Robotics: Science and Systems 2020
Corvalis, Oregon, USA, July 12-16, 2020

In-Situ Learning from a Domain Expert for Real World Socially Assistive Robot Deployment

Katie Winkle
and Séverin Lemaignan
and Praminda Caleb-Solly
and Paul Bremner
Bristol Robotics Laboratory
Email: k.winkle@bristol.ac.uk

Ailie Turton
Department of Allied Health Professions
University of the West of England
Bristol, UK

Ute Leonards
School of Psychological Science
University of Bristol
Bristol, UK

***Abstract*—The effectiveness of Socially Assistive Robots (SAR) relies on their ability to motivate particular user behaviours, e.g. engagement with a task, requiring complex social interactions tailored to the needs and motivations of the user. Professionals from human-centred domains such as healthcare are experts in such interactions, but their ability to contribute to SAR development has traditionally been limited to the identification of applications and key design requirements. In this work we demonstrate how interactive machine learning offers a way for such experts to be involved at every stage of design *and* automation of a robot, as well as the value of taking this approach. We present a novel technical framework for in-situ, online interactive machine learning that can be used in ecologically-valid human-robot interactions. Using this framework, we were able generate fully autonomous, appropriate and personalised robot behaviour in a high-dimensional application of assistive robotics.**

I. Introduction

Socially assistive robots (SARs) can be defined as robots which provide assistance through social interaction alongside or instead of physical aid [5]. Example applications include exercise instruction and encouragement (for general fitness/sports [18] and in various types of therapy [13][6]) as well as weight loss coaching [10] and other forms of positive behaviour change (e.g. reducing energy consumption [8]). Domain experts in such human-centred professions (teaching, healthcare etc.) actively effect social influence in the people under their care (students, patients, etc.) through intelligent use of social behaviours. For example, a study with therapists undertaken to inform SAR design demonstrated how they knowingly leverage their credibility and relationship with the patient, as well as personalise their approach (with regards to e.g. style of approach, use of feedback) in an attempt to maximise compliance [19]. How to replicate this domain-expert level of social intelligence on a robot, for autonomous generation of such socially complex behaviour, is an open research question crucial for the development and real-world deployment of SARs.

Many previous works on automating social robot behaviour have attempted to emulate lifelike behaviour by employing models based on human or animal psychology (e.g. [12], [1]), or through observing and then attempting to replicate human-human interaction [18]. An alternative approach is to have a human *teach* the robot how to behave. This is typically achieved using Learning from Demonstration methods in which a human controls the robot to demonstrate the desired behaviour, resulting in a training dataset to which machine learning can then be applied offline (e.g. [11], [17], [4]). State of the art work has gone past this to utilise interactive machine learning (IML) which allows the learning process to occur in realtime, such that the robot can be used and trained simultaneously. Such an approach was recently demonstrated as a feasible method for generating autonomous, socially assistive robot behaviour via the framework of Supervised Progressively Autonomous Robot Control (SPARC) [16].

This work addresses two key interaction features not yet considered in the previous application of SPARC: long-term interaction and personalisation of robot behaviour, such that the resultant system should be able to identify *what* actions to do, *when* and for *who*. As an exemplar longitudinal SAR application, we consider the use of a robot exercise coach to support the UK National Health Service (NHS) Couch to 5km (C25K) programme. The programme consists of undertaking 3x weekly exercise sessions for 9 weeks, with sessions building up from a combination of short runs and walks to a full 30 minutes running, typically delivered via podcast[1]. The walk-run schedule of each session directly provides the basic, functional instructions the robot, our 'C25K coach', must deliver in order to guide users through the programme. From an engagement point of view, the programme combines a relatively boring task (walking/running on a treadmill with no music/video entertainment) with the physical challenge of gradually increasing running time/speed. This requires long-term commitment and engagement from users, hence placing great importance on the robot's ability to provide task feedback and encouragements in a socially intelligent, engaging and supportive manner.

II. Technical Approach

We adopt the IML approach developed by Senft et al. [16], as it was demonstrated to be adequate to efficiently learn

[1] https://www.nhs.uk/live-well/exercise/get-running-with-couch-to-5k/

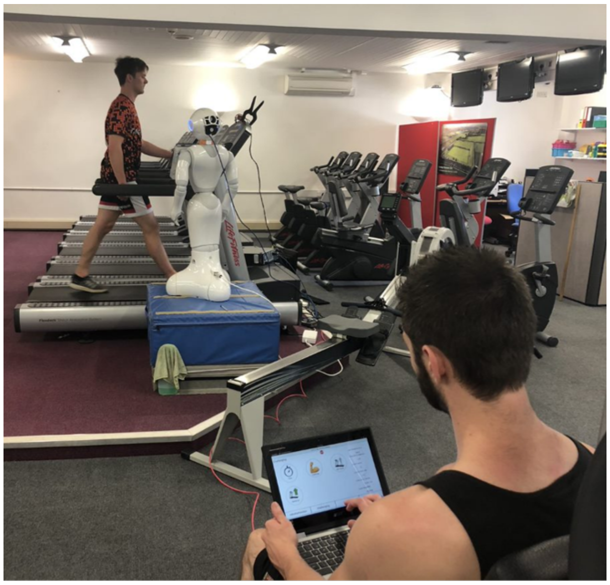

Fig. 1. Our C25K robot coach guiding a participant through an exercise session, supervised by the fitness instructor who can initiate robot actions, and respond to actions suggested by the machine learning system, through the teaching interface.

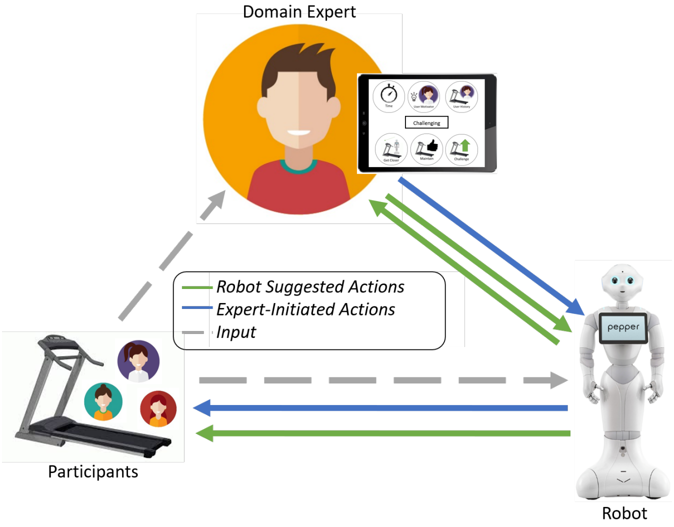

Fig. 2. Expert-in-the-loop machine learning: during the robot-participant interaction, the expert teacher can either *initiate* suitable robot actions (expert exemplars), or the robot itself can *suggest* actions, that are then *validated* by the expert, resulting in $\{$*input space*; *action*; *reward*$\}$ tuples which are added to the training set of the robot's learning algorithm. In our work specifically, the expert's action choice/validation is informed not only by real-time observation of the participant (as per [16]) but also by their overall knowledge of that participant (i.e. their needs/what works well for them) built up from getting to know them pre-training and observing/interacting with them over multiple sessions

a rich action policy (including both task-specific and task-independent social support actions) from a high-dimensional input space, in a relatively short period of time. Figure 2 provides a simplified overview of the interaction flow underpinning this IML technique. Two main operational modes can be distinguished: *supervised* operation, corresponding to the training of the system (pictured), followed by *autonomous* operation. During the supervised phase, the robot interacts with the participant directly, but under the close supervision of the expert. Initially, the robot does not have any action policy, and the expert effectively teleoperates the robot by providing action *exemplars* that are directly executed by the robot, and simultaneously added to the training dataset. After each example, the robot incrementally trains its own model, in order to progressively learn its own action policy. Early on in the process, the robot starts to generate action suggestions, that are sent to the expert for validation. If accepted, these suggestions are positively rewarded; if rejected, they are negatively rewarded. Combined with the expert-initiated exemplars, this helps the robot IML algorithm to quickly converge toward an appropriate action policy. Once the expert is confident that the robot's suggestions are "good enough to be trusted", he/she can 'switch' to the autonomous mode, in which the robot's suggestions are automatically accepted, without any human intervention, resulting in a fully autonomous behaviour.

We propose that such an approach to automation can be considered an extension of traditional participatory design techniques [15]. Typically, techniques such as focus groups, interviews and prototyping are used to allow non-roboticist, expert stakeholders to contribute to robot design, e.g. resulting in proposed use cases, design features and guidelines (e.g. [9], [19], [2]). The automation of robot behaviour however, which requires a significant technical understanding of robotics/artificial intelligence, is not typically considered during such activities. The IML approach specifically addresses this, allowing expert stakeholder(s) to be involved at every stage of the design and development of the autonomous robot, from the initial framing of the input and output spaces of the robot, to the *learning* of an appropriate (autonomous) action policy.

For our C25K coach, the exact role of robot, as well as its action and input spaces, were co-designed with a domain expert (a fitness instructor from the UWE Centre for Sport) over 6 co-design sessions between the first author, fitness instructor and occasionally an additional member of the research team. These sessions were conducted over a period of 5 weeks and represent a total of 12.5 hours direct co-design work as described in Table I.

A. Modelling the Robot Action Space

Conceptually, it was identified that SARs provide two key action types:

1) Task Actions: those which provide direct task-specific instructions to the user, e.g. when to transition between walking and running in the C25K scenario.
2) Social Support Actions: those designed to facilitate and encourage user engagement with the task (essentially socially persuasive behaviours leveraging the social influence of the robot) e.g. providing feedback on task performance, demonstrations of sympathy etc. Note these

Session (length)	Design Activities
1 (2 hours)	Interaction scenario was presented to fitness instructor for initial/un-biased recommendations, researcher then shared pre-prepared suggestions based on previous work to brainstorm initial key actions/inputs
2 (2.5 hours)	Visited gym that will be used for the study Fitness instructor conducted mock Couch to 5km session with SL observed and filmed by KW Physical prototyping of the teaching interface tablet
3 (2 hours)	Took draft tablet teaching interface to the gym, fitness instructor put KW through mock session choosing actions via the tablet interface and verbalising what he was doing/why (action choices were stored via the tablet interface and the instructor's use of the tablet was also video recorded) KW and fitness instructor went through resulting video footage to further discuss what participant information may have been informing his action choice
4 (2 hours)	Went through the action space to discuss specific utterances/examples for each action
5 (1 hour)	Dictionary of specific utterances worked on predominantly by the fitness instructor alone
6 (3 hours)	Tested the experimental set-up with the fitness instructor trying out both the teaching and participant role Made final practical study decisions e.g. fitness instructor placement during the running sessions etc.

TABLE I
KEY PARTICIPATORY/CO-DESIGN ACTIVITIES UNDERTAKEN BETWEEN THE RESEARCH TEAM AND DOMAIN EXPERT TO DEVELOP THE IML ROBOT SET-UP INCLUDING THE NAIVE ROBOT ACTION SPACE, INPUT SPACE AND TEACHING INTERFACE.

might also include low level/non-verbal behaviours, e.g. a change in proxemnis.

These actions can be additionally described and/or shaped by an an overall *mood* or *style*. As such, both Task and Social Support actions can be described by an {*action-type, style-modifier*} pairing, as shown for our C25K coach in Table II. For example, a {*Run, Sympathetic*} action might have the robot say *"Ok now I know you can do this, next is a run for 5 minutes"* whereas a {*Run, Challenge*} action might have the robot say *"Right I want to see you push hard on this next run for 5 minutes"* and a {*Social, Positive*} pairing results in a *Humour* action which might have the robot say *"You can call me Terminator because I'm going to make you run for your life!"*. Further, it was noted that style-modifiers could also inform lower level, non-action specific robot behaviour such as proxemics and non-verbal communication cues. For our C25K coach, styles were used to set robot eye colour as shown in Table II. A brief description of each robots action is as follows:

- Time: encouragingly referring to the amount of time remaining on the run/walk
- Social: a demonstration of 'social support'/interaction such as giving encouragement or telling a joke
- Performance: giving task-specific feedback to speed up, down or stay the same
- Reward: praise the user for their effort/performance
- Check User: ask the user how they are feeling (with user response submitted via icons displayed on the chest-mounted touch screen tablet
- Animation: perform one of Pepper's stock, positively valenced, non-verbal animations (i.e. utilsiing sound, movement and eye colour)
- Get Closer: Pepper 'leans' forward (chest tilts forward from the waist)
- Back Off: Pepper 'leans' backward (chest tilts backward from the waist)
- Run: introducing the length of the next run and counting down into the transition
- Walk: introducing the length of the next walk and counting down into the transition
- Eye Colour: change of eye colour

B. Interaction Features and Input Space

Four key interaction features were identified in considering what input space might be required to inform a socially intelligent action policy covering both Task and Social-Supporting Actions:

1) Task State: describes non-performance related task information e.g. timing information, task state the user should be undertaking (walking versus running).
2) Task Performance: describes if/how well the user is performing the prescribed task.
3) Task Engagement: captures to what extent the user is engaged with the prescribed task, recognising effort as being distinct from performance.
4) User Personality: e.g. big five personality traits [7] and other measures of attitude/motivation.

Full consideration of these categories requires an input space that combines *static* data, that do not change over the course of interaction, and *dynamic* data, that update during the interaction. This is shown for our C25K coach in Table III. Of particular interest are the *Task Engagement* measures, which include both static and dynamic measures. Specifically, these aim to capture an overall motivation with regards to the task and longer term interaction scenario, as well as more instantaneous, in-session engagement/effort.

C. Dual Learner System

The composition of actions as {*action-type, style-modifier*} lends itself to a dual learning system, with the overall *learner* actually containing a *'style learner'* and *'action-type learner'*; with each of these learners wrapping a classification algorithm suitable for use in IML. The output of these can be combined to generate actions, and the output of the style learner specifically can additionally be applied directly to low level behaviours as per Table II.

	Social-Supporting Actions								Task Actions		Low Level
	Time	**Social**	**Performance**	**Reward**	**Check User**	**Animation**	**Get Closer**	**Back Off**	**Run**	**Walk**	**Eye Colour**
P	Time	Humour	Maintain	Praise	-	Animation	-	-	Run	Walk	Green
C	Time	Challenge	Speed Up	-	-	-	-	-	Run	Walk	Yellow
S	Time	Challenge Sympathise	Speed Down	Praise	Check PRE	-	-	-	Run	Walk	Blue
N	-	-	-	-	-	-	Get Closer	Back Off	Run	Walk	White

TABLE II

Full listing of Task and Social-Supporting Actions for our C25K robot coach, all of which can be described by a {*action-type, style-modifier*} pairing, plus application of style modifiers to eye colour as an example of using Style to modify low level behaviour independent of specific actions. Style modifiers: P = Positive; C = Challenging; S = Sympathetic; N = Neutral.

Type	**Feature**	**Values**	**Description**
(Dynamic) Task State	Task Action Type	0, 0.5, 1	Whether participant is in warm-up, walk or run
	Session Progress	0-1	Time spent in session/session duration
	Programme Progress	0-1	Time spent on programme/programme duration
	Programme Action Progress	0-1	Time spent on current walk or run action/action duration
	Programme Action Duration	0, 0.5, 1	Current walk/run action length as $\leq$ 3 mins, $\geq$ 20 mins or other
	Time Since Last Action	0-1	Time since last action/60; capped at 1
Dynamic Performance	Relative Speed: Average	0-1	Current speed/(2 x average speed)
	Relative Speed: Best	0-1	Current speed/(2 x personal best speed)
Dynamic Engagement	Heart Rate	0-1	Heart rate/2x resting heart rate capped at 1
	Motivation/Effort	0, 0.5, 1	Self-reported measure in warmup/on check PRE action
	Facial Expression: Lip Pull*	0-1	Normalised action unit returned by OpenFace
	Facial Expression: Mouth Open*	0-1	Normalised action unit returned by OpenFace
Static Engagement	Elaboration level (self)	0-1	Normalised sum of 3 Likert questions (derived from [anon. ref])
	Elaboration level (expert)	0-1	as above but rated by fitness instructor
	Activity Level	0-1	Likert question response
Static Personality	Extroversion	0-1	Big Five measure normalised with respect to max score
	Agreeableness	0-1	Big Five measure normalised with respect to max score
	Conscientiousness	0-1	Big Five measure normalised with respect to max score
	Emotional Stability	0-1	Big Five measure normalised with respect to max score
	Openness to Experience	0-1	Big Five measure normalised with respect to max score

TABLE III

Input space of the 20 state features implemented for the dual learning system (both style and action class learners utilised the same input space). The facial features marked * were later removed due to unreliability during testing.

D. IML System Architecture

A simplified schematic of the system control architecture is shown in Figure 3. All nodes communicate through the Robot Operating System (ROS) [14] with a number of custom, study-specific ROS message types being implemented to describe e.g. the different types of actions. Communication with/control of the Pepper robot is done using NAOqi, Softbank's multi-platform operating system[2]. All source code is open-source & available online.[3]

1) Teacher Interface: The Teacher Interface (1) is used by the fitness instructor to i) initiate robot actions directly (providing *expert exemplars* as per Figure 2) and ii) respond to learner suggestions (including the suggested styling of Task Actions). Actions allowed to time-out at the interface, receiving no response from the fitness instructor within the given timeframe, are considered *passively accepted* and allowed to auto-execute. The teaching interface is coded in QML and runs on a touch-screen tablet held by the teacher during exercise sessions (shown in Figure 1).

2) External Sensors: Dynamic task data requiring the use of external sensors were addressed as follows:

Heart rate: captured via a polar H10 Bluetooth heart rate sensor[4] worn on the user's chest. Raw output (in beats per minute) was displayed on the teacher interface, but for learning purposes was normalised with respect to user resting heart rate (see Table III)

Treadmill speed: read and automatically digitised from treadmill display using a treadmill mounted camera. Raw output (in miles per hour) was displayed on the teacher interface, but for learning purposes was normalised with respect to users' average and personal best speeds, which were iteratively updated during robot use.

User Perceived Rate of Exertion (PRE): users were asked e.g. 'How are you feeling' by the robot, and asked to respond via the robot-mounted tablet (described below).

Facial expression: real-time extraction of 'lip pull' and 'mouth open' facial action units using OpenFace [3] via a treadmill mounted camera. The robot's camera was not utilised due to the robot's positioning.

An external tablet was used in place of the robot's tablet to

[2] doc.aldebaran.com/2-5/index_dev_guide.html

[3] https://caidin.brl.ac.uk/k2-winkle/engagement_architecture

[4] polar.com/uk-en/products/accessories/polar_h10_heart_rate_sensor

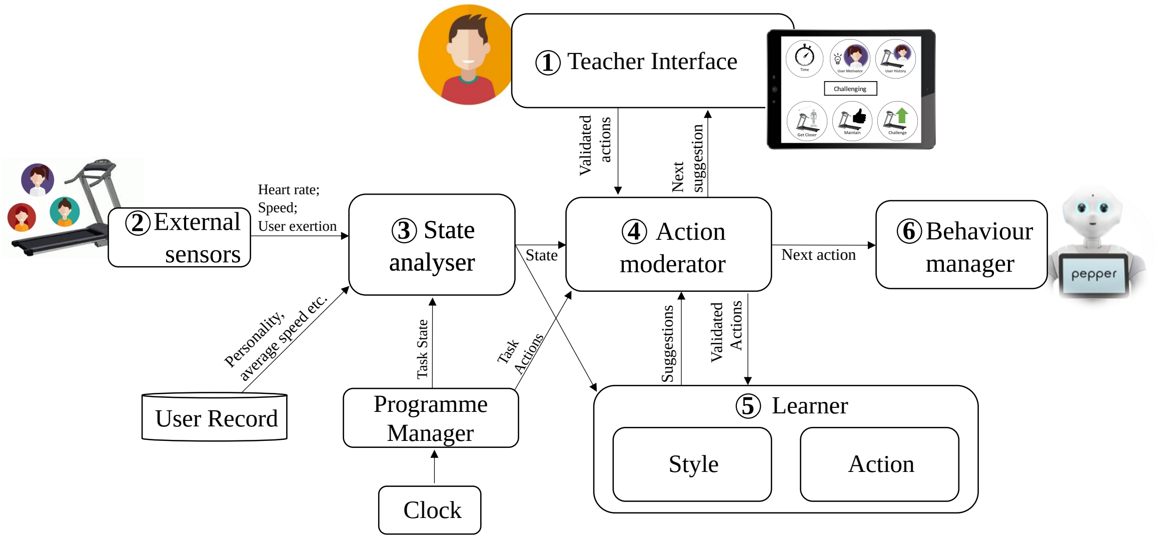

Fig. 3. A representation of the system architecture used for learning from the fitness instructor and ultimately generating autonomous robot behaviour in guiding users through the C25K programme.

allow for direct connection to the ROS architecture. A simple user interface was coded in QML, through which users could trigger start of sessions by selecting their user icon, respond to the robot's Check PRE action (responding to e.g. 'How are you feeling' by selection an appropriate *good, ok, not great* response icon) and view subtitles of the robot's speech (published via ROS messages by the Behaviour Manager (5) at the time of action execution).

Additionally, there are two key databases within the architecture. Firstly, the User Record contains both static (pre-hoc collected data such as personality scores, activity level) and dynamic (e.g. time spent on programme, average and personal best speeds) user data. Secondly, the Programme Database (within the Programme Manager) identifies the timing of Task Actions for each exercise session according to the C25k programme.

3) State Analyser: The State Analyser (3) collects data from all input sources: external sensors, programme manager, internal clock etc. in order to produce the 20-dimensional input state used to describe the interaction (see Table III). The state analyser receives data from sources at whatever rate those sources publish, but is set to publish specifically at 2Hz. This also therefore represents the rate at which the Learner (5) is passed new input states and asked to make suggestions.

4) Action Moderator: The Action Moderator (4) facilitates the passing of actions between the fitness instructor, learner and robot. This includes:

- receiving Task Actions from the programme manager and applying the latest learner-suggested style before sending to the teacher interface for validation
- publishing appropriate *style* and *action* training examples to the Learner (5) based on actions initiated and/or validated at the Teacher Interface (1)
- relaying Learner (5) suggestions to the Teacher Interface (1) managing a priority queue based on action type and limiting the rate of action suggestion
- relaying fitness instructor initiated/accepted learner-suggested actions to the Behaviour Manager (6) managing a priority queue based on action type

5) Learner: The Learner is responsible for accepting training examples and generating suggested Styles and Social-Support Actions. It wraps generalised machine learning algorithm instances (one for *style* and one for *action*), interpreting the input/output between them and the wider system. This makes it easy to switch between different classification algorithms. The Learner is event driven primarily by the receipt of training examples and input states, demonstrated by the example code excerpt presented in Algorithms 1 and 2. We trialled two classification algorithms during testing - a multi-layer perceptron and an adapted KNN algorithm (detailed below); ultimately using the adapted KNN.

Receipt of a new state calls the prediction functions of the style and action learning algorithm instances. The resulting suggestions are then used to compose style and action suggestions. Social-Supporting Action suggestions are composed through combination of the style and action suggestions according to the pairings in Table II; if such a pairing does not exist, then no action is returned. The resultant output for one action and two different styles is shown in the final code excerpt presented in Algorithm 2.

Following [16], on receipt of an expert-initiated/evaluated action, a reward value is generated according to its validation status (see Algorithm 1). Teacher-initiated actions and teacher-accepted learner-suggestions are given a reward of 1. Passively accepted suggestions are given a reward of 0. Learner-suggested actions that are refused by the expert are given a reward of -1. The *action-type* or *style* is then enumerated via a dictionary to create a $\{$*state-label-reward*$\}$ tuple which is added to the collection of instances representing classifier training data. Use of these rewards in the context of making suggestions is shown in Algorithm 3.

Algorithm 1 Processing of an action training example.

```
Input: Action e $(\{action, style\}, state)$
$a$ = e.action
$s$ = e.style
$x$ = e.state
if a.validation == EXPERT_INITIATED or ACCEPTED then:
    $r = 1$
else if a.validation == PASSIVE_ACCEPTED then:
    $r = 0$
else if a.validation == REFUSED then:
    $r = -1$
end if
$c_a$ = (a,x,r)
$c_s$ = (s,x,r)
actionclass_learner.add_instance($c_a$)
style_learner.add_instance($c_s$)
```

Algorithm 2 Example creation of a Social-Support Action suggestion on receipt of a new input state.

```
Input: new state $x$:
suggested_style = style_learner.predict($x$)
suggested_action = action_learner.predict($x$)
if suggested_style ! = NONE then
    create_update_style_action(style)
    if suggested_action ! = NONE then
        if action == SOCIAL then
            if style == POSITIVE then
                suggested_action = [HUMOUR]
                ...
            else if style == SYMPATHETIC then
                action = random_choice[SYMPATHISE, CHALLENGE]
                suggested_action = [SYMPATHETIC, action]
```

Algorithm 3 Adapted KNN algorithm logic for generating suggestions; used for both style and specific action class (shown here as applied to style suggestions). Note that the *threshold* used to decide whether suggestions get proposed to the supervisor is dynamically updated as new tuples are added to the collection of instances (following [16]).

```
Input: $x'$ current state ; $C_S$ collection of style instances $c_s$ = (s,x,r) $S$ ensemble of styles present in $C_S$
Output: suggested style $\pi_S(x)$
for all s ∈ S do
    for all p = (x,r) ∈ $C_S$ do
        compute similarity $\Delta$ between $x$ and $x'$:
        $\Delta$(p) = 1 - $\frac{\sum_{i=1}^{n}(x'(i)-x(i))^2}{n}$

        find closest pair $\hat{p} = argmax_p \Delta(p)$

        compute expected reward $\hat{r}(s)$ for applying $s$ in state $x'$:
        $\hat{r}(s) = \Delta(\hat{p}) \cdot r(\hat{p})$
        where $r(\hat{p})$ is the reward $r$ of the pair $\hat{p} = (x, r)$

        Select the style with the maximum expected reward:
        $\pi_S(x') = argmax_s \hat{r}(s)$

        if $\hat{r}(\pi_S(x')) > threshold$ then
            propose $\pi_S(x')$ to supervisor
```

6) Behaviour Manager: The behaviour manager turns validated actions into explicit robot commands and executes them via the NAOqi ROS bridge. Dialogues to be used in speech-based actions are stored in dictionaries for each {*action, style*} combination. To identify which specific dialogue to execute for a given action, the behaviour logs are checked to identify which dialogue was used last time the current user saw this action. The next listed dictionary entry is then selected, until all dictionary listings have been exhausted in which case the first dictionary entry is used and iteration through the dialogue begins again.

III. Evaluation

The robot was installed in a university gym which was closed to the general public for the duration of the study. The experimental setup is depicted in Figure 1. The fitness instructor involved in co-designing the action/input space was also employed to be the system *teacher* and to observe/facilitate all sessions. 10 participants (4 male/6 female, age range 26 to 60 with mean 36.2) were recruited to take part in the study, with 1 participant (female) dropping out midway through. They were not compensated for their participation. The study was approved by the UWE Research Ethics Committee.

A. Procedure & Testing Schedule

Over the course of the in-the-wild testing, participants worked with the ML system and a heuristic based control. In total, participants saw three versions of the robot coach:

1) ML-Supervised (ML-S): The ML system as controlled/supervised by the fitness instructor, i.e. with him generating unprompted actions and responding to system generated action suggestions to generate training data.
2) ML-Autonomous (ML-A): The ML system allowed to run autonomously, i.e. with no additional actions generated nor suggested actions refused by the fitness instructor.
3) Heuristic (H): A heuristic based system derived through an iterative participatory design process with the fitness instructor, and updated between Phases 1 and 3. This was designed to represent a control condition to generate autonomous, expert-informed behaviour, to compare our machine learning approach against. Note that the Heuristic system utilised a smaller action space than the IML system (as evidenced by Figures 4 and 5). The

fitness instructor was simply unable to include certain actions (e.g. humour, speed up) within the heuristics because: i) no obvious, universal conditions for using such actions could be identified and/or ii) such actions were considered 'risky' by the fitness instructor; e.g. telling a joke at the wrong time could severely negatively impact on user experience and incorrect use of speed up could be unsafe.

The experimental protocol was designed around delivery of the C25K Programme. Exposure to the three versions of the coach was split across three key testing phases. An example for one participant, showing how each test phase related to specific C25K sessions and robot conditions, is given in Table IV. Training data was collected during all ML-S training sessions in Phases 1 and 2 of the study.

- Phase 1 [8 sessions per participant]: Participants alternated between the ML-S and H robots each session, for a total of 4 sessions with each. It was made explicitly clear that the two robots were programmed differently (each robot was colour coded either *orange* or *purple*) but not it was not explained *how* they were different. Condition ordering and colour labelling were counterbalanced across participants.
- Phase 2 [9-13 sessions per participant]: Participants worked exclusively with the ML-S robot as it continued to be trained by the fitness instructor. The robot was still labelled and referred to as either the *purple* or *orange* accordingly.
- Phase 3 [3+ sessions per participant]: Participants unknowingly worked out with ML-A robot for two sessions before the H robot was explicitly re-introduced. Again, to hide the difference between the IML and H systems, participants were told that while they'd been working with e.g. the *purple* robot the 'other half of the group' had been working with the *orange* robot, or vice versa, and now they were once again being given another opportunity to test and compare the two systems.

B. Technical Limitations

1) Facial Expression Input: As the study progressed and participants were running faster, for longer, it became clear that the facial expression tracking started to fail; likely due to vibrations from the treadmill causing the camera image to blur. As such, the two facial expression features listed in Table III were removed from the input vector and the learning agent was re-trained based on all previous training data with those two features removed.

2) Rate of Action Suggestion: As noted previously, successful automation would require the learning system to identify *what* actions, *when* and *for who*. Preliminary testing of autonomous behaviour produced by the system suggested a failure to properly learn the *when*. Specifically the system failed to learn that *do nothing* might sometimes be the appropriate action. Dynamic updating of the policy suggestion threshold failed to impact the rate of suggestions such that suggestions were made every time the learner was passed an input state. This resulted in action suggestions generally being too frequent and/or repetitive.

A 'safety limit' constraining suggestion rate to 1 action suggestion every 10s had already been coded within the Action Moderator to prevent the teacher tablet being rushed with suggestions, as experienced during early testing of the supervised system. So, for Phase 3 of the study we increased this limit to 30s. This matched the action rate of the heuristic system, resulting from iterative testing with the fitness instructor, with our motivation being that it would allow fairer testing of participant experience of the action choices (the *what* and *for who*) made by the learner compared to the heuristic control system.

C. System Performance

In Phase 3 testing we successfully ran 2-4 fully autonomous exercise sessions, i.e. with no fitness instructor and/or researcher intervention, per participant. A full review of system performance within the context of the use case, including e.g. detailed analysis of participant experience, interactions between the learner and the fitness instructor and specific learning algorithm performance etc. is out of scope for this article. However, we present preliminary results suggesting our system successfully learned and automated the choice of *what* actions *for who;* and produced more appropriate behaviour than a heuristic based control.

The nature of our interaction scenario makes it difficult to compare action distributions across conditions as a measure of performance, as exercise sessions dynamic (participant state, like energy level that day, and task requirements, e.g. lots of short run/walks versus longer runs). As such, two 'good' sessions, where the robot acts appropriately, may have very different action distributions. However, the distributions in Figures 4 and 5 provide insight into 1) how well the system learned to replicate instructor behaviour and 2) the personalised behaviour seen across different participants.

Figure 4 shows that the distribution of actions produced by the system when running autonomously (ML-A) was very similar to that produced under supervision/control by the fitness instructor (ML-S); demonstrating the robot successfully learned an appropriate action policy that emulated that of the fitness instructor. Figure 5 specifically shows the action distribution for two participants that the fitness instructor identified as needing different support strategies. It can be seen that the ML-A robot behaved differently with each of them; a Fisher's exact test applied to these two distributions demonstrates that they were indeed significantly different ($p < 0.01$). In contrast, for these particular participants, the heuristic system produced almost identical action distributions.

D. Participant Experience

Preliminary analysis of qualitative participant and domain expert data further supports our conclusion that our system successfully learned and automated the choice of *what* actions *for who;* and produced more appropriate behaviour than a

	Phase 1								Phase 2			Phase 3				
S#	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27
Cdn	H	IML-S	H	IML-S	H	IML-S	H	IML-S	IML-S			IML-A	IML-A	H	IML-A	IML-A

TABLE IV
Experimental testing schedule for one of the participants who completed all 27 Couch to 5km sessions. *S#* is the C25K session number, and *Cdn* is the robot condition.

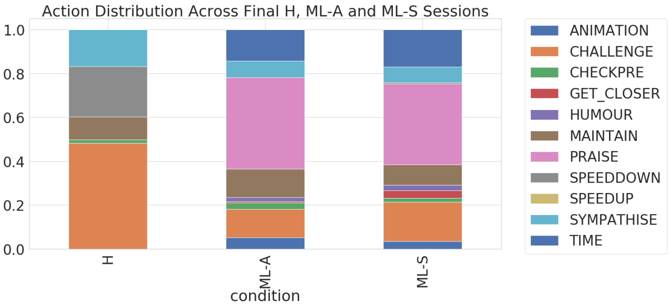

Fig. 4. Normalised action distribution across the Phase 3 heuristic, ML-A and ML-S sessions across all participants. The similarity between the ML-A and ML-S distributions shows to what extent the action policy of the autonomous system replicated that of the fitness instructor. The heuristic system action policy clearly differed greatly, even though it was produced through co-design with the same fitness instructor.

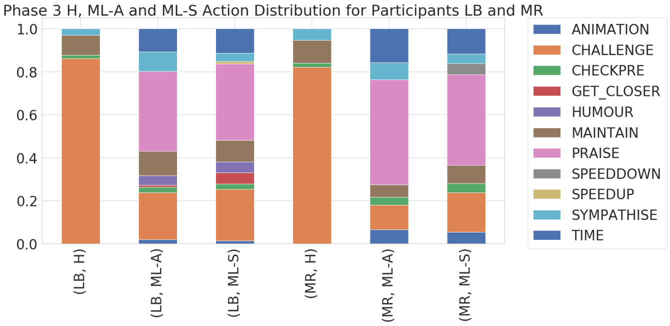

Fig. 5. Normalised action distribution across the Phase 3 heuristic, ML-A and ML-S sessions for two participants identified by the fitness instructor as needing different approaches.

heuristic based control. For example, on the choice of *what* action:

[User PT]: *I think now [the ML robot] is more informed and gives more accurate feedback (i.e.to slow down when I am really tired)*

[User FB]: *After the initial sessions, I felt [the ML robot] included more encouraging/supportive phrases and had a good balance with the phrases designed to push you. They were also well placed within the runs (often towards the end where you might need more motivation).*

[User JF]: *[the ML robot] seemed like it got more helpful in getting me to find a flow*

Such comments suggest that the fitness instructor increasingly tailored the robot's behaviour based on his understanding of what worked well for these participants as he got to know them. This learning is reflected in notes he took for himself through the programme, e.g. *push her/keep her interested, challenge further, starts off way too fast* etc. and the personalised action distributions discussed previously. This represents exactly the kind of intuitive, domain expert social intelligence which is 'unlocked' and captured into the robot system specifically using this IML approach.

IV. Conclusion

In this article we present a socially assistive robot system closely co-designed and then automated with a domain expert using a state-of-the-art interactive machine learning approach. Building on previous works using this approach to go beyond state-of-the-art we:

- extended traditional participatory design techniques to have a domain expert involved throughout design and development of the system
- designed an action/input space that represents a complex model of the social interaction space; considering personality traits, subjective experience and a range of task and engagement metrics - resulting in robot behaviours that were personalised to each participant
- utilised a dual learning approach to split the classification problem across robot *action-types* and *styles*
- tackled a real-world longitudinal interaction scenario in which social support is crucial to engagement, successfully generating autonomous, personalised social robot behaviour

We demonstrated the system in an 11-week longitudinal and in-the-wild study, towards the end of which we successfully ran 28 fully autonomous, robot-led exercise sessions across 9 participants. The robot demonstrated full functional and social autonomy; i.e. providing both task specific instructions as well as social supporting behaviours. On appropriateness of these autonomous behaviours, initial results suggest our system successfully learnt the *what* and *for who* but not did not master the *when*. We suggest this is related to our choice of learning algorithm (a simple extension to a KNN) and our approach (or lack thereof) to encoding *do nothing* as sometimes being an acceptable action policy. We will attempt to address this in future work using data collected during the study.

Acknowledgments

To the fitness instructor, Don, whose enthusiasm, excitement and energy was contagious. Also to Jamie, Adam and all at the UWE Centre for Sport staff who made this possible.

References

[1] R. C. Arkin, M. Fujita, T. Takagi, and R. Hasegawa. Ethological modeling and architecture for an entertainment robot. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 1, pages 453–458 vol.1, May 2001. doi: 10.1109/ROBOT.2001.932592.

[2] S. Azenkot, C. Feng, and M. Cakmak. Enabling building service robots to guide blind people a participatory design approach. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 3–10, March 2016. doi: 10.1109/HRI.2016.7451727.

[3] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 59–66. IEEE, 2018.

[4] Madison Clark-Turner and Momotaz Begum. Deep Reinforcement Learning of Abstract Reasoning From Demonstrations. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 372–372. ACM, 2018.

[5] David Feil-Seifer and Maja J Matarić. Defining socially assistive robotics. In *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference On*, pages 465–468. IEEE, 2005.

[6] Rachel Gockley and Maja J Mataric. Encouraging physical therapy compliance with a hands-off mobile robot. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pages 150–155. ACM, 2006.

[7] Samuel D Gosling, Peter J Rentfrow, and William B Swann. A very brief measure of the Big-Five personality domains. *Journal of Research in Personality*, 37(6): 504–528, December 2003. ISSN 0092-6566. doi: 10.1016/S0092-6566(03)00046-1.

[8] J. Ham and C. J. H. Midden. A Persuasive Robot to Stimulate Energy Conservation: The Influence of Positive and Negative Social Feedback and Task Similarity on Energy-Consumption Behavior. *International Journal of Social Robotics*, 6(2):163–171, April 2014. ISSN 1875-4791, 1875-4805. doi: 10.1007/s12369-013-0205-z.

[9] Simon Jenkins and Heather Draper. Care, Monitoring, and Companionship: Views on Care Robots from Older People and Their Carers. *International Journal of Social Robotics*, 7(5):673–683, 2015.

[10] Cory D Kidd and Cynthia Breazeal. Robots at home: Understanding long-term human-robot interaction. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference On*, pages 3230–3235. IEEE, 2008.

[11] W Bradley Knox, Samuel Spaulding, and Cynthia Breazeal. Learning Social Interaction From the Wizard: A Proposal. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[12] Séverin Lemaignan, Mathieu Warnier, E. Akin Sisbot, Aurélie Clodic, and Rachid Alami. Artificial cognition for social human–robot interaction: An implementation. *Artificial Intelligence*, 247:45–69, June 2017. ISSN 0004-3702. doi: 10.1016/j.artint.2016.07.002.

[13] Norjasween Abdul Malik, Fazah Akhtar Hanapiah, Rabiatul Adawiah Abdul Rahman, and Hanafiah Yussof. Emergence of Socially Assistive Robotics in Rehabilitation for Children with Cerebral Palsy: A Review. *International Journal of Advanced Robotic Systems*, 13(3): 135, June 2016. ISSN 1729-8814. doi: 10.5772/64163.

[14] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an Open-Source Robot Operating System. In *ICRA Workshop on Open Source Software*, volume 3, page 5. Kobe, Japan, 2009.

[15] Selma Sabanovic. Robots in Society, Society in Robots. *International Journal of Social Robotics*, 2(4):439–450, December 2010. ISSN 1875-4791, 1875-4805. doi: 10.1007/s12369-010-0066-7.

[16] E. Senft, S. Lemaignan, P. Baxter, M. Bartlett, and T. Belpaeme. Teaching robots social autonomy from in situ human guidance. *Science Robotics*, 2019.

[17] Pedro Sequeira, Patrícia Alves-Oliveira, Tiago Ribeiro, Eugenio Di Tullio, Sofia Petisca, Francisco S Melo, Ginevra Castellano, and Ana Paiva. Discovering Social Interaction Strategies for Robots From Restricted-Perception Wizard-of-Oz Studies. In *The Eleventh ACM/IEEE International Conference on Human Robot Interation*, pages 197–204. IEEE Press, 2016.

[18] L. Sussenbach, N. Riether, S. Schneider, I. Berger, F. Kummert, I. Lutkebohle, and K. Pitsch. A robot as fitness companion: Towards an interactive action-based motivation model. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 286–293, August 2014. doi: 10.1109/ROMAN.2014.6926267.

[19] Katie Winkle, Praminda Caleb-Solly, Ailie Turton, and Paul Bremner. Social Robots for Engagement in Rehabilitative Therapies: Design Implications from a Study with Therapists. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, pages 289–297, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-4953-6. doi: 10.1145/3171221.3171273.