Learning and Personalizing Socially Assistive Robot Behaviors to Aid with Activities of Daily Living

CHRISTINA MORO and GOLDIE NEJAT, Autonomous Systems and Biomechatronics Laboratory, Department of Mechanical and Industrial Engineering, University of Toronto, and AGE-WELL Network of Centres of Excellence
ALEX MIHAILIDIS, AGE-WELL Network of Centres of Excellence, and Department of Occupational Science and Occupational Therapy, University of Toronto and Toronto Rehabilitation Institute, University Health Network

15

Socially assistive robots can autonomously provide activity assistance to vulnerable populations, including those living with cognitive impairments. To provide effective assistance, these robots should be capable of displaying appropriate behaviors and personalizing them to a user's cognitive abilities. Our research focuses on the development of a novel robot learning architecture that uniquely combines learning from demonstration (*LfD*) and reinforcement learning (*RL*) algorithms to effectively teach socially assistive robots personalized behaviors. Caregivers can demonstrate a series of assistive behaviors for an activity to the robot, which it uses to learn general behaviors via *LfD*. This information is used to obtain initial assistive state-behavior pairings using a decision tree. Then, the robot uses an *RL* algorithm to obtain a policy for selecting the appropriate behavior personalized to the user's cognition level. Experiments were conducted with the socially assistive robot Casper to investigate the effectiveness of our proposed learning architecture. Results showed that Casper was able to learn personalized behaviors for the new assistive activity of tea-making, and that combining *LfD* and *RL* algorithms significantly reduces the time required for a robot to learn a new activity.

CCS Concepts: • **Theory of computation** → **Reinforcement learning**; • **Computing methodologies** → **Cognitive robotics**; **Learning from demonstrations**; *Online learning settings*; • **Human-centered computing** → User centered design; • **Hardware** → Emerging architectures;

Additional Key Words and Phrases: Human-robot interaction, socially assistive robots, robot behavior learning

ACM Reference format:
Christina Moro, Goldie Nejat, and Alex Mihailidis. 2018. Learning and Personalizing Socially Assistive Robot Behaviors to Aid with Activities of Daily Living. *ACM Trans. Hum.-Robot Interact.* 7, 2, Article 15 (October 2018), 25 pages.
https://doi.org/10.1145/3277903

Authors' addresses: C. Moro and G. Nejat, Department of Mechanical and Industrial Engineering, 5 King's College Road, University of Toronto, Toronto, ON, Canada M5S 3G8, and AGE-WELL Network of Centres of Excellence Inc., 550 University Ave., Toronto, ON, Canada M5G 2A2; emails: c.moro@mail.utoronto.ca, nejat@mie.utoronto.ca; A. Mihailidis, AGE-WELL Network of Centres of Excellence Inc., 550 University Ave., Toronto, ON, Canada M5G 2A2, and Department of Occupational Science and Occupational Therapy, University of Toronto and Toronto Rehabilitation Institute, University Health Network, Toronto, ON, Canada M5G 1V7; email: alex.mihailidis@utoronto.ca.

2573-9522/2018/10-ART15
https://doi.org/10.1145/3277903

INTRODUCTION

For robots to effectively function in human-centric environments, they must have the social behaviors necessary for interacting with people. This is especially true of socially assistive robots that use social interactions to assist vulnerable populations, such as individuals living with cognitive impairments, as the quality of a robot's behaviors, particularly its speech and gestures, directly influences the effectiveness of the assistance provided [1]. Socially assistive robots can assist with a variety of activities, including aiding seniors with meal eating [2], facilitating cognitively stimulating activities in long-term care facilities [3, 4], assisting stroke patients with their rehabilitation programs [5], and providing social therapy to autistic children [6].

The behaviors of socially assistive robots have traditionally been designed using one of three methods: (1) manually hand-crafting combinations of speech, gestures, and other communication modes necessary to display a behavior [7–10]; (2) teaching a robot multimodal behaviors through learning from demonstration (*LfD*) [11, 12]; or (3) autonomously learning multimodal behaviors via reinforcement learning (*RL*) algorithms [13, 14]. Manually preprogramming robot behaviors involves tedious annotation, without the potential for expanding the robot's skillset once the robot is deployed in an environment. *LfD* and *RL* algorithms allow robots to learn behaviors without having to preprogram them. However, they may require large numbers of interactions with demonstrators (e.g., *LfD*) or intended users (e.g., *RL*) for training purposes, which may not always be available or feasible. With respect to the latter, it is not always safe for vulnerable users to engage with a robot that has not been fully trained. In addition to learning general assistive behaviors, socially assistive robots may also have to adapt their behaviors to their specific users, as behavior personalization can positively affect robot acceptance [5, 7] and increase its use over time [7].

Only a handful of work has focused on personalizing assistive robot behaviors to user profiles [5, 15–17]. Behaviors have been personalized to either a general user group, for example, extroverted versus introverted users [5], or to a user state during an activity, such as stress level during a memory game [15]. Personalization of assistive robot behaviors to a single user's cognitive model has yet to be investigated. This form of personalization allows a robot to effectively assist cognitively impaired users by adapting the level of assistance provided to a user's cognitive requirements and abilities. Our aim is to develop an effective method for teaching a robot personalized assistive behaviors to provide person-centered care.

Research has shown that challenging behaviors in persons with cognitive impairments, for example, distress and apathy, occur more frequently in care settings lacking person-centered care [18]. Person-centered care involves providing assistance that is adapted to the individual's personality and physical, psychological, and social needs. We aim to develop an assistive robot to provide such person-centered care by personalizing the robot's behaviors to the user's cognition, with the goal of maintaining overall well-being.

A user with cognitive impairment also has a higher likelihood of using the robot in the long term if the robot adapts to his or her physical and mental needs, which can change over time. Research in human-robot interaction (HRI) has found that a technology's ability to adapt to users directly affects their attitude toward the technology [7], which in turns influences whether they will use the technology. Robot behavior personalization, therefore, contributes to promoting positive, sustained use of the technology and can lead to higher levels of engagement and compliance during assistive activities. This has been demonstrated through user studies involving cognitively impaired seniors and robots assisting in recreational activities. For example, the Bandit robot adapted the difficulty of a music game to the cognitive state of seniors with dementia in order to maintain high levels of engagement during the activity and maximize their performance levels [19].

The majority of robots in development today have been given behaviors generalized across groups of people. These robots typically do not consider a user's individual needs. However, generalized behaviors may not be appropriate for individuals with cognitive impairments, as they each have their own set of symptoms and limitations. Incorporating a person-centered care [18] approach to assist individuals with cognitive impairments allows a robot to personalize its behaviors and functions to an individual's cognitive needs and preferences. This promotes the development of a robot that is both engaging and easy to use.

Without personalization, the user may not enjoy using the technology and stop using the robot to assist with activities of daily living. The user may also become disengaged in the activity if it does not meet his or her personal needs [19]. In the event that a senior perceives the robot as being unhelpful, he or she will not use the technology adequately and may encounter substantial difficulties in accomplishing basic activities of daily living essential to his or her well being. It is therefore important for the robot to adapt and personalize its behaviors to the senior's cognitive state to promote the robot's continued use in the long term.

In this article, we present a novel architecture for a robot to learn personalized, socially assistive behaviors to help users living with cognitive impairments complete activities of daily living (ADLs). The architecture is used to learn the assistive behaviors as well as the verbal and nonverbal communication modes used to implement them. Furthermore, it allows the robot to learn to personalize these behaviors to an individual user's cognition. The uniqueness of our approach consists of developing a learning architecture combining *LfD* and *RL* algorithms for learning assistive robot behaviors personalized to a user's cognition when the number of user-robot interactions is limited. While other HRI work has used both *LfD* and *RL* algorithms to teach robots behaviors, these have typically focused on game-based activities requiring a large number of training iterations [20] or on personalizing behaviors to general user groups [16], rather than an individual user's cognition. To the best of our knowledge, this work is the first to investigate the combined use of *LfD* and *RL* algorithms for person-specific personalization of assistive robot behaviors.

In the proposed architecture, a set of generalized robot behaviors are first learned from expert demonstrations using an *LfD* approach. The learned behaviors are labeled according to their verbal and nonverbal content; for example, a demonstrated behavior may contain assertive speech with a large number of gestures, whereas another demonstration of the same behavior may contain suggestive speech with very few gestures. The robot then learns a personalized policy for selecting the appropriate labeled behavior to achieve a desirable user state using a user cognitive model and a *Q-learning approach*. We first present the architecture and subsequently investigate the validity of the proposed architecture though an assistive tea-making activity for seniors with dementia.

RELATED WORK

In this section, we discuss the existing approaches that have been used for the design of robot behaviors in HRI applications.

Manually Preprogramming Robot Behaviors

Robot behaviors can be preprogrammed by manually generating robot behavior models based on large quantities of annotated human behavior data. For example, in [21], patterns of gaze aversion in two-person dialogues were investigated to obtain the exact spatial (i.e., gaze direction) and temporal (i.e., proportion of time spent in a gaze direction) parameters used by humans during conversation. In total, twelve 45-minute videos of single speakers and four 180-minute videos of three speakers conversing were manually annotated. A follow-up study implemented the

identified sequences of gaze aversion and direction on an NAO robot by manually defining the speed and direction of the robot's head rotations during conversation with a human [9]. The robot was given three gaze behaviors: cognitive (to give the impression the robot was thinking about a response), floor management (which occurred at the beginning of a speaking turn or during a pause), and intimacy modulating (all other instances). During interview-type conversations with 30 participants, the robot could either display the correct gaze aversion behaviors, no gaze aversion, or poorly timed gaze aversion. The robot that showed appropriate gaze aversion seemed more intentional and natural to participants. In [10], the synchronization between gestures and speech from videos of TED Talk speakers were investigated to identify a model of multimodal communication patterns used by humans. Several videos were manually annotated for gesture classification, gesture duration, grammar, sentence components, and style (e.g., excited vs. calm). A speech-gesture synchronization model was created by parsing sentences into common keywords and associating these keywords to gestures. The model was then implemented on the humanoid ASIMO robot to evaluate its ability to expressively communicate.

Manually replicating human behaviors onto social robots from large quantities of annotated videos can be a time-consuming process. Furthermore, the existing approaches do not allow a robot to learn new behaviors or adapt its behaviors over time to different individuals once the behaviors have been programmed.

Teaching Robot Behaviors through Demonstrations

While the majority of work in *LfD* has focused on teaching robots physical tasks [22–25], a handful of researchers have also considered teaching robots social interactions from demonstrations [11, 12]. For example, in [11], videos of 16 demonstrators narrating paper making to another person were recorded and manually coded for four classes of gestures (deictic, iconic, metaphoric, and beat) and four classes of gaze directions (reference, recipient, narrator's own gesture, and other). Speech was also coded into lexical affiliates for each gesture class. The coded demonstrations were used as input to a dynamic Bayesian network (DBN) that learned the most probable gesture and gaze direction given a segment of speech. The authors noted large variances in the way demonstrators displayed each behavior, making it difficult to identify a single appropriate gesture and gaze direction with high probability. Therefore, to replicate the behaviors onto the human-like Wakamaru robot, they identified the most common gesture made and created a similar gesture for the robot by manually moving its arms and tracking the gesture trajectory. Given a segment of speech, the robot learned which gesture and gaze direction to display using the learned probabilities from the DBN. In [12], a Robovie II humanoid robot was taught how to interact with customers in a camera store using data from 178 interactions between a seller and customer. Both the seller's and customer's speech, motion, and spatial formations were autonomously clustered into joint behavior states using dynamic hierarchical clustering. The clusters were used in a variable-length Markov model to predict possible behaviors for the robot.

For the aforementioned approaches, the robots learned the manner in which to display a behavior based on the demonstration having the highest probability. In general, the limitation with *LFD* is that there may only be a few demonstrations and the demonstrations can be suboptimal [26].

Teaching Robot Behaviors through Reinforcement Learning

Reinforcement learning can be used to allow a robot to learn optimal behaviors through interactions with users. For example, with respect to behavior learning, in [14], the Furhat robot, a tabletop robotic head, used Q-learning to determine the optimal combination of communication modes (speech, head gestures, gaze direction, and facial expressions) required to direct a person's attention in a memory game. The participant's gaze and speech were used as state inputs along

with the game state. Combinations of communication modes were selected according to an ϵ-greedy exploration strategy. Costs were assigned to each of the communication modes, while a positive reward was assigned if the user redirected his or her attention to the game. The robot's goal was to minimize the overall cost while increasing the number of positive rewards. In [13], the Pepper robot used Q-learning to learn to gain a person's attention in a public space using a combination of speech, gestures, and gaze direction. The robot was placed in a public environment for 14 days, where it interacted with passersby. Then, a deep Q-network was trained offline by sampling from the interaction data. The robot successfully learned which combination of communication modes had the maximum likelihood of getting a person's attention after approximately 14,000 interactions. While these Q-learning approaches have been used to learn general behaviors for all users, *RL* techniques have also been used to learn personalized robot behaviors as discussed in the next subsection.

Robot Behavior Personalization Using an RL Algorithm

RL algorithms can be used to learn a user-specific policy, which can improve the effectiveness of HRI through personalization [23, 24]. For example, in [17], Q-learning was used by the human-like ARIO robot to identify the optimal combination of its gestures (e.g., head shake, arm wave), speech (e.g., call the person's name, make a sound), and navigation (e.g., move to user, move in user's field of view) to obtain a user's attention while he or she read. A Hidden Markov Model was trained to identify the user's state based on face direction, body direction, and speech. After 26 interactions, the robot developed user-specific policies based on the way users preferred to be interrupted while reading. For instance, one user preferred having his name called out continuously, whereas another user preferred having his name called out once followed by a wave. In [15], a MAXQ hierarchical reinforcement learning (HRL) approach was used by the human-like Brian 2.0 robot to learn assistive behaviors for a cognitive training game. Learning occurred in two stages: first, the robot learned appropriate assistive behaviors for the different game states through offline learning, and then it used online learning during interaction with a user to personalize its behaviors to the user's stress levels by measuring heart rate. These studies adapted the robot's behavior according to specific activity parameters; however, they are difficult to generalize to other types of activities.

Other research has focused on adapting the robot's behavior to general user types. For example, policy gradient reinforcement learning was used to adapt the child-like Bandit robot's behaviors to extroverted or introverted user personalities during a stroke rehabilitation activity [5]. The robot's speech, speed, proximity to the user, and gender of voice were manually designed to represent extroverted and introverted personalities. During interactions with users, the robot learned to select extroverted or introverted versions of each parameter to yield the highest compliance rates while developing a model for personality matching.

Rather than personalizing behaviors to general user types, focus groups conducted with seniors suggest that socially assistive robots capable of adapting their behaviors to a user's specific level of impairment could lead to increased technology acceptance, higher intentions for use, and a more overall positive attitude toward these robots [7]. Our research focuses on the personalization of assistive robot behaviors to a user's level of cognition to provide person-centered assistance. The use of *RL* algorithms for both behavior learning and personalization would require a large number of interactions with users to develop an optimal policy. For example, in [13], 14,000 interactions were required to learn a policy, and in [5] and [14], the policies never fully converged due to a limited number of interactions. As socially assistive robots interact with vulnerable populations, it is not feasible to train a robot through such a high number of interactions with users. To minimize the number of iterations required for behavior learning, *LfD* can be used to quickly learn an initial

policy for selecting assistive robot behaviors to display based on the user's state, while an *RL* algorithm can be used to optimize the learned policy.

Behavior Learning Using Both *LfD* and *RL*

The use of both *LfD* and *RL* techniques have either focused on learning (1) agent/robot actions during games such as Atari [28] and Angry Birds [20], (2) robot navigation [26], or (3) robot behaviors to facilitate group activities [16]. In the first two cases, the objective has been to reduce the computation time of traditional *RL* algorithms by using suboptimal human demonstrations of the activity to initially shape the policy and then find the optimal policy using an *RL* algorithm. For example, in [28], human demonstrations were used with deep Q-learning to teach a computer to play several Atari games. Initially, the algorithm randomly samples from a set of human demonstration data that include the state, action taken, reward received, and next state at every time step. The demonstration samples are placed in a batch, which is used to iteratively update the neural network and shape the weights. The game is played using the current policy, while the state, actions, and rewards found using the *RL* algorithm slowly replace the demonstration samples in the batch. This algorithm was tested on a series of Atari games and outperformed both other Deep Q-Networks and the highest-performing human demonstrations. In [26], an algorithm called Approximate Policy Iteration with Demonstrations (APID) was developed and tested in both a car driving simulation and a real robot navigation task. APID was initially given a sample of human demonstrations relevant to the activity being learned, which include the state, action taken, reward received, and next state at every time step. The demonstrations were initially used to shape the value function in the Approximate Policy Iteration technique by imposing a set of linear constraints during policy evaluation, after which a policy improvement step was applied. The algorithm outperformed both Least-Squares Policy Iteration and supervised learning in both experiments, irrespective of the quantity of demonstrations provided.

With respect to HRI applications, in [16], *LfD* was used to teach a robot assistive behaviors for facilitating a group Bingo game. An *RL* algorithm was then used to personalize the robot's speech content using persuasion strategies and Thompson Sampling. Nonexpert demonstrators used a GUI to define the sequence of assistive behaviors using known behavior types. Only three activity demonstrations were needed for the robot to learn the appropriate behavior sequence given the user activity state and user assistance request state. The robot learned which of its four persuasion strategies (i.e., praise, suggestion, scarcity, and neutral) was most likely to achieve compliance from users in the Bingo game.

To the best of the authors' knowledge, no robot-behavior-learning architecture has been developed to learn the combination of verbal and nonverbal communication modes necessary to display assistive behaviors and behavior sequences and personalize these behaviors to a user. To address this challenge, we propose to leverage the strengths of both *LfD* and *RL* algorithms. Namely, *LfD* will be used to teach the robot how and when to display assistive behaviors provided by expert demonstrations. The expert demonstrations teach the robot combinations of speech and gestures required to display a behavior and map the behaviors to robot and user states. As previous research using *LfD* has shown, there can exist differences across demonstrators in how such behaviors are displayed [11]. Typically, existing work has focused on identifying the most frequent behavior displayed by the demonstrators. Rather than using the most frequent behaviors, our work focuses on using the unique differences in speech and gestures across demonstrators to personalize the behaviors to a user's cognition. Behavior personalization is done using an *RL* algorithm that learns which demonstrated behavior, labeled according to its speech and gesture types, is most likely to transition the user into a desirable state (i.e., focused on the activity and completing the correct step).

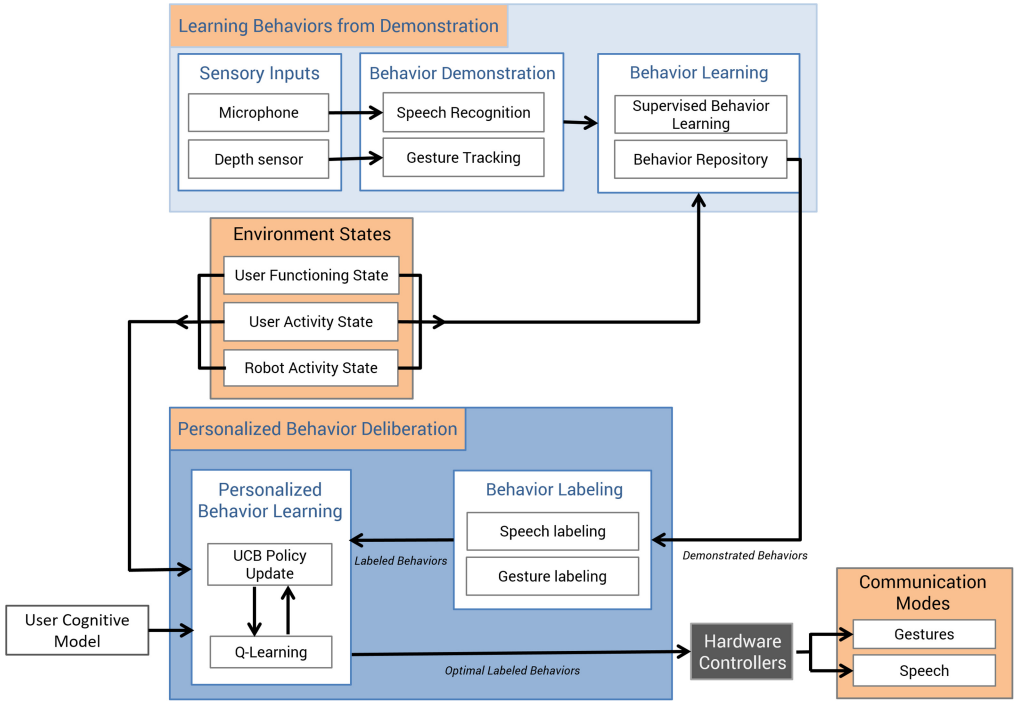

Fig. 1. Socially assistive robot-behavior-learning architecture.

SOCIALLY ASSISTIVE ROBOT BEHAVIOR LEARNING

The proposed robot-behavior-learning architecture is presented in Figure 1. In the *Learning Behaviors from Demonstration* subsystem, the robot uses expert demonstrations to first learn the combination of speech and gestures that make up specific assistive behaviors and, second, the environment state-assistive behavior pairs identifying when the robot should display each assistive behavior based on both the robot activity state and user state. The set of all environment state-assistive behavior pairs is stored in a behavior repository. These demonstrated behaviors are labeled according to their speech content and levels of movement activity in the *Personalized Behavior Deliberation* subsystem. In this subsystem, Q-learning coupled with an Upper Confidence Bound (UCB) strategy for exploring behavior selection is used to teach the robot which of the labeled behaviors transition the user to desirable states based on the *User Cognitive Model*. A policy for selecting the appropriate labeled behaviors based on user cognition is learned, and the selected behavior is displayed on the robot by imitating the speech and gestures initially learned from demonstrations.

Learning Behaviors from Demonstration

To teach the robot new behaviors, expert demonstrators physically perform a demonstration of each behavior required for a given assistive activity in front of the robot. The demonstrator's speech is recorded through a microphone, while a depth sensor is used to track and record their gestures. These demonstrated assistive behaviors are stored in a behavior repository, along with the respective environment states. The robot learns to display behaviors by imitating the combinations of speech and gestures from the behaviors stored in the behavior repository. Supervised learning, in the form of a Classification and Regression (CART) decision tree classifier, is used to

learn the environment state-assistive behavior pairs, i.e., a policy to select behaviors during an assistive activity. The robot learns how to display behaviors for the assistive activity by imitating the demonstrated speech and gesture combinations, while also learning when to display assistive behaviors based on the environment states.

User States

The user model represents two states: user functioning state, s_{fnc}, and user activity state, s_{ac}, such that $s_u = \{s_{fnc}, s_{ac}\}$. The user functioning state is one of five mental functioning states known to be displayed by seniors with cognitive impairments while performing ADLs [29–31]: focused, distracted, having a memory lapse, showing misjudgment, or being apathetic. The user activity state is defined as one of the possible actions performed by seniors with cognitive impairment during ADLs [29–31]: successfully completing a step, being idle, repeating a step, conducting a step incorrectly, or declining to continue the activity. The desired user state is $s_u = \{focused,\ successfully\ completing\ a\ step\}$. The cognitive and activity states defined herein are the characteristics identified in persons with dementia that inhibit them from independently completing ADLs [29–31].

Activity Model

The activity is represented as a set of M sequential robot activity states, $s_r = \{s_r^1, s_r^2, \ldots, s_r^M\}$. The robot activity states represent individual steps in the assistive activity, which can include, for example, initiating the activity and instructing a particular activity step. Transitions from one robot activity state to another are a function of both the robot activity state and the user state, i.e., $s_r' = f(s_r, s_u)$. The robot will transition to the next activity step if the user is in a desirable state; otherwise, it may remain in the same state or choose to skip a step.

Robot Model

The robot is equipped with a set of N behaviors it needs to learn, $B = \{b^1, b^2, \ldots, b^N\}$. Each behavior i is composed of a set of n communication mode combinations, each containing a combination of speech and gestures displayed by an expert demonstrator, i.e., $b^i = \{cm_1^i, cm_2^i, \ldots, cm_n^i\}$. Combination cm_j^i is represented as a function of the robot's arm joint angles (θ) and speech (sp), i.e., $cm_j^i = f(\theta, sp)$.

Environment State-Assistive Behavior Mapping

The *Behavior Learning* submodule uses supervised learning, in particular a CART decision tree [32], to learn the environment state-assistive behavior mapping for the given assistive activity. A CART decision tree was used as it can provide accurate results even with a small number of demonstrations [33] and can easily handle outliers due to different interpretations or variations across multiple demonstrators without overfitting [34].

In our *Learning from Demonstration* module, the demonstrated behaviors b^i are the targets (i.e., classes) and the environment states, $s = \{s_r, s_u\}$, are the features. CART samples from the pairs of demonstrated behaviors and environment states stored in the *behavior repository*, i.e., $\{s^i, b^i\}$, to learn the behavior classifications.

An example CART decision tree is shown in Figure 2. The root node contains all the environment state-assistive behavior pairs stored in the *behavior repository*. The environment state-assistive behavior pairs contained in each node will be referred to as the node sample. At the root node, a splitting feature is selected to classify behaviors, which consists of either the robot activity state, user functioning state, or user activity state. The samples that satisfy the splitting feature are moved down the left branch; otherwise, they are moved down the right branch into two new nodes, shown in Figure 2. Each node is characterized by its impurity H, which measures the homogeneity

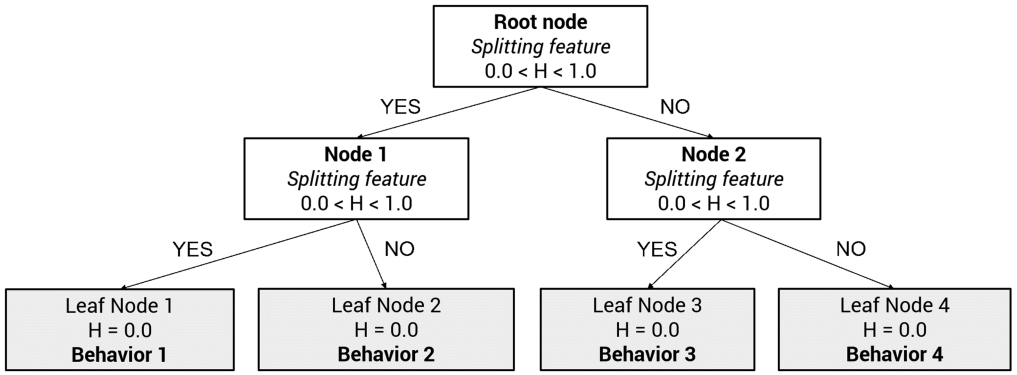

Fig. 2. Example CART decision tree with four behaviors.

of behaviors contained in a node's sample [32]:

$$H = \frac{z_l}{Z} G(X_l) + \frac{z_r}{Z} G(X_r), \tag{1}$$

where z_l and z_r are the number of environment state-assistive behavior pairs moved down the left and right branches, and Z is the total node sample size. The node impurity is based on the Gini Index, $G(X)$, which measures the sample impurity. Both the node and sample impurities vary between [0.0, 1.0]: if the sample, X, only holds one behavior (i.e., is pure), then $G(X) = 0.0$; if a sample contains two equally partitioned behaviors, then $G(X) = 0.5$. The Gini Index for each behavior b^i in node m is [32]

$$G(X_m) = 1 - \sum_{b^i} p^2_{mb^i}, \tag{2}$$

where p_{mb^i} is the proportion of samples containing behavior b^i in node m:

$$p_{mb^i} = \frac{1}{Z} \sum_{b^i}^{B} I(b^i). \tag{3}$$

A leaf node, or terminal node, is reached when $H = 0.0$, i.e., when there is only one behavior contained in the node sample, indicating a behavior has been fully classified (Figure 2). In the end, we have at least one leaf node for each possible behavior.

The developed CART decision tree provides a list of binary rules for classifying behaviors. Given an environment state, the tree predicts the appropriate behavior for the robot to display. However, as each expert demonstrator demonstrated each behavior differently, the robot also needs to determine which behavior variation is most appropriate for a given user. A user personalization model is obtained in the *Personalized Behavior Deliberation* subsystem.

Personalized Behavior Deliberation Using Reinforcement Learning

Once a set of demonstrated behaviors is learned in the *Learning Behaviors from Demonstration* subsystem, each learned behavior is autonomously labeled according to its speech content and gestures. The labeled behaviors can then be used to personalize the robot's behavior selection to the user's cognition. The robot's personalized behaviors are inspired by the Eysenck Personality Model [35], which defines personality in terms of extroversion and introversion. The model has been successfully used in previous HRI studies. For example, in [5], a study involving a robot using such personality types to coach individuals during a stroke rehabilitation exercise found that participants preferred robots with similar personalities to theirs. In our work, rather than define

binary personality types, we use a spectrum for speech and movement activity to define user preferences. Namely, speech and gestures are used to represent demonstrativeness, enthusiasm, and shyness associated with extraversion or introversion. Some individuals may respond better to high levels of demonstrativeness with less enthusiasm when providing directives, while others might prefer high levels of both demonstrativeness and enthusiasm. In our model, the spectrum includes low to high levels of assertive speech and low to high levels of movement activity.

The *Personalized Behavior Deliberation* subsystem uses Q-learning with a UCB exploration strategy to personalize behaviors by learning a policy that selects the most appropriate labeled behaviors for that specific user's cognitive model.

Speech Labeling

The sequence of speech used in each demonstration is labeled as either *suggestive, assertive*, or *other*. These labels, l_s, are chosen as users have shown personal preferences for robots that speak suggestively or assertively to them during an assistive activity [5]. An utterance is labeled *suggestive* if it contains propositional wording such as "can you," "if you want," "try," and "maybe," whereas an utterance is labeled as *assertive* if it contains exclusively imperative wording such as "pull" or "fill." Utterances are labeled as *other* if they are neither suggestive nor assertive, for example, asking general social questions such as "How are you doing today?"

Gesture Labeling

Gestures are labeled according to the level of movement activity, l_{ma}. Movement activity refers to the amount of movements shown, i.e., how many gestures a person makes in a given time period [36]. Gestures enhance dialogue [37] and are particularly important for directing attention and establishing the context about an activity between two people [1]. The amount of gestures or movement made reflects the degree to which a demonstrator is directing the user's attention and emphasizing aspects of the environment to explain a concept.

Movement activity levels are defined as high, medium, and low and are measured by calculating the change in joint angles of each arm across two sequential timeframes, where each frame contains the 3D joint coordinates at a given time step, over the course of the entire demonstration. The average change in all arm angles is taken over the entire interaction and is used as the measure of movement activity for that behavior demonstration:

$$\Delta\theta_j = \left\{\left|\theta_{a,p}^{t-1} - \theta_{a,p}^{t}\right|\right\} \forall\ t,\ a,\ p, j, \tag{4}$$

$$l_{ma} = \sum_{j}^{J} \frac{\Delta\theta_j}{J}, \tag{5}$$

where a represents the selected arm, j is the joint, θ_j is the joint angle, p is the joint position, t is the timeframe, and J is the sum of all arm joints.

User Cognitive Model

The user cognitive model represents the cognitive processes governing the user's functioning and activity state transitions. The user functioning transition probability, $T_{fnc} = P(s'_{fnc}|s_{fnc},\ s_r, b_l^i)$, depends on the current user functioning state, robot activity state, and labeled behavior displayed by the robot, $b_l^i = \{b^i,\ l_s,\ l_{ma}\}$. The user activity state transition probability, $T_{ac} = P(s'_{ac}|s'_{fnc},\ s_{ac}, s_r)$, in turn depends on the new user functioning state, previous user activity state, and robot activity state. The user model regulates the state transition probabilities in the *Personalized Behavior Learning* module, where the Q-learning algorithm learns which labeled behavior transitions the user to a desirable functioning state and activity state based on the transition probabilities.

Personalized Behavior Learning

In the *Personalized Behavior Learning* submodule, the robot uses the speech and gesture labels of each behavior learned in the *Learning Behaviors from Demonstration* module to personalize its behaviors to user cognition. Given a particular user cognitive model, Q-learning is used to learn the value of selecting each labeled behavior in an environment state. Q-learning was chosen as it is a model-free strategy that does not require learning the exact state transition probabilities [38], which can be very complex for modeling a person's cognitive processes. Rather, only the value of selecting behaviors in a given state is required.

Q-learning uses a Markov Decision Process (MDP) formulation [38]. Our MDP consists of a tuple $< S,\ B_s,\ R,\ T,\ \gamma,\ \alpha >$, where S is the set of environment states, $s = \{s_r,\ s_u\}\ \forall\ s\ \in S$; B_s is the set of possible labeled behaviors at state s, i.e., $b_l^i = \{b^i,\ l_s,\ l_{ma}\}\ \forall\ b_l^i \in B_s$; $R(s, b_l^i)$ is the reward received for selecting labeled behavior b_l^i while in state s; $T = P(s'|s, b_l^i)$ is the transition probability function based on the environment state (including the robot activity state, user functioning state, and user activity state); γ is the discount factor; and α is the learning rate.

The robot's goal is to choose a labeled behavior that will maximize the probability of the user being in a desirable user state, and thus maximize its reward. The value of all environment state-labeled behavior pairs (s, b_l^i) is given by $Q(s, b_l^i)$. At every step in the activity, the environment state-labeled behavior values are updated according to the Bellman equation:

$$Q\left(s, b_l^i\right) = \alpha\left(R\left(s, b_l^i\right) + \gamma\ argmax_{b_l^i}, Q\left(s', b_l^{i\prime}\right) - Q\left(s, b_l^i\right)\right). \tag{6}$$

The learned environment state-labeled behavior values, (s, b_l^i), are used to develop a policy π through which the robot chooses the appropriate (highest value) labeled behavior b_l^i at every state s. To learn the (s, b_l^i) values, the robot must explore all possible labeled behaviors while still selecting behaviors that yield high rewards. One strategy for doing so is to select all possible labeled behaviors at least once in each state, and subsequently select behaviors based on the probability of yielding a high reward. The Upper Confidence Bound (UCB) algorithm uses such an exploration-exploitation strategy.

The UCB algorithm adapted for *RL* [39, 40] is used here to select the behavior with the highest probability of transitioning to the desired user state. Initially, UCB selects all behaviors with equal probability. Over time, it learns which behavior has the highest probability of yielding the highest reward. UCB is an optimistic policy that minimizes regret, i.e., the difference between the reward from the optimal behavior and the reward received. It has been proven to achieve near-optimal regret and faster convergence than traditional strategies such as decaying epsilon-greedy [39]. Even after convergence, UCB guarantees that all labeled behaviors will be explored at some point in the future, irrespective of how poorly they have performed in the past. No labeled behaviors are permanently ruled out, which is a desirable trait if the user's cognition changes over time.

UCB initially attributes the maximum reward $r_{max}/(1-\gamma)$ to each environment state-labeled behavior pair, where r_{max} is the maximum possible reward from $R(s, b_l^i)$ at any time step. Each labeled behavior is selected once, providing an initial approximation for the empirical mean reward $\widehat{\mu_\iota}$ of each labeled behavior b_l^i. At each time step t, the labeled behavior b_l^i that maximizes $\widehat{\mu_\iota}$ + $\lambda \cdot \sigma$ is selected, where λ is a hyperparameter, and its empirical mean σ is updated by the reward observed [40]:

$$b_{l_t}^i\ =\ \text{argmax}_{b_l^i}\left\{\widehat{\mu_i}\left(s_t, b_l^i\right)\ +\ \lambda \cdot \sigma_i\left(s_t, b_l^i\right)\right\}. \tag{7}$$

The result of the proposed Q-learning with UCB exploration algorithm is a policy determining which labeled behavior is appropriate to display at each state. By using Q-learning with UCB exploration in conjunction with the LfD algorithm presented previously, the robot's assistive behaviors are optimized for both the activity at hand and the user's cognitive profile. By learning

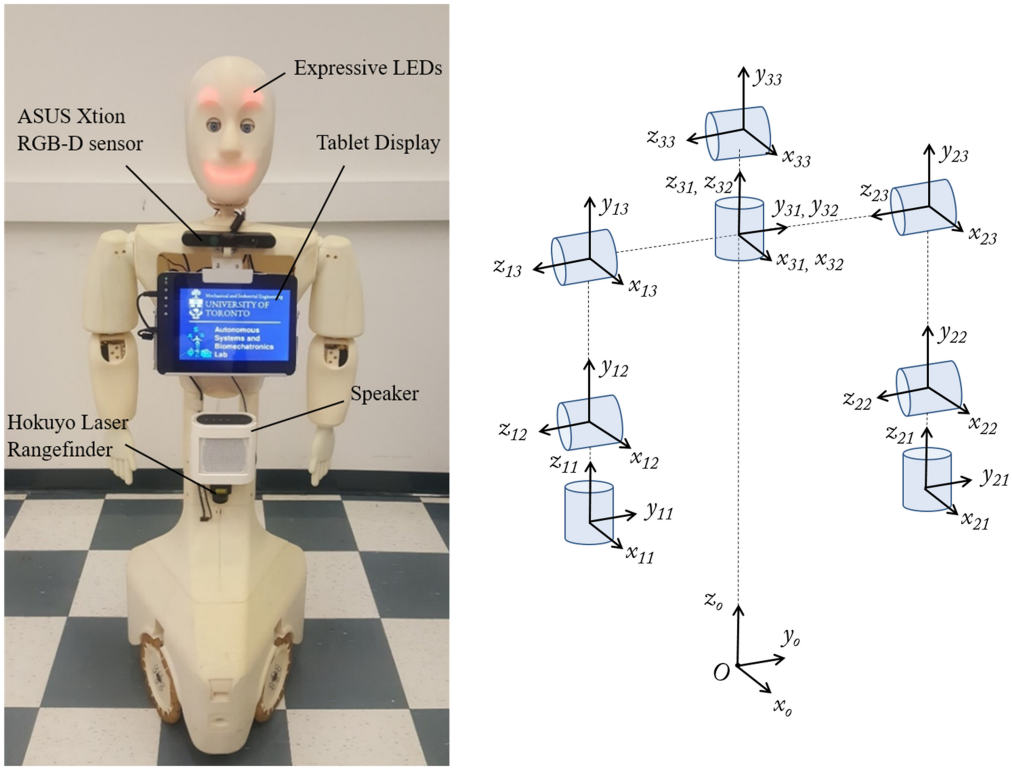

Fig. 3. The Casper robot and its kinematic model.

task-specific behaviors directly from allied health care professionals, the robot learns to perform behaviors appropriate for the task at hand. Subsequently, the robot learns to personalize these task-based behaviors specifically for user preferences. In doing so, we aim to provide optimal robot behaviors.

THE CASPER ROBOT

Our behavior learning architecture was integrated with the socially assistive robot Casper to provide activity assistance. The Casper robot [41, 42] (Figure 3) is a human-like robot with an expressive face, two arms, and a torso mounted on an omnidirectional base. It uses LEDs for its mouth and eyebrows to display five facial expressions: happy, sad, surprised, angry, and neutral. The robot uses the Amazon Polly Text-to-Speech API [43] speech synthesizer. The robot's neck has 2 degrees of freedom (DOF), which allow for nodding and shaking of its head (Figure 3). Casper's two 3-DOF arms are used to display different gestures. An ASUS RGB-D sensor on the robot's torso can be used for person detection and tracking during assistance, while a Hokuyo laser rangefinder is used for environment mapping, robot localization, and navigation. Casper also has a 10" touchscreen tablet mounted on its chest for displaying multimedia such as videos, images, and text.

Robot Gestures

The Asus depth sensor and the ROS OpenNI tracker package [44] are used for skeleton tracking of the demonstrator. OpenNI uses both RGB and depth images to identify the demonstrator and track his or her joint positions in 3D space at a rate of 30Hz. Vectors representing the

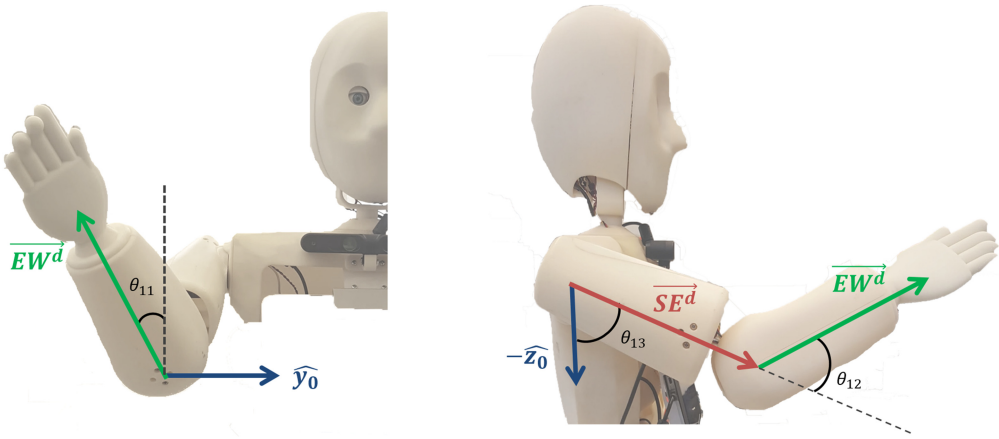

Fig. 4. Representation of the demonstrator's arm joint vectors mapped onto the robot and the corresponding robot joint rotations of its right arm.

demonstrator's wrist ($\overrightarrow{W^d}$), elbow ($\overrightarrow{E^d}$), and shoulder ($\overrightarrow{S^d}$) positions are evaluated with respect to the torso's reference frame. To determine the robot's required shoulder rotation, a vector, $\overrightarrow{SE}^d$, aligned to the demonstrator's upper arm is defined as the distance between the demonstrator's elbow and shoulder positions. The z-component is set to zero as Casper's shoulder has no rotation about the z-axis (Figure 4). Similarly, to determine the robot's elbow rotations, a vector between the demonstrator's wrist and elbow, $\overrightarrow{EW^d}$, is defined as the distance between the demonstrator's wrist origin and elbow origin vectors. These vectors are represented as:

$$\overrightarrow{SE^d} = \left(S_x^d - E_x^d,\ S_y^d - E_y^d, 0\right) \tag{8}$$

$$\overrightarrow{EW^d} = \left(E_x^d - W_x^d,\ E_y^d - W_y^d,\ E_z^d - W_z^d\right). \tag{9}$$

The vectors mapped onto the robot are shown in Figure 4. The dot products between unit vectors $\widehat{y_0}$ and $\widehat{z_0}$, defined according to the robot's fixed reference frame, and vectors $\overrightarrow{SE^d}$ and $\overrightarrow{EW^d}$ are used to compute the required rotation angles (Figure 4). The mapping to Casper's shoulder (θ_{13}, θ_{23}) and elbow (θ_{12}, θ_{22} for elbow pitch and θ_{11}, θ_{21} for elbow yaw) angles for each arm is determined by:

$$\theta_{13}, \theta_{23} = \cos^{-1}\left(\frac{\overrightarrow{SE^d} \cdot -\widehat{z_0}}{\|\overrightarrow{SE^d}\|}\right) \tag{10}$$

$$\theta_{12}, \theta_{22} = \cos^{-1}\left(\frac{\overrightarrow{EW^d} \cdot \overrightarrow{SE^d}}{\|\overrightarrow{EW^d}\|\ \|\overrightarrow{SE^d}\|}\right) \tag{11}$$

$$\theta_{11}, \theta_{21} = \cos^{-1}\left(\frac{\overrightarrow{EW^d} \cdot \widehat{y_0}}{\|\overrightarrow{EW^d}\|}\right) - \pi/2. \tag{12}$$

Speech

Speech recognition is performed using the IBM Watson Speech-to-Text API [45], recorded using an Acoustic Magic Voice Tracker II microphone array. Speech by the demonstrator is segmented into a string of text (i.e., utterance) by detecting at least 1 second of silence between utterances.

ROBOT LEARNING STUDY

In order to investigate the performance of our architecture for robot learning of assistive behaviors, we conducted a teaching-by-demonstration study where teachers demonstrated assistive behaviors to the robot Casper. Participants were asked to demonstrate how they would assist a senior with dementia in preparing a cup of tea in a kitchen environment under different scenarios, while their movements and speech were recorded. The objective of the study was to determine if Casper could effectively learn its assistive behaviors from the different teachers.

Participants

The participants recruited were graduate students enrolled in allied health care programs at the University of Toronto who have been trained in assisting vulnerable populations with everyday tasks. We use this demographic as they will be the future teachers of such socially assistive robots in health care settings. A recruitment flyer was distributed to each respective department and sent to students through a mailing list. Ethics approval from the University of Toronto Research Ethics Board was obtained prior to commencement of the study. The inclusion criteria for the participants were (1) to be a graduate student (master's or PhD) in allied health care fields, (2) to not have any prior robotics experience, and (3) to have clinical training with vulnerable populations (including people with dementia). In total, 15 participants were recruited (*Occupation Therapy = 9, Physical Therapy = 1; Clinical Psychology = 1; Biomedical Communications = 1; Speech-Language Pathology = 2; Kinesiology = 1*).

The Tea-Making Activity Demonstrations

Tea-making was chosen as the assistive activity as it has been identified as an activity of daily living that requires assistance from a caregiver [46]. The overall tea-making activity was composed of 12 discretized steps, which are presented in Table 1, along with five additional behaviors in scenarios where the senior is having difficulty completing the task. The participants were asked to demonstrate to the robot how they would assist a senior with dementia, named Ms. Potts, in preparing a cup of tea.

Procedure

The experiment took place in the kitchen environment shown in Figure 5. Speech was recorded using the microphone array placed on the table. A depth sensor positioned behind the robot was used to track the joint angles of the participants. 2D videos of the experiment were recorded to observe the overall interaction between each participant and robot. Participants were asked to show the robot how they would implement the behaviors for each of the tea-making steps as well as for several additional scenarios as listed in Table 1. As Casper is a socially assistive and noncontact robot, the participants were instructed not to touch or grasp any of the objects. However, they were encouraged to use gestures, such as pointing. Casper then replicated the behavior displayed by the demonstrator after every demonstration (Figure 6). The demonstrator's gestures were mapped onto the Casper robot using the procedure discussed in the subsection "Robot Gestures." The participants were asked to validate the behavior or make corrections to it, if necessary, by redemonstrating the behavior.

Table 1. The Assistive Tea-Making Behaviors to Demonstrate

1.	Invite the senior to make tea.
2.	Instruct the senior to turn the faucet on.
3.	Instruct the senior to fill the kettle.
4.	Instruct the senior to turn the kettle on.
5.	Converse socially with the senior, e.g., asking about his or her day, tea preferences, etc.
6.	Ask if the senior would like sugar in his or her tea.
7.	Instruct the senior to add sugar to his or her tea (if appropriate).
8.	Ask if the senior would like milk in his or her tea.
9.	Instruct the senior to add milk to his or her tea (if appropriate).
10.	Instruct the senior to put a teabag in his or her cup.
11.	Instruct the senior to add boiling water to the cup.
12.	Instruct the senior to stir the contents of his or her cup.
13.	Re-engage a senior who is distracted.
14.	Motivate a senior to finish making tea if he or she wants to end the activity before completion.
15.	Motivate a senior who no longer wants to make tea and did not respond positively the first time.
16.	Correct a senior who is putting a teabag in the kettle (doing a step incorrectly).
17.	Correct a senior who is putting a second teabag in his or her cup (doing an incorrect step).

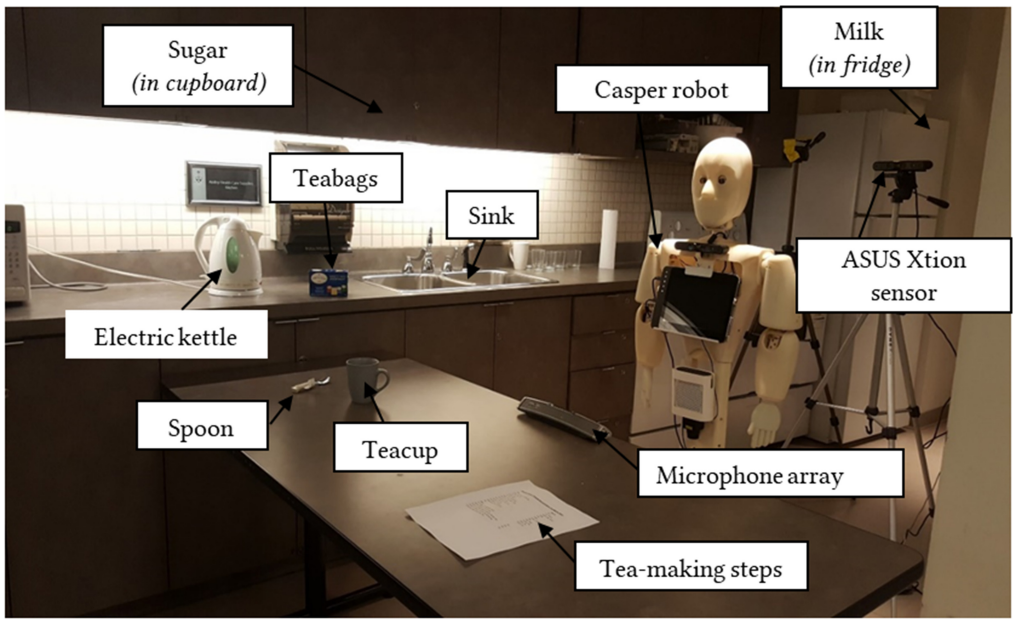

Fig. 5. Study setup displaying the Casper robot, sensors, and items used to make tea.

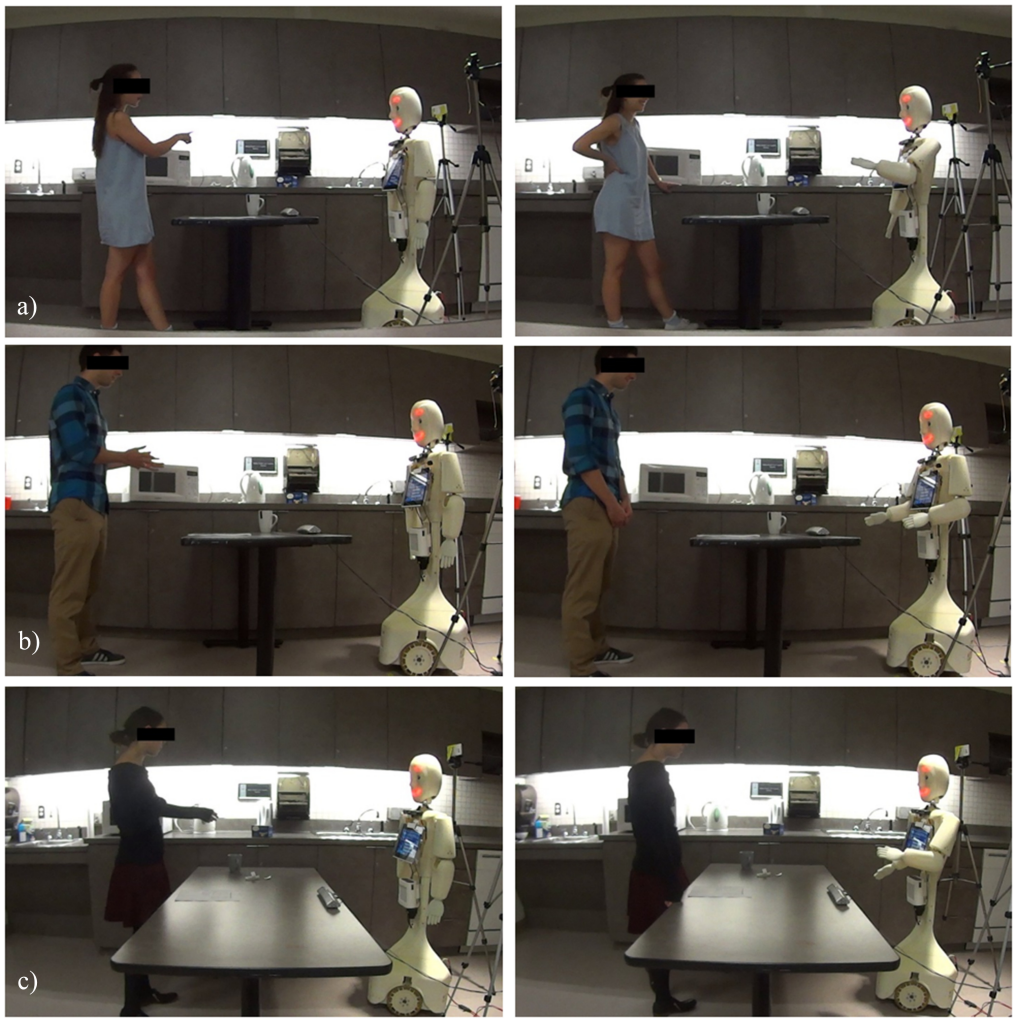

Fig. 6. Examples of demonstration learning sessions with the teacher performing a demonstration (left) and the Casper robot replicating the demonstrated behavior (right). The behaviors demonstrated include (a) pointing to the sugar in the cabinet, (b) conversing socially with the user by asking about his or her day, and (c) pointing to the box of tea on the counter.

In this experiment, interaction refers to the actions taken by the demonstrator and senior at a given activity step, whereas demonstration refers to the specific display of a demonstrator's speech and gestures during an interaction.

Questionnaire

At the end of the overall demonstration session, the participants were asked to complete a short questionnaire on their overall experience (Table 2). In particular, the questions were designed so that participants could provide their opinion on teaching the robot, potential use cases based on their clinical experiences, and areas of improvement during the teaching process. The questions were left open-ended to avoid biasing their responses.

Table 2. Questionnaire

1.	What was your overall experience like in teaching the robot to do this type of activity?
2.	Can you think of any other techniques/tools you could use to teach Casper to assist with an activity?
3.	As a health care professional, how helpful would it be to have a robot take on one of these repetitive activities of daily living?
4.	Can you think of other activities the robot could assist with?

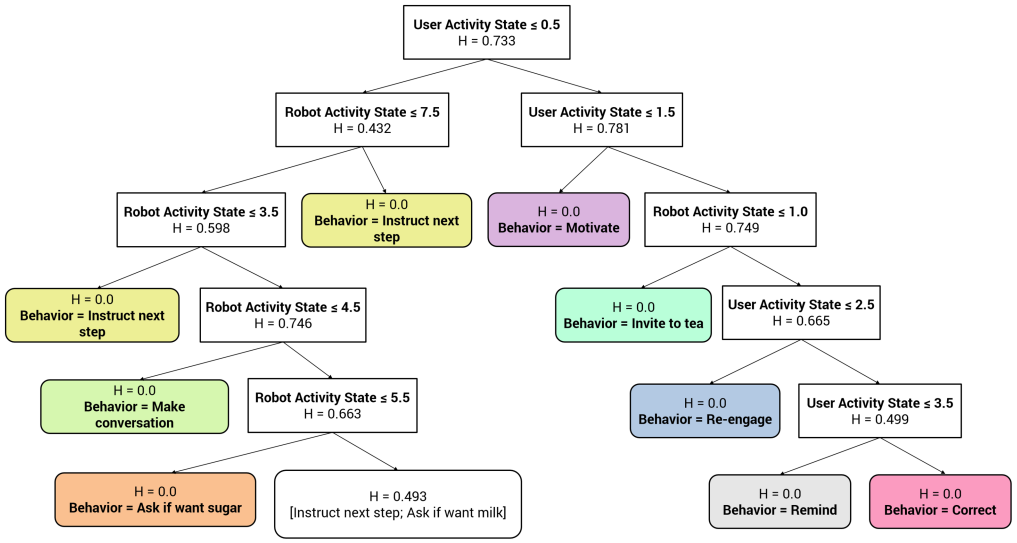

Fig. 7. The decision tree generated by the CART algorithm showing the splitting nodes as rectangles with their splitting attribute and node impurity, and the color-coded nodes representing leaves showing the predicted class.

EXPERIMENTAL RESULTS

Learning from Demonstration

Of the 255 demonstrations provided, 233 demonstrations were usable and therefore were included for training. The CART decision tree was evaluated using 10-fold cross-validation. The cross-validation results show that the appropriate behavior was selected in 93% of the interactions. The final decision tree is presented in Figure 7. A rectangular node represents a branch split, including the respective splitting attribute (robot state, user functioning state, or user activity state) and node impurity. The color-coded nodes are leaves and represent the robot behaviors that were classified. These behaviors are taken from Table 1 where behaviors 2 through 4, 7, and 9 through 12 are grouped together as "Instruct next step."

A sensitivity and specificity analysis was conducted on the behaviors predicted by the decision tree during the k-fold validation. The true-positive rates (TPRs), false-negative rates (FNRs), and false-positive rates (FPRs) are listed in Table 3. For each behavior b_i, the true positives (TP) are the instances where behavior b_i was correctly predicted; the true negatives (TN) are all the other correctly predicted behaviors, the false negatives (FN) are the instances where behavior b_i was incorrectly predicted, and the false positives (FP) are the instances where another behavior was

Table 3. True-Positive Rate (TPR), False-Negative Rate (FNR), and False-Positive Rate (FPR) for the Decision Tree

Behavior	TPR	FNR	FPR
Invite to tea	100%	0%	0%
Instruct next step	100%	0%	12.8%
Ask if the user wants sugar	41.7%	58.3%	0%
Ask if the user wants milk	0%	100%	0%
Make social conversation	100%	0%	0%
Motivate the user to make tea	100%	0%	0%
Re-engage a disengaged user	100%	0%	0%
Remind the user of a step	100%	0%	0%
Correct a user	100%	0%	0%

predicted to be behavior b_i. The TPR, FNR, and FPR were calculated as follows:

$$TPR = \frac{\sum TP}{TP + FN} \tag{13}$$

$$FNR = 1 - TPR \tag{14}$$

$$FPR = \frac{\sum FP}{FP + TN}. \tag{15}$$

The decision tree predicted all behaviors with 100% accuracy except for the "Ask if the user wants sugar" and "Ask if the user wants milk," resulting in a 93% overall behavior prediction rate. The CART decision tree classified the "Ask if the user wants milk" and "Ask if the user wants sugar" behaviors as "Instruct the next step" in five out of the 12 instances. The last leaf node contained both "Instruct the next step" and "Ask if the user wants milk" in its sample set. It also had a greater number of samples containing the "Instruct the next step" behavior, such that the tree has a higher probability of choosing this behavior during classification. Moreover, it is possible that the behavior "Ask if the user wants sugar" also had a splitting threshold close to that of the last leaf node, as they both originate from the same node, which could result in it being classified as "Instruct the next step."

Questionnaire Results

The questionnaire results showed that 93% of participants said it was easy to teach the robot and believed the robot would be helpful to them in assisting older adults. While 80% of participants were satisfied with the teaching process, five participants provided alternative suggestions such as having the robot observe two people making tea or using a hand-over-hand technique where the demonstrator physically moves the robot's arms to demonstrate a gesture. Several additional activities were suggested with respect to the robot assisting. These activities are presented in ranked order: meal preparation, personal toileting, dressing, housekeeping, providing reminders, doing laundry, and bathing.

Personalization Model Results

Each of the demonstrated behaviors stored in the *behavior repository* was labeled according to speech content and movement activity levels. Six user cognitive models were created, each with its own preference in labeled behaviors. The Q-learning algorithm was trained for each of these

Table 4. Label Distribution for Speech Assertiveness and Movement

	Speech			Movement Activity		
	Assertive	Suggestive	Other	High	Medium	Low
Number of interactions	103	95	35	30	122	101
Percentage of interactions	44.21%	40.77%	15.02%	11.59%	48.07%	40.34%
Standard deviation		N/A			2.27	

Table 5. Reward Distribution Based on the Robot State, User Functioning State, and User Activity State

Robot State	User Functioning State	User Activity State	Reward
Any	Any	Idle, Repeating a step, Conducting a step incorrectly, or Declining to continue the activity	−0.4
Terminal state (i.e., Instruct the user to stir their tea)	Focused	Completed the correct step	1.0
Any except terminal state	Focused	Idle or Completing a step correctly	0.0

models, where the value of selecting a labeled behavior in each environment state was learned. The results from our reinforcement learning algorithm are presented in the next sections.

Behavior Labeling

Each behavior demonstration was labeled according to its speech assertiveness and movement activity as shown in Table 4. The speech label distribution was identified to be 44.21% assertive, 40.77% suggestive, and 15.02% other, according to the classification approach discussed above in the "Speech Labeling" subsection. The movement activity label distribution was 11.59% high movement, 48.07% medium movement, and 40.34% low movement. The movement values were defined by the respective joint angles from our demonstrators. These distributions show that the participants greatly varied in how they provided assistance.

Q-Learning Setup

The environment states consisted of the robot activity state and the user state, $s = \{s_r, s_u,\}$, and the actions consisted of the labeled behaviors, $b_l^i = \{b^i, l_s, l_{ma}\}$. In total, there were 300 possible states, based on all possible combinations of user activity state, user functioning state, and robot activity state, and nine possible behavior labels, based on all possible speech and gesture combinations for each behavior. Rewards were provided based on the state the user transitioned into (Table 5). A positive reward was given when the user was focused and completed the activity. A negative reward was given when the user transitioned into an undesirable state. No reward was given in the remaining scenarios. The discount factor γ was 0.8, the learning rate α was 0.3, and the hyperparameter λ was 0.1.

User Personalization

The Q-learning algorithm was trained on all six combinations of different user cognitive models: *User 1* performed a step correctly given assertive, high-movement robot behaviors; *User 2* performed a step correctly given assertive, medium-movement robot behaviors; *User 3* performed a step correctly given assertive, low-movement behaviors; *User 4* performed a step correctly given suggestive, high-movement robot behaviors; *User 5* performed a step correctly given suggestive, medium-movement robot behaviors; and *User 6* performed a step correctly given suggestive, low-movement robot behaviors. Each user was given a different probabilistic cognitive model with its own unique set of probabilities for T_{fn} and T_{ac}. All six user cognitive models were tested to verify that the algorithm would converge across all models.

The labeled behaviors were used as input to the user cognitive model. Given a labeled behavior, the user's next cognitive state was probabilistically chosen from the five possible cognitive states (focused, distracted, having a memory lapse, showing misjudgment, or being apathetic) according to his or her own user-specific cognitive state transition function. The user's cognitive state also probabilistically determined the next activity state. Noise was added to the user cognitive and activity transition probabilities to model the unpredictability that can be caused by dementia [47]. The robot's goal is then to select the most appropriate labeled behavior to transition the user to a desirable state, i.e., "Focused" and "Successfully completing the correct step." All other combinations of user functional and activity states were considered undesirable states.

LfD and Q-Learning Results

The Q-learning algorithm was trained five times on each user cognitive model. The accumulated rewards at each time step in the five training sessions were averaged for each user. The overall cumulative reward per episode, where an episode refers to the entire sequence of making tea starting with inviting the user to make tea and ending by instructing the user to stir the tea, is shown in Figure 8. The Q-learning algorithm was trained for two cases: (1) the user always complied with the robot's behaviors when the appropriate labeled behavior was selected (Figure 8(a) and Figure 9(a)), and (2) the user complied with the robot's behaviors 90% of the time when the appropriate labeled behavior was selected (Figure 8(c) and Figure 9(c)). For the latter, the user entered an undesirable random state the remaining 10% of the time. In Case 1, as can be seen in Figure 8(a), the approach converged to a cumulative reward (upper bound) of 1.0 after an average of 10 episodes (min = 2 episodes, max = 17 episodes) across the user cognitive models. For Case 1, the average number of steps required to complete the tea-making activity was 53 steps (min = 36 steps; max = 86 steps) across user cognitive models. In Case 2, the approach converged, on average, after approximately 30 episodes (min = 21 episodes, max = 45 episodes) and required an average of approximately 57 steps to complete the tea-making activity (min = 15 steps, max = 75 steps) across user cognitive profiles. These results show that the robot was successfully able to determine an appropriate policy for selecting labeled behaviors based on the state, regardless of the user cognitive model.

Comparison of LfD and RL with Only RL

We conducted a comparison study to investigate the performance of using both *LfD* with *RL* algorithms for robot assistive behavior learning versus using only an *RL* algorithm. The same aforementioned experiment was conducted using only an *RL* algorithm (Q-learning with UCB) to learn both the appropriate behaviors and labels with the set of user cognitive models. In the *RL*-only case, the robot initially randomly selects both a behavior and a label and observes the next user state. Based on the rewards received, the robot iteratively learns to select the appropriate labeled behavior. The results of the *RL*-only experiments are presented in Figure 8(b) and Figure 9(b) for Case 1, and Figure 8(d) and Figure 9 for Case 2.

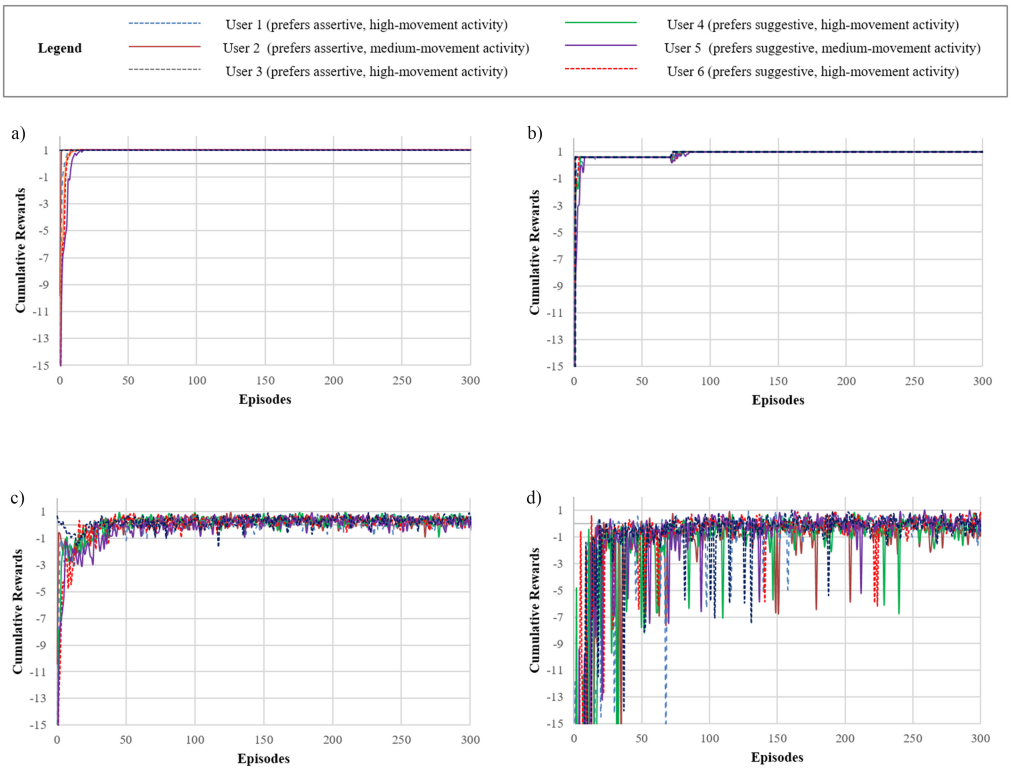

Fig. 8. Cumulative rewards per episode for the six user cognitive models. With 100% user compliance: (a) *LfD* and *RL* and (b) *RL*-only. With 90% user compliance: (c) *LfD* and *RL* and (d) *RL*-only.

The *RL*-only learning approach required an average of 78 episodes (min = 72 episodes, max = 85 episodes) to converge in the case of 100% user compliance, compared to an average of 10 episodes when using both *LfD* and *RL*. Similarly, in the case of 90% user compliance, the *RL*-only approach required an average of approximately 251 episodes to converge (min = 184 episodes, max = 330 episodes), compared to only 30 episodes for the *LfD* and *RL* approach. A similar trend can be seen in the number of steps required to complete a tea-making activity: approximately 676 steps in the case of 100% user compliance and 1,203 steps in the case of 90% user compliance were required on average during the first learning episode for the *RL*-only approach. These results show that the use of both *LfD* and *RL* algorithms can provide a significantly faster learning rate by reducing the number of episodes required for convergence.

DISCUSSION AND CONCLUSIONS

We have developed a novel robot-behavior-learning architecture using a combination of *LfD* and *RL* algorithms for a robot to learn and personalize assistive behaviors. *LfD* was implemented with professionally trained allied health care students and our socially assistive robot Casper. The CART decision tree successfully classified behaviors and learned environment state-assistive behavior mapping with an identification rate of 93%. We found that demonstrators showed a high degree of variability in how behaviors were displayed, especially for speech, where 44.21% spoke assertively and 40.11% spoke suggestively. Less variation was found in gestures, with demonstrators mainly showing low (40.34%) to medium (48.07%) movement activity. Still, these findings demonstrate

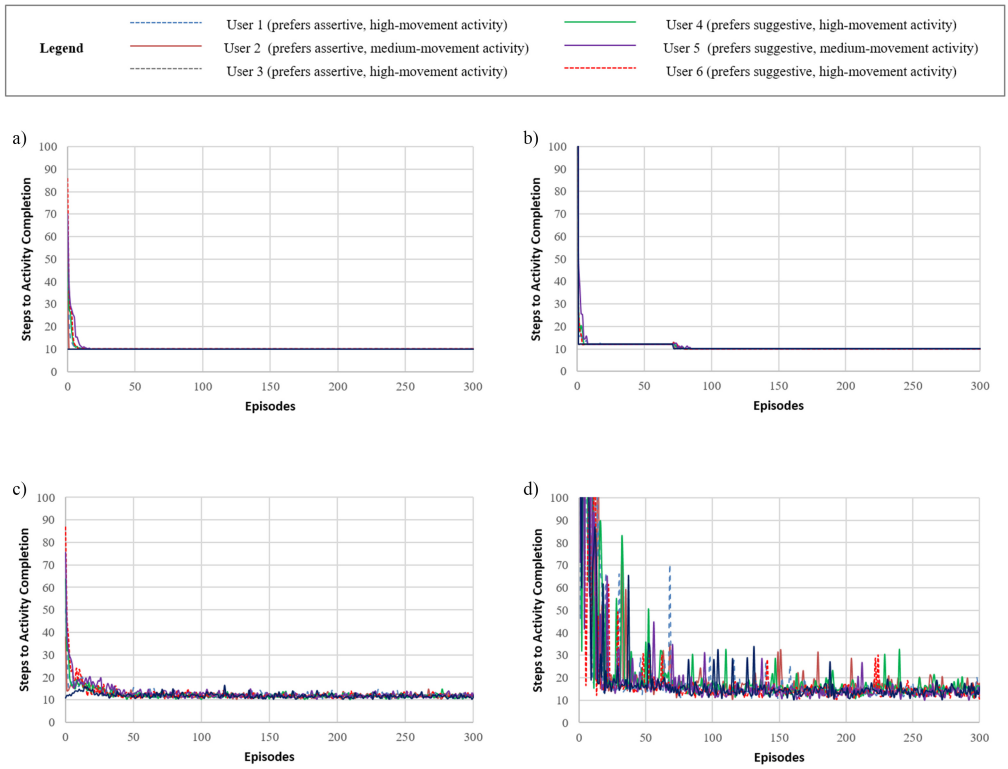

Fig. 9. Number of steps required to complete the tea-making activity per episode for the six user cognitive models. With 100% user compliance: (a) *LfD* and *RL* and (b) *RL*-only. With 90% user compliance: (c) *LfD* and *RL* and (d) *RL*-only.

the variability in demonstrator approaches, which is aligned with findings from previous research using *LfD* to learn behaviors [11].

Overall, the demonstrators were positive about the teaching method employed during *LfD* training. One recommendation for future implementations was to also have a professional actor play the role of the senior with dementia during the demonstrations. This type of interaction may also allow for the learning of robot behaviors with respect to facial expressions [48–50], proximity [51, 52], and the use of backchannels [53] with respect to the senior the robot is assisting.

The labeled behaviors for six different user cognitive models were learned using Q-learning with UCB exploration. A comparison study was then done with an *RL-only* approach, which showed that the proposed architecture combining *LfD* and *RL* significantly increased the rate of convergence. Both the number of episodes required for convergence and the number of steps required to complete the tea-making activity were lower when incorporating both *LfD* and *RL* into our learning model. The proposed robot-behavior-learning architecture provides a promising method for robots to learn to personalize their behaviors to a user and can allow for learning of this personalization in a shorter number of interactions with vulnerable users.

Future work will involve the integration of our robot-behavior-learning architecture with an activity perception system in order for the robot to detect the activity items in the environment while providing assistance to seniors with dementia. Then we will conduct user studies with this

demographic. We will also investigate the generalizability of the learned behaviors with other activities. As the user maintains the same cognitive model across all ADLs, it can be expected that a similar policy for behavior personalization would hold across different activities and we will investigate this assumption.

ACKNOWLEDGMENTS

This work is supported by the Canadian Consortium on Neurodegeneration in Aging (CCNA), AGE-WELL NCE Inc., the Canada Research Chairs (CRC) Program, and the Ontario Graduate Scholarship (OGS) Program. We would like to thank Shayne Lin for his assistance with the user studies, and Carrie Yang for her assistance in robot speech synthesis.

REFERENCES

[1] D. Feil-Seifer and M. J. Matarić. 2005. Defining socially assistive robotics. In *IEEE 9th International Conference on Rehabilitation Robotics*. 465–468.

[2] D. McColl and G. Nejat. 2013. Meal-time with a socially assistive robot and older adults at a long-term care facility. *J. Human-Robot Interact.* 2, 1 (2013), 152–171.

[3] J. Li, W.-Y. G. Louie, S. Mohamed, F. Despond, and G. Nejat. 2016. A user-study with tangy the bingo facilitating robot and long-term care residents. In *IEEE International Symposium on Robotics and Intelligent Sensors (IRIS'16)*. 109–115.

[4] C. Thompson, S. Mohamed, G. Louie, J. C. He, J. Li, and G. Nejat. 2017. The robot tangy facilitating trivia games: A team- based user-study with long-term care residents. In *IEEE International Symposium on Robotics and Intelligent Sensors (IRIS'17)*. 173–178.

[5] A. Tapus, C. Tapus, and M. J. Mataric. 2008. User-robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy. *Intell. Serv. Robot* 1, 2 (2008), 169–183.

[6] K. Dautenhahn. 2003. Roles and functions of robots in human society: Implications from research in autism therapy. *Robotica* 21, 4 (2003), 443–452.

[7] M. Heerink, B. Krose, V. Evers, and B. Wielinga. 2010. Assessing acceptance of assistive social agent technology by older adults: The Almere model. *Int. J. Soc. Robot* 2, 4 (2010), 361–375.

[8] E. Torta, J. Oberzaucher, F. Werner, R. J. Cuijpers, and J. F. Juola. 2012. Attitudes towards socially assistive robots in intelligent homes: Results from laboratory studies and field trials. *J. Human-Robot Interact.* 1, 2 (2012), 76–99.

[9] S. Andrist, X. Z. Tan, M. Gleicher, and B. Mutlu. 2014. Conversational gaze aversion for humanlike robots. In *ACM/IEEE International Conference on Human-Robot Interaction*. 25–32.

[10] V. Ng-Thow-Hing, P. Luo, and S. Okita. 2010. Synchronized gesture and speech production for humanoid robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10)*. 4617–4624.

[11] C.-M. Huang and B. Mutlu. 2014. Learning-based modeling of multimodal behaviors for humanlike robots. In *ACM/IEEE International Conference on Human-Robot Interaction*. 57–64.

[12] P. Liu, D. F. Glas, T. Kanda, H. Ishiguro, and N. Hagita. 2014. How to train your robot - teaching service robots to reproduce human social behavior. In *IEEE International Symposium on Robot and Human Interactive Communication*. 961–968.

[13] A. H. Qureshi, Y. Nakamura, Y. Yoshikawa, and H. Ishiguro. 2016. Robot gains social intelligence through multimodal deep reinforcement learning. In *IEEE-RAS 16th International Conference on Humanoid Robots*. 745–751.

[14] J. Hemminghaus and S. Kopp. 2017. Towards adaptive social behavior generation for assistive robots using reinforcement learning. In *ACM/IEEE International Conference on Human-Robot Interaction*. 332–340.

[15] J. Chan and G. Nejat. 2012. Social intelligence for a robot engaging people in cognitive training activities. *Int. J. Adv. Robot. Syst.* 9, 4 (2012), 113.

[16] W.-Y. G. Louie and G. Nejat. 2016. A learning from demonstration system architecture for robots learning social group recreational activities. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'16)*. 808–814.

[17] Y. S. Chiang, T. S. Chu, C. D. Lim, T. Y. Wu, S. H. Tseng, and L. C. Fu. 2014. Personalizing robot behavior for interruption in social human-robot interaction. In *IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO'14)*. 44–49.

[18] D. Brooker and I. Latham. 2015. *Person-Centred Dementia Care : Making Services Better with the VIPS Framework*. Jessica Kingsley Publishers.

[19] A. Tapus, C. Tapus, and M. J. Mataric. 2009. The use of socially assistive robots in the design of intelligent cognitive therapies for people with dementia. In *IEEE International Conference on Rehabilitation Robotics*. 924–929.

[20] H. W. Park and A. M. Howard. 2015. Retrieving experience: Interactive instance-based learning methods for building robot companions. In *IEEE International Conference on Robotics and Automation (ICRA'15)*. 6140–6145.

[21] B. Mutlu, T. Kanda, J. Forlizzi, J. Hodgins, and H. Ishiguro. 2012. Conversational gaze mechanisms for humanlike robots. *ACM Trans. Interact. Intell. Syst.* 1, 2 (2012), 1–33.
[22] A. D. Brenna, S. Chernova, M. Veloso, and B. Browning. 2009. A survey of robot learning from demonstration. *Rob. Auton. Syst.* 57, 5 (2009), 469–483.
[23] S. Manschitz, J. Kober, M. Gienger, and J. Peters. 2015. Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations. *Rob. Auton. Syst.* 74, 5 (2015), 97–107.
[24] J. J. Steil, F. Rothling, R. Haschke, and H. Ritter. 2004. Situated robot learning for multi-modal instruction and imitation of grasping. *Rob. Auton. Syst.* 47, 2 (2004), 129–141.
[25] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato. 2004. Learning from demonstration and adaptation of biped locomotion. *Rob. Auton. Syst.* 47, 2 (2004), 79–91.
[26] B. Kim, A. Massoud Farahmand, J. Pineau, and D. Precup. 2013. Learning from limited demonstrations. In *Advances in Neural Information Processing Systems*. 2859–2867.
[27] C. Breazeal. 2003. Toward sociable robots. *Rob. Auton. Syst.* 42, 3 (2003), 167–175.
[28] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, G. Dulac-Arnold, I. Osband, J. Agapiou, J. Z. Leibo, and A. Gruslys. 2017. Learning from demonstrations for real world reinforcement learning. *Arxiv Prepr. ArXiv1704.03732*.
[29] Alzheimer's Association. 2009. Memory Loss & 10 Early Signs of Alzheimer's. Retrieved from https://www.alz.org/alzheimers_disease_10_signs_of_alzheimers.asp. Accessed December 22, 2017.
[30] Alzheimer Society of Canada. 2017. 10 Warning Signs. Retrieved from http://www.alzheimer.ca/en/Home/About-dementia/Alzheimer-s-disease/10-warning-signs. Accessed December 22, 2017.
[31] Alzheimer's Society. 2017. Symptoms. Retrieved from https://www.alzheimers.org.uk/info/20064/symptoms. Accessed December 22, 2017.
[32] L. Breiman, J. Friedman, R. Olshen, and C. Stone. 1984. *Classification and Regression Trees*. Belmont, CA: Wadsworh & Brooks Cole.
[33] B. Gupta. 2017. Analysis of various decision tree algorithms for classification in data mining. *Int. J. Comput. Appl.* 163, 8 (2017), 15–19.
[34] S. Singh and M. Giri. 2014. Comparative study Id3, CART and C4. 5 decision tree algorithm: A survey. *Int. J. Adv. Inf. Sci. Technol.* 3, 7 (2014), 97–103.
[35] H. J. Eysenck. 1991. Dimensions of personality: 16, 5 or 3?—Criteria for a taxonomic paradigm. *Pers. Individ. Dif.* 12, 8 (Jan. 1991), 773–790.
[36] H. G. Wallbott. 1998. Bodily expression of emotion. *Eur. J. Soc. Psychol.* 28, 6 (1998), 879–896.
[37] D. Morris. 1981. *Gestures: Their Origins and Distribution*. Stein & Day Pub.
[38] R. S. Sutton and A. G. Barto. 1998. *Introduction to Reinforcement Learning*. MIT Press.
[39] P. Auer, N. Cesa-Bianchi, and P. Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.* 47, 2/3 (2002), 235–256.
[40] R. Y. Chen, S. Sidor, P. Abbeel, and J. Schulman. 2017. UCB exploration via Q-ensembles. arXiv preprint arXiv: 1706.01502.
[41] P. Bovbel and G. Nejat. 2014. Casper: An assistive kitchen robot to promote aging in place. *J. Med. Device.* 8, 3, Article 30945 (2014).
[42] Autonomous Systems and Biomechatronics Lab. 2013. Casper - Socially Assistive Humanoid Robot. *Youtube*. Retrieved from https://www.youtube.com/watch?v=noSJ9qWt_f0. Accessed May 16, 2017.
[43] Amazon Web Services. 2017. Amazon Polly – Lifelike Text-to-Speech. Retrieved from https://aws.amazon.com/polly/. Accessed November 9, 2017.
[44] ROS. 2013. openni_tracker - ROS Wiki. Retrieved from http://wiki.ros.org/openni_tracker. Accessed November 24, 2017.
[45] IBM Watson. 2017. Watson Speech to Text. Retrieved from https://www.ibm.com/watson/services/speech-to-text/. Accessed November 9, 2017.
[46] S. Czarnuch and A. Mihailidis. 2011. The design of intelligent in-home assistive technologies: Assessing the needs of older adults with dementia and their caregivers. *Gerontechnology* 10, 3 (2011), 169–182.
[47] M. C. Silveri, G. Reali, C. Jenner, and M. Puopolo. 2007. Attention and memory in the preclinical stage of dementia. *J. Geriatr. Psychiatry Neurol.* 20, 2 (2007), 67–75.
[48] T. Fong, I. Nourbakhsh, and K. Dautenhahn. 2003. A survey of socially interactive robots. *Rob. Auton. Syst.* 42, 3–4 (2003), 143–166.
[49] T. Tojo, Y. Matsusaka, T. Ishii, and T. Kobayashi. 2000. A conversational robot utilizing facial and body expressions. In *IEEE International Conference on Systems, Man and Cybernetics*. 858–863.
[50] F. Ferland, D. Létourneau, A. Aumont, J. Frémy, M.-A. Legault, M. Lauria, and F. Michaud. 2012. Natural interaction design of a humanoid robot. *J. Human-Robot Interact.* 1, 2 (2012), 118–134.

[51] E. T. Hall. 1966. *The Hidden Dimension*. New York: Doubleday & Co.
[52] M. L. Walters, K. Dautenhahn, K. L. Koay, C. Kaouri, R. Boekhorst, C. Nehaniv, I. Werry, and D. Lee. 2005. Close encounters: Spatial distances between people and a robot of mechanistic appearance. In *IEEE-RAS International Conference on Humanoid Robots*. 450–455.
[53] C. Rich, B. Ponsler, A. Holroyd, and C. L. Sidner. 2010. Recognizing engagement in human-robot interaction. In *ACM/IEEE International Conference on Human-Robot Interaction*. 375–382.

Received April 2018; accepted August 2018