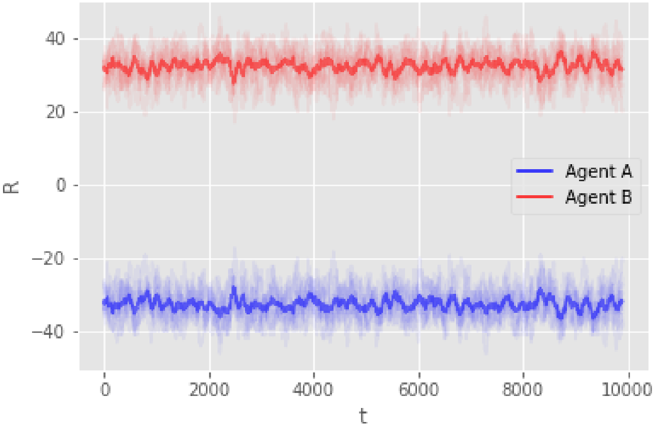


Figure 8: Level 3 vs Level 1