Pattern Recognition Letters 000 (2017) 1–10

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Supervised autonomy for online learning in human-robot interaction

Emmanuel Senft[a,*], Paul Baxter[b], James Kennedy[a], Séverin Lemaignan[a], Tony Belpaeme[a,c]

[a] Plymouth University, Drake Circus, Plymouth PL4 8AA, United Kingdom
[b] Lincoln Centre for Autonomous Systems, University of Lincoln, Brayford Pool, Lincoln LN6 7TS, United Kingdom
[c] Ghent University, imec – IDLab, Department of Electronics and Information Systems, Ghent, Belgium

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Human-Robot interaction
Reinforcement learning
Interactive machine learning
Robotics
Progressive Autonomy
Supervised autonomy

ABSTRACT

When a robot is learning it needs to explore its environment and how its environment responds on its actions. When the environment is large and there are a large number of possible actions the robot can take, this exploration phase can take prohibitively long. However, exploration can often be optimised by letting a human expert guide the robot during its learning. Interactive machine learning, in which a human user interactively guides the robot as it learns, has been shown to be an effective way to teach a robot. It requires an intuitive control mechanism to allow the human expert to provide feedback on the robot's progress. This paper presents a novel method which combines Reinforcement Learning and Supervised Progressively Autonomous Robot Competencies (SPARC). By allowing the user to fully control the robot and by treating rewards as implicit, SPARC aims to learn an action policy while maintaining human supervisory oversight of the robot's behaviour. This method is evaluated and compared to Interactive Reinforcement Learning in a robot teaching task. Qualitative and quantitative results indicate that SPARC allows for safer and faster learning by the robot, whilst not placing a high workload on the human teacher.

1. Introduction

In the not too distant future robots will be expected to have social skills, leaving the factory to interact with people in environments designed exclusively for use by humans [1]. Their users will not be academics or engineers but the elderly, therapists, children or simply non-experts in technology and science. Each user will have specific needs that cannot be totally anticipated at the robot's design stage. Many researchers have argued that this issue can be best addressed by having the user involved in generating the behaviour [e.g. 2,3]. However, we cannot assume that users will have the technical knowledge required to make changes to the code controlling the robot. Therefore, we believe that robots need to have a mechanism allowing a human to teach the robot in an easy, natural and efficient manner.

One way to provide a robot with such learning capability is to use machine learning. Classic machine learning is often designed by experts to be used by experts, its interface being often too complex for people not involved in the design process [4]. Many methods also suffer from practical issues: Deep Learning [5] relies on having large datasets to train networks, while Reinforcement Learning [6] uses extensive and costly exploration to gather data points used for learning. As we aim at allowing a non-expert end-user to personalise the robot's behaviour, complex interfaces are not desirable, large dataset are not available and random exploration can lead to undesired actions by the robot. This suggests two main challenges: how to empower the user with the ability to teach the robot and how to gather safe training experiences for the robot. A solution aiming to solve these two challenges is *interactive machine learning* [4,7,8]. In this framework, the human is part of the machine learning process. By providing ground truth labelling or guiding the agent during exploration to the interesting parts of the environment, the human can bootstrap and guide the learning. Furthermore, the human can provide more information than simply labelling the samples, bringing further improvements to the learning [9,10] and if enough control is provided, the human teacher can also prevent the robot from making undesirable or potentially dangerous errors.

In this paper, we present a novel approach to combine reinforcement learning with interactive machine learning following the Supervised Progressively Autonomous Robot Competencies (SPARC) method proposed in [11]. By giving control of the robot's actions to a teacher, we aim to maximally use the human's knowledge and transfer it to a robot in a quick, safe and efficient manner. This method is compared to *Interactive Reinforcement Learning* (IRL), described in [12], using a study involving 40 participants interacting

[*] Corresponding author.
E-mail address: emmanuel.senft@plymouth.ac.uk (E. Senft).

http://dx.doi.org/10.1016/j.patrec.2017.03.015
0167-8655/© 2017 Elsevier B.V. All rights reserved.

with both approaches in Sophie's Kitchen, the environment used to demonstrate IRL.

The reminder of the paper is organised as follows. Section 2 presents different approaches used to teach robots in an interactive fashion. We then describe the scope of the study, including our hypotheses (Section 3) and methodology (Section 4). Results are presented in Section 5 and are discussed in Section 6. We also propose guidelines for designing robots which interactively learn from people. Finally, we conclude by summarising the main results and the guidelines in Section 7.

2. Related work

In human-robot interaction, the expected behaviour of the robot is often solely known by the users: for therapies, therapists are the experts and they know how the robot is supposed to behave when interacting with patients. For assistive robots in homes, each user has his own desires and preferences concerning the robot's behaviour. Consequently, these users have to be able to adapt the behaviour of the robot in a way which suits them without requiring technical skills. One approach to allow non-technical persons to teach a robot an action policy is *Learning from Demonstration* [13,14]. In this framework, a human provides a robot with demonstrations of the expected behaviour and the robot learns the correct action policy. This methods is often used for teaching motor trajectories to a robot, but is also applicable to high level action policy learning in robotics [15]. The conventional approach consists of a set of demonstrations from the teacher followed by additional learning without supervision until reaching an appropriate action policy. However, human-robot interactions are not a static process, the learning should happen during all interactions and be interactive: the user should at all times be able to correct the robot when it selects a suboptimal action.

In *interactive machine learning* a human is included in the learning loop, allowing him to provide input during the learning process, this approach has received increased attention over the last decade. One of the main domains being extensively researched is active learning [16]. Active learning has been used in a range of fields: from medical image classification [17] to robotics [18]. In this framework, an agent has to classify points in a dataset and an 'oracle' is present and available. The oracle, often a human, can provide ground truth labelling, but its use has a cost (time or money for example) and consequently should be minimised. As such, the conventional challenge of active learning is to find how to optimise the use of the oracle to improve the learning. Multiple approaches have been tested, such as requiring labels for the points with the higher uncertainty or which categorisation would provide the best improvement of the learning.

However, as pointed out by [19], one of the main limits of active learning is that the robot is in control of the interaction: the robot takes initiative to request training data from the user, regardless of what the human wants the robot to do, potentially leading to frustration or incomprehension on the human side. For this reason, methods have been developed to give the initiative back to the human, placing the human in a teaching role. For example, when set in a reinforcement learning framework, the human teacher can provide additional feedback [12,20] and actively decides to reward or not to reward a specific action.

In human robot interactions, the robot's actions can have a real impact on the world and some actions, if executed at an incorrect moment, can create discomfort for the user or even cause physical or psychological harm. These errors can be the result of an incorrect action policy or a sensor failure for example, but they have to be prevented. When using a robot in real human-robot interaction applications, a safeguard should therefore be present to prevent the robot from executing undesirable actions, especially when working with vulnerable users, where some actions would have severely negative effects. It is on this basis that the concept of *supervised autonomy* was introduced [21]: a safeguard is provided by a human supervising the robot in a semi-autonomous setup. The robot is mainly autonomous, but a human teacher has enough control over the interaction to step in at any time to correct the action about to be executed by the robot. This approach ensures that only desired actions will be executed by the robot whilst not relying completely on a human to control the robot as with Wizard of Oz [22]. The challenge is then the incorporation of robot learning into this scheme to facilitate progressive performance improvement: this approach can be combined with interactive machine learning to let the robot learn from its errors without requiring the robot to actually make them. At the same time, the human is used to bootstrap the learning with their knowledge, but also to ensure that the robot behaviour is always appropriate. This would allow the robot to improve its behaviour over time, while reducing the frequency of human interventions, having the robot learning without needing to face the consequence of its actions.

An analogous system is predictive texting on mobile phones: as a user types a message, possible words are suggested, but the user has full control over which word to select. All the while, the algorithm learns: it adopts new words, spellings and tunes its predictive models to suit the user's particular language use and preferences. We propose a similar mechanism for Human-Robot Interaction, and in this context we introduced the Supervised Progressive Autonomous Robot Competencies (SPARC) [11,23].

By combining interactive machine learning and supervised autonomy, SPARC provides an agent with online learning whilst keeping the control of the agent's actions in the user's hand. This method based on a suggestion/correction mechanism allows the robot to adapt its behaviour to the user whilst ensuring, due to the presence of the human teacher, that the actual actions executed by the robot are suited to the current interaction. This approach is especially useful in context where the cost of having the robot making errors is high, such as when interacting with vulnerable population.

3. Scope of the study

Following on from our earlier research on using people to teach an action policy to a robot during interaction [11], we seek to evaluate SPARC when combined with the widely used learning paradigm of Reinforcement Learning (RL) [6]. We compare this approach to an alternative method combining interactive machine learning and reinforcement learning: IRL [12]. To this end we tested both learning methods in the environment initially used by Thomaz and Breazeal and described in Section 4.

3.1. Interactive Reinforcement Learning

IRL implements the principles presented in [12]. In IRL the human teacher can provide positive or negative feedback on the last action executed by the robot. The robot combines this with environmental feedback into a reward which is used to update a Q-table: a table with a Q-values (the expected discounted reward) assigned to every state-action pair and used to select the next action. Three additions to the standard algorithm have been proposed and implemented by Thomaz and Breazeal and are used here as well: guidance, communication by the robot and an undo option.

The guidance emerged from the results of a pilot study where participants assigned rewards to objects to indicate that the robot should do something with these objects. With the guidance, teachers can direct the attention of the robot toward certain item in the environment to indicate the robot that it should interact with them.

The robot can communicate its uncertainty by directing its gaze toward different items in the environment with equally high probability of being used next. The aim of this communication of uncertainty is to provide transparency about the robot's internal state, for example indicating when a guidance should be provided.

Finally, after a negative reward, the robot tries to cancel the effect of the previous action (if possible), resulting in a undo behaviour. As shown in the original paper, these three additions improve the performance on the task.

3.2. SPARC

SPARC (Supervised Progressively Autonomous Robot Competencies) uses a single type of input similar to the guidance present in IRL. However with SPARC, it is used to control the actions of the robot. The robot communicates every of its intentions (i.e the action it plans to execute next) to its teacher. The teacher can either not intervene and let the robot execute the suggested action or he can step in and force the robot to execute an alternative action. This combination of suggestions and corrections gives the teacher full control over the actions executed by the robot. This also makes the rewards redundant: rather than requiring the human to explicitly provide rewards a positive reward can directly be assigned to each action executed by the robot as it has been either forced or passively approved by the teacher.

3.3. Differences of approaches

Unlike IRL, SPARC offers full control over the actions executed by the robot. SPARC changes the learning paradigm from learning from the environment's response to learning from the users preferences. We use an expert in the task domain to evaluate the appropriateness of actions before their execution and we use this evaluation and control provided to the expert not to rely on observing negative effect of an action to learn that this action should be avoided, but rather what the best action is for each state. Even in a non-deterministic environment such as HRI, some actions can be expected to have a negative consequence. The human teacher can stop the robot from ever executing these actions, preventing the robot from causing harm to itself or its social or physical environment.

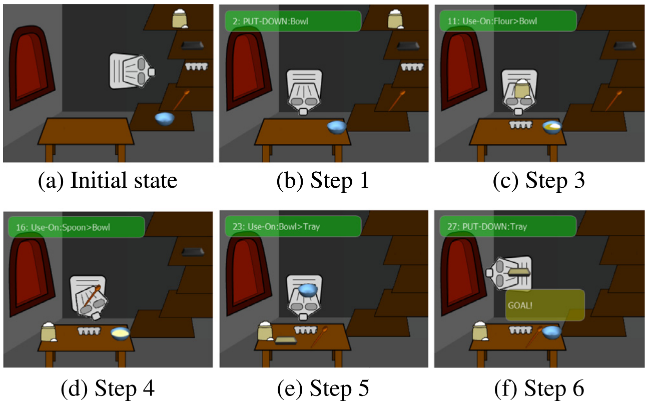
Another noticeable difference is the way in which the robot communicates with the user: in IRL, the robot communicates its uncertainty about an action and with SPARC its intention of executing an action.

It should also be noted that the quantity of information provided by the user to the robot is similar for both IRL and SPARC: in SPARC the user can offer the whole action space as commands to the robot, but removes the need for explicit rewards. While in IRL, the teacher can guide the robot toward a subset of the action space but has to manually provide feedbacks to evaluate the robot's decisions.

3.4. Hypotheses

Three hypotheses are tested in this study:

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



(a) Initial state (b) Step 1 (c) Step 3

(d) Step 4 (e) Step 5 (f) Step 6

Fig. 1. Presentation of different steps in the environment. (a) initial state, (b) step 1: the bowl on the table, (c) step 3: both ingredients in the bowl, (d) step 4: ingredients mixed to obtain batter, (e) step 5: batter poured in the tray and (f) step 6 (success): tray with batter put in the oven. Step 2: one ingredient in the bowl has been omitted for clarity.

4. Methodology

4.1. Task

The task used in this study is the same as [12]: Sophie's kitchen, a simulated environment on a computer where a virtual robot has to learn how to bake a cake in a kitchen. As the source code was not available, the task was reimplemented to stay as close as possible to the description in the paper and the online version of the task.[1]

The scenario is the following: a robot, Sophie, is in a kitchen with three different locations (shelf, table and oven) and five objects (flour, tray, eggs, spoon and bowl) as shown in Fig. 1a. Sophie has to learn how to bake a cake and the user has to guide the robot through a sequence of steps while giving enough feedback so the robot can learn a correct series of actions. As presented in Fig. 1, there are six crucial steps to achieve a successful result:

1. Put the bowl on the table.
2. Add one ingredient to the bowl (flour or eggs).
3. Add the second ingredient.
4. Mix the ingredients with the spoon to obtain batter.
5. Pour the batter in the tray.
6. Put the tray in the oven.

The environment is a deterministic Markov Decision Process, defined by a state, a set of actions (move left, move right, pick up, drop and use), a deterministic transition function, absorbing states (success or failure) after which the simulation is restarted in its initial state and an environmental reward function (+1 for success and −1 for failure and −0.04 for every other step to penalise long sequences). Different action policies can lead to success, but many actions end in a failure state, for example putting the spoon in the oven results in a failure. As argued by Thomaz and Breazeal, this environment provides a good setup to evaluate teaching methods to a robot due to the large number of possible states (more than 10,000), the presence of success and failure states and the sparse nature of the environmental reward function which increases the need for a teacher to aid the learning. More details on the environment are available in the original paper.

[1] http://www.cc.gatech.edu/~athomaz/sophie/WebsiteDeployment/.

Please cite this article as: 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

4.2. Implementation

In this experiment two systems are tested: IRL and SPARC. The underlying learning algorithm is strictly identical for both system, only the way of interacting with it is different: participants have more control in SPARC, implicitly reward action rather than explicitly and evaluate the intention of the action rather than its results. The learning algorithm (see Algorithm 1) is a variation on Q-learning, without reward propagating.[2] This guarantees that any learning by the robot is only due to the teaching by the human, and as such provides a lower bound for the robot's performance. By using Q-learning, the performance of the robot would be higher.

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 $s'Q(s_t, a_t) \leftarrow$
    $Q(s_t, a_t) + \alpha(r_{t+1} + \gamma(\max_a Q(s_t, a)) - Q(s_t, a_t))$
end
```

4.2.1. Interactive Reinforcement Learning

We have implemented IRL following the principles presented in [12]. The user can use the left click to display a slider in order to provide rewards. The guidance is implemented by right-clicking on objects: it directs the robot's attention to the object if facing it (a click on objects in different locations has no effect). Following the guidance, the robot will execute the candidate action involving the object. The action space is not entirely covered by this guidance mechanism: for example, it does not cover moving from a location to another. This guidance if used correctly, limits the exploration for the current step to the part of the environment evaluated as more interesting by the user without preventing the robot to explore in further steps. The robot can communicate its uncertainty by looking at multiple objects having similarly high probability of being used.

Some modifications were required to the original study due to the lack of implementation details in the original paper, one of them being the use of a purely greedy action selection instead of using softmax, due to the absence of parameters descriptions. The reliance on human rewards and guidance limits the importance of autonomous exploration, and thus, the greediness of the algorithm should assist the learning by preventing the robot to explore outside of the guided policy. Additionally, as the human teacher can vary the rewards provided to the system, they have full control of the convergence or divergence of the algorithm.

4.2.2. SPARC

SPARC uses the gaze of the robot toward objects or locations to indicate which action the robot is suggesting to the teacher. Similarly to the guidance in IRL, the teacher can use the right click of the mouse on objects to have the robot execute the action associated to this object in the current state and this has been extended to also cover locations. With SPARC, the command covers all the action space: at every time step, the teacher can specify, if desired, the next action executed by the robot. If an action is not corrected, a positive reward of 0.25 is automatically received (as it has the implicit approval from the teacher) and if the teacher selects another action, a reward of 0.5 is given to the correcting action (the corrected action is not rewarded). That way, actions actively selected are more reinforced and participants can still have give higher rewards when using IRL. This system allows for the use of reinforcement learning with implicit reward assignation, which simplifies the Human-Robot Interaction.

4.3. Experimental design

Participants are divided into 2 groups and interact first either with IRL or SPARC as shown in Fig. 2. Before interacting, participants receive an information sheet explaining the task (describing the environment and how to bake a cake) and one explaining the system they are interacting with. Then they interact for three sessions with the assigned system. Each session is composed of a training phase and a testing phase. The training phase is composed of as many teaching episodes as the participant desires, a teaching episode ends when a success or failure state has been reached which returns the environment to the initial state. In the same way as in the initial experiment by Thomaz and Breazeal, participants can decide to terminate the training phase whenever they desire by clicking on a button labelled ‘Sophie is ready’, however it is also terminated after 25 min to impose an upper time limit to the study. After the end of a training phase, the robot will run a testing phase where the participant's inputs are disabled and which stops as soon as a ending state is reached or the participants decide to stop it (for example if the robot is stuck in a loop). This testing phase is used to evaluate the performance of the participants for this session. The interaction with a system consists of three repeated independent sessions with their own independent training and testing phases to observe how the interactions evolve as participants are getting used to the system.

After participants completed their three sessions with the first system, they are asked to interact for three more sessions with the other system. This way, every participant interacts three times with each system (IRL and SPARC) and the order of interaction is balanced. Additionally, a demographic questionnaire is given before the first interaction, a first post-interaction questionnaire after the interaction with the first system, a second identical one after the interaction with the second system and a final post-experiment questionnaire at the end of the experiment. All information sheets and questionnaires can be found online.[3]

This experimental design prevents the risk of having an ordering effect by having a symmetry between conditions. Both conditions having an identical experimental procedure only with the order of interaction varying.

4.4. Participants

A total of 40 participants have been recruited using a tool provided by the university to reach a mixed population of students and non-student members of the local community. All participants gave written informed consent, and were told of the option to withdraw at any point. All participants received remuneration at the standard U.K. living wage rate, pro rata. Participants were distributed randomly between the groups whilst balancing gender and age (age *M*=25.6, *SD*=10.09; 24F/16M). Participants were

[2] In Q-learning the update function is $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma(\max_a Q(s_{t+1}, a)) - Q(s_t, a_t))$.

[3] http://www.tech.plym.ac.uk/SoCCE/CRNS/staff/esenft/experiment2.html.

Please cite this article as: 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

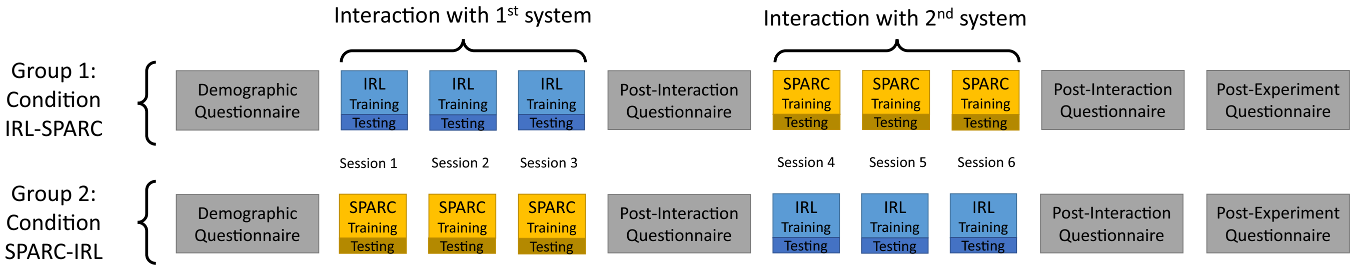

Interaction with 1st system
Interaction with 2nd system
Group 1: Condition IRL-SPARC
Group 2: Condition SPARC-IRL
Demographic Questionnaire
IRL Training Testing
SPARC Training Testing
Post-Interaction Questionnaire
Post-Experiment Questionnaire
Session 1
Session 2
Session 3
Session 4
Session 5
Session 6

Fig. 2. Participants are divided into two groups. They first complete a demographic questionnaire, then interact for three independent sessions (with a training and a testing phase each) with a system (IRL or SPARC). After a first post-interaction questionnaire, participants interact for another three sessions with the other system before completing the second post-interaction questionnaire and a final post-experiment questionnaire.

mostly not knowledgeable in machine learning and robotics (average familiarity with machine learning M=1.8, SD=1.14; familiarity with social robots M=1.45, SD=0.75 - Likert scale ranging from 1 to 5).

In addition to naive non-expert users, an expert user (one of the authors) interacted five times with each system following a strictly optimal strategy in both cases. These results from the expert are used to evaluate hypothesis 2 and show the optimal characteristics of each system (IRL and SPARC) when used by trained experts such as therapist in a context of assistive robotics.

4.5. Metrics

4.5.1. Objective metrics

We collected three metrics during the training phase: the number of times a participant reached a failure state while teaching, which can be related to the risks taken during the training and the teaching time (from 0 to 25 min) and the number of inputs provided during the training, which can be seen as the efforts invested in the teaching. The testing phase being only a single run of the taught action policy ending as soon as the robot reaches an ending state (failure or success) or if stopped by the participants. We only use the performance achieved during this single test as evaluation of the success of training. As not all participants reached a success during the testing phase, we used the six key steps defined in Section 4.1 as a way to evaluate the performance ranging from 0 (no step has been completed) to 6 (the task was successfully completed) during this testing run: for example a testing where the robot puts both ingredients in the bowl but reaches a failure state before mixing them would have a performance of 3.

4.5.2. Subjective metrics

The post-interaction and post-experiment questionnaires provide additional subjective information to compare with the objective results from the interaction logs. Two principal metrics are gathered: the workload on participants and the perception of the robot.

Workload is an important factor when teaching robots. As roboticists, our task is to make the teaching of the robot as undemanding as possible, meaning that the workload for user should be minimal. Multiple definitions for workload exist and various measures can be found in the literature. Due to its widespread use in human factors research and clear definition and evaluation criteria, we decided to use the NASA-Task Load Index (TLX) [24]. We averaged the values from the 6 scales (mental, physical and temporal demand, performance, effort and frustration) to obtain a single workload value per participant for each interaction. So we have two measures for each participant, after interaction with the first system (IRL or SPARC) and after the interaction using the other system.

Finally, the perception of the robot has been evaluated in the post-interaction and post-experiment questionnaires using subjective questions (measured on a Likert scale), binary questions (which robot did you prefer interacting with) and open questions on preference and naturalness of the interaction.

5. Results

Most of the results are non-normally distributed. Both ceiling and floor effects can be observed depending on the conditions and the metrics. For the teaching time, some participants preferred to interact much longer than others, resulting in skewed data. Likewise for the performance: often participants either reached a successful end state or did not hit any of the sub-goals of the task ending often in two clusters of participants: one at a performance of 6 and one at 0. Similarly, some participants who interacted a long time with the system did not complete any step, while others could achieve good results in a limited time. Due to the data being not normally distributed, non-parametric statistical tests have been used. We use a combination of Friedman test for one way comparison with repeated measures, Wilcoxon rank sum test for between subject comparisons and the Wilcoxon signed rank test for within subject pairwise comparisons. Additionally, as each interaction consists of three sessions, a Bonferroni correction has been applied to pairwise comparison between sessions. A similar correction was used when comparing between systems to account of the two different groups. To apply the Bonferroni correction, we multiply the p-values by the correcting factors, which allows us to keep a global significance level at $p = .05$.

Initial results of the first interaction of the participants have been reported in [25].

5.1. Effectiveness and efficiency with non-experts

Four objective metrics (performance, teaching time, number of inputs used and number of failures) and one subjective metric (workload) have been used to evaluate the efficiency of IRL and SPARC.

5.1.1. Performance

Fig. 3 presents the performance of participants during the interaction. In the first three sessions participants interacted with either IRL or SPARC, and swapped for the remaining three sessions. There is a significant difference of performance between systems; a Friedman test shows a significant difference between systems during the first three sessions ($\chi^2 = 50.8,\ p < .001$) and during the next three sessions ($\chi^2 = 36,\ p < .001$). Similarly, a significant difference in performance is noted within participants (Group 1: $\chi^2 = 37.9,\ p < .001$ - Group 2: $\chi^2 = 55.3,\ p < .001$). So in all the cases, participants interacting with SPARC achieved a significantly higher performance than those interacting with IRL, regardless of

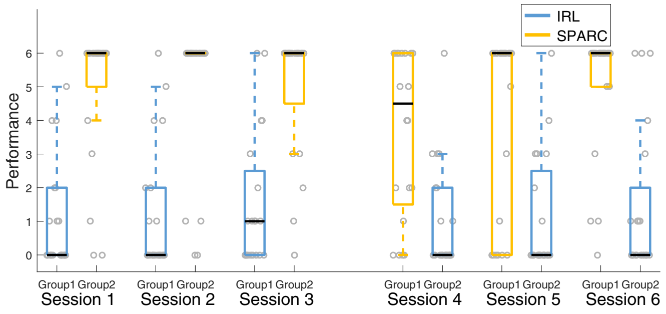


Fig. 3. Comparison of the performance for the six sessions (three with each system, IRL and SPARC, with interaction order balanced between groups). A 6 in performance shows that the taught policy leads to a success. The circles represent all the data points (n=20 participants per group), the black horizontal line the median and the top and bottom of the boxes the first and third quartiles. The learning is consistently better when using SPARC.

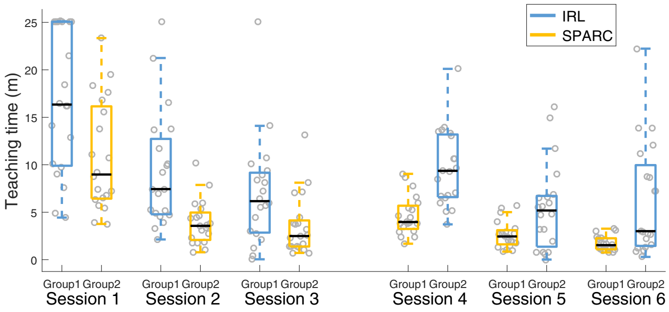


Fig. 4. Comparison of the teaching time (in minutes) for all the interactions. Participants spent less time teaching the robot when using SPARC than IRL.

Table 1
Medians of the teaching time. In the first three sessions, group 1 interacted with IRL and group 2 with SPARC and participants interacted with the other system for the next three sessions.

	$\tilde{X}_1$	$\tilde{X}_2$	$\tilde{X}_3$	$\tilde{X}_4$	$\tilde{X}_5$	$\tilde{X}_6$
Group 1	16.3	7.44	6.17	3.97	2.45	1.53
Group 2	8.97	3.57	2.49	9.36	5.18	3.01

the order in which they interacted ($p < .05$ for all pairwise comparison). No difference of performance has been observed when using Wilcoxon signed rank test on the three repetitions between participants when interacting with the same system, so interacting for a second or third session with the same system does not have a significant impact on participants' performance.

It must be noted that in our study, only a limited number of participants managed to teach the robot to complete the task using IRL, this observation will be discussed in more details in Section 6.

5.1.2. Teaching time

The teaching times for all the interactions are shown in Fig. 4. Regardless of the order in which they used SPARC or IRL, participants needed significantly less time to teach the robot when using SPARC than with IRL (Friedman test between participants for the first three sessions: $\chi^2 = 9.77$, $p = 0.0018$ - next three sessions: $\chi^2 = 20.2$, $p < .001$). Pairwise comparison also show significance ($p < .05$) except for sessions 3 and 5 which can be explained by the floor effect observed when teaching with SPARC and a potential loss of motivation when using IRL.

Additionally, when interacting multiple times with the same system, participants interacted significantly less in the second interaction with a system than during the first one (cf. Table 1) and only for SPARC the teaching time significantly decreases again between the second and the third session.

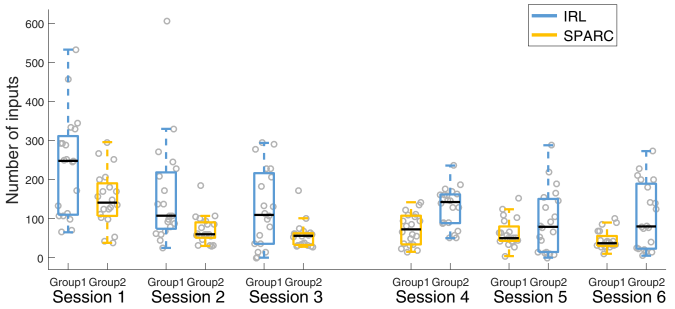


Fig. 5. Comparison of the number of inputs used during the teaching phases.

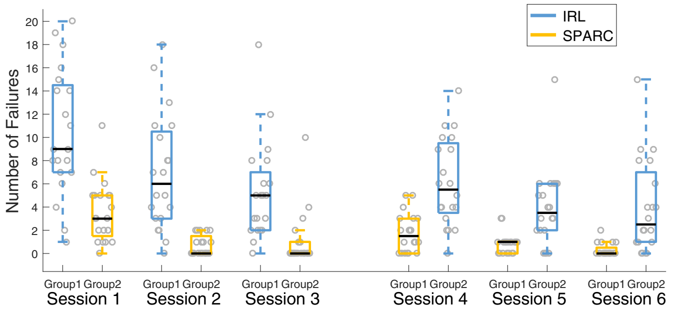


Fig. 6. Comparison of the number of failure states reached during the teaching process. Due to the ability to stop the robot from executing a suggested action, there are fewer failure states when using SPARC.

5.1.3. Number of inputs

The number of inputs used in both system is presented in Fig. 5. For IRL, this represents every time a participant provided guidance or a reward to the robot, and for SPARC every time a participant provided a command. The number of inputs used is lower when teaching with SPARC than with IRL (Friedman test between participants for the first three sessions: $\chi^2 = 11.7$, $p < .001$ - next three sessions: $\chi^2 = 11$, $p < .001$). However with pairwise comparisons only session 2 ($p = .008$) and session 4 ($p < .001$) present a significantly different number of inputs used.

5.1.4. Number of failures

Fig. 6 shows the number of failures observed with both systems for every session. In all the interactions, participants interacting with SPARC faced fewer failures during the training of the robot than those interacting with IRL (Friedman test between participants for the first three sessions: $\chi^2 = 47.8$, $p < .001$- next three sessions: $\chi^2 = 41.8$, $p < .001$ - within participants in group 1: $\chi^2 = 56.6$, $p < .001$ - group 2: $\chi^2 = 20.7$, $p < .001$ - all pairwise comparison: $p < .002$).

5.1.5. Workload

The average workload felt by participants after each interaction with a system is shown in Fig. 7. As the workload data is normally distributed, a student t-test has been used. Participants interacting with IRL first reported an average workload of 12.9 (*SD*=2.33), with SPARC first this was 8.95 (*SD*=3.02). With SPARC after having interacted with IRL the reported workload was 7.44 (*SD*=3.33) and with IRL after SPARC it was 13.9 (*SD*=2.85). We found a significant difference between the reported workload when interacting with IRL or SPARC regardless of the order of interaction. This was also observed between participants (interaction with system 1, independent t-test: $t(38) = 4.63$, $p < .001$ - system 2, independent t-test: $t(38) = -6.5$, $p < .001$ - Group 1, paired t-test: $t(19) = 9.82$, $p < .001$ - Group 2, paired t-test: $t(19) = -6.8$, $p < .001$). Regardless of the interaction order, participants rated SPARC as having a lower workload than IRL.

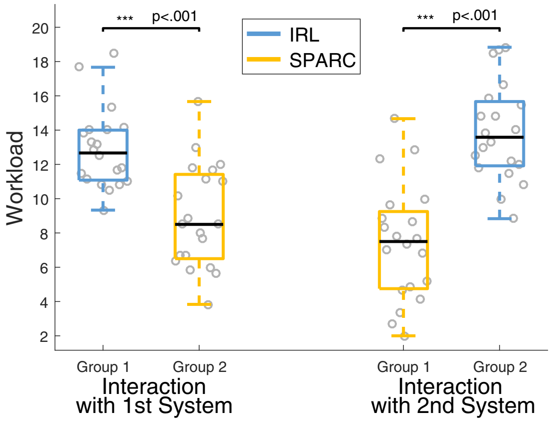


Fig. 7. Comparison of the workload experienced by participants. SPARC was perceived as having a lower workload. Results being normal, student t-test has been used for the comparisons.

5.1.6. Validation of the hypothesis

The objective data (performance, teaching time, number of inputs and number of failures) show that despite spending a shorter time interacting with SPARC and using less inputs, participants reached a higher performance than with IRL whilst facing fewer failures during the teaching. Additionally, when interacting with SPARC, participants' time required to teach the robot decreased with successive sessions, without affecting the performance. This indicates that after the first session, participants understood the interaction mechanism behind SPARC and consistently managed to achieve a high performance whilst requiring less time to teach the robot the task. On the other hand, when interacting with IRL, participants' performance remains low over the session, and their teaching time decreases between session 1 and 2 but not between session 2 and 3. This might be due to a loss of motivation after session 1 where often participants did not succeed to teach the robot, reducing the desire to further interact in successive sessions.

The results suggest that teaching the robot using SPARC allows the robot to achieve a higher performance than with IRL, in a shorter time, without requiring more inputs, while making fewer errors when teaching. These objective results are also supported by subjective measures: the workload on the teacher is lower when using SPARC than when using IRL. For these reasons, H1 ('Compared to IRL, SPARC can lead to higher performance, whilst being faster, requiring fewer inputs and less mental effort from the teacher and minimising the number of errors during the teaching when used by non-experts.') is supported.

5.2. Safety with experts

To evaluate the safety offered by SPARC and IRL, an expert (one of the authors) interacted five times with each systems. In both cases, the expert followed a strictly optimal strategy. This shows the expected behaviours in optimal conditions, the best metrics achievable. Results of the interactions are presented in Table 2. In both cases, the expert successfully taught the robot (as indicated by a performance of 6), which indicates that both systems can be used to teach a robot an action policy. However the time required to teach the robot with IRL is significantly higher than with SPARC.

Additionally, when using IRL, even an expert cannot prevent the robot from reaching failure states during the training due to the lack of control over the robot's action. This is prevented when interacting with SPARC, due to the full control and clear communication, the teacher can ensure that only desired actions are executed. So with sufficient knowledge, an expert can teach the robot to behave safely without having to explore undesired states. This has real world applications, as random exploration is often impossible or undesirable, SPARC offers a way for the teacher to stop the robot from executing actions with negative consequences.

Table 2
Results of an expert interacting 5 times with each system following an optimal strategy. Both IRL and SPARC reached a success during all the testing phase, but the time required to teach SPARC was significantly shorter, and unlike IRL, not a single failure was reached during the training with SPARC. Data following a normal distribution, student t-test has been used.

	IRL *M*(*SD*)	SPARC *M*(*SD*)	*t*(8)	*p*
Perf.	6 (0)	6 (0)	NA	NA
Time (mn)	4.5 (0.67)	0.60 (0.03)	13.1	< .001
# of Fail.	3.2 (0.84)	0 (0)	8.55	< .001

Similar results have been observed with the non-expert participants: in their last interaction with SPARC, both groups had a median of 0 failures for a performance of 6, meaning that more than half of the participants taught the robot the task without ever hitting a failure state. These results support H2 ('SPARC can be used by experts to teach an action policy safely, quickly and efficiently').

5.3. Control

One of the main differences between the two methods is the way in which the concept of teaching is approached. With IRL an exploratory individual learning approach is followed: the robot has freedom to explore, and it can receive feedback on its actions and hints about actions to pursue next from a teacher. This is to some extent inspired by how children are taught, where the learning process can be more important than the achieved results. This is supported by the behaviours observed by Thomaz and Breazeal: their participants gave motivational rewards to the robot, just as one would to do to keep children motivated during learning, despite the absence of effect or use in classical reinforcement learning.

The post-experiment questionnaire included the open question: 'which robot did you prefer interacting with and why?'. Almost all the participants (38 out of 40) replied that they preferred interacting with SPARC. Half of all the participants used vocabulary related to the control over the robot actions ('control', 'instruction', 'command', 'what to do' or 'what I want') to justify their preferences without these words being used in the question. Furthermore, multiple participants reported being frustrated to have only partial control over the robot's actions with IRL, they would have preferred being able to control each action of the robot.

To the question 'which interaction was more natural?', 10 participants rated IRL as being more natural, using justifications such as: 'The robots thinks for itself', 'Some confusion in the [IRL] robot was obvious making it more natural', 'More like real learning', 'Because it was hard to control the robot' or 'People learn from their mistakes faster'. But despite acknowledging that IRL is more natural, closer to human teaching, participants still preferred teaching using SPARC. This suggests that when humans teach robots, they are focused on the results of the teaching: can the robot do the new task requested. This relates to the role of robots, they often interact in human-centred scenario where they have to complete a task for their users. And due to the absence of life-long learning for robots today, it is not worth investing time and energy to allow the robot to improve its learning process or explore on its own. These comments from the participants show support for H3 ('Teachers prefer a method providing more control over the robot's actions.').

Please cite this article as: 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

6. Discussion

Despite not being originally designed to be used in combination with Reinforcement Learning, SPARC does achieve good results. This shows that principles covered by SPARC (control over the robot's actions, communication and evaluation of intentions and automatic execution of proposed actions) are agnostic to the learning algorithm and promote efficient teaching. Furthermore, SPARC achieves a higher performance, in a shorter time and facing less failures than IRL, whilst requiring a lower workload from the human teacher (supporting H1). Finally, when used by experts, SPARC demonstrates that teaching can be safe and quick: the full control over robot's action in the teacher's hands ensures that only desired actions will be executed (validating H2). These results show an interesting feature of teaching; as robots mainly interact in task oriented, human-centred environments, human teachers seem to prefer direct approaches focused on commands rather than letting the robot explore on its own (partial support for H3).

6.1. Comparison with original interactive reinforcement learning study

Unlike in the original experiments evaluating IRL [12], in the study presented in this paper most of the participants did not succeed in teaching the robot the full cake baking sequence using feedback and guidance. In the Thomaz and Breazeal [12] study, the participants were knowledgeable in machine learning (M=3.7, SD=2.3 - range: 1 to 7), but the population in the current study was drawn from a more general public having little to no knowledge of machine learning (M=1.8, SD=1.13 - range: 1 to 5). This can explain why a much larger number of participants did not achieve success with IRL in this study whereas Thomaz and Breazeal only reported 1 participant out of 13 failing the task. In our study, 12.5% of the participants and the expert did manage to train the robot using IRL. This seems to be largely due to participants not consistently rewarding correct actions, preventing the reinforcement learning algorithm from learning. This is why implicit rewards –every action allowed by the teacher is positively rewarded– tend to work better than explicit ones. This is consistent with [26] who note that feedback is not well suited for teaching an action policy from scratch, but better for fine tuning. For teaching the basis of the action policy, they recommend using demonstrations, the method used by SPARC.

6.2. Advantages and limitations of SPARC

In the SPARC implementation for this study, SPARC reproduces actions selected by the teacher. So one can argue that no learning algorithm is required, instead the actions could just be blindly reproduced by the robot. However SPARC combined with reinforcement learning does provide advantages: due to the Q-Table, all the loops in the demonstration are removed when the robot interacts on its own and it provides a way to deal with variations in teaching. It also allows the robot to continue from any state in the trajectory. And finally, due to the suggestion/correction mechanism, the teacher can leave the robot to act on its own as long as it attempts correct actions, and the human to intervene only when the robot is about to execute an incorrect action.

Over the 79 successful trials using SPARC, participants used 47 different strategies to teach the robot the task of baking a cake. This shows how SPARC, as a single control mechanism, allows for different action policies to be learnt depending on the person teaching the robot. With SPARC the robot can adapt its behaviour to the human it is interacting with, profiling the user to find the desired way of behaving.

However SPARC also has limitations in the current implementation, related to the quality of the human supervised guidance. If the teacher allows an action to be executed by mistake (through inattention or by not responding in time), this action will be reinforced and will have to be corrected later on. This might lead to loops when successive actions are cancelling each other (such as move left, then right). In that case, the teacher has to step in and manually guide the robot to break this cycle. Furthermore, due to the automatic execution of actions, the teacher has to be attentive at all times and ready to step in when a wrong action is suggested by the robot.

In this version, SPARC has been applied to a scenario where a clear strategy with optimal actions is present. The interaction also takes place in a virtual environment with a discrete time. Real HRI are stochastic, happen in real time and often there is no clear strategy known in advance. However, we argue that human experts in the application domain can know what type of actions should be executed when, and which features of the environment they used for their decision. As this knowledge can not be available to the robot's designers, robots should be able to learn from a domain user in an interactive fashion. In the current implementation, SPARC mainly receives inputs from a teacher at predefined discrete times and still does not use the human knowledge to it's fullest: the learning algorithm is still simple and with limited inputs, but as described in Section 6.4, we are working on improving SPARC to suit real-world HRI.

Nevertheless, we argue that SPARC allows for easy and safe teaching due to the presence and control by the teacher. And the suggestion/correction mechanism with automatic execution of actions allows for a smooth teaching process where the workload on the teacher can decrease over time as shown in [11]. The workload of the teacher when starting is relatively high, when the robot has no information on which actions to take yet, and decreases over time requiring only limited intervention by the teacher.

6.3. Recommendations for designing interactive machine learning for human-robot interactions

From observing the participants interacting with both systems, we derived four recommendations for future designs of interactive learning robot. Although the study here used a simulated robot, we believe these to be also relevant for real-world, physical installations.

6.3.1. Clarity of the interface

Algorithms used in machine learning often need precisely specified inputs and outputs and require an internal representation of the world and policies. These variables are often not accessible to a non expert: the weights of a neural network or the values in a Q-table are not easily interpreted, if at all. The inner workings of the machine learning algorithms are opaque, and people only have access to input and output of the black box that is machine learning. As such, care needs to go into making the input and output intuitive and readable. For example, in this study (following Thomaz and Breazeal's original study), the communication between the robot and the teacher occurred through the environment: using clicks on objects rather than buttons on a graphical user interface. This design decision has important consequences as participants first have to familiarise themselves with the interface: how to interpret the robot's behaviour, what actions are available for each state and what is the exact impact of the actions? This lack of clarity leads to a high number of failures and high teaching time during the first session in our study. So we argue that to avoid this precarious discovery phase for the teachers, roboticists have to design interfaces taking into account results from the Human Factors community as advocated by [27].

6.3.2. Limits of human adaptability

Human-Robot Interaction today is facilitated by relying on people adapting to the interaction, often making use of anthropomorphisation [28]. Roboticists use people's imagination and creativity to fill the gaps in the robot's behaviour. However, human adaptivity has its limits: in our study, often participants adopted one particular way of interacting with the system and they hold on to it for a large part of the interaction. For example, participants clicked on an object requiring two actions to interact with, assuming that the robot had planning capabilities which it did not. Or when the robot was blocked in some cycles (due to constant negative reward in IRL or due to a loop created and not stopped with SPARC), participants kept on trying the same action to break the loop, without really exploring alternatives. For these reasons, if robots are to be used with a naive operator, they need a mechanism to detect these 'incorrect' uses and either adapt to these suboptimal human inputs or they need to inform the user that this type of input is not supported and clarify what human behaviour is appropriate instead.

6.3.3. Importance of keeping the human in the learning loop

Other methods have been used to provide a robot with an action policy, for example [29] argue that instead of having a human teach the robot, interactive behaviours can be extracted from observing human experts interacting and by using big data machine learning techniques on these observations. This approach has shown some promise [30], but we argue that an action policy for human-robot interaction should be able to be modified online by a human. Furthermore, the presence of a human in the loop can allow the machine learning to deal with sensor errors or imperfect action policies. An expert supervising the robot should also be able to prevent the execution of specific actions or force the execution of others. This was one of the important points we considered when proposing SPARC: there is no distinction between a teaching and a testing phase, they are merged into a single phase. The teacher can correct the robot when needed and let it act when it behaves correctly. Participants used this feature of SPARC in this study: many participants corrected SPARC only when required rather than forcing every action, 37.5% of the participants even let the robot complete the task without giving a single command before starting the test to be sure that the robot is ready. So SPARC has been used as a tool to provide online learning to a robot whilst keeping the teacher in control, but reducing the need of intervention over time.

6.3.4. Keeping people in control

Most of the scenario where a robot has to learn how to interact with humans are human-centred: the robot has to complete a task to help a human (such as in socially assistive robotics). In these scenarios, the goal of the learning is to ensure that the robot can complete the task assigned to it, not to provide the robot with tools to learn more efficiently in further interactions. Similarly, participants in our study did not desire to have the robot exploring on its own and learn from its experience, they wanted to be able to direct the robot. Furthermore, a lack of control over the robot's actions can lead to frustration and loss of motivation for the teacher. This human control is especially critical when the robot is designed to interact with other people as undesired actions can have a dramatic impact, such as causing harm for the interaction partners or bystanders. For these reasons, we argue that when designing an interactively learning robot for Human-Robot Interaction in human-centred scenario, it is critical to keep the human in control.

However, a drawback of Interactive Machine Learning is that the human can prevent the algorithm from converging if feedback is not provided correctly. This was also a limitation in the original study [12], as participants can break a converged policy or not create the gradient of Q-Values required for convergence with Q-Learning.

It should be noted that this control does not mean that the robot cannot learn and become autonomous. We take stronger inspiration from Learning from Demonstration, using human input more efficiently to guide the learning, speeding it up and making it safer, especially in the early stages of the learning. The human is in control mainly when the robot is prone to making exploratory mistakes, and can prevent them before they occur, but once the action policy is appropriate enough, the teacher can leave the robot to learn mostly on its own and refine its action policy with limited supervision from a human.

6.4. Future work

We are currently working on a new experiment in which people interacting with a robot in a continuous time and non-deterministic environment. In this experiment, the teacher is able to send commands to the robot, provide rewards and identify features in the environment they consider important. The learning algorithm will take these inputs into account and combine them with interaction metrics to learn. An approach could be to use the actor-critic paradigm: the critic being an objective evaluation of the action results (environmental rewards), and the actor using results from the critic and teacher's guidance to update the action policy.

7. Conclusion

SPARC has been proposed to address the problem of providing a robot with adaptive behaviour whilst guaranteeing that the behaviour expressed by the robot remains suitable for task at hand. To achieve this, a suggestion and correction system has been used to allow a teacher to be in control of the robot at all times whilst not having to manually select every single action. This approach has been combined with reinforcement learning and was compared to IRL, where the operator manually provides feedback and guidance to the learning agent. The results from a user study involving 40 participants show that SPARC can be used to let naive participants successfully teach an action policy. While doing so SPARC requires less teaching time and limits undesired actions during the teaching phase when compared to IRL. Additionally, the workload on users was lower when using SPARC. Based on these results and other observations, we propose four guidelines to design interactive learning robots: (1) the interface to control the robot has to be intuitive, (2) the limits of human adaptability have to be taken into account (robots should detect deadlocks in human behaviours and adapt their way to be controlled or inform the human about it), (3) the operator should be kept in the learning loop and (4) teachers should stay in control of the robot behaviour when interacting in sensitive environment. The first two points can be seen to apply to all robot teaching methods, and should be addressed at the time of designing the interface. By definition, SPARC aims to address these last two points: maintaining the performance of an adaptive system by remaining under progressively decreasing supervision.

Acknowledgements

This work was supported by the EU FP7 DREAM project (grant no. 611391) and EU H2020 Marie Sklodowska-Curie Actions project DoRoThy (grant 657227). Additionally, the authors would like to thank the reviewers and editors for the valuable comments in improving this article.

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] 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.
[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] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Robot Programming by Demonstration, in: Springer Handbook of Robotics, Springer, 2008, pp. 1371–1394.
[14] B.D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Rob Auton Syst 57 (5) (2009) 469–483.
[15] M.E. Taylor, H.B. Suay, S. Chernova, Integrating reinforcement learning with human demonstrations of varying ability, in: The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, 2011.
[16] B. Settles, Active Learning Literature Survey, University of Wisconsin, Madison 52 (55–66) (2010) 11.
[17] D. Chyzhyk, B. Ayerdi, J. Maiora, Active learning with bootstrapped dendritic classifier applied to medical image segmentation, Pattern Recognit. Lett. 34 (14) (2013) 1602–1608.
[18] S. Chernova, M. Veloso, Interactive policy learning through confidence-based autonomy, J. Artif. Intell. Res. 34 (1) (2009) 1.
[19] 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.
[20] 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.
[21] S. Thill, C.A. Pop, T. Belpaeme, T. Ziemke, B. Vanderborght, Robot-assisted therapy for autism spectrum disorders with (partially) autonomous control: challenges and outlook, Paladyn, J.Behav. Rob. 3 (4) (2012) 209–217.
[22] L.D. Riek, Wizard of oz studies in hri: a systematic review and new reporting guidelines, J Human-Rob. Interact. 1 (1) (2012).
[23] 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.
[24] 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.
[25] E. Senft, S. Lemaignan, P.E. Baxter, T. Belpaeme, Sparc: an efficient way to combine reinforcement learning and supervised autonomy, FILM Workshop at NIPS'16, 2016.
[26] T. Kaochar, R.T. Peralta, C.T. Morrison, I.R. Fasel, T.J. Walsh, P.R. Cohen, Towards understanding how humans teach robots, in: International Conference on User Modeling, Adaptation, and Personalization, Springer, 2011, pp. 347–352.
[27] J.A. Adams, Critical considerations for human-robot interface development, in: Proceedings of 2002 AAAI Fall Symposium, 2002, pp. 1–8.
[28] J. Złotowski, D. Proudfoot, K. Yogeeswaran, C. Bartneck, Anthropomorphism: opportunities and challenges in human–robot interaction, Int. J. Soc. Robot 7 (3) (2015) 347–360.
[29] P. Liu, D.F. Glas, T. Kanda, Learning interactive behavior for service robots the challenge of mixed-initiative interaction, in: Proceedings of the Workshop on Behavior Adaptation, Interaction and Learning for Assistive Robotics, 2016.
[30] P. Liu, D.F. Glas, T. Kanda, H. Ishiguro, N. Hagita, How to train your robot-teaching service robots to reproduce human social behavior, in: Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on, IEEE, 2014, pp. 961–968.

Please cite this article as: 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