Retrieving Experience: Interactive Instance-based Learning Methods for Building Robot Companions

Hae Won Park[1] and Ayanna M. Howard[1]

***Abstract*— A robot companion should adapt to its user's needs by learning to perform new tasks. In this paper, we present a robot playmate that learns and adapts to tasks chosen by the child on a touchscreen tablet. We aim to solve the task learning problem using an experience-based learning framework that stores human demonstrations as task instances. These instances are retrieved when confronted with a similar task in which the system generates predictions of task behaviors based on prior actions. In order to automate the processes of instance encoding, acquisition, and retrieval, we have developed a framework that gathers task knowledge through interaction with human teachers. This approach, further referred to as interactive instance-based learning (IIBL), utilizes limited information available to the robot to generate similarity metrics for retrieving instances. In this paper, we focus on introducing and evaluating a new hybrid IIBL framework using sensitivity analysis with artificial neural networks and discuss its advantage over methods using k-NNs and linear regression in retrieving instances.**

I. Introduction

For robots to become life-long companions gathering information and generalizing them to learn new skills is important. The interactive instance-based learning (IIBL) project studies how robots can learn through accumulating experience during interaction with users who don't necessarily have the skills to program robots. Using an instance-based approach as an overarching framework for learning, "snapshots" of experience instances are stored in memory formulated as task state and action pairs. These instances provide guidelines to solve similar problems in the future, mimicking the cognitive process of problem solving in humans (e.g., prototype theory [1], [2]).

In our previous work [3], we have discussed the three issues of automating the processes of instance-based learning: instance encoding, acquisition, and retrieval. The encoding is the problem of "what to learn", meaning what features the robot needs to extract while observing the teacher's demonstrations. While the teacher is providing a demonstration of the task, the robot has to identify and pair teacher's actions to task states and store them in memory, which is the acquisition problem. We discussed how interactive methods can aid in solving these issues through keyword mapping of robot's skills. In this paper, we provide further discussion on the retrieval problem. We reviewed two methods for increasing the performance of k-nearest neighbors (k-NN) to compute similarities between instances and retrieve relevant experiences. To determine the distance, we have adopted regression methods assuming linear and nonlinear relationships between the task features to recommend a vector of feature weights used towards designing a weighted distance function for k-NNs. In this process, we introduce locally weighted linear regression and sensitivity analysis using artificial neural networks to measure how much each task feature contributes to maximizing the task objective. Later, an integrated systematic approach is presented that illustrates the interaction between IIBL and these regression modules. The tasks that are presented here place the robot learner, acting as a playmate, with children as teachers in a shared tablet workspace [4]. This setting allows the teacher to closely monitor and evaluate the robot's learning, and provide necessary instances at the moment learning is happening. For system evaluation, we have used a humanoid robot, Darwin, in open-house events, exhibitions, play-therapy centers, and at homes, in which the system performed robustly while learning and improving from demonstrations provided by children and adults.

In the next section, previous works on instance-based learning and interactive machine learning are reviewed. In Section III, the two approaches to training an instance retrieval function is presented, and their performance is evaluated and discussed in Section IV. Conclusion and future work are presented in Section V.

II. Related Work

Instance-based learning (IBL) utilizes an analogical reasoning process that uses previous knowledge stored in memory to solve new problems. Case-based reasoning (CBR) is a popular instance-based learning method that provides predictions to new problems from a baseline that similar problems have similar solutions [5]. By retrieving and reusing past solutions, the system can avoid the time necessary to derive solutions from scratch. IBL methods store training instances in their raw form and postpone generalization until the query time. When a new instance is introduced, its classification relies on the stored data and its similarity with respect to previous instances. Machine learning with rule-based methods perform explicit generalization during training, and the data used in the process are discarded afterwards. In IBL, the generated prediction can be traced back to the original instances that provided the generalization, which in turn improves the explanatory behavior of the overall learning system. Prior works used IBL for robot learning, planning and action-selection problems [6], [7], [8]. These systems

This work is supported by the National Science Foundation, under grant CNS-0940146 and grant IIS-0705130.

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

were targeting specific problems, and therefore the domain dependent models for instance representation and retrieval were given by the expert.

The main limitation of instance-based learning, that the generalization can only be made to an extent of a task space covered by its instances, can be improved through an interactive system that monitors the system's state in real time and provides necessary inputs to efficiently cover the given problem space. Interactive machine learning (IML) places humans in the process of designing, training, and evaluating machine-learning systems [9], [10]. In IML, humans provide inputs to the system and monitor the output that in turn influences what inputs they will provide next. The action of providing human input is often referred to as "teaching", and the teaching occurs during a system deployment which makes the learning process interactive. Therefore, learning from demonstration (LfD) with social robots naturally fosters an IML setting. LfD is a method for programming new skills into the robot by providing human demonstrations [11]. Research efforts in agents learning from human teachers are not only limited to teachers providing inputs. Robots can also ask for help after a failure [12] or actively request to resolve an ambiguity [13].

In most knowledge-intensive systems, CBR is used as a knowledge retrieval and management tool in which the case base is loaded prior to system deployment, for example using ontology [14]. Recently, there have been successful efforts in applying LfD techniques to automate the process of case acquisition in CBR, sometimes referred to as a *lazy-LfD* approach. In [15], the authors solved the issue of populating a case base with plans through LfD for generating planners for real-time strategic games. A data-driven CBR was developed in [16] through crowdsourcing. In this work, the proposed system collected 82,479 cases during human-human collaborative tasks 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.

This paper proposes an interactive instance-based learning (IIBL) framework that utilizes IML to actively engage the user in the process of knowledge encoding and acquisition during human-robot interaction on a shared workspace. By providing a visualization of the system's current state and performance, such as through a robot's social behavior, programming a machine-learning system becomes more accessible to the end-users. Compared to previous work using LfD for providing batch instances prior to an actual system deployment, IIBL collects and maintains instances while interacting with the teacher.

III. Approach

The methods for instance encoding (selecting task features) and acquisition (extracting task states and actions) are presented in our previous work [3]. Here, we present the two proposed approaches to modeling and training an instance retrieval metric. IIBL's generalization is provided by a distance function that measures similarity between instances. The most popular form is the k-nearest neighbors (k-NN) algorithm that predicts a query's label using the k number of nearest instances in memory [17]. In order to improve the performance of k-NNs, we model the distance function as a linear sum of locally weighted feature distances. In the following, weight-vector prediction methods using linear regression and sensitivity analysis with neural networks are presented.

A. Locally Weighted Regression (LWR-IIBL)

Linear regression is the problem of fitting a linear function to a set of input-output pairs given a set of training examples. 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 [18]. The distances between the feature pairs become the input variables:

$$\mathbf{d} = \{\delta_1(v_{1_i}, v_{1_j}), \delta_2(v_{2_i}, v_{2_j}), \cdots, \delta_n(v_{n_i}, v_{n_j})\}^T,$$

where n is the number of task features, v_{k_i} and v_{k_j} are the values of the query point and retrieved instance's k-th feature, and $\delta_k(v_{k_i}, v_{k_j})$ is the output of the k-th feature distance metric. The distance $\delta_k(v_{k_i}, v_{k_j})$ will be abbreviated as δ_{ij}^k for simplicity. The target function models a retrieval function assuming a weighted linear relationship of the feature distances:

$$g(\mathbf{w}, \mathbf{d}) = \sum_{k=0}^{n} w_k \cdot \delta_{ij}^k, \text{ where } w \in [0, 1] \text{ and } |\mathbf{w}| = 1$$

where $\mathbf{w} = \{w_0, w_1, \cdots, w_n\}$ is the regression coefficient vector, and $\delta_{ij}^0 = 1$. A set $\mathbf{E}$ is defined as instances around the query point. 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 [19]. This overall process is a 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.

In the following, the second approach to training the feature weight vector is presented that relies on a sensitivity analysis with neural networks. We attempt to address the limitation that the linear regression methods possess — their

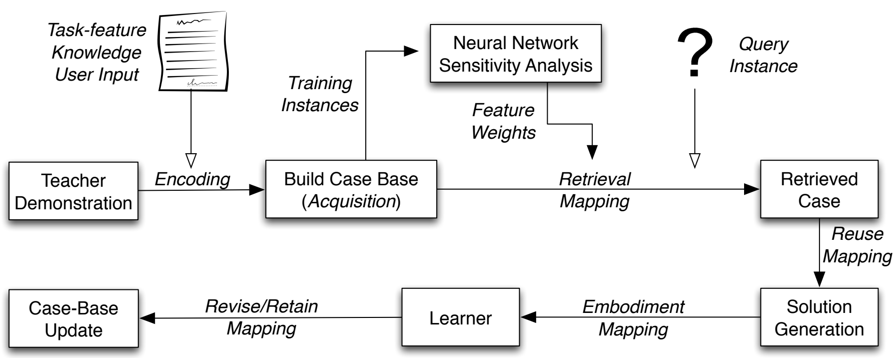


Fig. 1. A hybrid approach to IIBL with neural networks system overview. A feedforward NN is generated using instances collected from initial user demonstration. The NN supplies a prediction of the feature weight vector to IIBL obtained through analyzing sensitivity of each task feature.

incapacity to model nonlinear behavior of the dependent variables.

B. Sensitivity Analysis with a Feedforward Neural Network (NNSA-IIBL)

The second method is using a hybrid approach to IIBL with an artificial neural network (ANN). The hybrid system utilizes neural network sensitivity analysis (NNSA) for recommending a feature weight vector $\mathbf{w}$. ANN is a powerful supervised learning method for solving classification and regression problems, and a multilayer feedforward neural network can represent a broad set of nonlinear functions. [20], [21]. However, ANN can only provide little comprehensible knowledge about how it arrived at a given result. Discarding the dataset used for training makes the system difficult to trace back to the specific source that contributed to providing the prediction. When the prediction is inaccurate, instance-based methods can replace or update the instances that caused these results and re-train the system when necessary. In human-robot interactive learning environments, knowing which demonstrations led to which robot behavior can provide persuasive information for explaining the robot's decision. This encourages the human teacher to provide supplemental or more accurate examples.

The goal of the sensitivity analysis is to evaluate the relative importance of the input features [22]. It measures to what degree each input feature contributes to the change in the prediction result. Some information obtained from the sensitivity analysis can also tell the form of relationship between the variables [21]. Some analysis can also report the form of relationship in the context of other variables [22]. NNSA-IIBL reviews the change in the system performance when each input node is removed to measure sensitivity. The larger the change is, the higher contribution the feature makes to the system. A similar approach was taken by Shin, et al. [23]. They adopted methods of neural network pruning for feature weighting. Network pruning is a practical method to minimize the size of the network, while maintaining good performance [24]. The sensitivity of each feature is calculated by removing the input node from the trained neural network. Instead of physically removing the neuron and connected synapses, the weights connected to the input neuron are set to zero. Then the sensitivity is measured by the difference in the prediction result when the feature is present and when it is removed. The sensitivity S_i of a given input feature x_i is:

$$S_i = \frac{\sum_{\forall observations} \frac{|P - \hat{P}|}{P}}{N} \quad (1)$$

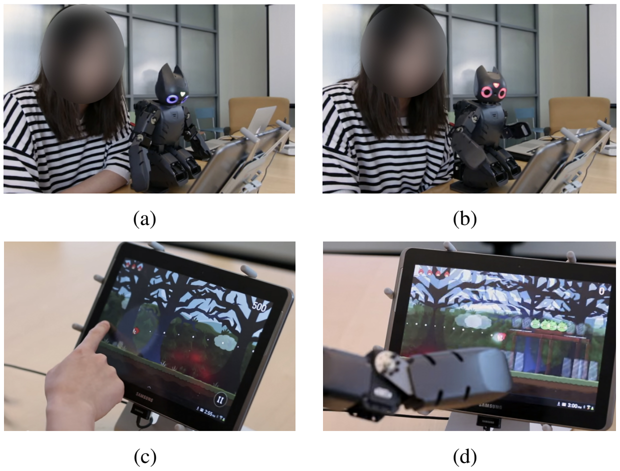
where P is the normal prediction value for each training instance after training, and $\hat{P}$ is the modified prediction value when the input node i is removed. N is the number of training datasets. In IIBL, we compute P as the system performance of the retrieved solution after the initial training, and $\hat{P}$ as the system performance after the input node i has been removed. Afterwards, feature weights w_i are assigned proportion to their sensitivity value:

$$w_i = \frac{S_i}{\sum_{m=1}^{n_I} S_m}. \quad (2)$$

C. A Hybrid Approach to IIBL with Neural Networks

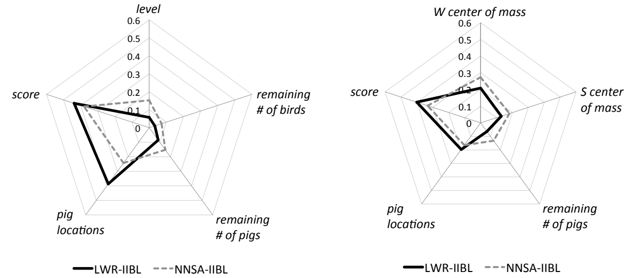
Shin, et al. [23] suggested utilizing ANN's outputs as solutions to given problems as well as using the network as a feature-weighting mechanism for instance retrieval. The basic idea is to compare the solution generated by ANN and the solution retrieved from the case base using the weights ANN suggested to the system. If the solutions match in a classification problem (or the difference is within a threshold in a regression problem), then the system reports the result, and if not, both solutions are rejected.

In IIBL, a neural network is solely used for training the feature weights (Fig. 1). Since an IIBL framework is targeting the generalization of task modeling, providing absolute comparison between an ANN generated solution and an instance-based retrieved solution across various tasks is impossible. It would be possible to find a comparison



(a) (b) (c) (d)

Fig. 2. (a),(b) Robot learning a task from the participant in a shared tablet workspace. (c) Participant providing a demonstration. (d) Robot executing retrieved solution.



(a) Feature set I (b) Feature set II

Fig. 3. Two feature sets programmed by the participants during the interaction and an average result of the trained weights using IIBL methods.

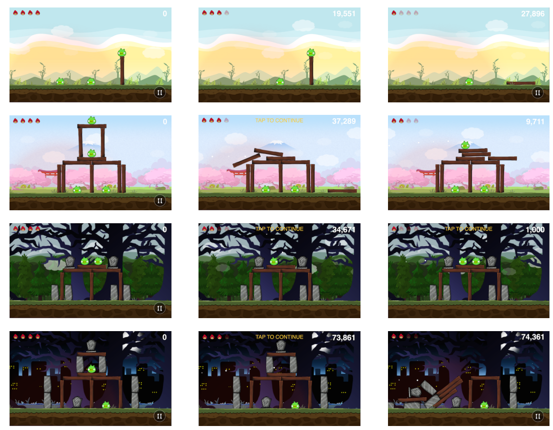


Fig. 4. Twelve query scenarios of the task used for evaluation.

threshold value that caps the rejection rate under a limit through iteration. However, instead of rejecting possible solutions, IIBL prefers to make mistakes that encourages the human user to observe the robot behavior and provide better demonstrations at the right time and improve the case base. If the frequency of the user providing demonstration increases, ANN re-generates the feature weights with the current case base.

IV. Evaluation Results and Discussion

A. *Task Description and Data Collection*

We have applied the proposed IIBL framework during our *Angry Darwin Expedition*, in which our robot, Darwin, learned to play a strategic game "Angry Birds" on a shared tablet workspace from various users. During a six-month period, over 130 people interacted with our robot learner including over 90 children, among which 33 participated in the formal experiment. The task's objective was to shoot the bird to destroy all enemies, the pigs, either by directly aiming at them or knocking down the surrounding structures to collapse them. A total of 66 instance databases, two sets per participant, were collected that consisted of 1,596 demonstrations with an average of 24.18 demonstrations ($\sigma = 6.82$) per participant per set (Fig. 2).

Among the feature sets programmed by the participants, two sets were selected for evaluation as shown in Fig. 3. In the figure, average results of the feature-weights are depicted that were trained using the proposed IIBL methods. In both methods, the score was the most dominant feature. This aligns with the fact that the success of the task was driven by the highest game score, which had high correlation to how many enemies were destroyed.

In the following, the performances of the IIBL methods are evaluated against k-NN regarding their performance and efficiency. Twelve problem scenarios in Fig. 4 were used in the evaluation using instance sets collected from the participants.

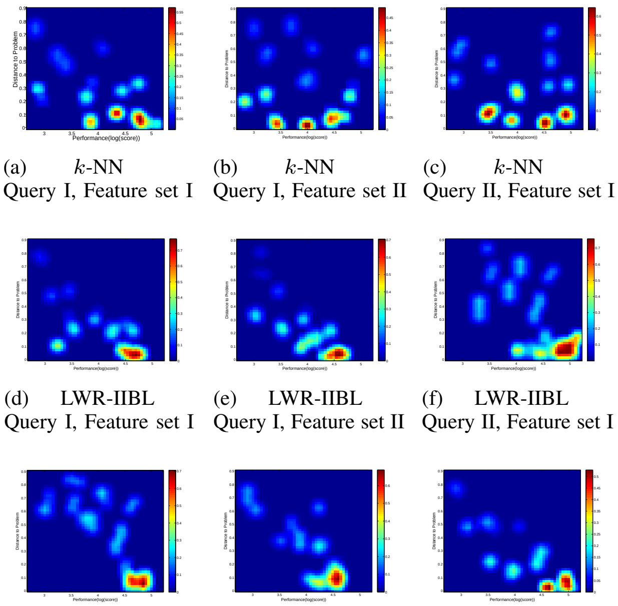
B. *Evaluation I: Performance*

The first hypothesis was that the IIBL methods will produce comparable task performance to the average performance of the demonstrator. In Table I, performances of generated solutions from k-NN, LWR-IIBL, and NNSA-IIBL methods are compared to an average performance of the demonstrations provided by the participants. With varying k (number of retrieved cases), distances are computed between the query point and the problems in the case base using each retrieval method. Then the performance of each retrieved solution is evaluated using a logarithm of the earned game score.

TABLE I

Mean performance (log(score)) of instance-retrieval methods compared to participant demonstrations with the feature set I.

k	k-NN	LWR-IIBL	NNSA-IIBL	Participants
1	4.14±2.23	5.12±0.93	5.49±0.61	3.75±2.02
2	4.02±2.02	4.97±0.76	5.15±0.43	-
3	4.13±1.72	4.78±1.08	4.91±0.21	-
4	3.96±1.46	4.89±0.86	4.76±0.09	-
5	3.11±1.87	4.12±0.82	4.34±0.12	-
6	2.79±0.92	3.82±0.44	3.86±0.13	-



(a) k-NN Query I, Feature set I (b) k-NN Query I, Feature set II (c) k-NN Query II, Feature set I

(d) LWR-IIBL Query I, Feature set I (e) LWR-IIBL Query I, Feature set II (f) LWR-IIBL Query II, Feature set I

(g) NNSA-IIBL Query I, Feature set I (h) NNSA-IIBL Query I, Feature set II (i) NNSA-IIBL Query II, Feature set I

Fig. 5. x axis: distance to given query, y-axis: performance ($log(score)$). Retrieval probability of instances depicted along the distance from given query points I and II, and their corresponding performance when the retrieved solutions are applied to each problem. Both IIBL methods successfully retrieved instances that perform best, while k-NN method's retrieved cases are scattered across the performance range.

Overall, the result shows that the average performance gradually decays as k increases, and the confidence interval is larger when the solution depended on less retrieved instances. IIBL methods outperformed k-NN and produced more stable results (smaller confidence interval). When $k = 4$, IIBL performed on average 21.84% better than k-NN and 28.67% better than the average performance of the participants' demonstrations. The performance of the participants was averaged over all demonstrations given throughout the experiment which also included unsuccessful demonstrations. The large confidence interval of the teacher's performance reflects such fact.

Table II shows the evaluation result using a second feature set. The result is consistent with the previous feature set. At $k = 4$, IIBL performed 11.87% better than k-NN and 15.90% better than the participants' demonstrations on average.

When distances between the query point and the instances in the case base are plotted, we can clearly visualize which instances are most likely to be retrieved. In Fig. 5, the regions of instances with the highest probability of being retrieved are more red than others, gradually fading into blue regions that are less likely to be selected. While k-NN in both feature sets shows scattered plots of likely retrieved instances, IIBL methods show better consistency between the likely retrieved solutions and their resulting performance when applied to a given problem. In other words, the instances retrieved by the IIBL methods have higher probability of producing the best performance, while k-NN doesn't guarantee that the retrieved instances will generate good performance. The instances retrieved by k-NN differed from those retrieved by the IIBL methods, while the instances retrieved by the two IIBL methods were much similar.

According to the plots, NNSA-IIBL produced more stable results (standard deviation σ_d from the linear fitting line = 1.15) compared to LWR-IIBL (σ_d = 5.67) among different query points. The evaluation results suggest that NNSA-IIBL is more stable and performs slightly better than LWR-IIBL.

TABLE II

MEAN PERFORMANCE (LOG(SCORE)) OF INSTANCE-RETRIEVAL METHODS COMPARED TO PARTICIPANT DEMONSTRATIONS WITH THE FEATURE SET II.

k	k-NN	LWR-IIBL	NNSA-IIBL	Participants
1	4.26±1.92	4.72±1.21	5.04±0.83	3.05±1.94
2	3.98±2.13	4.26±0.86	4.67±0.42	-
3	3.72±1.74	3.92±0.86	4.15±0.27	-
4	3.16±1.32	3.32±0.72	3.75±0.32	-
5	2.67±1.21	3.25±0.43	3.25±0.21	-
6	2.42±0.86	3.06±0.32	3.19±0.15	-

C. Evaluation II: Efficiency

The second hypothesis was that the IIBL methods will reduce workload, i.e., reduce the number of demonstrations required to achieve the same amount of system performance, compared to the k-NN approach. We gradually increased the size of the case base with the instances collected from the participants and measured each method's performance. A total of 1,596 instances were added four at a time in a random order. Each method's performance was measured against the twelve problem scenarios in Fig. 4. The result with $k = 4$ is depicted in Fig. 6. According to the experiment, IIBL methods' performance increased faster then k-NN. LWR-IIBL took 162 instances, NNSA-IIBL took 153 instances, and k-NN took 212 instances to reach within the 95% convergence for all query points. On average, IIBL required 34.60% less instances to solve all twelve problems with the best performance. If a sufficient number of cases populate the problem space, IIBL and k-NN's performance will eventually converge. However, exploring all possible problems will increase the teacher's workload significantly.

V. CONCLUSION AND FUTURE WORK

Accumulating experiences and retrieving them at the right time are important for building lifelong robot learners. Here, we defined experience as task state and action pairs that are extracted during human teacher's demonstrations. Using an IIBL framework, our project focuses on automating the processes of experience instance encoding, acquisition, and retrieval. This paper presented the approach to training a retrieval metric using linear regression and sensitivity analysis with neural networks. We have evaluated IIBL methods by comparing them to the teacher's demonstration and k-NN. The two hypotheses regarding the performance and

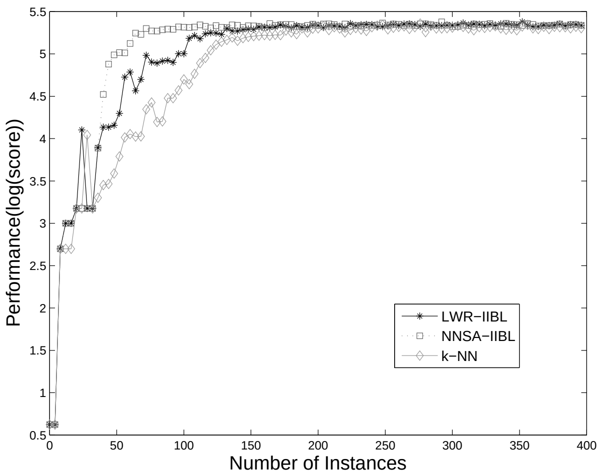


Fig. 6. Evaluation of the case-base size and the performance of IIBL methods compared to k-NN. NNSA-IIBL learned the fastest while k-NN generally required more instances to produce similar performance.

efficiency of the IIBL methods were analyzed, and we have observed that IIBL produces better performance with fewer instances than traditional instance-based learning using k-NN. Both IIBL methods generated successful predictions for task-feature contributions, and the retrieval functions recommended by the two methods produced superior outcomes compared to traditional k-NN. Throughout the application with varying conditions presented in this paper, NNSA-IIBL showed more stable performance compared to LWR-IIBL. While NNSA-IIBL was capable of modeling the unknown nonlinear behaviors of the input variables, LWR-IIBL required less demonstrations to train.

The IIBL project is currently working on deploying our robot playmate to local lending libraries and play centers. During this process, the system will be used for human-robot interaction studies to answer the following research questions: How can robots use social behaviors to inform the teacher of its reasoning process? Can robot playmates improve children's target behavior, such as eye contact, collaboration, and turn taking?

REFERENCES

[1] D. L. Medin, M. W. Altom, and T. D. Murphy, "Given versus induced category representations: Use of prototype and exemplar information in classification," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 10, no. 3, p. 333, 1984.

[2] E. Rosch, "Principles of categorization," *Concepts: core readings*, pp. 189–206, 1999.

[3] H. W. Park, R. A. Coogle, and A. M. Howard, "Using a shared tablet workspace for interactive demonstrations during human-robot learning scenarios," in *Robotics and Automation, 2014. ICRA '14. IEEE International Conference on*, June 2014.

[4] 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.

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

[6] M. Veloso, J. Carbonell, A. Perez, D. Borrajo, E. Fink, and J. Blythe, "Integrating planning and learning: The prodigy architecture," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 7, no. 1, pp. 81–120, 1995.

[7] M. W. Floyd, B. Esfandiari, and K. Lam, "A case-based reasoning approach to imitating robocup players." in *FLAIRS Conference*, 2008, pp. 251–256.

[8] R. Ros, J. L. Arcos, R. Lopez de Mantaras, and M. Veloso, "A case-based approach for coordinated action selection in robot soccer," *Artificial Intelligence*, vol. 173, no. 9, pp. 1014–1039, 2009.

[9] M. Ware, E. Frank, G. Holmes, M. Hall, and I. H. Witten, "Interactive machine learning: letting users build classifiers," *International Journal of Human-Computer Studies*, vol. 55, no. 3, pp. 281–292, 2001.

[10] J. A. Fails and D. R. Olsen Jr, "Interactive machine learning," in *Proceedings of the 8th international conference on Intelligent user interfaces*. ACM, 2003, pp. 39–45.

[11] 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.

[12] M. N. Nicolescu and M. J. Mataric, "Learning and interacting in human-robot domains," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 31, no. 5, pp. 419–430, 2001.

[13] M. Cakmak and A. L. Thomaz, "Designing robot learners that ask good questions," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM, 2012, pp. 17–24.

[14] J. A. Recio-Garía and B. Díaz-Agudo, "Ontology based cbr with jcolibri," in *Applications and Innovations in Intelligent Systems XIV*. Springer, 2007, pp. 149–162.

[15] S. Ontanón, 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," in *Artificial Intelligence (IJCAI) Workshop on Learning Structural Knowledge From Observations, The 2009 International Joint Conferences on*, 2009.

[16] 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.

[17] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[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.

[19] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: an approach to regression analysis by local fitting," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988.

[20] D. F. Specht, "A general regression neural network," *Neural Networks, IEEE Transactions on*, vol. 2, no. 6, pp. 568–576, 1991.

[21] S. Lek, M. Delacoste, P. Baran, I. Dimopoulos, J. Lauga, and S. Aulagnier, "Application of neural networks to modelling nonlinear relationships in ecology," *Ecological modelling*, vol. 90, no. 1, pp. 39–52, 1996.

[22] M. Gevrey, I. Dimopoulos, and S. Lek, "Review and comparison of methods to study the contribution of variables in artificial neural network models," *Ecological Modelling*, vol. 160, no. 3, pp. 249–264, 2003.

[23] C.-K. Shin, U. T. Yun, H. K. Kim, and S.-C. Park, "A hybrid approach of neural network and memory-based learning to data mining," *Neural Networks, IEEE Transactions on*, vol. 11, no. 3, pp. 637–646, May 2000.

[24] R. Reed, "Pruning algorithms-a survey," *Neural Networks, IEEE Transactions on*, vol. 4, no. 5, pp. 740–747, 1993.