Multi-Agent Learning from Learners

Mine Melodi Caliskan, Francesco Chini, Setareh Maghsudi

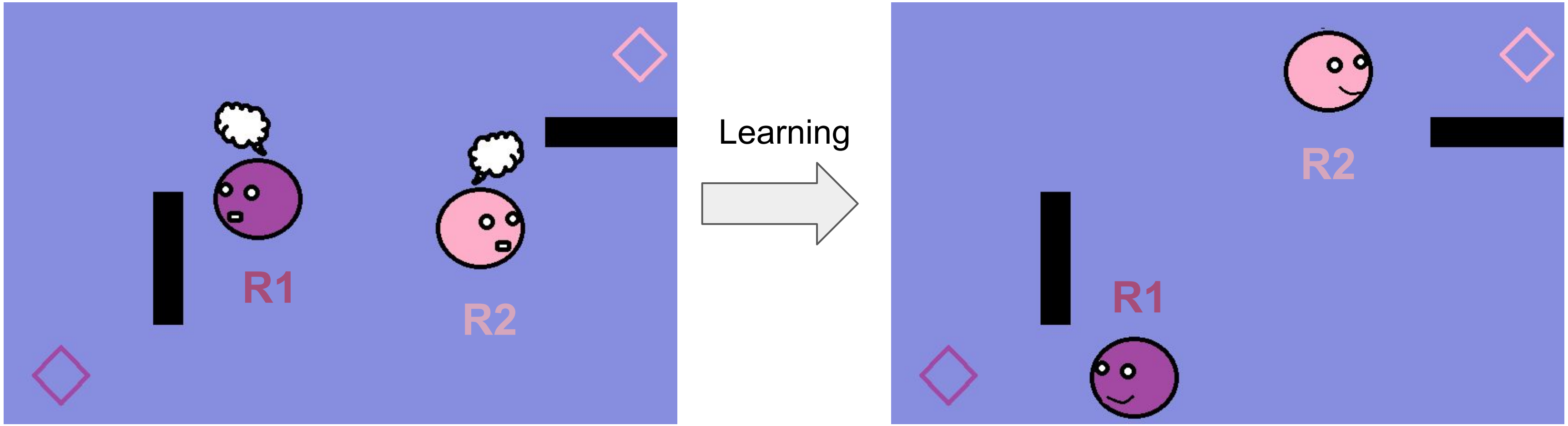
Cyber Valley

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Introduction

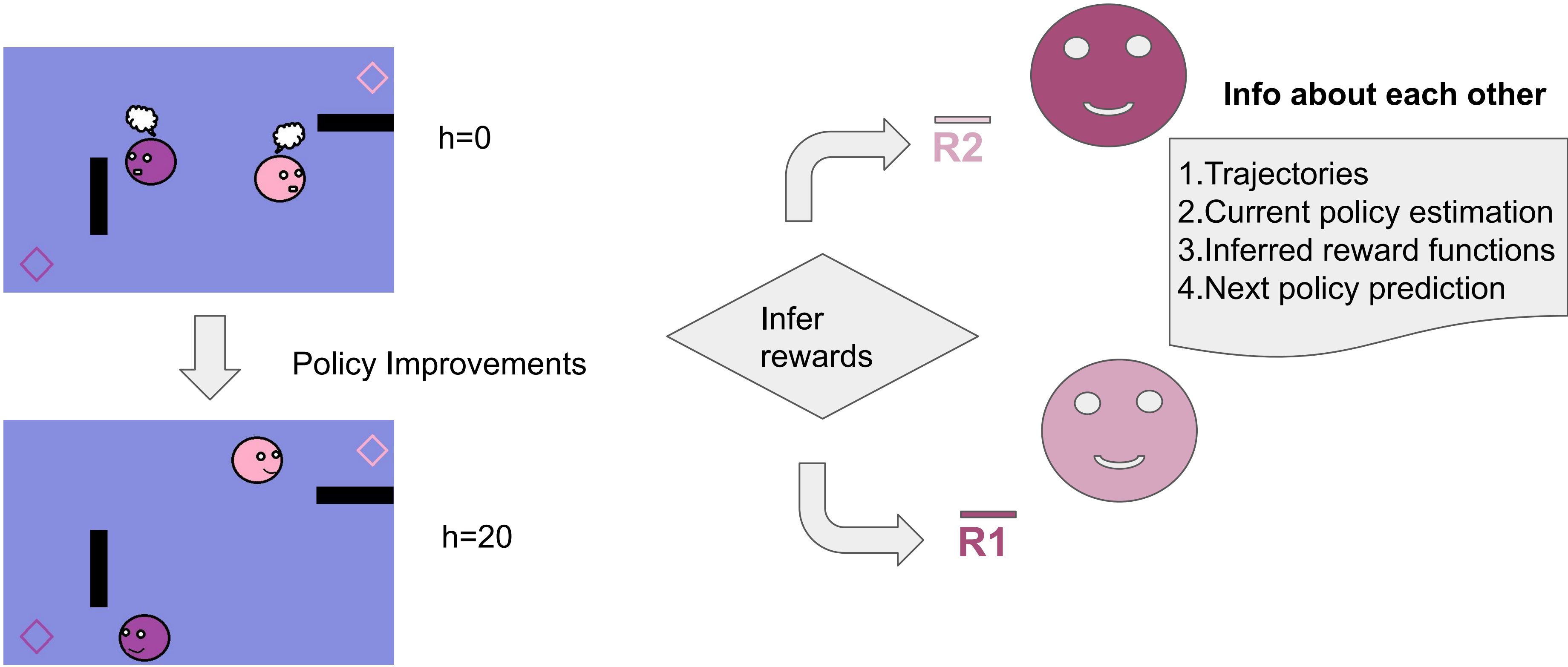
We study the "Learning from a Learner" problem in multi-agent setting

➔ **Goal:** Infer the reward functions of other agents that you interact with who are **not experts** but are **still learning**



Jacq, A., Geist, M., Paiva, A., and Pietquin, O. Learning from a learner. In International Conference on Machine Learning, pp. 2990–2999. PMLR, 2019

Introduction



Introduction

Potential applications:

- **Autonomous cars:** Cars from different companies might have different reward functions e.g safety or energy efficiency, shared environment and no equilibrium

➔ Predict behaviour using recovered reward functions

- **Fairness:** The agents might use the information about other agents' rewards in order to learn altruistic behaviours
- **Decentralization:** Use the information about the reward function to decentralize MARL algorithms that requires reward information e.g. Nash Q-learning

Problem Setting

- N agents acting together in the same environment
- Each agent i is trying to maximize its own reward R_i (general-sum)
- Agents can only observe the state s the actions $a_1, \ldots, a_N$ performed by the other agents and their own reward $R^i(s, a_1, \ldots, a_N)$

➔ Assume agents are optimizing entropy-regularized objective (individually):

$$\mathcal{J}(\pi^i) = \mathbb{E}_{\pi^i, \boldsymbol{\pi}^{-i}} \left[\sum_{t \geq 0} \gamma^t \left(R^i(s_t, \mathbf{a}_t) + \alpha \mathcal{H}\left(\pi^i(\cdot | s_t) \right) \right) \right]$$

Modeling other agents while optimizing your own policy

1. **Policy Improvements:** Multi-Agent Soft Policy Iteration (MA-SPI)

- Evaluate

$$\widetilde{Q}^{\pi^i}_{\mathrm{soft}}(s, a^i) = \tilde{R}^i\left(s, a^i\right) + \gamma \mathop{\mathbb{E}}_{\boldsymbol{\pi}}\left[\widetilde{Q}^{\pi^i}_{\mathrm{soft}}(s', a^i_{\mathrm{new}}) + \alpha \mathcal{H}\left(\pi^i\left(\cdot|s'\right)\right)\right]$$

- Improve

$$\pi^i_{\mathrm{new}}(a^i|s) \propto \exp\left(\frac{1}{\alpha}\widetilde{Q}^{\pi^i}_{\mathrm{soft}}(s, a^i)\right)$$

Modeling other agents while optimizing your own policy

2. Recovering Reward Functions:

- Estimate policies of the other agents from trajectories

$$\begin{tikzcd}
\boldsymbol{\pi}_0 \arrow[r, "SPI"] \arrow[d, dashed] & \boldsymbol{\pi}_1 \arrow[r, "SPI"] \arrow[d, dashed] & \dots \arrow[r, "SPI"] & \boldsymbol{\pi}_H \arrow[d, dashed] \\
\mathcal{D}_0 & \mathcal{D}_1 & \dots & \mathcal{D}_H
\end{tikzcd}$$

- Infer reward functions

$$\mathop{\mathbb{E}}_{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i}}\left[\overline{R^i}(s,\boldsymbol{a}^{-i},a^i)\right] = \alpha \ln \pi^i_{\text{new}}(a^i|s) + \alpha\gamma \mathop{\mathbb{E}}_{\substack{\boldsymbol{a}^{-i}\sim\boldsymbol{\pi}^{-i} \\ s'\sim P(\cdot|s,\boldsymbol{a}^{-i},a^i)}}\left[D_{\text{KL}}(\pi^i(\cdot|s')|\pi^i_{\text{new}}(\cdot|s'))\right]$$

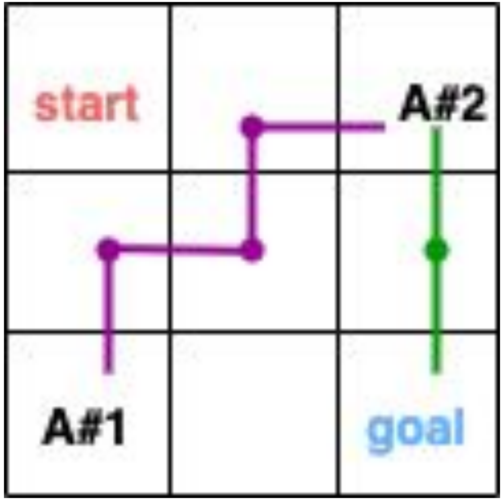
Error Bounds

In the paper, we provide error bounds

- For the reward recovery in terms of policy estimations
- For the predicted policy improvement in terms of recovered rewards

➔ These are novel contributions even in the single-agent case.

Experiments



$$M^i_{\text{hom}} = -d(\text{agent}_i, \text{goal}) + d(\text{agent}_i, \text{agent}_j) \text{ for } i = 1, 2$$

$$M^i_{\text{het}} = \begin{cases} -d(\text{agent}_i, \text{goal}) - d(\text{agent}_i, \text{agent}_j) & i = 1 \\ -d(\text{agent}_i, \text{goal}) + d(\text{agent}_i, \text{agent}_j) & i = 2 \end{cases}$$

Results

Metric	M_{hom}	M_{het}
PCC #1	0.48 ± 0.06	0.45 ± 0.04
PCC #2	0.59 ± 0.02	0.42 ± 0.02
$\hat{P}$	$\mathbf{0.54} \pm 0.03$	$\mathbf{0.44} \pm 0.01$
SCC #1	0.44 ± 0.14	0.51 ± 0.02
SCC #2	0.60 ± 0.04	0.43 ± 0.03
$\hat{S}$	$\mathbf{0.52} \pm 0.06$	$\mathbf{0.47} \pm 0.01$

Wed 26 Jul 2 p.m. HST — 3:30 p.m. HST
Exhibit Hall 1 #606