Reinforcement Learning algorithms analysis and their performance in Human Robot Social Interaction

Alina Munir (9munir@informatik.uni-hamburg.de)

***Abstract*— A robot must learn by exploring its environment and how the environment rewards for its actions. In complex and huge environments, the number of possible actions become large hence making it very time consuming for the robot to explore completely. By letting a human teacher guide the robot in this phase optimises the exploration hence the goal of this paper is to study the effect of real-time human interaction in machine-learning algorithms in robots. We study two approaches; firstly, Interactive Reinforcement Learning (IRL) an approach to train robots by natural interaction by treating human rewards as explicit and secondly, combination of Reinforcement Learning and Supervised Progressively Autonomous Robot Competencies (SPARC) an approach which allows the human to fully control the robot and by treating rewards as implicit hence learns an action policy while maintaining human supervisory. According to findings of Qualitative and quantitative results SPARC manifests safer and faster learning while keeping the human input optional hence reducing the workload on human teacher.**

I. Introduction

In near future robots are expected to be an integrated participants of the society in complex environments that are previously meant for humans only [1] . Users of these robots will be non-expert population such as elderly, children and people from all sorts of backgrounds rather than just expert engineers and the usage would require individualistic functionality that cannot be anticipated at the time of designing of these intelligent agents [2], [3]. We can assume that users would not have enough knowledge or expertise in the field of robotics to change the coded algorithm of the robot hence, it is vital that robot learns from human in more natural and easy way known to humans e.g.: by speech dictation etc.

Machine learning will certainly play an important role in the development of robot assistants for human environments such as homes, schools, hospitals and offices. Several approaches have been tested, such as requiring a single label with greater uncertainty or determining which category would provide the best learning enhancement. Given the complexity and unpredictability of the environment and the tasks these robots are expected to perform; we expect a significant amount of training to be provided by (possibly unqualified) field personnel. Furthermore, for robots to play a long-term role in human life, we must expect the robot to learn new tasks and improve its skills on a regular basis. Conventional machine learning is often developed for use by specialists, and the user interface is often too complex for people not involved in the development process [4]. Many methods also suffer from practical problems: deep learning [5] depends on the availability of large datasets to form networks, while Reinforcement learning [6] requires extensive and expensive research to collect the data used for training.

In goal to customization of Robot's behaviour it is observed that, complex interfaces are undesirable, large data sets are not accessible and random exploration can lead to false robot actions. This creates two main challenges: how to teach the robot to the user and how to collect safe training experiences for the robot. The solution to both these problems is interactive machine learning [4], [7], [8].The human is part of the machine learning process so by providing labelling or guiding the agent during exploration, the human can guide the learning. Human guide can be of more benefit in the robot's learning and potentially saving it from dangerous situations or making undesired decisions [9], [10] but in this process of active learning a huge limitation of robot always being in control is faced regardless of fragility of the situation [11]. Hence, it is important to give the teaching role back to the human to avoid frustration and disappointment on human side. In Reinforcement Learning framework human is allowed to actively contribute to the reward of the actions made by the robot [12] and where humans can give feedback to the actions taken and hence decide to reward or not [12], [13].

In this paper we study a mechanism similar to Supervised Progressive Autonomous Robot Competencies (SPARC) [14], [15]. It combines interactive machine learning and supervised autonomy. SPARC utilizes Reinforcement Learning for action-policy learning but gives a complete control of the robot's actions to the human teacher. This can be an essential technique in vulnerable situations where agent's one wrong move is opting for huge losses hence involving the human presence in the loop ensures the correction, in case when suggestive move by agent is not acceptable.

The remaining of the paper is followed by section Approaches where we discuss both IR and SPARC in detail. In next section 'Experiment' we discuss the experimental setup, algorithms and working of both techniques and discuss the findings and results in later sections 'Results' and 'Discussion' we get critical about our own experiment and describe some possible improvements. Lastly we conclude our study in 'Conclusion'.

II. Approach

In this paper we study the comparison between SPARC approach [16] and an alternative method combining interactive machine learning and reinforcement learning: IRL [12]. As both learning methods are experimented in the same environment initially used by Thomaz and Breazeal.

A. *Interactive Reinforcement Learning (IRL):*

In IRL human teacher provides a positive or negative feedback on the robot's last executed action which is utilized along with the action-policy reward to get the next action. Three additional parameters are introduced to the standard algorithm: guidance, communication by the robot and an undo option.

Firstly, Guidance helps the teacher to direct the agent's attention to a specific object by allotting the reward value to that object. Secondly, Robot's gaze communication indicates the certainty or uncertainty by either looking at a particular object or in between different items respectively. This feature helps indicate the transparency of robot's internal state. Thirdly, to correct the effect of negative reward agent tries to cancel the previous action by performing the opposite of that if possible.

B. *Supervised Progressive Autonomous Robot Competencies (SPARC):*

SPARC is used to control the actions of the robot. The robot communicates all its intentions (i.e. the action it intends to perform) to its teacher. The teacher cannot intrude and let the robot perform the suggested action, or intrude and force the robot to perform an alternative action. This combination of suggestions and corrections gives the teacher full control over the actions performed by the robot. It also eliminates the need for rewards: instead of having to explicitly assign rewards by the human, a positive reward can be assigned directly to each action performed by the robot, since it was passively imposed or approved by the teacher.

Comparing IRL and SPARC:

- SPARC enables the full control over the actions performed by the robot unlike IRL. A human can stop a robot from causing harm to itself or others in complex social dynamics where consequences of minor mistakes could be of high-risk.
- Robot communication is different in both techniques, in IRL gaze is used to indicate the uncertainty of decision by robot while in SPARC intention-based communication takes place.
- Reward is allotted manually and explicitly to the robot by IRL for its every decision while in SPARC explicit reward is removed.
- IRL teacher can only guide the robot towards a subset of action space while SPARC teacher can command the whole action space that robot must execute.

Hypotheses:

Three hypotheses are tested in this study [16]:

- H1: Compared to IRL, SPARC can lead to higher performance, whilst being faster, requiring fewer inputs and less mental effort from the teacher and minimizing the number of errors during the teaching when used by non-experts.
- H2: SPARC can be used by experts to teach an action policy safely, quickly and efficiently.
- H3: Teachers prefer a method in which they can have more control over the robot's actions.

III. Experiment

A. *Experimental Setup*

In order to study above hypothesis, an experimental setup is developed using Java-based online simulation [12]. 'Sophie's World' is a setup based on State-Action Markov Decision Process where an agent has a limited number of objects and action to choose from in order to perform the task. Platform can be used for different tasks but for the purpose of this experiment, it is specified to perform 'baking the cake' in a 'kitchen Scenario' [16].

*1) **Sophie's World**:* Sophie's World is an interactive platform where an agent performs the task based on State-Action Markov Decision Process and gets the simultaneous feedback from the human teacher. The world W = {L, O, E, T} is a Set of k locations $L = \{l_1, l_2, ..., l_k\}$, set of n objects $O = \{o_1, o_2, ..., o_n\}$, set of set of legal states E from object specified configuration and a transition function $T : E = E * A$, where A is a fixed action set $A = \{MOVE-LEFT, MOVE-RIGHT, PICK-UP, PICK-DOWN, USE\}$.

An agent at any step can perform one of the predefined actions; either it can move left or right from its current location or pick up the object in its current location or drop down the object its holding or use the object in its possession on the object in its current location.Each action updates the transition function T of the world hence changing the world state E.

*2) **Kitchen Scenario**:* Paper [16] implemented the Sophie's World for a 'Kitchen Scenario' where the task of the agent was to bake a cake. Sophie is the agent which utilizes three locations (TABLE, SHELF, OVEN), five actions (MOVE-LEFT , MOVE-RIGHT , PICK-UP, PICK-DOWN, USE) and five objects (FLOUR, EGGS, SPOON, BOWL, TRAY). Each Object has some possible states e.g: 'SPOON' has only one possible object state while 'BOWL' has five states (EMPTY, FLOUR, EGGS, UNSTIRRED, STIRRED) and 'TRAY' has three states (EMPTY, BATTER, BAKED). At a particular time, agent can only perform one action on object for example when USE is manipulated on BOWL while the agent possesses FLOUR, the BOWL's state changes to 'FLOUR'. An 'UNSTIRRED' BOWL changes to 'STIRRED' when SPOON is USEd on the BOWL. State of TRAY is changed from 'BATTER' to 'BAKED' when PICK-DOWN is performed on TRAY in location 'OVEN'.

Initially the state S_0 starts from everything being on 'SHELF' and BOWL and TRAY being 'EMPTY'. There are several action that can contribute to the ultimate target because of the vast number of same-state transitions and the simplicity of state space hence an optimal set of six actions are defined to reach the goal fastest in following order:

1) PUT-DOWN BOWL on location 'TABLE'
2) USE an ingredient from FLOUR or EGGS on 'BOWL'
3) USE the other ingredient on 'BOWL'
4) USE SPOON on BOWL to change the state from 'UNSTIRRED' to 'STIRRED'
5) USE BOWL on TRAY to put the batter from bowl to tray
6) PUT-DOWN TRAY in location 'OVEN' to bake the cake

The Kitchen Scenario is a deterministic Markov Decision Process (MDP), which has a state, a set of actions, a transition function, absorbing states (SUCCESS or FAILURE) and a reward function. Reward function is +1 for successful completion of task, -1 for failure (disaster like putting the SPOON in the OVEN) and -0.04 for every other move to keep a limit on long sequence of actions.

The goal of the experiment is to teach the agent to bake a cake with the feedback of the human teacher because, as mentioned by Thomaz and Breazeal [12], there could possibly be more than 10,000 states that the agent can entertain in this environment with only a few reward values, that make the learning slow and difficult without the intervention of human feedback.

B. *Algorithms*

The underlying algorithm used in IRL and SPARC are same with different human reward integration. IRL utilizes the explicit reward by human user of every action based on its result while SPARC uses implicit reward by evaluating the intention of action.

1) ***IRL Implementation:*** Standard Q-learning Algorithm is used for Interactive Reinforcement Learning [12]. Additionally, for interactive rewards by the user, left click on the mouse can be used to reward r = [-1,1] by gliding on the slider on the Sophie's World screen as shown in Fig.[1]. Before allotting the human reward to the agent, visual feedback is displayed on the screen to enable the user to readjust the reward. This runs simultaneously and asynchronously with the operation of the agent. IRL utilizes two main features of interaction between agent and the user in this environment; gaze and guidance.

Fig.1 Sophie's Kitchen : Vertical bar is a slider used for human reward and the green visual box at the top left displays the current state-based reward. [12]

Gaze: Gaze is used to introduce internal state transparency which is opted to benefit the learning. Agent uses gaze to communicate the certainty and uncertainty of its actions to the human user; either the agent looks directly at one object communicating its surety or looks at different objects which have the same probability to be used to translate its uncertainty of the possible action.

Guidance: This attribute of IRL is used to limit the exploration of the agent and converge the action policy to the objects that human user mostly desires to be explored. It does not stop the agent from exploring further but highly influences its focus based on guidance by the human teacher. Right click on the object in the Sophie's World diverts the agent's attention to it if it is facing. Guidance does not completely control the action policy as agent's location cannot be guided by human.

IRL uses greedy policy which helps the learning in algorithm by restricting it from exploring outside the guided policy provided by human reward and guidance. Additionally, human reward fully controls the convergence and divergence of the algorithm.

2) ***SPARC Implementation:*** SPARC is developed on Q-learning algorithm without any reward propagation to bound the learning of the agent by human teaching hence developing a minimum limit of robot's performance as shown in Fig.[2]. Whereas robot performs better using Q-learning. SPARC uses the intention of the agent to help human understand the agent's situation. Agent suggests the action to the human user by using the gaze at objects and locations. Similar to IRL, human teacher can use right click on objects to guide the agent to perform action associated to that object in its present state. Moreover, SPARC extends this guidance to locations too, unlike IRL. At any given time, SPARC can specify the next action that should be executed by the agent. Teacher can choose to correct the executed

Algorithm 1: Algorithm used in SPARC.

```
while learning do
    a = action with the highest Q[s, a] valuelook at object or
    location used with awhile waiting for correction (2
    seconds) do
        if received command then
            a = received commandreward, r = 0.5
        else
            reward, r = 0.25
        end
    end
    execute a, and transition to s'Q(s_t, a_t) ←
    Q(s_t, a_t) + α(r_{t+1} + γ(max_a Q(s_t, a)) − Q(s_t, a_t))
end
```

Fig.2 SPARC Algorithm [16]

action or just not interrupt at all, a reward of 0.25 is assigned to the uncorrected action, on contrary, when the teacher chooses to correct an action the corrected action is not given any reward whereas the action that is manipulated by user to correct that action executed by the algorithm is rewarded 0.5. The actively selected actions are performed more without utilizing high human-robot interaction by only having implicit rewards.

C. *Experimental Design*

Participants were divided into two groups of 20 each. Firstly, they were made familiar with the Kitchen Scenario and how to bake a cake, then they interact with the environment in three individual sessions. Each session contains training and testing phase. In the training phase, participant can have as many episodes of interaction as desired whereas each episode terminates when a 'SUCCESS' or 'FAILURE' is reached or the participant quits the episode or by default after 25 minutes. After the training, there is a testing phase for every session where participant is unable to teach the agent anymore. In testing, the agent performs based on policies learnt during its training phase. The testing episode ends when the agent reaches 'SUCCESS' or 'FAILURE' or participant forcedly terminates it, to stop a continuous loop of actions. Testing phase determines the performance of the participant in its training phase. In this manner, participant does three sessions of the particular system and this helps evaluate the change in interaction based on their familiarity of the setup over time.

After doing three sessions, participant is asked to do three sessions of the other algorithm, IRL or SPARC. For both groups the order of these systems are shuffled i.e.: 20 participants first interact with 3 sessions of SPARC and then IRL whereas other 20 participants first interact with 3 sessions of IRL and then SPARC. This shuffling controls the order bias in interaction.

To test Hypothesis 1, where number of inputs and mental effort of human teacher are the testifiable variables, NASA-Task Load Index (TLX) [17] is used to calculate the work load experienced by the participant.

Individual performance of the participant is calculated by how far the agent could perform in the testing phase; 0 where no step from the six optimal steps (mentioned in subsection 'Kitchen Scenario') have been reached, 6 where all steps have been performed leading to 'SUCCESS' or 2 where only one ingredient has been added to the bowl before termination of the testing phase. Three major criterions of evaluating the degree and quality of interaction are measured with each session and participant; the number of input given during training phase, interaction time (0-25 minutes) and number of 'FAILURES' reached by the agent while training.

In order to study Hypothesis 2, one of the developers of the system is assigned as an 'expert' user to interact with an optimal strategy. This gives a comparison against other naïve participants that interacted as 'non-expert' users.

To study the preference of the type of interaction for Hypothesis 3, perception of robots is measured in the post-experiment questionnaire by binary and Likert Scale.

IV. Results

In order to testify hypotheses, we measured three types of metrics i.e.: Effectiveness and efficiency with non-experts, Safety with Experts and Control.

A. *Effectiveness and Efficiency with Non-experts*

To determine the efficiency of IRL and SPARC, 4 objective measuring criterion i.e.: Performance, Teaching time, Number of inputs and Number of failures, and 1 subjective measuring criteria i.e.: Workload, have been utilized.

1) ***Performance:***

Participants switch between systems (SPARC and IRL) for three sessions each, Friedman test showed a huge difference between performance of participants (regardless of their Group) during first three sessions and next three sessions with $p < 0.01$. Mega difference between performance of both groups is also observed by the test with $p < 0.01$, where Group1 interacted with IRL first and after three sessions swapped to SPARC and Group2 interacted with SPARC first and after three sessions swapped to IRL. Although no difference in performance has been observed in between three sessions of same system, which means there was no significant familiarity developed between the participant and the platform by increasing interaction. Overall, SPARC outperformed IRL for every pairwise comparison, which is determined by $p < 0.05$. The performance graph can be seen in Fig. [3] where y-axis donates the number of crucial steps being performed in the testing phase by the agent.

2) ***Teaching Time:***

SPARC required less teaching time than IRL which is observed by Friedman Test. It is important to note that in between three sessions of same system, only for

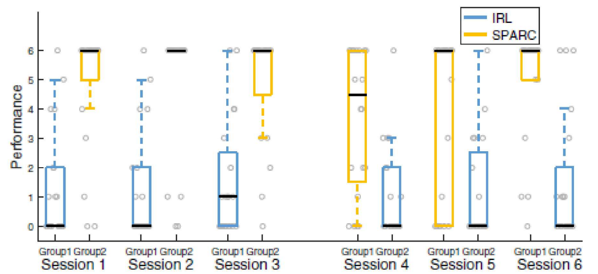

Fig.3 Completion of crucial steps to achieve the goal in testing phase. [16]

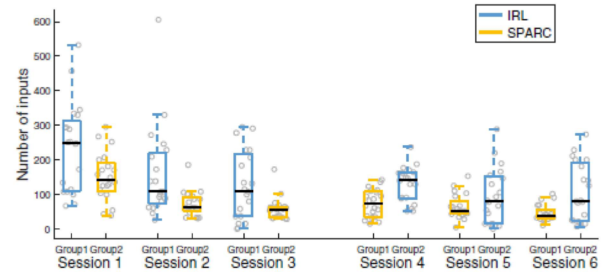

Fig.5 Number of Inputs given by human user in the Training phase. [16]

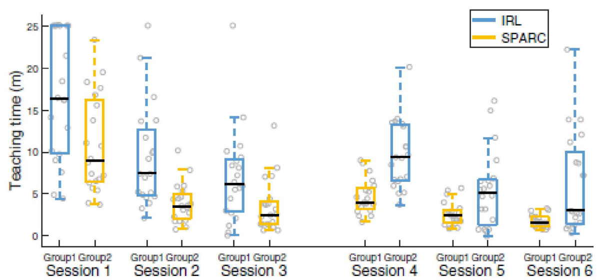

Fig.4 Time utilized for Training phase from (0-25 minutes). [16]

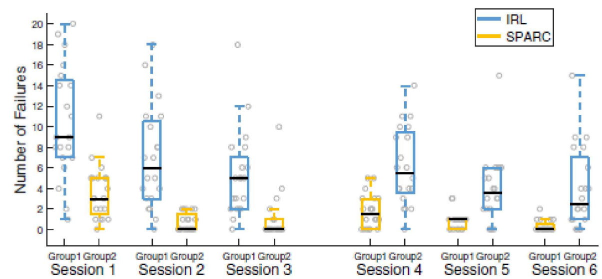

Fig.6 Number of Failures occurred in Training Phase [16]

SPARC time decreased between session 2 and session 3 as in counter case of IRL user got less interested in the interaction as sessions proceeded. The time used for training phase to teach the agent by interaction with human participant, for both systems, are shown in Fig. [4].

*3) **Number of Inputs:***

Every time an interaction was made by IRL or SPARC is seen as an input given to the system for teaching the agent, this is shown in Fig. [5]. Friedman Test with $p < 0.001$ is used to determine that SPARC needed lesser number of inputs than IRL.

*4) **Number of Failures:***

Number of failures occurred during the training phase of every session in both IRL and SPARC are shown in Fig. [6]. Friedman Test showed that lesser number of failures occurred with SPARC than IRL with $p < 0.001$.

*5) **Workload:***

Student-t Test has been used on normally distributed data of average workload experienced by participants which reported significant difference between the workload on human-teacher in interaction with IRL and SPARC. Regardless of groups, participants or sessions, IRL has been detected as more tiring system than SPARC, the graph in Fig. [7] shows the average workload reported by participants.

*B. **Safety with Experts***

One of the authors [16] performed optimal policy on both systems and played as an 'expert' user which optimized the performance in a way that 6 crucial steps were achieved in testing phase of both systems after training with the 'expert' user. Although it is observed that IRL required immensely higher training time than SPARC. Another important observation was that even with an 'expert' user IRL reached 'FAILURE' states due to the lack of control of human teachings on the agent whereas SPARC is fully safe with agent's intention communication and under the guidance of 'expert' user.

*C. **Control***

In the post-experiment questionnaire 38 / 40 participants voted for SPARC as a 'Preferred' system to interact with rather IRL, although IRL has been rated as 'Natural' by 10 / 40 participants. IRL has been a reporting reason of frustration by some participants due to the lack of control which emphasises that when humans teach robots, they look for task completion and not child-like natural learning strategies using exploration.

V. Discussion

SPARC was not originally designed to work in collaboration with RL but still it outperformed IRL and performed well, which supports its adaptability and flexibility in interactive teaching scenarios. As discussed in section('Results') SPARC can lead to higher performance, whilst being faster, requiring fewer inputs and less mental effort from the teacher and minimizing the number of errors during the teaching when used by non-experts (supporting H1). SPARC is observed to perform

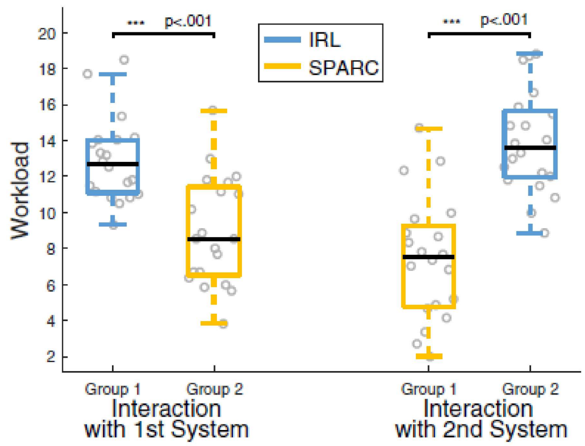

Fig.7 Workload on human user for interactive teaching to the agent [16]

its action policy safely, quickly and efficiently when taught by expert users (supporting H2). In the end, with the help of post-experiment questionnaires it is learned that human teachers prefer a method in which they can have more control over the robot's actions hence preferring SPARC and being frustrated by lack of control on IRL (supporting H3).

SPARC is fully controllable by human teacher which makes it easily adaptive to the teacher's personal way of performance in a particular scenario which means that it is a personalizable learning system. Meanwhile, it is important to note the limitations of the system. If a human reinforces a wrong action my mistake and manually corrects it by executing the altering next action, it can start a never ending loop of altering actions e.g: MOVE-LEFT and MOVE-RIGHT which has to be halted by the teacher immediately hence requiring a full time attention by the human teacher even if it is not obliged to intervene with a reward after every action performed unlike IRL. Another important point to ponder upon is that in this virtual simulation, SPARC was guided by the teacher with an optimal action-policy and six crucial steps still it took some time to perform the task but in real human-robot interaction scenarios task has to be performed in real-time without any guidance by a teacher with optimal policies, in most of the cases. Nevertheless, we insist that SPARC enables a safe, relatively fast, easy interactive robot learning with a full control of the teacher that significantly decreases the workload of the teacher. With time SPARC needs less and less guidance from the teacher hence justifies its importance in future human-robot interactive dynamics.

Additionally, four suggestions are developed should be kept in mind while designing an interactive robot; 1) Teacher should stay in learning loop, 2) Teacher should have control over robot in serious scenarios, 3) Robot should communicate its intuitions before performing, 4) Human adaptability should be limited by the robot. SPARC manages to suffice to first two criterion mentioned by having a complete human control over the agent and staying in the loop while decreasing the teaching role progressively as the agent's learning improves over time.

VI. Conclusion

Supervised Progressively Autonomous Robot Competencies (SPARC) is suggested to solve the issue of creating adaptive actions for a robot while ensuring that the behavior displayed by the robot remains adequate for the task at hand. A guidance and correction method has been used to do so, to allow human to be in charge of the robot at all times while not having to perform every single action manually. SPARC with additional Reinforcement Learning has been compared with Interactive Reinforcement Learning (IRL). In IRL human teacher manually gives reward and guidance for every action performed by the agent whereas in SPARC teacher has option to overwrite the action performed by the agent using RL or not intervene at all. In the experiment performed by 40 participants, it is observed that SPARC is a better teaching approach while interacting with 'non-expert' users. In comparison to IRL, SPARC outperformed in terms of lesser teaching time and lesser unproductive steps while interacting to achieve a goal. Based on results from evaluating criterion of workload, SPARC requires less human effort than IRL hence more human friendly human-robot interaction system.

References

[1] T. Fong , I. Nourbakhsh , K. Dautenhahn , A survey of socially interactive robots, Rob. Auton. Syst. 42 (3) (2003) 143–166 .

[2] J.F. Gorostiza , M.A. Salichs , End-user programming of a social robot by dialog, Rob. Auton. Syst. 59 (12) (2011) 1102–1114 .

[3] G. Hoffman , Openwoz: A runtime-configurable wizard-of-oz framework for human-robot interaction, 2016 AAAI Spring Symposium Series, 2016 .

[4] S. Amershi , M. Cakmak , W.B. Knox , T. Kulesza , Power to the people: the role of humans in interactive machine learning, AI Magazine 35 (4) (2014) 105–120 .

[5] Y. LeCun , Y. Bengio , G. Hinton , Deep learning, Nature (2015) .

[6] R.S. Sutton , A.G. Barto , Reinforcement learning: an introduction, 1, MIT press Cambridge, 1998 .

[7] J.A. Fails , D.R. Olsen Jr , Interactive machine learning, in: Proceedings of the 8th International Conference on Intelligent User Interfaces, ACM, 2003, pp. 39–45 .

[8] D. Olsen , Building interactive systems: principles for human-computer interaction, Cengage Learn., 2009 .

[9] A. Holzinger , Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain Inf. (2016) 119–131 .

[10] S. Stumpf , V. Rajaram , L. Li , M. Burnett , T. Dietterich , E. Sullivan , R. Drummond , J. Herlocker , Toward harnessing user feedback for machine learning, in: Proceedings of the 12th International Conference on Intelligent User Interfaces, 2007 .

[11] M. Cakmak , A.L. Thomaz , Designing robot learners that ask good questions, in: Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction, ACM, 2012, pp. 17–24 .

[12] A.L. Thomaz , C. Breazeal , Teachable robots: understanding human teaching behavior to build more effective robot learners, Artif. Intell. 172 (6) (2008) 716–737 .

[13] W.B. Knox , P. Stone , Combining manual feedback with subsequent mdp reward signals for reinforcement learning, in: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 5–12 .

[14] E. Senft , P. Baxter , J. Kennedy , T. Belpaeme , Sparc: Supervised progressively autonomous robot competencies, in: International Conference on Social Robotics, Springer, 2015, pp. 603–612 .

[15] E. Senft , P. Baxter , T. Belpaeme , Human-guided learning of social action selection for robot-assisted therapy, 4th Workshop on Machine Learning for Interactive Systems, 2015 .

[16] E. Senft et al., Supervised autonomy for online learning in human-robot interaction, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.03.015

[17] S.G. Hart , L.E. Staveland , Development of nasa-tlx (task load index): results of empirical and theoretical research, Adv. Psychol. 52 (1988) 139–183 .