Social-Task Learning for HRI

Anis Najar, Olivier Sigaud, and Mohamed Chetouani

Institut des Systèmes Intelligents et de Robotique
CNRS UMR 7222
Université Pierre et Marie Curie
Paris, France
{anis.najar,olivier.sigaud,mohamed.chetouani}@isir.upmc.fr

Abstract. In this paper, we introduce a novel method for learning simultaneously a task and the related social interaction. We present an architecture based on Learning Classifier Systems that simultaneously learns a model of social interaction and uses it to bootstrap task learning, while minimizing the number of interactions with the human. We validate our method in simulation and we prove the feasibility of our approach on a real robot.

Keywords: Human-Robot Interaction; Interactive Reinforcement Learning; Learning Classifier Systems

1 Introduction

Endowing robots with social interaction capacities in addition to task related skills is an important challenge that would facilitate the integration of robots in human environments for task collaboration whether in domestic or industrial contexts. Social interaction and task execution are two different aspects of Human-Robot collaboration that some recent works begin to point the necessity to distinguish [1]. However, these two aspects are still strongly mutually related, so it is not always possible to treat them independently from each other.

In our work, we are interested in studying the relationship between social interaction and task execution from a machine learning perspective: how can we use social interaction for task learning and inversely how could task learning contribute to learning social interaction? In the literature, these two questions are rarely addressed simultaneously, but often one aspect is determined in order to learn the other: social interaction mechanisms are predefined to learn a new task, or a known task is used to learn new social interaction mechanisms. In this paper, we propose to learn simultaneously a model of social interaction (Social Model) and a model of the task (Task Model) in a robot teaching scenario. In one way, the Task Model is used for grounding the interpretation of teaching signals and for learning how to behave according to them within the Social Model. In return, the Social Model is used for bootstrapping the learning process of the Task Model. This looped process is performed online while minimizing the number of interactions with the human.

In a previous work [6], we proposed a first model that learns a social reward function on the teaching signals from the rewards provided by the environment, and uses it to bootstrap task learning. One limitation of this model is that it does not take into account the long-term information provided by some teaching signals, so the Social Model is not able to learn anything about them. In this paper, we propose a model that overcomes this limitation by learning the state-action values instead of the direct rewards for grounding the meaning of teaching signals. We show that this model is able to boost task learning in addition to learning a more complete model of social interaction.

In the next section, we present some related work. In Section 3, we introduce our model. Section 4 describes the scenario and the experimental set-up. In Section 5, we provide a validation of our approach in simulation. Finally, we present an implementation of our model within a robotic architecture and we report the results of experiments performed on a real robot in Section 6.

2 Related work

Interactive Reinforcement Learning (IRL) provides a wide range of techniques for teaching RL-based systems [10] by the means of social feedbacks. These works differ in three main aspects: the interaction protocol used for providing feedbacks, the way feedbacks are interpreted for learning and the autonomy of the system with respect to the human.

Some works rely on artificial interfaces for interacting with the learning system. In [3], a virtual agent is trained by human feedbacks within a text-based environment. [9] and [12] use a clicking interface while [5] and [7] rely on push-buttons for providing human feedbacks. In contrast, other works rely on natural interaction protocols for delivering feedbacks such as spoken words [11] and speech features [2, 4]. Similarly, in our work we use a natural interaction protocol and we focus specially on non verbal cues such as head movements and pointing for providing feedback.

While most of these works associate human feedback with predetermined scalar values [3, 5, 7, 9, 11, 12], few works address the question of learning the meaning of teaching signals [2, 4]. In [4], a binary classification on prosodic features is performed offline before using it as a reward signal for task learning. In [2], however, the system learns simultaneously to interpret feedbacks and to perform the task. Our work, similarly to [2], tackles both questions at the same time by grounding the meaning of the teaching signals in the task and by using them in return to bootstrap task learning.

The autonomy of the learning agent with respect to the human is an important feature for evaluating Interactive Learning systems in terms of human load. In [2, 3, 7, 9], the learning agent is guided only by human feedbacks, so it is not able to learn without the presence of the human. By contrast, in our work like in [5, 11, 12], the system is able to learn autonomously through task related rewards while the human can choose the degree of its involvement in the interaction, for guiding the system in order to accelerate its learning process.

3 Model

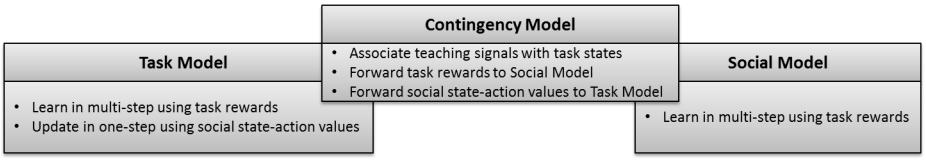


Fig. 1: Social-Task learning model

The idea of our method is to learn simultaneously two separate models, one for the task and one for social interaction. The model of the task would serve to ground the model of social interaction while minimizing the number of interactions with the human and the model of social interaction is used in order to bootstrap the learning of the task. We use an architecture based on three main components: a Task Model, a Social Model and a Contingency Model (Figure 1). The Task Model and the Social Model are represented by two different Markov Decision Processes (MDP) based on XCS[1] and the Contingency Model represents the contingency between states of both MDPs.[2]

In [6], we proposed a first model in which the Social Model is used for learning a social reward function on teaching signals based on task rewards. This function is then used online as an additional reward signal for boosting the learning process within the Task Model. We will refer to this first model as SRXCS (for Social Reward XCS). In this paper, we rather learn state-action values within the Social Model and use them as state-action values for the Task Model. We refer to this model as SVXCS (for Social Value XCS). From an algorithmic point of view, the difference between SVXCS and SRXCS resides in two points: the way the Social Model is updated by task rewards and the way the Task Model is updated by the Social Model. In SRXCS, the Social Model is updated by task rewards in a single-step fashion, while the Task Model is updated in a multi-step manner by using both social and task rewards. In SVXCS, however, the Social Model is learned in a multi-step way, whereas the Task Model is updated by the Social Model in a single-step fashion. In addition, in SVXCS, the Task Model still uses task rewards in multi-step as in SRXCS, so it is able to learn the task independently from the human.

[1] XCS is an RL system endowed with a generalization capability that allows learning general rule representations over state features, in a way that features that are not relevant for a given rule are replaced by a '#' symbol [8].

[2] We refer to [6] for a more detailed description of the model.

4 Scenario

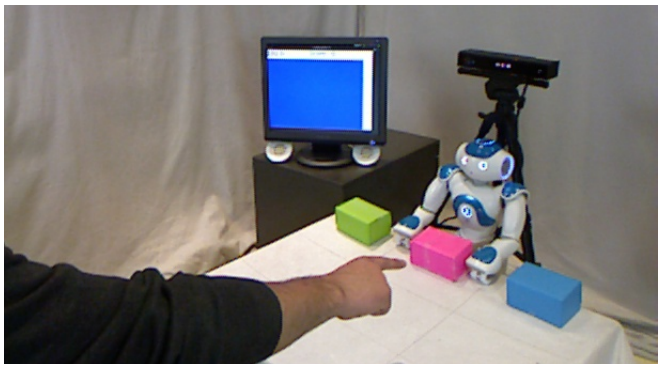


Fig. 2: Scenario: The robot must press the button corresponding to the information displayed on the screen. A human can help it through head movements and pointing.

We consider a simple task (Figure 2), in which a robot has to learn to press buttons of different colors, according to the information displayed on a screen. In a real-world scenario, the information displayed on the screen would represent the state of the physical environment and the action of pressing a button could represent any other elementary action that the robot could perform on objects.

4.1 Experimental setup

The experimental set-up is composed of a humanoid robot (Aldebaran Nao) facing a table on top of which there is a set of three buttons of different colors. At each moment, the screen displays the color of the button that the robot has to press. The robot is able to perform two kinds of actions: gazing to one of the different buttons or pressing the one it is facing. The task is a multi-step problem, meaning that in order to press the right button, the robot has to look for it first, and then to perform the action of pressing. The action of gazing to an object triggers a null reward. Pressing a button, however, triggers either a positive or a negative reward, represented by two different sounds, depending if the robot pressed the right or the wrong button. When the robot presses the right button, the task progresses and a new color is displayed on the screen.

While the robot is learning to perform the task, a human can sit in front of it in order to help it by using head nods, head shakes and pointing. Head nods and head shakes tell if the robot is looking to the right or the wrong button, while

pointing is meant to indicate the button it has to press. A Microsoft Kinect[3] V2 sensor is used to track the skeleton of the human[4].

4.2 Teaching protocol

We adopt an active learning procedure as a teaching protocol. When the robot encounters a new task situation (defined by the displayed color and the robot gazing state), the robot asks the human for help in two steps: First, it asks if it is looking to the right color, immediately looks at the person for a brief moment, before looking back to the button. Then, it repeats this process one more time by asking the person to point the right button.

This active learning procedure is motivated by two main reasons. First, the Contingency Model as it is currently designed stores the contingency between whole states. It means that all teaching features must be determined before sending the whole social state to the Contingency Model. So, this procedure fulfils this constraint, by actively asking the human to provide it with a value for each type of feature. Second, we have argued in [6] that in order to reach optimal performance, the human needs to interact with the system only in newly encountered situations, in the case of a perfect teacher. So, this protocol is meant to verify this assumption in a real set-up.

5 Model performance in simulation

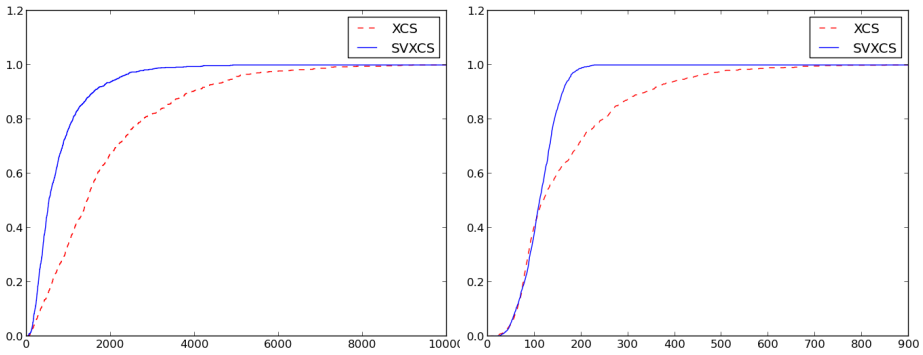
In this section, we present the performance of our model in simulation. We compare SVXCS to the standard XCS algorithm over 1000 experiments, to show how our model accelerates task learning. Then, we present the learned rules in the Social Model.

5.1 Task Model performance

We report the result of the experiments in two different settings: with and without genetic generalization. Figure 3 reports the probability for the model to converge before n steps. With genetic generalization (Figure 3.a), SVXCS needs at most 4903 iterations to learn the task, while the standard XCS needs at most 9184 iteration. Beyond this threshold, we are sure that the model converges. However, below this number of steps, the model converges only with a certain probability. It is worth noting that even with this gain, the number of steps is still considerable for a real-world scenario. Figure 3.b reports the performance of the models without genetic generalization. We can see that XCS converges in at most 851 steps, while SVXCS reduces this number to 227 which is more reasonable for a real robot.

[3] https://www.microsoft.com/en-us/kinectforwindows/, Last accessed 20-12-2014

[4] We use a modified version of the Kinect V2 client/server provided by the Personal Robotics Laboratory of Carnegie Mellon University. https://github.com/personalrobotics/, Last accessed 20-12-2014



(a) (b)

Fig. 3: Probability to converge before n steps. (a) With genetic generalization (b) Without genetic generalization.

5.2 Social Model

Table 1 shows the learned rules in the Social Model. We can see that the model found correct generalizations on the teaching signals. The first two lines correspond to the rules related to the action of pressing the button. They predict with maximum accuracy a reward of -1000 for pressing a button when there is a head shake and a reward of 1000 when there is a head nod, whatever the pointing information. The remaining rules correspond to the predicted values of gazing to the different objects. We can see that these rules represent a joint attention behaviour which leads the robot to gaze the button that the human is pointing, whatever the head movement information. It is worth noting that in RBXCS, we could not obtain this behaviour of joint attention because it is not able to take into account the long term information provided by pointing [6].

Table 1: Learned classifiers in the Social Model: the first two bits encode head movement information. The remaining bits represent the pointing information.

Condition	Action	Prediction	Rule meaning
#1###	0	-1000	head shake $->$ do not press the button
#0###	0	1000	head nod $->$ press the button
##0##	1	484	
##1##	1	698	pointing at button 1 $->$ gaze at button 1
###0#	2	485	
###1#	2	696	pointing at button 2 $->$ gaze at button 2
####0	3	485	
####1	3	694	pointing at button 3 $->$ gaze at button 3

6 Experiments on the real robot

In this section, we present our robotic architecture. Then, we report the experimental results on the real robot.

6.1 Robotic architecture

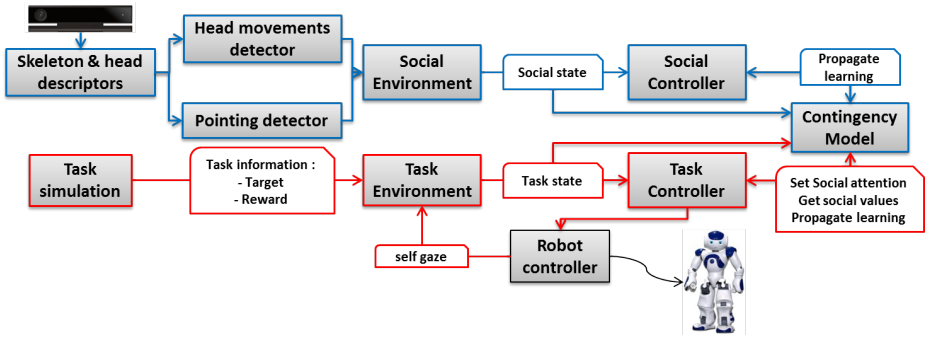


Fig. 4: Robotic architecture for Social-Task learning

To implement our model on the real robot, we developed a software architecture in ROS (Figure 4) including a set of modules for perception, decision making and control. The architecture is organized in two main layers, one related to the task (bottom) and another related to social interaction (top). The core of the architecture is composed of the Task Controller, the Social Controller and the Contingency module that implement the three components of our model. Task Environment and Social Environment modules encapsulate the representation of task and social states. In addition, we have a set of perception modules for detecting pointing and head movements and a module for controlling the robot.

6.2 Experimental results

Without genetic generalization, SVXCS converges within 227 steps (Figure 3.b). To validate these results, we performed experiments of 227 learning steps without genetic generalization with 10 different subjects.

Table 2 reports the results of these experiments (The sequence of the experiments has been changed in the table for better readability). These results show that when subjects provided correct teaching signals ($3-10$), the robot always succeeded in learning the task. Otherwise, the system failed at learning the task (1 and 2). In both experiments, the subjects sometimes performed a head nod instead of a head shake. In (1), a false positive detection of head nod has also

occurred. This resulted in an incoherent interpretation of head nods within the Social Model that hindered the Task Model from converging properly.

In other situations $(2-5)$, the system did not detect feedback from the user when the robot asked for. This was either because of a detection failure (5) or because the subjects were hesitant and did not provide a complete feedback $(2-4)$. In this case, the Contingency Model did not record any teaching signal for the corresponding situation; so the robot asked for feedback one more time for the same situation and this did not prevent from learning the task. In these experiments, the number of interactions with the robot was of 10, while in the other experiments the user was solicited only 9 times, which corresponds to the number of the different task states. Moreover, when all task situations have been explored, the subjects had no obligation to stay and the robot continued to learn autonomously. The longest interaction lasted for about 13 minutes, while the whole learning process lasted for about a half an hour.

To conclude, these experimental results validate our assumption that without genetic generalization, a perfect teacher interacting with the system only in new situations has the guarantee that the robot will learn the task within 227 steps.

Table 2: Experimental results: experiment id (ID), number of interactions with the subject (NI), number of detection failures (DF), number of times the subject did not provide feedback when needed (NF), number of times the system detected the wrong feedback (DE), number of times the subject provided wrong feedback (WF), time in minutes until the last interaction (IT), total duration of the experiment in minutes (TD), success of task learning (S).

ID	NI	DF	NF	DE	WF	IT	TD	S
1	9			1	1	6.20	31.30	No
2	10		1		2	8.15	32	No
3	10		1			7.50	30	Yes
4	10		1			12.30	31.30	Yes
5	10	1				5.30	31	Yes
6	9					9	31.30	Yes
7	9					12.45	31.30	Yes
8	9					6	31.30	Yes
9	9					5.15	31.30	Yes
10	9					7.30	31.30	Yes

7 Discussion

In this section, we discuss the limitations of our system and we propose some alternative solutions.

Teaching protocol: The active learning mechanism that we implemented presents many advantages. First, it is difficult to provide a rigorous definition of the

state of a person. Unlike physical objects, a person is proactive and his/her actions are extended over varying time windows. In addition, the limitation of perception devices may lead to false positive detection that could decrease system performance. So, actively asking for teaching signals one by one while gazing at the human for a determined duration makes it possible to control the acquisition of whole social states. Moreover, this active learning mechanism serves at engaging the person to prevent it from being passive or disengaging from the interaction.

However, this process is a burden for the human as it imposes a fixed protocol for the interaction. In addition, asking for teaching signals only for new situations makes not possible for the user to correct himself if he gives wrong teaching signals in a given situation. One possibility to make the interaction more natural would be then to define a contingency between state features. In this case, the human would be more free in the way he provides teaching signals and it would be more easier for him to correct wrong feedbacks.

Transparency: Another limitation of this teaching protocol is the lack of transparency. In fact, the human has no way to know if he has given wrong teaching signals to correct them or if the robot is not learning correctly. A solution for this would be to implement a mechanism asking for clarification whenever it detects incoherence within the Social Model or incoherence between the Task Model and the Social Model.

Social Model application: In the current definition of our model, the Social Model contributes to boosting task learning in only one way. It serves as an alternative space with reduced complexity that allows to learn state-action values more rapidly. So, the Social Model is employed only in a passive way through the interpretation of teaching signals but never for decision-making. However, the rules evolved by the Social Model could also be used for decision-making. For example, the joint attention mechanism could be useful for guiding the exploration strategy within the Task Model and so it could further optimize task learning.

8 Conclusion and future work

In this paper, we presented a model for learning simultaneously a task and a model of social interaction. We showed that our model SVXCS is able to accelerate task learning while minimizing the number of interactions with the human. We presented an implementation of our model within a robotic architecture and proved the feasibility of our approach on a real robot.

In future work, we propose to modify the Contingency Model by storing the contingency between state features instead of whole states in order to make the interaction more natural. We also intend to explore the possibility of using the Social Model for guiding the exploration strategy within the Task Model in order to accelerate task learning. Finally, we propose to enrich our model with additional actions like gazing to the human and asking for feedback to evolve more complex social behaviours.

9 Acknowledgments

This work is funded by the Romeo2 project.

References

1. L. J. Corrigan, C. Peters, G. Castellano, F. Papadopoulos, A. Jones, S. Bhargava, S. Janarthanam, H. Hastie, A. Deshmukh, and R. Aylett. Social-task engagement: Striking a balance between the robot and the task. In *Embodied Commun. Goals Intentions Workshop ICSR*, volume 13, pages 1–7, 2013.
2. J. Grizou, M. Lopes, and P.-Y. Oudeyer. Robot learning simultaneously a task and how to interpret human instructions. In *Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on*, pages 1–8, Aug 2013.
3. C. Isbell, C. R. Shelton, M. Kearns, S. Singh, and P. Stone. A social reinforcement learning agent. In *Proceedings of the fifth international conference on Autonomous agents*, pages 377–384. ACM, 2001.
4. E. Kim and B. Scassellati. Learning to refine behavior using prosodic feedback. In *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, pages 205–210, July 2007.
5. W. B. Knox and P. Stone. Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, May 2010.
6. A. Najar, O. Sigaud, and M. Chetouani. Socially Guided XCS: Using teaching signals to boost learning. In *GECCO'15 Companion*. ACM, 2015.
7. P. M. Pilarski, M. R. Dawson, T. Degris, F. Fahimi, J. P. Carey, and R. S. Sutton. Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–7. IEEE, 2011.
8. O. Sigaud and S. W. Wilson. Learning classifier systems: a survey. *Soft Computing*, 11(11):1065–1078, 2007.
9. H. B. Suay and S. Chernova. Effect of human guidance and state space size on interactive reinforcement learning. In *RO-MAN, 2011 IEEE*, pages 1–6. IEEE, 2011.
10. R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 1998.
11. A. C. Tenorio-Gonzalez, E. F. Morales, and L. Villaseñor-Pineda. Dynamic reward shaping: training a robot by voice. In *Advances in Artificial Intelligence–IBERAMIA 2010*, pages 483–492. Springer, 2010.
12. A. L. Thomaz and C. Breazeal. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *AAAI*, volume 6, pages 1000–1005, 2006.