From Misunderstanding to Cooperation: Understanding and Expressing Intentions Through Non-Verbal Actions

# # #

ABSTRACT

Solving situations of misunderstanding requires two abilities: to build a coherent model of others in order to understand them, and to build a model of "me" perceived by others in order to be understood. Having an image of me seen by others requires two recursive orders of modeling, known in psychology as first and second orders of theory of mind. It becomes especially difficult to find an understanding when agents dont have a common language to communicate and have to learn and share each others intentions through their behaviors. In this paper, we present a cognitive architecture based on both Reinforcement Learning and Inverse Reinforcement Learning that aims to reach mutual understanding in multi-agent scenarios. We study different conditions of empathy or gratitude that lead to cooperation in prisoner's dilemma.

CCS Concepts

•**Computing methodologies** → **Multi-agent systems;**

Keywords

AAMAS proceedings, LaTeX, text tagging

1. INTRODUCTION

2. MUTUAL MODELING

2.1 Non-recursive approach

Many ToM-based architecture have been developed in order to study multi-agent behaviors in social games. But so far, most of these applications where limited to first order of modeling (agents does not take into account how they are modeled by others) while higher orders leads to better performances in a range of simple social games (cite). However, 2nd order (an agent has a model of itself viewed by others) seams sufficient to generate rich social behaviors (cite david). Higher orders (an agent has a model of its own theory of mind imagined by other agents), although they outperform 2nd order in some cases (especially fourth cite Mod) do not seams to bring important advantages (cite nego).

We introduce a cognitive architecture enabling second order of modeling. In contrast with previous approaches (cite all highers), this one is not recursive. In recursive modeling (cite rigorous), an agent re-use its own architecture (that allows theory of mind) to model other agents. This would lead to an infinite loop of mutual modeling, known as infinite regress in epistemic logic (clark). In mutli-agent framework, recursions need to be stopped at a given depth. Such approaches have limits: it is difficult to process in parallel reasoning of all agents, it becomes heavy in computation beyond second order of modeling, and different agents or their images (perceived by others) may have different reasoning and may adopt different behaviors facing similar observations.

In our non-recursive approach, one agent has three models: a model of itself, a model of others and a model of itself seen by others. None of these models are performing theory of mind. At any instant, the agent updates its three models given his observations and use them to make predictions and decisions.

2.2 Model of itself

An agent i models itself as a RL agent: at a time t, it chooses an action a_i^t. Depending on this decision and all other agent's decisions $\{a_j^t\}_{j\neq i}$, it receives an observation $o_i^t = O(a_1^t, a_2^t, ..a_n^t)$ (n being the number of agents) and a reward that only depend on this observation $r_i^t = R_i(o^t)$. Each agent has its own reward function that is unknown by other agents.

As in (ref sequira14), this framework is simplified by the agent as a *Markovian Decision Process* (ref) (MDP) where the observations are assumed to be states that just depend on its previous observation and action following an unknown probability distribution:

$$o_i^t = O(a_1^t, a_2^t, ..a_n^t) \sim \mathbb{P}[o_i^t | a_i^t, o_i^{t-1}]$$

Hence, at the beginning, the decision making of the agent is performed by Q-learning (ref). Given the observation o^t, the agent learns the best new action a^{t+1} in order to maximize its future rewards (ref to algo 1).

algo 1: QL

2.3 Model of others

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

At the same time, it receives actions and observations of other agents $\{a_j^t\}_{j\neq i}$ and $\{o_j^t\}_{j\neq i}$. Given this information, it can infer their reward functions $\{R_j\}_{j\neq i}$ by IRL. It this setup, the IRL must be performed on-line. In (ref onlineIRL), Jin and *al* provide an incremental algorithm for on-line IRL in a MDP framework. This method is efficient but not so intuitive. As our final goal is to develop agents that could interact with humans, we want to adopt a less efficient but more intuitive approach that looks like how any human (or even a child) would infer the intentions of others. And maybe the simplest way is the following:

If I liked I repeat, otherwise I change: Supposing agent j receives an observation *"light"* and chooses action *"press-the-button"*. Then it receives new observation *"glass-of-juice"*. If it liked the juice, probably the next time it will observe the signal *"light"* it would again choose action *"press-the-button"*. Otherwise it would choose another possible action.

In order to formalize this approach, we denote as $\hat{r}_{i:j}^t = \hat{R}_{i:j}(o_j^t)$ the reward of agent j at time t inferred by agent i. Agent i memorizes, for each possible observation o_j of agent j, the last action $A_{i:j}(o_j)$ it chose facing o_j. Agent i also memorizes, for each observation o_j, the next observation $O_{i:j}(o_j)$ perceived as a consequence of choosing action $A_{i:j}(o_j)$. If at time t, agent j observes o_j^t and chooses once again the action $a_j^t = A_{i:j}(o_j^t)$, it means agent j "liked" the previous consequence of this choice, says $O_{i:j}(o_j^t)$. In that case, agent i increments its inferred reward function $\hat{R}_{i:j}(o_j^t)$ for agent j as follow:

$$\hat{R}_{i:j}(o_j^t) \leftarrow (1-\frac{1}{n}).\hat{R}_{i:j}(o_j^t) + \frac{1}{n}$$

Where n is the number of times agent i observed agent j choosing $A_{i:j}(o_j^t)$ while observing o_j^t. Contrariwise, if it chooses a different action $a_j^t \neq A_{i:j}(o_j^t)$, agent i decrements the estimated reward function $\hat{R}_{i:j}(o_j^t)$ for agent j:

$$\hat{R}_{i:j}(o_j^t) \leftarrow (1-\frac{1}{n}).\hat{R}_{i:j}(o_j^t) - \frac{1}{n}$$

Then, given the inferred reward functions, an agent can predict the next action of other agents. Such a prediction can be used to adapt its own next decision in consequence, and also to evaluate how it is able to model other agents. This intuitive IRL process is summarized in (ref to algo2).

algo 2: intuitive IRL

2.4 Model of itself seen by others

In order to model itself perceived by other agents, an agent processes exactly the same way used to model other agents. Thus, it infers its own reward function R_i given its previous actions and observations in order to estimate how other agents would infer its reward function. In the following sections, we denote as $\hat{R}_{i:(j:i)}$ this estimated function. As in (ref section 2.3), agent i uses its memories of its previous choices of action $A_{i:(j:i)}(o_i)$ and consequences $O_{i:(j:i)}(o_i)$ observed by another agent j for all possible observations o_i in order to update $\hat{R}_{i:(j:i)}$. Note that if all agents are aware of all the true observations of others and have the same initial estimations of others rewards (for instance, $R^0_{i:(j:i)}(o_i) = R^0_{j:i}(o_j) = 0\ \forall i, j, o_i, o_j$), we then have the equality:

$$\hat{R}^t_{i:(j:i)} = \hat{R}^t_{j:i} \quad \forall t$$

3. EXPRESSING INTENTIONS

At this moment, our agents are just behaving in an "egoist" way, trying to maximize their own rewards and optionally they model others and themselves seen by others. But in order to promote cooperation, we provide any agent with the possibility to help other agent to infer its reward function. In that purpose, an agent can, each time it has not the expected observation and reward, move next time to another action even if the last one was in average the optimal choice. In other words, imagine the agent has learned that when it sees an observation *"light"* the optimal action is *"press-the-button"* and leads with probability 0.5 to *"glass-of-wine"* associated with positive reward ($R(wine) = 1$) and with probability 0.5 to *"electric-chock"* that is associated with negative reward ($R(chock) = -0.9$), while another action *"do-nothing"* always leads to *"nothing"* associated with a null reward ($R(nothing) = 0$). In that case, a RL-based behavior would always chose action "press-the-button" that leads in average to a positive reward ($\hat{r} = 0.1$). But another agent observing this behavior, with no additive information, would infer that both *"glass-of-wine"* and *"electric-chock"* are associated with positive rewards while *"nothing"* is associated with negative (or null) reward. Now, if the agent, each time it receives the electric chock, do nothing the very next time it sees the light, it becomes possible for an observer to guess it does not like the chock but tried the action because it wanted the glass of wine.

Formally, the agent is using its model of itself seen by others: when agent i is perceiving o_i, it looks at the true reward associated with the last consequence $O_{i:(j:i)}(o_i)$:

$$r = R_i\left(O_{i:(j:i)}(o_i)\right)$$

if this reward was acceptable (*e.g.* superior to a fixed threshold) the agent repeat the last action it did after observing o_i, hence $A_{i:(j:i)}(o_i)$. Otherwise, it chooses the best of the remaining actions (according to Q-values).

That way we enable agents to help each others in inferring their reward functions. Now our agents have the choice between two possible behaviors: the classical Q-learning or this expressing-intentions behavior (described step by step in ref-to-algo3).

algo 3 : expressing-intentions behavior

4. EMPATHY AND GRATITUDE

We finally provide our agents with intrinsic rewards (ref intrinsic motivation) that depend on how they estimate the rewards of other agents. We define two different intrinsic rewards that an agent can feel observing the other ones:

Empathy $e^t_{i:j}$ of an agent i observing an agent j at a time t is proportional to its estimation of the reward that j received:

$$e^t_{i:j} \propto \hat{R}_{i:j}(o_j^t)$$

Gratitude $g^t_{i:(j:i)}$ of an agent i observing an agent j at a time t is proportional to its estimation of *how j would infer i's own reward*:

$$g^t_{i:(j:i)} \propto \hat{R}_{i:(j:i)}(o_i^t)$$

Our model of empathy is based on de Waal's *Action-Percept Model* framework (ref Waal). In this context, agents

have a common set of possible actions or observations. Then empathy describes the capacity to be affected by and share the emotional state of another (inferred through this common set of action-perception).

The intrinsic reward for gratitude is based on the idea that "*it's the thought that counts*", expression used to indicate that it is the kindness behind an act that matters, however imperfect or insignificant the act may be.

Now, at time t, as agent i observes a signal o_i^t, it receives a total reward $\mathfrak{r}_i^t$, that is sum of extrinsic ($R_i(o_i^t)$) and intrinsic rewards (empathy and gratitude):

$$\mathfrak{r}_j^t = R_i(o_i) + \sum_{j \neq i} \alpha_i \, e_{i:j}^t + \beta_i \, g_{i:(j:i)}^t$$

Where α and β are control parameters that are used to try different situations. For example, we can compare agents that only feel empathy ($\beta = 0$) or only gratitude ($\alpha = 0$). We can also explore negative values of α and β that could lead to aggressive behaviors.

5. PRISONER'S DILEMMA

The Prisoner's Dilemma (PD) is an ideal game to study the social behavior of our agents. In that context, we focus on a 2-agents system. Each agent has the choice between two actions *defect* or *cooperate*. If both agent choose to cooperate, they receive a reward $\mathcal{R}$. If they both defect, they receive a smaller reward $\mathcal{P}$. If one agent defect while the other cooperate, the agent that defected receives the highest reward $\mathcal{T}$ and the agent that cooperated receives the smallest reward $\mathcal{S}$. The generalized form of PD requires following conditions:

$$\mathcal{T} > \mathcal{R} > \mathcal{P} > \mathcal{S}$$

The payoff relationship $\mathcal{R} > \mathcal{P}$ implies that mutual cooperation is superior to mutual defection, while the payoff relationships $\mathcal{T} > \mathcal{R}$ and $\mathcal{P} > \mathcal{S}$ imply that defection is the dominant strategy for both agents. We implemented the iterated version of this game (IPD), where agents successively play this game and remember previous actions of their opponent. Classical RL-learning would always tend to the Nash equilibrium (sandholm1996multiagent) that consists in always choosing defection.

We implemented an IPD with payoff $\mathcal{T} = 1$, $\mathcal{R} = 0.6$, $\mathcal{P} = 0$, $\mathcal{S} = -1$. Table 5 displays the payoff matrix of this game. Each game last 1000 iterations. At each iteration t, agent i

	Cooperate	Defect
Cooperate	0.6, 0.6	-1, 1
Defect	1, -1	0, 0

Table 1: Prisoner's Dilemma payoff matrix

chooses action $a_i^t \in \{cooperate, defect\}$ and receives a signal $o_i^t \in \{\mathcal{O}_\mathcal{R}, \mathcal{O}_\mathcal{S}, \mathcal{O}_\mathcal{T}, \mathcal{O}_\mathcal{P}\}$ associated with the corresponding reward ($R_i(\mathcal{O}_\mathcal{S}) = \mathcal{S}$, etc). Agent i also receive the action and the observation of the other agent, a_j^t and o_j^t. But agent are not aware of the payoff matrix that defines the reward of the other (in fact, it is the same).

The Q-learning behavior of agents was implemented with parameters $\gamma = 0.8$, $\eta = 0.05$ and the action was chosen using the Gibbs softmax method:

$$a \sim \mathbb{P}[a|o] = \frac{e^{\theta Q(o,a)}}{\sum_b e^{\theta Q(o,b)}}$$

With temperature parameter $\theta = 5$.

6. RESULTS AND DISCUSSION

6.1 Pure Q-learning

We first tried to let our agents behave without expressing intention and with no intrinsic reward for empathy or gratitude ($\alpha = \beta = 0$). As expected, agents quickly tend to the Nash equilibrium and always defect (ref to fig1). Each agent learned a wrong reward function for the other. (ref to table 2) shows the average resulting reward functions learned by the agents over 50 IPD game with 1000 iterations.

	$\mathcal{T}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{S}$
Truth	1	0.6	0	-1
$\hat{R}_{1:2}$	0.34	-0.13	0.90	-0.75
σ^2	0.39	0.52	0.16	0.13
$\hat{R}_{2:1}$	0.25	-0.16	0.92	-0.77
σ^2	0.36	0.46	0.15	0.094

Table 2: Average learned other's rewards functions by agents 1 and 2 over 50 IPD games and variances. We can see that with a small variance agents successfully learned that the other has negative reward $\mathcal{S}$, but as the other was always defecting at the end, both thought that the other had strong positive reward $\mathcal{P}$ that is, in fact, null.

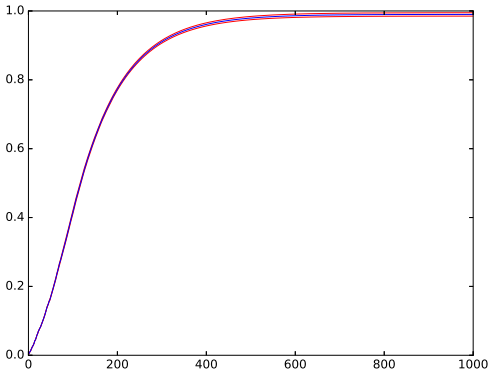

Figure 1: test

6.2 Q-learning with empathy & gratitude

6.3 Expressing intentions with empathy & gratitude

6.4 Playing with empathy

7. CONCLUSION

REFERENCES