2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Using a Shared Tablet Workspace for Interactive Demonstrations during Human-Robot Learning Scenarios

Hae Won Park[1], Richard A. Coogle[2], and Ayanna Howard[1]

Abstract— **One of the key elements for building a long-term robotic companion is incorporating the ability for a robot to continuously learn and engage in new tasks. Utilizing a defined workspace that provides various shared content between human and robot could assist in this learning process. Here, we propose integrating a touchscreen tablet and a robot learner for engaging the user during human-robot interaction scenarios. The robot learner's domain-independent core reasoner follows the structure of instance-based learning which addresses the issues of acquiring knowledge, encoding cases, and learning a retrieval metric. The system utilizes demonstrations provided by the user to auto-populate the knowledge base through natural interaction methods, encodes cases based on the feature structure provided by the user, and uses an adaptive-weighting technique to design a retrieval metric with linear regression in the feature-distance space. Through a tablet environment, the user teaches a task to the robot in a shared workspace and intuitively monitors the robot's behavior and progress in real time. In this setting, the user is able to interrupt the robot and provide necessary demonstrations at the moment learning is taking place, thus providing a means to continuously engage both the participant and the robot in the learning cycle.**

I. Introduction

Tablets such as the iPad easily attract attention, provide convenient access to daily computing tasks, and are supported by a huge collection of mobile applications. Tablets are commonly available and their intuitive touch-based interface has replaced many traditional entertainment and educational products, such as televisions, video-game consoles, and textbooks. Tablets in classrooms have proven their power to better motivate students and increase students' learning performance [1], [2]. Articles also report how tablet computers are used to help children with disabilities and learning issues by actively engaging them with the device's attractive touchscreen interface and design [3], [4].

In this paper, we discuss a robot learner that interacts with a human teacher by using the tablet as a shared workspace (Fig. 1). The role of robot learning for engagement is to increase the duration of engagement by incorporating a turn-taking scenario. Studies have shown that when people are required to teach others, they themselves become more engaged in the task [5]. The proposed robot learner observes the user while storing information about his/her situation-action responses (defined as a case), and then retrieves these cases to execute a corresponding behavior. This approach of storing example instances, known as lazy learning, provides a flexible structure for modeling unknown task domains with a single framework. Case-based reasoning (CBR) is one of the lazy-learning methods that solve new problems based on the solutions of similar past problems [6]. By comparing the current task to some past task cases stored in memory, the nearest cases and its solutions are retrieved and adapted to the current task.

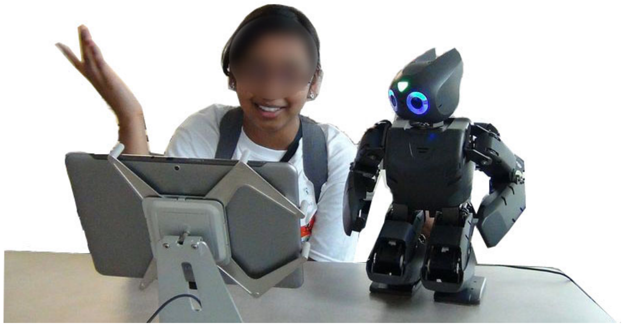


Fig. 1. A participant is teaching our robot a task on a tablet. The use of smart devices with touchscreens is a rising trend in education and entertainment thanks to their intuitive interfaces and convenient access to numerous apps. The robot learner learns to participate in a task through user demonstration and at the same time motivates user engagement.

Throughout the paper, we address the methods of *encoding* a given instance through reducing the task-feature space from an input from the user, *acquiring* cases that auto-populate the case base through encoding naturalistic demonstrations of the user, and *retrieving* case instances by determining the optimal similarity measure to find the nearest cases of the query instance.

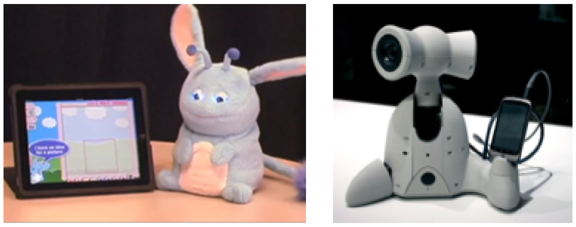
The overarching research statements are: 1) The shared tablet workspace provides an intuitive environment for engaging both the user and the robot learner, while the proposed framework effectively models tasks on the tablet by utilizing natural demonstration by the user even when an explicit model of the problem domain is difficult to elicit and is not amenable to complete mathematical modeling, 2) The robot's learning behavior and performance affects the length of interaction and the social behavior of the user. In order to support the statements, we measure how well and efficiently the robot learns a task from human demonstration on the tablet, measure the user's interaction time, analyze the emerging social behavior from the user depending on the robot's learning strategies, and conduct a post-experiment

This work is supported by the National Science Foundation, under grant IIS-1208287.

[1]Hae Won Park and Ayanna Howard are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. haewon.park@gatech.edu, ayanna.howard@ece.gatech.edu

[2]Richard A. Coogle is with Georgia Tech Research Institute, Atlanta, GA, USA. richard.coogle@gtri.gatech.edu

978-1-4799-3684-7/14/$31.00 ©2014 IEEE



(a) Popchilla [7]

(b) Robotic music listening companion [8]

Fig. 2. Robot companions enhance user's experience with smart devices.

survey to evaluate user experience.

In Section II, we review previous work and discuss why an instance-based learning supported by learning from demonstration (LfD) for case acquisition is effective in modeling tasks on the tablet. After addressing the issues of a domain-independent, instance-based learning and discussing the approaches to those challenges in Section III, we present the details of the approaches in Section IV. Section V and VI explains the experimental setup and reports the evaluation results and discussions. Finally, the conclusion is provided in Section VII.

II. Related Work

We first outline the studies that link robots with applications on touchscreen devices for enhancing user experience. Next, we review how LfD techniques are used in conjunction with CBR systems to automate the process of acquiring knowledge. We then introduce the issues of applying these techniques to modeling multiple tasks through a single framework. Finally, we summarize the benefits of using a shared tablet environment to facilitate naturalistic demonstration from the user and how engagement can help solve foreseen challenges of instance-based learning.

A. Tablets and Robots

It is reported that robots can enhance user experience through functioning in conjunction with applications (apps) on smart devices. Popchilla [7] (Fig. 2(a)) combines an interactive drawing application with a robot that generates motion and sound responses to user's input on the tablet. The robotic music-listening companion [8] (Fig. 2(b)) produces on-beat motions to the music playing from the smartphone. The robot's rhythmic behavior makes the person feel like they are sharing the experience, and the person perceives the event as more enjoyable.

As a research platform, touchscreen devices function as shared workspaces and reduce perceptual uncertainties. In LfD, there are several methods for recording teacher's demonstrations [9]. The uncertainty of environmental perception always poses a difficult challenge, including tracking the teacher's motions or recognizing the human's social cues. The tablet platform reduces such uncertainty since the touchscreen provides quantified sensor data from the gestural behavior of the user. The development environment and already available apps on the market facilitate the process of designing and implementing a task with controllable modalities. Such benefits of deploying a touchscreen-based medium for studying interactions between human and robot are discussed in [10]. In their work, the touchscreen setup provides context to unstructured social human-robot interaction.

Our research attempts to solve the limitations that the previous research haven't addressed. Until now, most tablet-based robots exhibited simple reactive behaviors, were tele-operated, or were limited to conducting a single task. Based on our previous work, which addresses the efficacy of coupling tablets as a shared workspace with a robot learner for HRI studies [11], we present a system that could be easily configured to engage robots with tablet apps.

B. Case-based Reasoning and Learning from Demonstration

Case-based reasoning (CBR) is a lazy-learning method where computation is performed at the instance-query time, compared to eager-learning methods in which generalization is conducted during training [12]. CBR is effective when an explicit model of the problem domain is difficult to elicit and is not amenable to complete mathematical modeling, such as general tasks on tablets.

In most knowledge-intensive CBR systems, the case base is preloaded. Recently, there have been successful efforts in applying LfD techniques to automate the process of case acquisition in CBR, sometimes referred to as lazy LfD approach. A data-driven CBR is used in [13], achieved through crowd-sourcing. In this work, the proposed system collected 82,479 cases during human-human collaborative task in a virtual reality environment. Afterwards, the case base was used towards a similar task conducted in the physical world with a human-robot team to generate robot behavior. In [14], the authors have solved the issue of populating case base with plans through LfD for generating planners for real-time strategic games. Similar to these works, we provide the tablet environment for the user to perform natural demonstration to the robot while the robot learner extracts the task state and the user's action to form cases. In addition, unlike the two examples above in which they separately populate the case base prior to application, our case acquisition happens at the same time the user and the robot carries out the task. The benefit of such setting is that the user monitors the robot's performance in real time and provides necessary demonstrations when needed. Such an approach is called just-in-time learning [15].

III. Issues Towards a Domain-independent, Instance-based Learning Framework for Task Modeling

Most CBR implementations, including the above mentioned works, rely on the expert to represent and retrieve cases. Since CBR relies on accumulated knowledge, it would be possible to build a robot that stores multiple knowledge libraries linked to different tasks. However, there are several hindrances to automating this approach even if we overlook the problem of handling large data. These problems are associated with the issues of acquisition, encoding, and retrieval of case instances, and our approach is summarized in the following:

- ***Acquisition*** is the problem of how cases are collected. Our robot learner acquires cases through interaction with the teacher who demonstrates the task on the shared tablet workspace. The robot receives task states from the tablet and associates the states with the teacher's behavior formalized as touch events on the tablet.

- ***Encoding*** is the problem of case representation. The challenges of feature selection during human-robot interaction have been studied by imitation-learning researchers as the issue of "what to imitate" [16], [17]. Unfortunately, when it is difficult to acquire a full view of the user and the task scene, or when an alternative method of tracking such cues is unavailable, the aforementioned methods are hardly applicable. Instead, we develop an interface for the users to input task-feature properties that they think are relevant to the task. The significance of each feature variable is determined during training a retrieval metric. Such a method of human intervention is widely used to mediate uncertainties in the environment.

- ***Retrieval*** is the problem of computing a similarity measure for finding nearest case instances. We adapt a linear regression method for computing a case-retrieval function with little or no domain-specific knowledge. Since our goal is to use a domain-independent framework to model tasks with different types of features, we convert the feature space into a real-valued feature-distance space and apply locally weighted regression (LWR). Basic LWR is a derivative of the k-nearest neighbor (k-NN) classifier and has been implemented extensively throughout CBR based applications.

IV. Approach

In Fig. 3, the overall learning-system process is depicted. The core reasoner provides the fundamental structure of a CBR system, i.e., retrieve-reuse-revise-retain of cases. In this paper, we address the issues of acquiring, encoding, and retrieving cases, and the approaches we took are introduced in the remainder of this section.

A. Acquisition

A case is defined as a tuple composed of a problem and a solution. When demonstrations are recorded as cases, the current state of the task is encoded as the problem, and the user's response to that state on the tablet is encoded as the solution. A UDP socket communication is established between the tablet and the robot, and a task state packet is sent from the tablet to the robot with some sampling rate. When the user initiates any touch event on the tablet, the start and end coordinates are sent to the robot; the robot then knows that a demonstration was given and creates a case. In the same way, the robot sends the start and end coordinates of a synthesized touch event to the tablet and computes inverse kinematics for its head and arm joints to generate a hand-eye coordinated motion.

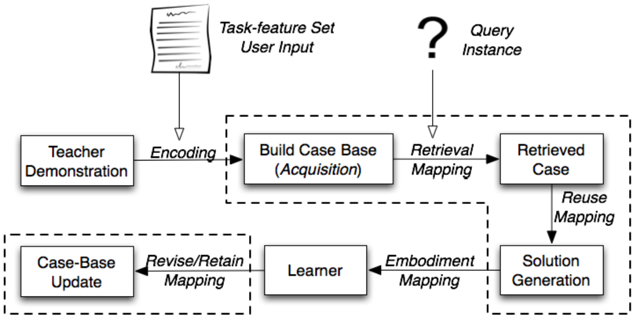


Fig. 3. Steps of case-based reasoning (CBR) incorporated within the overall structure of recording and encoding demonstrations, retrieving and reusing cases, and mapping a generated behavior to the robot's embodiment. CBR steps are depicted inside dashed boxes.

B. Encoding

A case is a 2-tuple model:

$$C = \{D_{prob}, D_{sol}\}$$

where D_{prob} is a problem descriptor and D_{sol} is a solution descriptor. The problem and solution descriptors consist of task features:

$$D_{prob} = \{f_1^p, f_2^p, \cdots, f_n^p\},$$

$$D_{sol} = \{f_1^s, f_2^s, \cdots, f_m^s\}$$

where n and m are the numbers of the problem features, f^p, and the solution features, f^s. Domain-dependent feature descriptors are:

$$f_i^p : \{x_i^p, attr_i^p\}, \text{ where } attr_i^p : \{T_i^p, M_{ex_i}^p, M_{dist_i}^p, w_i\},$$

$$\text{and } f_j^s : \{x_j^s, attr_j^s\}, \text{ where } attr_j^s : \{T_j^s, M_{ex_j}^s\}.$$

The feature space variables including the feature value x and feature attributes $attr$ are:

1) x: The feature *value* x of data type T is extracted with the method M_{ex}. The similarity between the two feature values is calculated by the distance function M_{dist}, and the resulting similarity measure influences the overall case similarity by the factor of w.
2) T: The feature *data type* in CBR could be in many different forms including string, integer, boolean, float, and vectors of these data types.
3) M_{ex}: The feature *extraction method* returns x of data type T. For tablet-based applications, this indicates the method of how to parse data packets sent from the tablet.
4) M_{dist}: The feature *distance metric* measures the distance between two feature values of data type T and returns a float value. This results in a real numeric value for all feature types and can now be represented

in an n-dimensional space. The returned float value is normalized to $\in [0, 1]$.

5) w: The regression weight coefficient can either be trained through the system or a real value can be assigned from previous trainings. A locally weighted regression (LWR) method is used with feature-distances as an input space, and the coefficients are specified such that they minimize the squared error summed over the nearest instances of the query feature-distance vector. The process is detailed in the next section. The LWR's target function doubles as the global retrieval function. Through training, coefficient of the feature distance that less likely influences the decision of case retrieval quickly diminishes.

The system receives the above feature-attribute information from the user through a simple Extensible Markup Language (XML) interface.

C. Retrieval

The retrieval stage is where the current problem states are compared to the problems of the cases in the case-base. The retrieval function is modeled as a linear sum of locally weighted task features. The weights are trained such that the overall function minimizes the cost function. This approach is similar to maximizing a reward function that penalizes deviations from a demonstrated motion trajectory for solving the swing-up inverted pendulum task in [18].

Linear regression is the problem of fitting a linear function to a set of input-output pairs given a set of training examples, in which the input and output features are numeric. The distances between the feature pairs become the input variables:

$$\mathbf{d} = \{\delta(x^p_{1_i}, x^p_{1_j}), \delta(x^p_{2_i}, x^p_{2_j}), \cdots, \delta(x^p_{n_i}, x^p_{n_j})\}^T,$$

where $x^p_{k_i}$ is the k-th feature, and $\delta(x^p_{k_i}, x^p_{k_j})$ is the output of M_{dist_k}. The distance $\delta(x^p_{k_i}, x^p_{k_j})$ will be abbreviated as δ^k_{ij} for simplicity. The target function models a retrieval function assuming a general linear relationship of the feature distances:

$$g(\mathbf{w}, \mathbf{d}) = \sum_{k=0}^{n} w_k \cdot \delta^k_{ij}$$

where $\mathbf{w} = \{w_0, w_1, \cdots, w_n\}$ is the regression coefficient vector, and $\delta^0_{ij} = 1$. A set $\mathbf{E}$ is defined as nearest-neighbor instances corresponding to $\mathbf{d}_q$. The regression coefficient vector $\mathbf{w}$ is then specified in order to minimize the squared error summed over the set $\mathbf{E}$.

$$Error(\mathbf{w}, \mathbf{d}) = \frac{1}{2} \sum_{\mathbf{d} \in \mathbf{E}} (g(\mathbf{d}) - \hat{g}(\mathbf{w}, \mathbf{d}))^2.$$

The gradient descent method is then used to compute $\mathbf{w}$ iteratively. This overall process is called locally weighted regression (LWR) and is a representative method of instance-based learning approaches, except that here, we have applied LWR in the feature-distance space instead of the feature space itself. This process is repeated for some number of query points, and for each query point the nearest neighbor set $\mathbf{E}$ is restated. Note that after training, the target function $g(\mathbf{w}, \mathbf{d})$ is used as the global similarity measure for retrieving cases.

D. Embodiment Mapping

The retrieved case and its solution are used to reproduce the task behavior on a robotic platform through a mapping from the adapted solution to the robot's state and action space. This includes generating a synthesized touch gesture that triggers a touch event on the tablet. Darwin also retrieves its emotion group depending on the user's state and performance, and generates a combination of speech and gesture primitives that enables engagement through behavioral interaction. In Fig. 4, Darwin is initiating a touch event on the tablet through wireless communication (Fig. 4(a)-(b)), making eye contact and providing feedback after the participant's demonstration (Fig. 4(c)-(d)), encouraging the participant (Fig. 4(e)-(f)), and expressing sadness after an unsuccessful attempt (Fig. 4(g)-(h)).

V. Experimental Setup

For validation, we recruited 33 participants (mean age m=18.27, standard deviation σ=8.56) including 19 children (m=12.26, σ=4.24). The participants were to teach a virtual game, shown in Fig. 5, to Darwin. We analyzed data collected during various events on campus during a two-month period. Groups of local school students and younger children, some with special needs, were invited to observe and participate in various experiments conducted in our research group.

For the task used for evaluations, participants were asked to teach the robot a strategic game on the tablet, in which the player has to control the launching angle and the power of a bird to destroy enemies either by directly aiming at them or knocking down the structures. The structure of the game makes various strategies possible to complete each level within a given number of attempts. In the following, a pilot study was conducted to observe what group of features the users think was sufficient to learn the task.

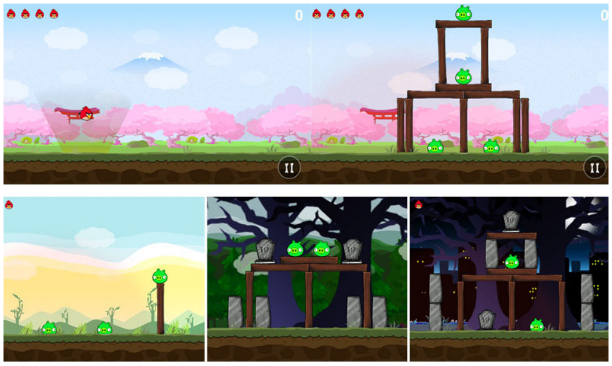


Fig. 5. The proposed framework was applied to a strategic game on the tablet.

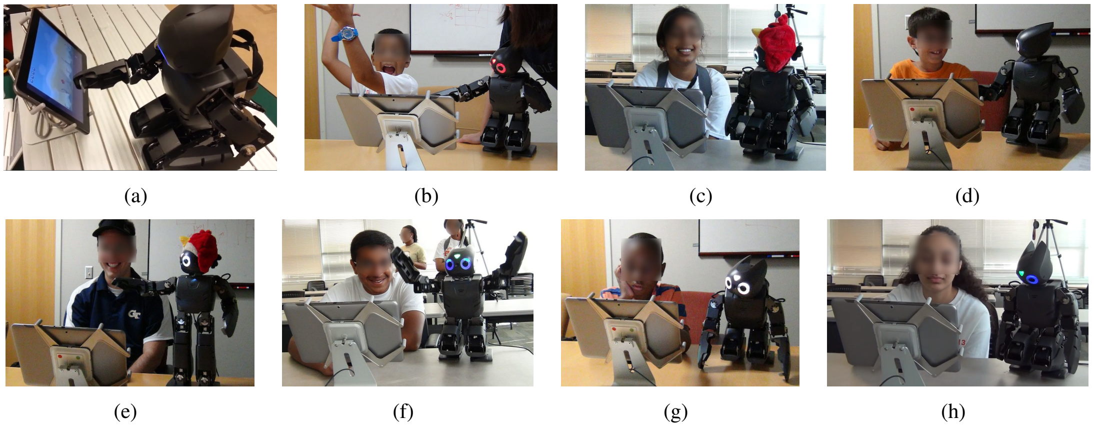


Fig. 4. The experiment was conducted in an open-house styled setting with a group of local school children. Each participant engaged in two experiments in which the robot demonstrated different case-retrieval strategies.

A. Pilot Study: Task-feature survey

Before the actual experiment, a pilot study was conducted with eight participants to select task features that they would want the robot learner to extract instances from. The participants each played four different game scenarios as shown in Fig. 5. The goal of the task was not just in completing each level, but to maximize the score. Fig. 6 shows the features the participants listed after conducting the task.

Among the listed features, the bird's launching angle and intensity are obviously the solution features of this task. The score, number of remaining pigs, and their locations received the most votes among the problem features. Participant #3's feature set, which includes the most voted features, was chosen for the remainder of the experiments. Fig. 7 shows the result of the trained weights using LWR.

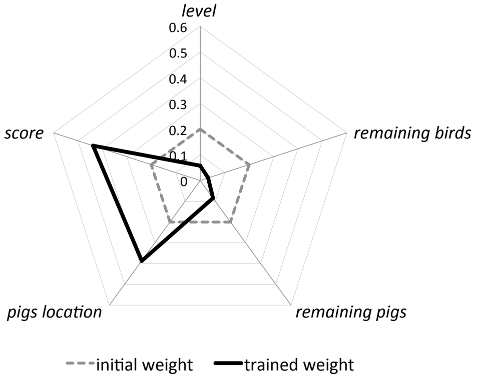


Fig. 7. Trained weights using the feature set of Participant #3 with locally weighted regression.

B. Experiment

Each experiment consisted of two sessions in which the participant taught the same task to two robots equipped with different case-retrieval methods (Fig. 4). The retrieval methods used by the robots were: Robot A (proposed adaptive weighting), Robot B (k-NN), and Robot C (random case retrieval). In modeling the retrieval function, the proposed method trains the weights of problem features as mentioned in Section IV, while k-NN assigns equal weights to all features. To prevent ordering effects, participants were grouped using a counterbalancing technique: 10 participants first interacted with Robot A then Robot B, another 10 Robot B then Robot A, 7 Robot A then Robot C, and 6 Robot C then Robot A.

The instruction given by the experimenter was strictly scripted to avoid any influence it might cause to the participant's experience. The script was as follows:

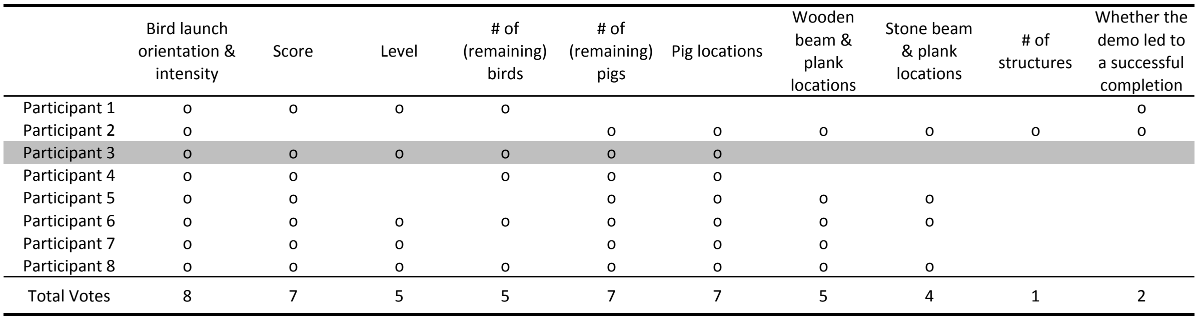
> *Now, I'd like you to teach Darwin to play the same game. Just teach him in the same manner you would teach your friend. Provide Darwin with demonstrations how to solve each level. Whenever you reach out to provide demonstration to Darwin, he will wait for his turn. Continue teaching each level until you are satisfied that Darwin had learned the level well enough, or think Darwin had stopped learning. Later, I want you to show me what you have taught Darwin, and collaboratively solve each level with him. Darwin may try to communicate with you, and he may not use human language. Afterwards, I will ask you some questions about your experience teaching a task to Darwin.*

The growth progress of the case base and any interaction with the tablet was logged, and two video cameras were placed to record the whole evaluation session. Later, the log was used to evaluate the system, and the videos were analyzed for interaction studies.

VI. Result and Discussion

In this section, we evaluate the result to test the following hypotheses:

- *Hypothesis 1*: The proposed learning framework produces comparable task performance against the average performance of the demonstrator.



	Bird launch orientation & intensity	Score	Level	# of (remaining) birds	# of (remaining) pigs	Pig locations	Wooden beam & plank locations	Stone beam & plank locations	# of structures	Whether the demo led to a successful completion
Participant 1	o	o	o	o						o
Participant 2	o				o	o	o	o	o	o
Participant 3	o	o	o	o	o	o				
Participant 4	o	o		o	o	o				
Participant 5	o	o			o	o	o	o		
Participant 6	o	o	o	o	o	o	o	o		
Participant 7	o	o	o		o	o	o			
Participant 8	o	o	o	o	o	o	o	o		
Total Votes	8	7	5	5	7	7	5	4	1	2

Fig. 6. Task features selected by pilot-study participants.

- *Hypothesis 2*: The proposed method of modeling a retrieval function through LWR on a feature-distance space reduces the workload, i.e., reduces the number of demonstrations required to achieve the same amount of system performance, compared to the k-NN approach.
- *Hypothesis 3*: The user's social behavior adapts to the robot learner's behavior.

A. Hypothesis 1

First, the learning performance of Robot A, Robot B, and Robot C are compared. In Table I, the performance of generated solutions is compared with varying k (number of retrieved cases). Distances are computed between a query problem and problems in the case base using each robot's retrieval method. Then the performance of each retrieved and adapted solution is evaluated using a logarithm of the earned game score. In this evaluation, it is shown that the Robot A's performance is more consistent compared to Robot B. When $k = 4$, Robot A's performance is better than Robot B's by 23.48%. When distances between the query point and cases in the case base are plotted, it clearly shows that in Robot A, the nearest cases are grouped together and produce the best performance, where as in Robot B, the nearest data points do not perform the best and are scattered (Fig. 8).

Relative to the teacher's performance, Robot A's performance was 1.32 ± 2.02 point better, and Robot B's performance was 0.21 ± 1.95 point better than the participant's average performance in logarithm scale. This result is due to the method we took to measure the performance of the teacher and the robots. While the participant's performance was averaged over all the demonstrations given throughout the session, the robot's performance was measured after the participant was done teaching. Therefore, the standard deviation of the teacher's performance was rather large (σ=2.86) while the robot's performance was consistent (σ=0.22). After making the measurements for each individual sessions, the difference between the teacher's and the robot's performance was averaged over all participants.

On a 5-point Likert scale, from strongly disagree (1) to strongly agree (5), post-experiment survey reports that participants felt both robots A (m= 4.67, σ=0.65) and B (m= 4.33, σ=0.82) were learning from them. Some participants still evaluated Robot C as "learning", but the majority responded Darwin was replaying demonstrations without any intelligence (m= 2.5, σ=1.05). When asked to compare the learning performance between the robots, participants responded that Robot A was the best learner (83%), which aligns with our finding above.

TABLE I
MEAN PERFORMANCE (LOG(SCORE)) OF CASE-RETRIEVAL METHODS

k	Robot A	Robot B	Robot C
1	5.12±0.52	4.14±2.23	-
2	4.97±0.76	4.02±2.02	-
3	4.78±1.08	4.13±1.72	-
4	4.89±0.86	3.96±1.46	2.85±0.84
5	4.12±0.82	3.11±1.87	-
6	3.82±0.44	2.79±0.92	-

B. Hypothesis 2

The average number of demonstrations ($k = 4$) given to each robot was: Robot A (m= 21.17, σ=6.44), Robot B (m= 29.17, σ=10.25), and Robot C (m= 24.15, σ=8.72). On average, participants provided 38% less demonstrations to Robot A than Robot B, while the average performance of Robot A was still better than that of Robot B. If a sufficient number of cases populate the problem space, Robot A and Robot B's performance will eventually converge. That is, if the number of high-performance demonstrations for each possible problem equals or exceeds k, Robot A and Robot B would retrieve the same set of cases for a given problem. However, exploring all possible problems will increase the teacher's workload significantly. Therefore, the proposed algorithm effectively increases the learning performance while reducing the user's workload in teaching a task to a robot.

C. Hypothesis 3

As discussed above, participants provided more demonstration to Robot B than Robot A. In the questionnaire asking when the participants stopped teaching each robot, majority of the participants answered "when Darwin clears each level several times" for Robot A (64%) and Robot B(61%), and "when Darwin stopped improving" for Robot

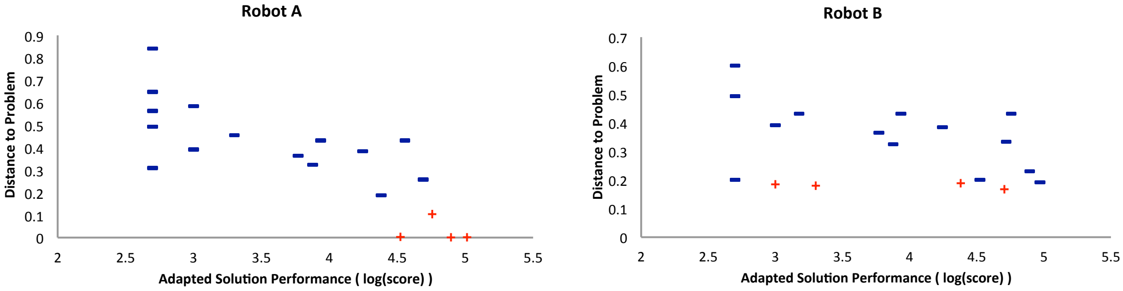


Fig. 8. A representative example of a case-retrieval result ($k = 4$) from the two robots: Robot A (adaptive weighting) and Robot B(standard k-NN). Distances from the query point to problems in case base are plotted against the performance in log(score). The retrieved cases that have the shortest distance to the query point are marked in red crosses.

C (52%). Participants also spent almost twice (90%) more time with Robot B than Robot A, and 26% more with Robot B than Robot C. Participants spent more time instructing the robot when the robot was improving slower (Robot B), but quickly lost interest when the robot wasn't responding to the demonstrations (Robot C). Through these results, we observed that the participant's behavior changes, e.g., the amount of interaction and when to end an interaction, based on the robot learner's learning ability and performance.

VII. Conclusion and Future Work

We present a novel system that couples a robot learner with a tablet that functions as a shared workspace. The goal of this research was to design a domain-independent learning system for a robot that continuously motivates engagement of the user. One of the limitations of commercially available robots is that they fail to provide new content when the user wants. We believe that continuous motivation comes from a continuous supply of new materials, and tablets provide such an environment.

By addressing the acquisition, encoding, and retrieval issues of designing an instance-based learning algorithm, our proposed system achieved the domain-independent property. First, task features can be encoded in any data type as long as a distance metric is specified. Second, a linear retrieval function is modeled by converting the task-feature space into a feature-distance space, and finding the set of regression weight coefficients that minimizes the squared output error summed over the nearest instances of the query feature-distance point. Lastly, case base is auto-populated through interaction with the user, who provides demonstrations through naturalistic interaction with the robot and the tablet.

As part of our future work, we plan to perform experiments with the therapists and their patients, evaluating the design of the framework user interface and the level of engagement individuals with cognitive disabilities exhibit. Regarding the learning algorithm, we plan to investigate an approach using neural networks within CBR to directly measure feature sensitivity when its node in the network is negated.

References

[1] B. McClanahan, K. Williams, E. Kennedy, and S. Tate, "A breakthrough for josh: How use of an ipad facilitated reading improvement," *TechTrends*, vol. 56, no. 3, pp. 20–28, 2012.

[2] W. Hu, "Math that moves: schools embrace the ipad," *The New York Times*, vol. 4, 2011.

[3] Á. Fernández-López, M. Rodríguez-Fórtiz, M. Rodríguez-Almendros, and M. Martínez-Segura, "Mobile learning technology based on ios devices to support students with special education needs," *Computers & Education*, 2012.

[4] N. Shah, "Special education pupils find learning tool in ipad applications." *Education Week*, vol. 30, no. 22, pp. 1–16, 2011.

[5] A. Gartner *et al.*, "Children teach children: Learning by teaching." 1971.

[6] J. L. Kolodner, "An introduction to case-based reasoning," *Artificial Intelligence Review*, vol. 6, no. 1, pp. 3–34, 1992.

[7] Bill. (2010, Aug. 13) iPad Controlled Robots Help Kids With Autism. [Online]. Available: http://www.padgadget.com

[8] G. Hoffman and K. Vanunu, "Effects of robotic companionship on music enjoyment and agent perception," in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, 2013, pp. 317–324.

[9] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[10] P. Baxter, R. Wood, and T. Belpaeme, "A touchscreen-based sandtrayto facilitate, mediate and contextualise human-robot social interaction," in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*. IEEE, 2012, pp. 105–106.

[11] H. W. Park and A. Howard, "Providing tablets as collaborative-task workspace for human-robot interaction," in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, 2013, pp. 207–208.

[12] D. Chen and P. Burrell, "Case-based reasoning system and artificial neural networks: A review," *Neural Computing & Applications*, vol. 10, no. 3, pp. 264–276, 2001.

[13] S. Chernova, N. DePalma, and C. Breazeal, "Crowdsourcing real world human-robot dialog and teamwork through online multiplayer games," *AI Magazine*, vol. 32, no. 4, pp. 100–111, 2011.

[14] S. Ontañón, K. Bonnette, P. Mahindrakar, M. A. Gómez-Martín, K. Long, J. Radhakrishnan, R. Shah, and A. Ram, "Learning from human demonstrations for real-time case-based planning," 2009.

[15] J. Handelsman, D. Ebert-May, R. Beichner, P. Bruns, A. Chang, R. DeHaan, J. Gentile, S. Lauffer, J. Stewart, S. M. Tilghman, *et al.*, "Scientific teaching," *Science*, vol. 304, no. 5670, pp. 521–522, 2004.

[16] C. Breazeal and B. Scassellati, "Challenges in building robots that imitate people," *Imitation in animals and artifacts*, p. 363, 2002.

[17] Y. Nagai, "From bottom-up visual attention to robot action learning," in *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*. IEEE, 2009, pp. 1–6.

[18] C. Atkeson and S. Schaal, "Robot learning from demonstration," in *Machine Learning: Proceedings of the Fourteenth International Conference (ICML '97)*, 1997, pp. 12–20.