Reputation-driven Decision-making in Networks of Stochastic Agents

David Maoujoud† and Gavin Rens†

†Declaratieve Talen en Artificiële Intelligentie, Katholieke Universiteit Leuven

Introduction

We propose a Markov Decision Process-derived framework, called RepNet-MDP, tailored to multi-agent domains in which agent reputation is a key driver of the interactions between agents. We address the mathematical inconsistencies and alleviate the intractability of RepNet-POMDP, a framework developed by *Rens et al.* [1] in 2018. We furthermore use an online learning algorithm for finding approximate solutions to RepNet-MDPs. We perform early experiments to showcase the feasibility of our approach. RepNet agents are shown to be able to adapt their own behavior to the past behavior and reliability of the remaining agents of the network. Finally, our work identifies a limitation of the framework in its current formulation that prevents its agents from learning in circumstances in which they are not a primary actor.

[1] Gavin Rens, Abhaya Nayak, and Thomas Meyer. "Maximizing Expected Impact in an Agent Reputation Network - Technical Report". In: *CoRR* abs/1805.05230 (2018). arXiv: 1805.05230.

The RepNet-MDP basics

- Multi-agent framework.
- The agents are assumed to be selfish.
- The interactions between agents are driven by their reputation.

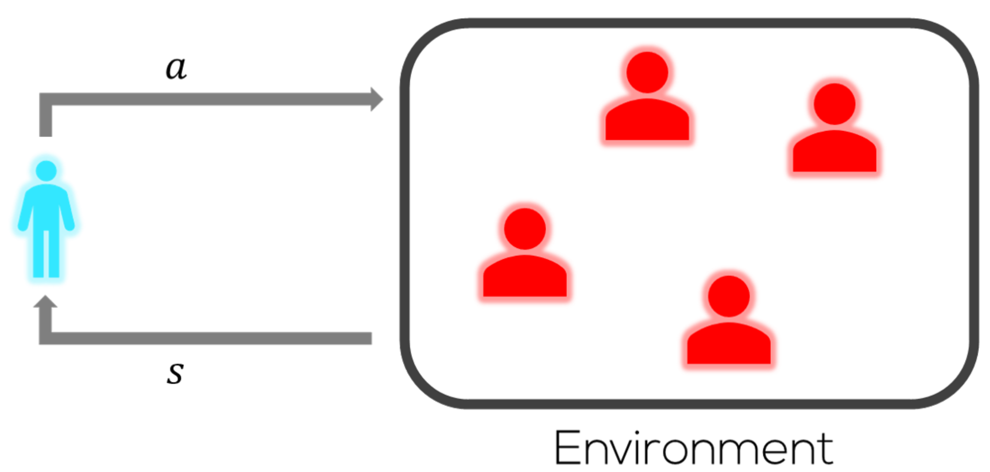

Fig. 1: RepNet agent interacting with the environment, which includes the other agents.

The RepNet-MDP loop

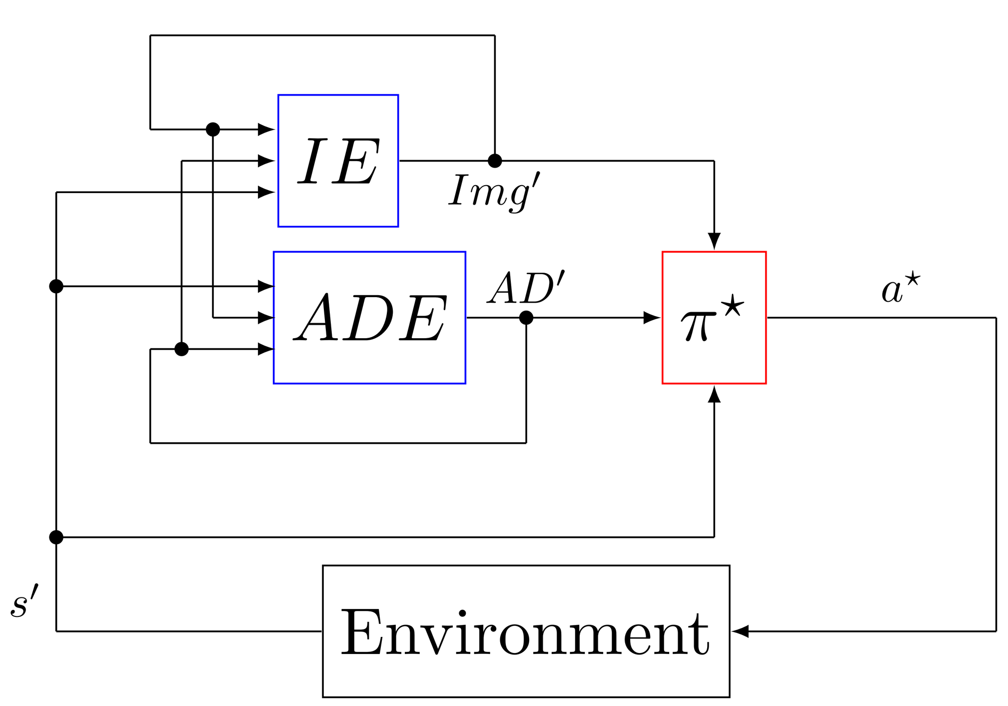

Fig. 2: The RepNet-MDP loop.

- s is the state of the environment.
- AD is called the *action distribution* of the RepNet agent. $AD(h, s)$ returns a probability distribution over actions in $\mathcal{A}$ for agent h in state s, according to the RepNet agent.
- ADE is called the *action distribution estimation* function. It updates the *action distribution* AD at each time-step.
- Img is called the *image function* of the RepNet agent. $Img(h, i)$ returns the image agent i has of agent h according to the RepNet agent.
- IE is called the *image estimation* function. It updates the *image* Img at each time-step.
- $\pi^\star$ is the policy followed by the RepNet agent, such that $a^\star = \pi^\star(s, AD, Img)$.

Reputation = Summary of the Image

The reputation of an agent h, according to agent g, is defined as:

$$REP_g(h, Img_g) := \frac{1}{|\mathcal{G}'|}\sum_{i\in\mathcal{G}'} Img_g(h,i) \times Img_g(i,g),$$

Similarities between MDPs and RepNet-MDPs

Standard MDP Bellman equations:

$$\begin{cases} V^\star(s,d) := \max_{a\in\mathcal{A}}\left\{\mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{T}(s,a,s')V^\star(s',d-1)\right\} & d>1 \\ V^\star(s,1) := \max_{a\in\mathcal{A}}\left\{\mathcal{R}(s,a)\right\} \end{cases}$$

RepNet-MDP Bellman equations ($\theta_g = \langle s, AD_g, Img_g\rangle$):

$$\begin{cases} V_g(\theta_g,k) := \max_{a\in\mathcal{A}}\left\{PI_g(s,AD_g,a) + \gamma \sum_{s'\in\mathcal{S}} T_g(g,s,a,s',r_g)V_g(\theta_g',k-1)\right\} & k>1 \\ V_g(\theta_g,1) := \max_{a\in\mathcal{A}}\left\{PI_g(s,AD_g,a)\right\} \end{cases}$$

$PI_g(s, AD_g, a)$ is the *immediate reward* of RepNet-MDPs.

$PI_g(s, AD_g, a) =$ "Self-impact + impact on the network"

Objective and subjective transition models

- *Objective Transition Model* OT → Describes the actual rules of the environment.
- *Subjective Transition Model* ST → Used by the RepNet agent to reason about the best course of action.

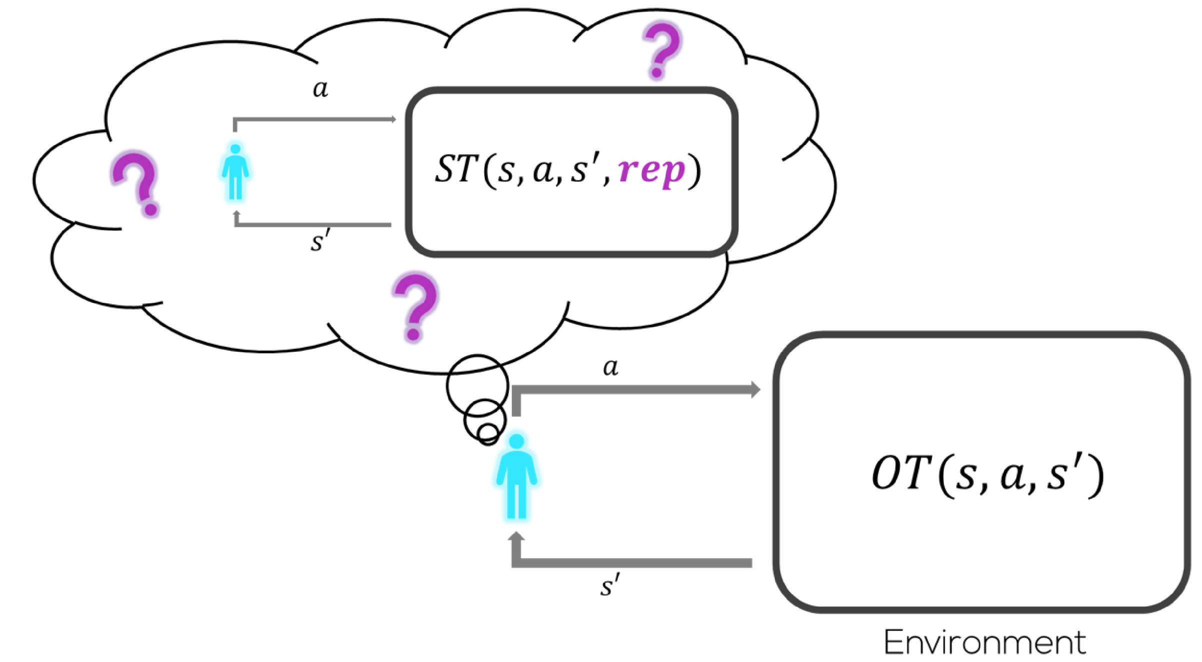

Fig. 3: Objective and subjective transition models.

Experiment: Trading between 3 agents

Agent A = RepNet Agent

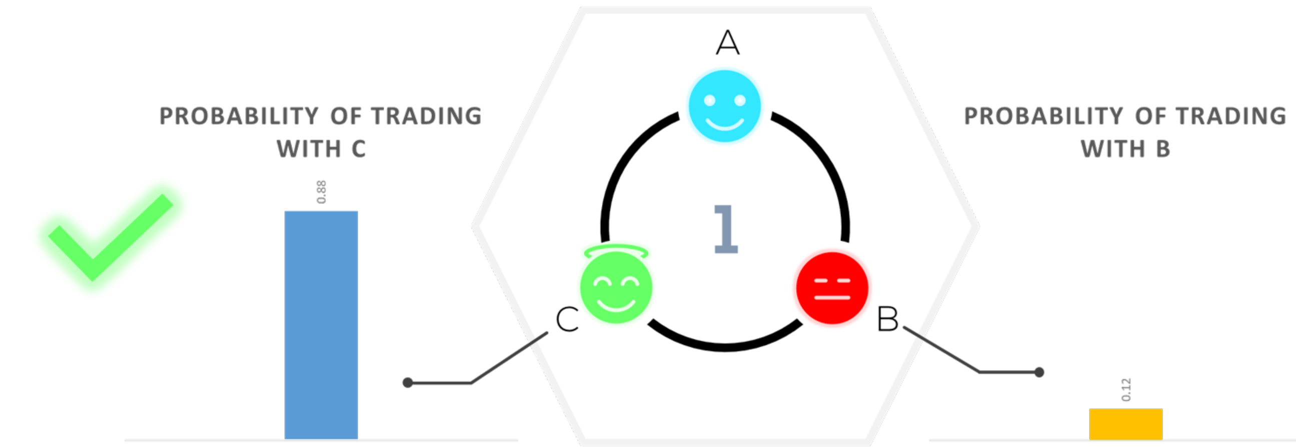

Fig. 4: Agent B rejects all trade offers, Agent C accepts all trade offers.

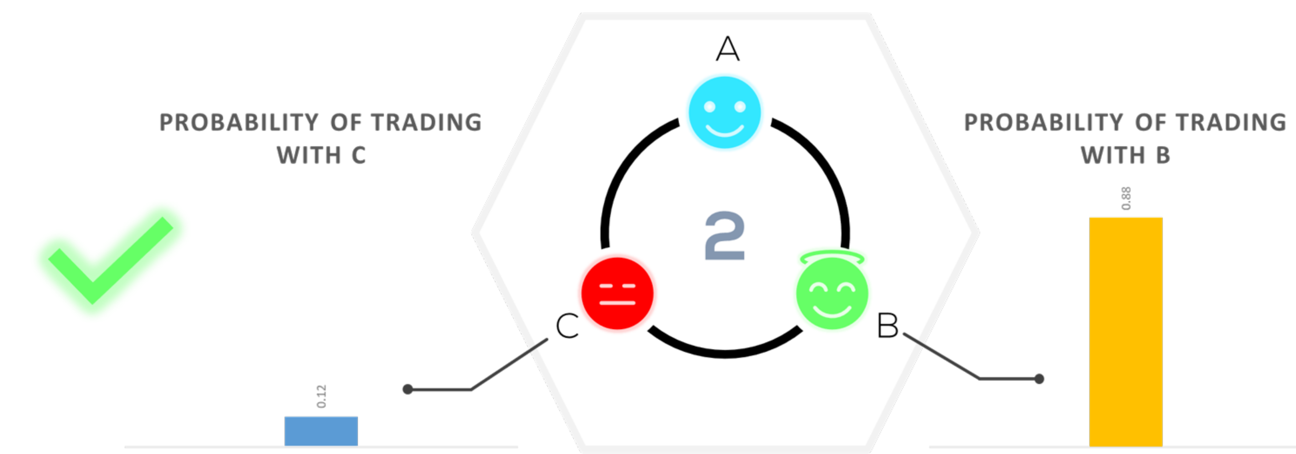

Fig. 5: Agent B accepts all trade offers, Agent C rejects all trade offers.

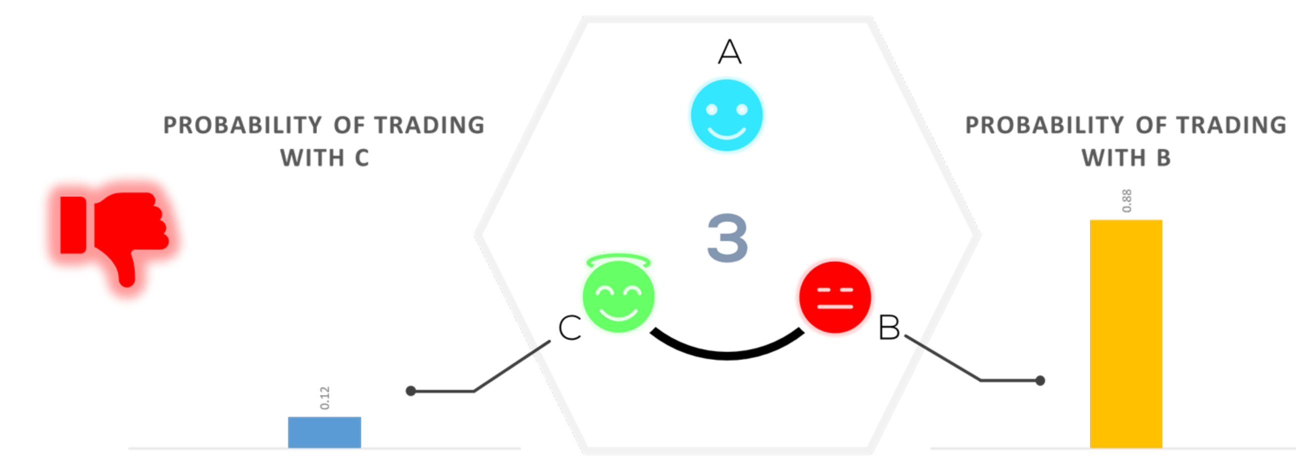

Fig. 6: Agent B rejects all trade offers, Agent C accepts all trade offers. Agent A is just an Observer.