Reinforcement Learning under Threats

Victor Gallego
ICMAT-CSIC
victor.gallego@icmat.es

Roi Naveiro
ICMAT-CSIC
roi.naveiro@icmat.es

David Rios Insua
ICMAT-CSIC
david.rios@icmat.es

Abstract

In several reinforcement learning (RL) scenarios, mainly in security settings, there may be adversaries trying to interfere with the reward generating process. However, when non-stationary environments as such are considered, Q-learning leads to suboptimal results (Busoniu, Babuska, and De Schutter 2010). Previous game-theoretical approaches to this problem have focused on modeling the whole multi-agent system as a game. Instead, we shall face the problem of prescribing decisions to a single agent (the supported decision maker, DM) against a potential threat model (the adversary). We augment the MDP to account for this threat, introducing Threatened Markov Decision Processes (TMDPs). Furthermore, we propose a level-k thinking scheme resulting in a new learning framework to deal with TMDPs. We empirically test our framework, showing the benefits of opponent modeling.

1 Threatened MDPs

A *Threatened Markov Decision Process* (TMDP) is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{B}, \mathcal{T}, r, p_A)$ in which $\mathcal{S}$ is the state space; $\mathcal{A}$ denotes the set of actions available to the supported agent; $\mathcal{B}$ designates the set of threat actions, or actions available to the adversary; $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \Delta(\mathcal{S})$ is the transition distribution; $r : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \Delta(\mathbb{R})$ is the reward distribution; and $p_A(b|s)$ models the beliefs that the DM has about his opponent move, i.e., a distribution over $\mathcal{B}$ for each state $s \in \mathcal{S}$.

We propose to replace the standard Q-learning rule by

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

and compute its expectation over the opponent's action argument

$$Q(s,a) := \mathbb{E}_{p_A(b|s)}\left[Q(s,a,b)\right]. \quad (2)$$

This may be used to compute an $\epsilon-$greedy policy for the DM, i.e., choosing with probability $(1-\epsilon)$ the action $a = \arg\max_a Q(s,a)$ or a uniformly random action with probability ϵ when the DM is at state s.

In general, we will consider both the DM and the adversary as rational agents that aim to maximize their respective expected cumulative rewards.

1.1 Non-strategic opponent

We assume that the supported DM is a joint action learner (i.e., she observes her opponent's actions after he has committed them). At every iteration, the DM shall choose her action using the Q-function from Eq. (2). However, she needs to predict the action b chosen by her opponent. A typical option is to model her adversary using fictitious play (FP), i.e., $p_A(b|s)$ is computed using the empirical frequencies of the opponent past plays.

Algorithm 1 Level-2 thinking update rule

Require: $Q_A, Q_B, \alpha_A, \alpha_B$ (DM and opponent Q-functions and learning rates, respectively).
Observe transition (s, a, b, r_A, r_B, s') from the TMDP environment
$Q_B(s,b,a) := (1-\alpha_B)Q_B(s,b,a) + \alpha_B(r_B + \gamma \max_{b'} \mathbb{E}_{p_B(a'|s')}\left[Q_B(s',b',a')\right])$ ▷ Level-1
Compute B's estimated $\epsilon-$greedy policy $p_A(b|s')$ from $Q_B(s,b,a)$
$Q_A(s,a,b) := (1-\alpha_A)Q_A(s,a,b) + \alpha_A(r_A + \gamma \max_{a'} \mathbb{E}_{p_A(b'|s')}\left[Q_A(s',a',b'))\right]$ ▷ Level-2

1.2 Level-k thinking

Now a level-k scheme (Stahl and Wilson 1994) will be introduced. The previous section described how to model a level-0 opponent, i.e. a non strategic opponent, which can be practical in several scenarios. However, if the opponent is strategic, he may model the supported DM as a level-0 thinker, thus making the adversary a level-1 thinker. This chain can go up to infinity, so we will have to deal with modeling the opponent as a level-k thinker, with k bounded by the computational or cognitive resources of the DM.

To deal with it, we introduce a hierarchy of TMDPs in which $TMDP_i^k$ refers to the TMDP that agent i needs to optimize, while considering its rival as a level-$(k-1)$ thinker. Thus, we have the following process:

- If the supported DM is a level-2 thinker, she may optimize for $TMDP_A^2$. She models B as a level-1 thinker. Consequently, this "modeled" B optimizes $TMDP_B^1$, and while doing so, he models the DM as level-0 (using Section 1.1).

- In general, we have the chain of TMDPs:

$$TMDP_A^k \rightarrow TMDP_B^{k-1} \rightarrow \cdots \rightarrow TMDP_B^1.$$

Exploiting the fact that we are in a repeated interaction setting (and by assumption that both agents can observe all past committed decisions and obtained rewards), each agent may estimate their counterpart's Q-function, $\hat{Q}^{k-1}$: if the DM is optimizing $TMDP_A^k$, she will keep her own Q-function (we refer to it as Q_k), and also an estimate $\hat{Q}_{k-1}$, of her opponent's Q-function. This estimate may be computed by optimizing $TMDP_B^{k-1}$ and so on until $k = 1$. Finally, the top level DM's policy is given by

$$\arg\max_{a_k} Q_k(a_k, b_{k-1}, s),$$

where b_{k-1} is now given by

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

and so on, until we arrive at the induction basis (level-1) in which the opponent may be modeled using the fictitious play approach from Section 1.1. Note that in the previous hierarchy of policies the decisions are obtained in a greedy, deterministic manner (i.e. just by maximizing the lower level $\hat{Q}$ estimate). We may add uncertainty to the policy at each level, for instance, by considering $\epsilon-$greedy policies.

Algorithm 1 specifies the approach for a level-2 DM. Because she is a level-2 DM, we need to account for her Q-function, Q_A (equivalently Q_2 from before), and that of her opponent (who will be level-1), Q_B (equivalently $\hat{Q}_1$).

2 Experiments and Results

To illustrate the TMDP's and level-k reasoning framework, we focus on the *friend or foe* environment, from a suite of RL safety benchmarks introduced in (Leike et al. 2017). The supported DM needs to travel a room and choose between two identical boxes, hiding positive and negative rewards, respectively. This reward assignment is controlled by an adaptive adversary, who estimates the DM's actions using an exponential smoother. Let $\boldsymbol{p} = (p_1, p_2)$ be the probabilities with which the DM will choose targets 1 or 2, respectively, as estimated by the opponent. Then, at every iteration he updates his knowledge through $\boldsymbol{p} := \alpha\boldsymbol{p} + (1-\alpha)\boldsymbol{a}$ where $0 < \alpha < 1$ is a learning rate, unknown from the DM's point of view, and $\boldsymbol{a} \in \{(1,0),(0,1)\}$ is a one-hot encoded vector indicating whether the DM has chosen target 1 or 2. We consider an adversarial opponent which places the positive reward in target $t = \arg\min_i(\boldsymbol{p})_i$.

In particular, we compare the independent Q-learner and a level-2 Q-learner against the adaptive opponent. Targets' rewards are delayed until the DM arrives at one of the respective locations, obtaining ± 50 depending on the target chosen by the adversary. Each step is penalized with a reward of -1 for the DM. Results are displayed in Figure 1. Note that the independent Q-learner is exploited by the adversary. In contrast, the level-2 agent is able to approximately estimate the adversarial behavior, modeling him as a level-1 agent, thus being able to obtain positive rewards.

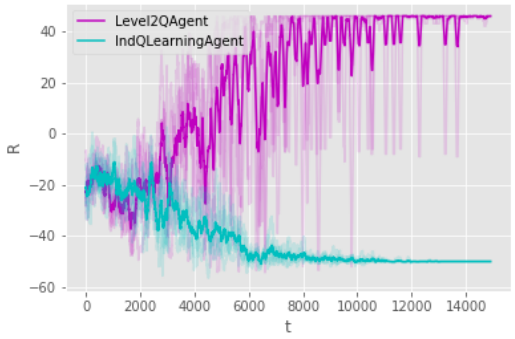

Figure 1: Rewards against the adversarial opponent

Adversary	R_{L1Q}	R_{L2Q}
WoLF-PHC	-2.05	$\mathbf{0.77}$
L2Q	-1.99	$\mathbf{-0.78}$
L1Q	-0.29	$\mathbf{0.87}$

Table 1: DM's rewards in iterated matrix game

In addition, we test our framework in the iterated variant of the classic Chicken game.We compare a FP Q-learner ($L1Q$) and a level-2 Q-learner ($L2Q$) DM against a WoLF-PHC (Bowling and Veloso 2001), $L1Q$ and $L2Q$ adversaries, reporting rewards (averaged over the last 100 iterations and 10 random seeds) for the DM in Table 1. Note how the higher level DM achieves greater rewards and even exploits other kind of opponents. Details and code can be found at https://github.com/vicgalle/ARAMARL.

3 Conclusions

We have introduced TMDPs, a novel framework to support decision makers who confront adversaries that interfere with the reward generating process in RL settings. In addition, we propose a scheme to model adversarial behavior based on level-k reasoning about opponents. Further empirical evidence is provided via experiments, with encouraging results. Further work shall study the properties of higher order adversaries.

References

Bowling, M., and Veloso, M. 2001. Rational and convergent learning in stochastic games. In *Proceedings of the 17th international joint conference on Artificial intelligence-Volume 2*, 1021–1026. Morgan Kaufmann Publishers Inc.

Busoniu, L.; Babuska, R.; and De Schutter, B. 2010. Multi-agent reinforcement learning: An overview. 310:183–221.

Leike, J.; Martic, M.; Krakovna, V.; Ortega, P. A.; Everitt, T.; Lefrancq, A.; Orseau, L.; and Legg, S. 2017. AI safety gridworlds. *arXiv preprint arXiv:1711.09883*.

Stahl, D. O., and Wilson, P. W. 1994. Experimental evidence on players' models of other players. *Journal of economic behavior & organization* 25(3):309–327.