Logical Team Q-learning:

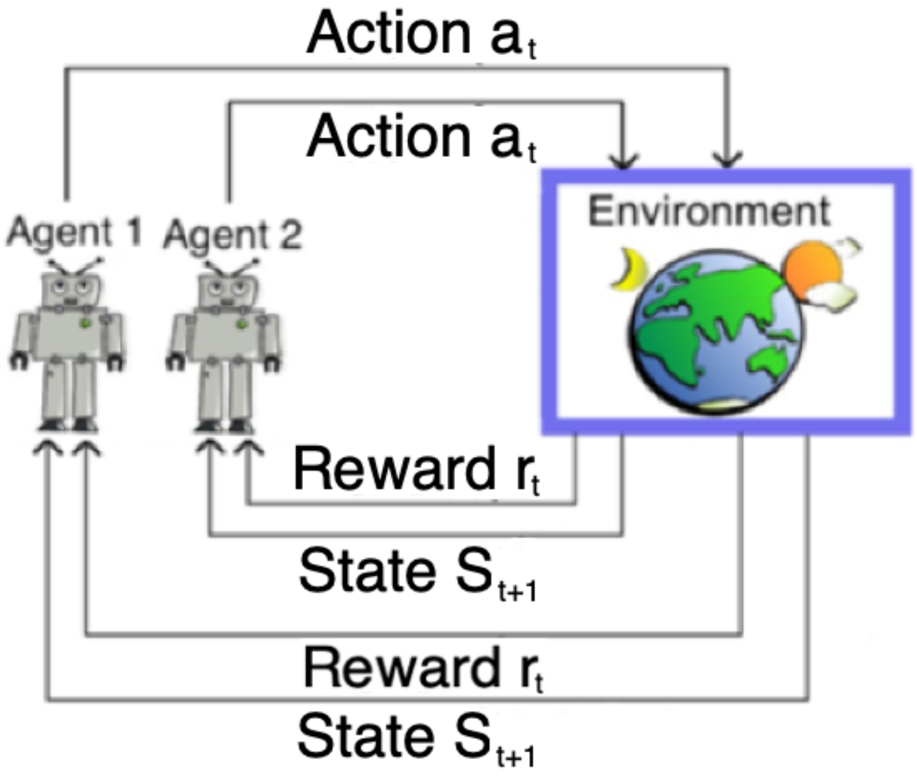
An approach towards factored policies in cooperative MARL

Lucas Cassano and Ali H. Sayed

MARL setting:

Problem: We address the challenge of learning factored policies in cooperative MARL scenarios.

Goal: To derive an algorithm that obtains factored policies that determine the individual behaviour of each agent so that the resulting joint policy is optimal.



Main idea:

Q-learning is derived as a stochastic approximation to a dynamic programming recursion that is provably convergent.

Can we take similar path to derive a MARL algorithm? In other words, our objective is to obtain a recursion that provably obtains optimal factored policies in the dynamic programming setting, and derive a MARL algorithm as a stochastic approximation to such procedure.

Contribution: We answer this question in the affirmative and derive such algorithms.

Dynamic programming case:

$$q^\dagger(s,a) = r(s,a) + \gamma \mathbb{E} \max_{a'} q^\dagger(s',a') \longrightarrow \text{Q-learning}$$

Dynamic programming case:

$$q^{\dagger}(s,a) = r(s,a) + \gamma \mathbb{E} \max_{a'} q^{\dagger}(s',a') \longrightarrow \text{Q-learning}$$

$$q^{k,\star}(s,a^k) = r(s,a^k,a^{-k}) + \gamma \mathbb{E} \max_{a'} q^{k,\star}(\boldsymbol{s'},a')\Big|_{a^n=\arg\max_{a^n} q^{n,\star}(s,a^n)\ \forall n\neq k} \qquad \forall\ \ k \in [1,K] \qquad (1)$$

$$\max_{a^k} q^{k,\star}(s,a^k) = \max_{a^1,\cdots,a^K} \left[r(s,a^1,\cdots,a^K) + \gamma \mathbb{E} \max_{a',k} q^{k,\star}(\boldsymbol{s'}, a'^{,k})\right] \qquad \forall\ \ k \in [1,K] \qquad (2)$$

Dynamic programming case:

$$q^{\dagger}(s,a) = r(s,a) + \gamma \mathbb{E} \max_{a'} q^{\dagger}(s',a')$$ → Q-learning

$$q^{k,\star}(s,a^k) = r(s,a^k,a^{-k}) + \gamma \mathbb{E} \max_{a'} q^{k,\star}(\boldsymbol{s}',a')\Big|_{a^n=\arg\max_{a^n} q^{n,\star}(s,a^n)\ \forall n\neq k} \quad \forall\ k\in[1,K] \tag{1}$$

$$\max_{a^k} q^{k,\star}(s,a^k) = \max_{a^1,\cdots,a^K}\left[r(s,a^1,\cdots,a^K) + \gamma \mathbb{E} \max_{a',k} q^{k,\star}(\boldsymbol{s}',a'^{,k})\right] \quad \forall\ k\in[1,K] \tag{2}$$

→ Bellman Team Optimality Operator (BTOO)

The paper includes a theorem that states that the BTOO can be used to obtain the desired factored q-functions with high probability.

Logical Team Q-learning:

Initialize: an empty replay buffer $\mathcal{R}$ and estimates $\widehat{q}_B^k$ and $\widehat{q}_U^k$.

for iterations $e = 0, \ldots, E$ **do**
  Sample T transitions $(s, \bar{a}, r, s')$ by following some behavior policy and store them in $\mathcal{R}$.
  for iterations $i = 0, \ldots, I$ **do**
    Sample a transition $(s, \bar{a}, r, s')$ from $\mathcal{R}$.
    for agent $k = 1, \cdots, K$ **do**
      if $a^n = \arg\max_{a^n} \widehat{q}_B^n(s, a^n)\ \forall n \neq k$ **then**
        $\widehat{q}_B^k(s, a^k) = \widehat{q}_B^k(s, a^k) + \mu\big(r + \max_a \widehat{q}_U^k(s', a) - \widehat{q}_B^k(s, a^k)\big)$
        $\widehat{q}_U^k(s, a^k) = \widehat{q}_U^k(s, a^k) + \mu\big(r + \max_a \widehat{q}_U^k(s', a) - \widehat{q}_U^k(s, a^k)\big)$
      end if
      if $\big(r + \max_a \widehat{q}_U^k(s', a) > \widehat{q}_B^k(s, a^k)\big)$ **then**
        $\widehat{q}_B^k(s, a^k) = \widehat{q}_B^k(s, a^k) + \mu\alpha\big(r + \max_a \widehat{q}_U^k(s', a) - \widehat{q}_B^k(s, a^k)\big)$
      end if
    end for
  end for
end for

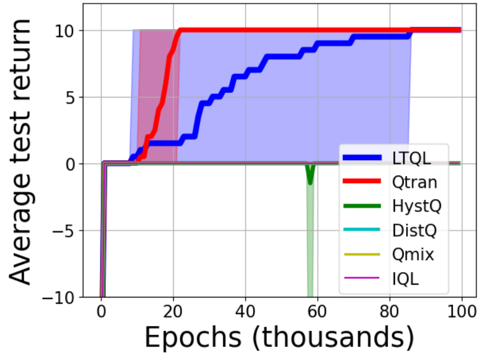
Experiments:

- A simple matrix game

		Agent 2		
		a_1	a_2	a_3
Agent 1	b_1	0	2	0
	b_2	0	1	2

$\longrightarrow q^{1,\star}(a^1) = [2, 1] \quad q^{2,\star}(a^2) = [0, 2, 0]$

- A finite state dec-POMDP



- A challenging predator-prey game