Active Learning on Service Providing Model: Adjustment of Robot Behaviors through Human Feedback

Shih-Huan Tseng, Feng-Chih Liu, and Li-Chen Fu, *IEEE Fellow*

***Abstract*— It is important for robots to personalize the interactions and provide user-desired services especially facing different elders. This paper, therefore, proposes a learning strategy on the service-providing model. Through human feedback, the strategy endows the robot to learn the users' needs as well as preferences and then to adjust its behaviors. Here, we assume that users' needs and preferences may vary with time, and hence the goal of this paper is to let the adjustment of robot behaviors be able to adapt to those variations. In turn, the service-providing model of the robot could be on-line adjusted as well. That is, it is to select a new action from those favorable actions that have been selected long before or which is not an unfavorable action that annoys humans recently. To implement our system, the service robot under discussion is applied to the office environment. For performance evaluation, we have made extensive experiments that satisfactorily demonstrate that our robot can indeed provide services to different users and adapt to their preference change.**

I. Introduction

Human-robot Interaction (HRI) [1] has become a popular research domain in recent years. The issue of long-term elderly care with robots has been conducted many years [2]. Due to the fact that elders' needs and preferences may vary with long-term care, rather than being fixed like typical environments that have been discussed profoundly in the related literature, humans in the interaction loop may cause the relevant reinforcement learning (RL) problem to be different from the traditional one. Rainer *et al.* presented a work on learning of a dialog strategy for identifying human-interested objects [3]. The dialog policy and reward function were trained offline with subjective user feedbacks. A Bayesian RL approach was developed in [4] for an autonomous wheelchair which was able to infer user's preference. The method there needs users to give direct commands to the robot. So far, the performance of these works heavily depends on the volume and significance of the collected data. Moreover, a method of adaptively adjusting physical parameters pertaining to the robot behaviors has been proposed in [5] where a pre-designed reward function was used. On the other hand, to adapt to user's preference in a game play, a Q-learning method was implemented on Aibo, a robotic pet dog, to learn the user-favored action sequence where the reward modality was human touch [6]. Finally, Jeanie and Goldie applied MAXQ learning [7] to the decision module of a socially assistive robot that was engaged in a memory game with a human. They took affective states of humans into online training whereas the offline training was performed with user simulation models.

To sum up, there is one issue which is not yet well addressed in the literatures of HRI, *i.e.*, how a robot learns the user's needs and acts to meet his/her preferences through human feedback during HRI. Here, the robot's actions favorable to human users should be selected again even when these actions have not been selected for a long time. As such, our contribution in this paper is that a new action selection policy are proposed to deal with the issue mentioned above.

This paper aims at developing a learning strategy for the robot through the help from human feedback. In this way, the robot's behaviors are able to adapt to users' needs and preferences. In other words, the robot can provide personalized services for different people at different times in a day.

II. Service Providing Model

In our system, a service-providing model is needed for the robot to interact with humans before serving them, which is as shown in Fig. 1. Here, the service mission is modeled as a Markov Decision Process (MDP) [8]. Specifically, an MDP is expressed as $\{S, A, T, R, D, \gamma\}$, where S represents a set of states, A represents a set of actions, T describes system dynamics, R gives the reward model, D is the initial state distribution, and γ is a discount factor for rewards.

The state space S is discretized by imposing a sequential decision making process. Note that S consists of all the progresses of service negotiations through which the human and the robot collaboratively catch respective intentions and attitudes about the underlying services.

The action space A is defined by categories of actions: *initiate interaction*, *pass interaction*, *respond to human request*, *infer*, *query*, and *performing services*. As an example, given observations, the robot either actively *initiates interaction* or *responds to the human request*. For the first case, if it is not rejected by the human, to *query* or *infer* the need of some services is the first step. Next, the interaction starts to loop, during which the robot first *queries* or *infers* the service type and then the details of the service in need. The human can terminate the dialog at any step or receive one of three services from the robot, namely, "*ProvideDrink*", "*BriefInformation*", or "*ArrangeSchedule*".

The transition function T encodes the probability of state transitions from one state s_i to another state s_{i+1} given an $P(s_i, a_i, s_{i+1})$

Shih-Huan Tseng is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan (e-mail: d96018@csie.ntu.edu.tw).

Feng-Chih Liu is with the Department of Electrical Engineering, National Taiwan University, Taiwan (e-mail: r99921015@ntu.edu.tw).

Li-Chen Fu is with the Department of Electrical Engineering and Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan (e-mail: lichen@ntu.edu.tw).

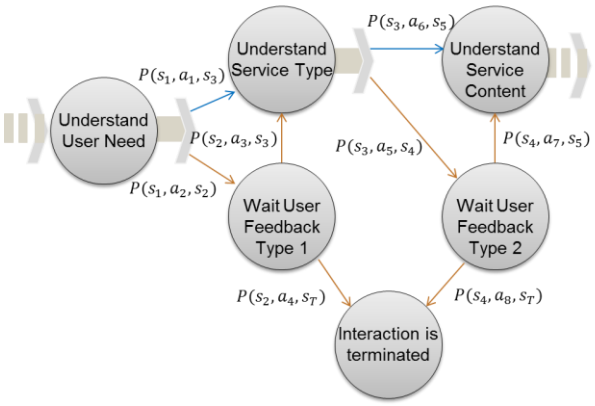

Fig. 1. A part of the service mission model.

action a_i, denoted as . In our system, the interaction is discretized by stages where the transition of states occurs only when valid information is collected.

The reward function *R* reflects the motivational structure of the service mission. A numerical reward, denoted as , is output at each step. By performing the user-desired services, the robot receives positive feedbacks from the user and also an additional reward when the service is accepted. Negative feedbacks from the user, either during the service negotiation or after the service is performed, punishes the robot. Each time the robot queries the user about anything, the robot receives a small negative reward as a punishment on raising questions and imposing the user's burden.

III. Learning through Human Feedback

A. HRI-Based Model Learning

Which side that initiates the interaction is of much importance since the human's activeness indicates the interest and preference. Without directly asking for true intention, the $r(s_i, a_i)$ robot observes the initiation and response from human during interaction and could still meet the human's preference. Before we specify the reward function for use in human-robot interaction, we first define three classes of interaction situations as follows :

$$I = \begin{cases} I_1, \text{if human initiated interation} \\ I_2, \text{if robot initaited interation and was accepted} \\ I_3, \text{if robot initiated interation but was rejected} \end{cases} \quad (1)$$

where I_1 is considered dominant because the human makes initiation, I_2 is the class showing moderate preference, and apparently I_3 shows negative signals.

To fulfill learning in an unknown environment, the reward function must be constructed first. The model-based methods hold the advantage of efficiency on samples over those model-free methods. Model-based methods first learn from experience the approximate models of state transitions and reward function and then use those models to estimate the value function for deciding the best policy. Here, we take some idea from R-MAX [9] and design a modified approach to approximate the reward function from interaction-based experience. To deal with the time-varying and ambiguous nature of human preference, we consider the recency of interaction. Recency factor pertains to the latest situation of interaction which belongs to one of the three classes I_i defined in eq. (1). Next, the approximate reward is computed using an exclusive manner. By introducing the newest update of human judgment and excluding the class of samples that is not favored, the robot can adapt more rapidly and is allowed to change to a completely different rewarding plan. The models of reward function and transition probability are given as:

$$R(s_i, a_j) = \begin{cases} V_{\max}, & \text{if } I_1 \\ \dfrac{\sum_t r(s_i, a_j)}{n(s_i, a_j)}, & \text{if } I_2 \\ \dfrac{\sum_t r'(s_i, a_j)}{n'(s_i, a_j)}, & \text{if } I_3 \end{cases} \quad (2)$$

$$P(s_i, a_j, s_k) = \begin{cases} \dfrac{n_t(s_i, a_j, s_k)}{n(s_i, a_j)}, & \text{if } I_1 \text{ or } I_2 \\ \dfrac{n_t'(s_i, a_j, s_k)}{n'(s_i, a_j)}, & \text{if } I_3 \end{cases} \quad (3)$$

where $V_{\max}$ is the upper bound on the optimal value function; $r(s,a)$ represents the one-step reward; $n(s,a)$ denotes the number of times executing the action *a* in the state *s*; $r'(s,a)$, $n'(s,a)$ and $n'(s,a,s')$ all denote the values of the corresponding functions that exclude the class of samples contradicting the human response during the last interaction. Note that there is a difference between the proposed interaction-based approach and R-MAX. The rewards of unknown action-state pairs are not magnified as those with R-MAX. We emphasize more on the exploration of actions follows a probabilistic distribution that is dependent on human feedback. In this way, learning in real-world interaction can be done more efficiently. The robot achieves a sub-optimal solution quickly and swiftly adapts to human feedback.

B. Action Selection Policy

The robot has to select an action from a set of available actions given the current state. Instead of choosing the action with maximal long-term rewards, actions should be selected in a way that balances between the current optimal action and unknown or sub-optimal actions. We contribute in a temporal measure of actions for guiding the action selection to fit human preference.

Considering the consistency of human preference in short-term period, the past experience of choosing actions is incorporated into the action selection policy. First, for each available action in the current state, we count the number of episodes from the last execution of the action till now, denoted

as $t(s,a)$. Second, the profit of selecting that action is taken to weight the precious value. The measure is defined as:

$$M(s,a) = t(s,a)^{\Delta Q(s,a)} \quad (4)$$

where $\Delta Q(s,a)$ is the difference between the action value and the average action value of the available action set in the state *s*. The action value $Q(s,a)$ can be calculated iteratively through one-step rewards and transition probability.

Note that the temporal measure is enlarged for relatively favorable actions with above-average returns, whereas the measure is reduced for relatively unfavorable action with below-average returns.

The robot has to select an action from a set of available actions given the current state. Instead of choosing the action with maximal long-term rewards, actions should be selected in a way that balances between the current optimal action and unknown or sub-optimal actions. We contribute in a temporal measure of actions for guiding the action selection to fit human preference.

The original action value is combined with the measure and becomes:

$$Pref(s,a) = Q(s,a) + M(s,a) \quad (5)$$

Finally, Gibbs-distribution is utilized to assign probability to actions based on the proposed measure of recency and profit. The resulting probability of selecting an action *a* in a state *s* is formulated as:

$$P(a \mid s) = \frac{e^{\frac{Pref(s,a)}{\tau(c)}}}{\sum_{a' \in A_s} e^{\frac{Pref(s,a')}{\tau(c)}}} \quad (6)$$

where *c* represents a context factor that contributes in the parameter setting of τ that controls the exploration rate.

By applying the temporal measure with the expected accumulated rewards of actions, those favorable actions that have not been selected for a long time will have a greater chance to be selected again, while the probability of selecting recently human unfavorable actions will be reduced.

IV. Experimental Results

Six users participated in three different experiments. Each user maintained 3~5 individual needs and preferences. The first experiment examines learning from no experience, that is, unknown users for the robot. The second experiment examines learning under changed environments, that is, known users with changed needs and preferences. The results of the first two experiments were investigated and we improved on the algorithm and did the third experiment.

In each experiment, there is a sequence of discrete events and each event has two features: *time* and *situation*. The first experiment ran for 50 events. Next, following the first experiment, users changed their needs and preferences and ran the second experiment for another 50 events. The third experiment was run independently where the user configurations were kept the same as the first two and the trial was run for a total of 100 events. As for performance analysis in detail, different algorithms were run for 200 events and the processes of interactions were analyzed.

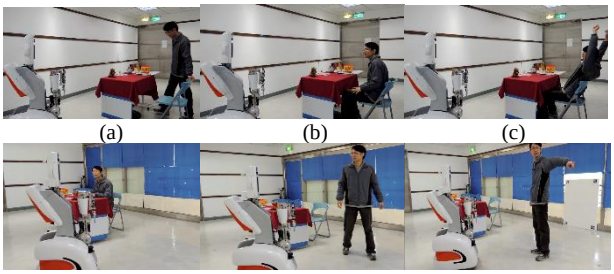

Fig. 8. The beginning stage of human-robot interaction that can be initiated by either a human or the robot. (a)-(c) The posture of *stretching the body* was observed. Later, the *robot initiated* a service mission to the person. (d)-(f) A *human initiated* the interaction while the robot stayed passive. The robot was asked for a *drink service*.

A. Learning User Needs and Preferences

The first experiment demonstrates the effectiveness of learning a user's needs and preferences. In the experiment, we let the robot initiate an interaction unless it decides to stay passive.

We compare the results of three approaches, of which the HRI-based is the one proposed in this paper, R-Max [9] is a model-based reinforcement learning method, and Q-learning [10] is a model-free reinforcement learning method. The number written after the HRI-based method represents the inverse of the parameter τ for the action selection. Overall, HRI-based approach collected the most reward, implying the effectiveness of learning the needs and preferences of an unknown user. The slope of a curve represents the trend of reward gaining and is also a quality index of the learned policy. As Fig. 2 shows, it can be found that HRI-based methods quickly learned good policies whereas the R-Max reaches a limit and its policy cannot compete with HRI-based methods. The possible reason is that R-Max relies heavily on the learned model and the model is generated by the samples with threshold on the sample size. Once the sample size is limited to a small number, the exploration cannot be performed fully and the learned policy will be insufficient. On the other hand, Q-learning explores freely and the performance depends on how fast the correct q-values are constructed. In Fig. 2, Q-learning rebounded back with a competent policy at event 30.

B. Adapting to Changes of User Needs and Preferences

In this experiment, we demonstrate the effectiveness of adjusting the learned policy to adapt to the changes of a user's needs and preferences. Continuing the previous experiment, the users modified their needs and preferences and the robot continued the service mission. Originally, the robot acted according to past experiences but soon negative feedbacks were received due to the changed needs and preferences. The performance at this stage depends on how fast the robot recovers from the impact of past experience which is partly faulty and on how the robot adjusts its behaviors to learns a

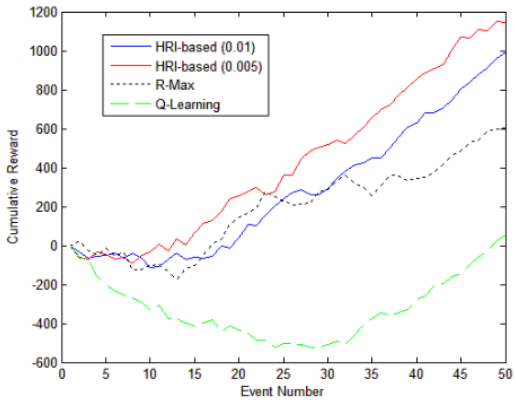

Fig. 2. Cumulative reward collected from service missions during the first 50 events. The robot knows nothing about the user and must learn from interactions.

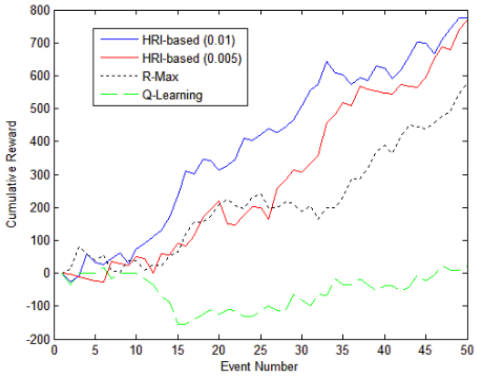

Fig. 3. Cumulative reward collected from service missions during the second 50 events. The user has changed his/her needs and preference. The robot knows a few about the user but must learn the changed user needs and preference from interactions.

good policy as well as maintains the performance of the unaffected part without doing a full exploration.

The effectiveness of learning to adapting to the changes of user needs and preferences is revealed by examining the rewards obtained over time. Given the same user, the more reward received from service missions indicates the robot is efficient and the provided services are suited to the user's needs and preferences. Figure 3 gives the result of different algorithms after the user changes part of his/her needs and preferences. It is obvious that the four algorithms performed poor initially when the change happened. In comparison, Q-learning failed to learn a new policy and collected rewards slowly. The other three methods recovered at event number 10 to 12 and increased performance over later events. It is interesting that HRI-based (τ =0.01) performed the best at this stage while HRI-based (τ =0.005) performed the best at learning at the initial stage. The explanation could be that the negative reinforcement takes the effect more quickly on the first one and the robot changes the policy of action selection more quickly. The model-based approach without considering HRI characteristics was comparatively weaker at first but the quality of its final policy looks as good as that of HRI-based methods. Finally, Table II shows that the comparison of performance with difference algorithms. The performance of our method with difference τ value is better than R-Max method.

TABLE II. PERFORMANCE COMPARISON

Method	Metrics		
	Interaction Rate (robot-initiated mission / 100 events)	Success Rate (accepted service / 100 missions)	Average Query (query/mission)
R-Max	49.92 %	71.48 %	1.2
HRI-Based (τ =0.01)	76.5 %	72.5 %	0.79
HRI-Based (τ =0.005)	80.5 %	72.04 %	1.095

V. CONCLUSION

In this paper, we proposed a model-based learning approach for application to human-robot interaction. We proposed an online algorithm by which the robot learns human's needs and preferences while interacting with users. The context and interaction situation are taken into the learning process to facilitate learning from interaction with humans. The action selection policy enabled the robot to learn the information more effectively by avoiding performing recently unfavorable services to users.

The experiments demonstrated that the robot could perform effective learning so that user-desired services car be provided at the right time and at the right situation. Furthermore, the trials also show that the robot can still provide adequate services despite the unknown users later changed their needs and preferences during the experiments.

REFERENCES

[1] M. Goodrich and A. Schultz, "Human-robot interaction: a survey," in *Foundations and Trends* in *Human-Computer Interaction*, 2007.

[2] K..Wada and T. Shibata, "Living With Seal Robots—Its Sociopsychological and Physiological Influences on the Elderly at a Care House," in *IEEE Transaction on Robotics*, vol. 23, pp. 972–980, 2007.

[3] R. Stiefelhagen et al., "Enabling multimodal human-robot interaction for the Karlsruhe humanoid robot," in *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 840–851, 2007.

[4] A. Atrash and J. Pineau, "A Bayesian reinforcement learning approach for customizing human–robot interfaces," in *Proceedings of the 13th international Conference on Intelligent user interfaces*, New York, NY, USA, pp. 355–360, 2009.

[5] N. Mitsunaga, C. Smith, T. Kanda, H. Ishiguro, and N. Hagita, "Adapting robot behavior for human–robot interaction," in *IEEE Transactions on robotics*, vol. 24, no. 4, pp. 911-916, 2008.

[6] A. Castro-Gonzalez, F. Amirabdollahian, D. Polani, M. Malfaz, and M. Salichs, "Robot self-preservation and adaptation to user preferences in game play, a preliminary study," in *Proceeding of the 2011 IEEE International Conference on Robotics and Biomechanics (ROBIO)*, Phuket, Thailand, Dec 7-11, 2011.

[7] J. Chan and G. Nejat, "A learning-based control architecture for an assistive robot providing social engagement during cognitively stimulating activities," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 3928-3933, 2011.

[8] M. L. Puterman, Markov Decision Processes, Wiley, 1994.

[9] R.I. Brafman and M. Tennenholtz, "R-max - a general polynomial time algorithm for near-optimal reinforcement learning," in *Journal of Machine Learning Research*, 2002.

[10] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, University of Cambridge, 1989.