The 23rd IEEE International Symposium on Robot and
Human Interactive Communication
August 25-29, 2014. Edinburgh, Scotland, UK,

Learning Something from Nothing: Leveraging Implicit Human Feedback Strategies

Robert Loftin[1], Bei Peng[2], James MacGlashan[3], Michael L. Littman[3], Matthew E. Taylor[2], Jeff Huang[3], and David L. Roberts[1]

[1]Department of Computer Science, North Carolina State University
[2]School of Electrical Engineering and Computer Science, Washington State University
[3]Department of Computer Science, Brown University

Abstract— **In order to be useful in real-world situations, it is critical to allow non-technical users to train robots. Existing work has considered the problem of a robot or virtual agent learning behaviors from evaluative feedback provided by a human trainer. That work, however, has treated feedback as a numeric reward that the agent seeks to maximize, and has assumed that all trainers will provide feedback in the same way when teaching the same behavior. We report the results of a series of user studies that indicate human trainers use a variety of approaches to providing feedback in practice, which we describe as different "training strategies." For example, users may not always give explicit feedback in response to an action, and may be more likely to provide explicit reward than explicit punishment, or *vice versa*. If the trainer is consistent in their strategy, then it may be possible to infer knowledge about the desired behavior from cases where no explicit feedback is provided. We discuss a probabilistic model of human-provided feedback that can be used to classify these different training strategies based on when the trainer chooses to provide explicit reward and/or explicit punishment, and when they choose to provide no feedback. Additionally, we investigate how training strategies may change in response to the appearance of the learning agent. Ultimately, based on this work, we argue that learning agents designed to understand and adapt to different users' training strategies will allow more efficient and intuitive learning experiences.**

I. Introduction

As the number of deployed robots grows, there will be an increasing need for end users to teach robots skills that were not pre-programmed. Despite initiatives to create user-friendly programming languages (*e.g.*, Scratch [12]), not everyone is able to become a proficient programmer, hence the need for an easily understood mechanism by which a robot can learn behaviors. Two common approaches are techniques based on *reinforcement learning* [15] of a reward function that assigns values to actions [8] and *learning from demonstration* [3] using demonstrations of behaviors. Unfortunately, the former requires that the reward function be fully defined, which may not be intuitive to non-technical users. The latter requires that the user be able to successfully complete the task using the robot's actions.

While both classes of techniques have their merits, we believe there is a third paradigm which has great potential for non-technically-trained users to easily teach robots or agents behaviors in situations where demonstrations may not be possible or feasible. In this paradigm, we focus on models of training and learning based on providing *categorical positive and negative feedback* to a learner. When providing feedback according to categories, trainers can give some form of positive feedback, some form of negative feedback, or withhold feedback. It is this latter case that proves most problematic from an algorithmic point of view.

A natural way to map these categorical feedbacks to the numerical rewards used in reinforcement learning is to encode categorical rewards as positive numerical values (for example, $+1$), categorical punishments as negative numerical values (for example, -1), and a lack of categorical feedback as 0. While there has been some success implementing reward-maximizing algorithms to learn from human feedback, treating feedback that is inherently categorical from humans as numerical has the potential to cause problems. First, it is unclear how a reinforcement-learning algorithm should use a feedback of zero. Should it be content with its decision (0 is better than -1) or change its decision (perhaps it could have gotten $+1$)? Second, consider a case where there is more than one way to provide positive feedback (*e.g.*, "ok" or "good"). Should one feedback have a numerical value of $+1$ and the other $+3$? What about $+1$ and $+17$? Absent a clearly defined reward function, any choice of assignment from categorical feedback to numerical value is likely to be arbitrary, and may prove a hindrance to learning. Third, people's teaching strategies may be non-stationary. For example, other work [9] has shown that users tend to increase their use of neutral feedback over time, in which case the most effective training algorithms will need to account for the teaching strategy being used and how it changes.

This paper discusses the results of three user studies in which participants trained a set of virtual agents, showing that different human trainers may take different approaches to teaching the same behavior. Based in part on these results, we argue that robots and virtual agents designed to learn from human trainers should treat human feedback as categorical input, rather than a numerical value, and should explicitly account for variations in training strategies. Furthermore, we discuss a probabilistic model that can be used to encode the categorical feedback, or lack thereof, that human trainers provide to agents while training. We present this model as

978-1-4799-6765-0/14/$31.00 ©2014 IEEE

a foundation for algorithmic development that is beyond the scope of this work.

II. Background and Related Work

A. Behaviorism

Behaviorism, a field of psychology, focuses on learning from feedback. Skinner introduced operant conditioning, a concept of providing feedback to modify the frequency of behaviors [13]. There are a number of ways in which training feedback can be provided. These so-called *operant conditioning paradigms* can be grouped into four categories [14]: positive reward (R$+$), negative reward (R$-$), positive punishment (P$+$), and negative punishment (P$-$). Rewards increase the frequency of the behavior they are associated with while punishments decrease the frequency. Positive refers to adding a stimulus and negative refers to removing a stimulus. An example of R$+$ is giving a dog a treat (rewarding by adding a desirable stimulus). An example of P$-$ is taking a prized toy away (punishing by removing a desirable stimulus).

Dog trainers have learned that using only positive reward (R$+$) results in fewer unintended side effects for dogs than when positive punishment (P$+$) is used to reduce undesired behavior [5]. Given the widespread prevalence of pet dogs in the United States and around the world, we anticipate that many human trainers will intuitively apply these concepts when training robots. Because of biases toward the R$+$/P$-$ operant conditioning paradigms, taking these strategies into account when designing learning algorithms may provide a contextual advantage for learning from human trainers. Earlier work supports this intuition [11].

B. Learning from Reward

In the machine-learning literature, one common goal is to learn to maximize an unknown reward function. In this problem domain, there are different actions an agent can choose. After selecting an action, the agent will receive a numerical reward based on the chosen action. The agent's goal is to maximize the long-term expected reward, and must balance exploring actions to better estimate their true payout with exploiting the currently estimated best action. While conceptually a simple problem, studies have shown that human learners behave sub-optimally on such tasks [1], [2], suggesting the problem is indeed non-trivial.[1] In a contextual setting, the reward for the different actions will depend on the world's current *state*, which the learner can observe. Further, if the agent's actions determine which state is reached next, the problem is a sequential decision problem and can be addressed in the framework of *reinforcement learning* [15].

In contrast to learning from a fixed numerical reward signal, our work is part of a growing literature that addresses learning from human feedback. Thomaz and Breazeal [16] treated human feedback as a form of guidance for an agent trying to solve a reinforcement-learning problem. The human feedback did not actually change the reward coming from the underlying problem, or the optimal policy, but improved exploration and sped-up learning. The study also found that humans would give reward in anticipation of good actions, rather than rewarding or punishing the agent's recent actions.

There is also a growing body of work that examines how humans *want* to teach agents by providing demonstrations in a sequential decision task [4], or by selecting a sequence of data in a classification task [7]. More similar to our work, Knox *et al.* [9] looks at how people want to provide feedback to learning agents. Knox *et al.* found that when a human reduces the amount of feedback they give over time, forcing the learning agent to make mistakes can increase the rate of feedback. Our work differs because we focus on how humans naturally provide feedback when training, not how to manipulate that feedback.

III. A Probabilistic Model of Human Feedback

This section presents a probabilistic model of human feedback that encapsulates differences in trainer's categorical feedback strategies. This model forms the basis of algorithms designed explicitly to learn in this paradigm [11]. We model the learning problem as a set of discrete observations of the environment and a set of discrete actions that can be taken. The behavior being trained is represented as a *policy* that is a mapping from observations to actions. At any time, the trainer may give explicit reward (R$+$), explicit punishment (P$+$), or do nothing (which corresponds to either R$-$ or P$-$ depending on the strategy being used).

We can classify a trainer's strategy by the cases in which they give explicit feedback. Under an R$+$/P$+$ strategy, a trainer typically gives explicit reward for correct actions and explicit punishment for incorrect actions. Under an R$+$/P$-$ strategy, correct actions typically get an explicit reward and incorrect actions typically get no response, while an R$-$/P$+$ strategy typically provides no response for correct actions and explicit punishment for incorrect actions. An R$-$/P$-$ strategy rarely gives explicit feedback of any type. Under an R$+$/P$-$ strategy, the lack of feedback can be interpreted as *implicitly* negative, and under an R$-$/P$+$ strategy, it can be interpreted as implicitly positive—the lack of feedback can be as informative as explicit feedback.

Our model can be used both to learn from human trainers through Bayesian inference, and to identify the strategies that those trainers are following. Our model assumes that the trainer first determines if the action taken was consistent with some target policy λ^* for the current observation, with some probability of error ϵ. The trainer then decides whether to give explicit feedback or simply do nothing. If the trainer interprets the learner's action as correct, then she will give an explicit reward with probability $1-\mu^+$, and if she interprets the action as incorrect, will give explicit punishment with probability $1-\mu^-$. So, if the learner takes a correct action it will receive explicit reward with probability $(1-\epsilon)(1-\mu^+)$, explicit punishment with probability $\epsilon(1-\mu^-)$, and will receive no feedback with probability $(1-\epsilon)\mu^+ + \epsilon\mu^-$.

The parameters μ^+ and μ^- encode the trainer's strategy. For example, $\mu^+ = 0.1, \mu^- = 0.1$ correspond to an R$+$/P$+$

[1] In our work, the human trainers define the correct policy and provide rewards, rather than attempt to learn from those rewards.

strategy where nearly every action receives explicit feedback, while $\mu^+ = 0.1, \mu^- = 0.9$ correspond to an R+/P− strategy, where only actions interpreted as correct tend to receive explicit feedback. Putting these elements together, for time step t (each time step corresponds to the agent observing the world, choosing an action, and receiving an explicit/implicit feedback from the trainer), we have a distribution over the feedback f_t conditioned on the observation o_t, action a_t, and the trainer's target policy λ^*,

$$p(f_t = f^+ | o_t, a_t, \lambda^*) = \begin{cases} (1-\epsilon)(1-\mu^+), & \lambda^*(o_t) = a_t \\ \epsilon(1-\mu^+), & \lambda^*(o_t) \neq a_t, \end{cases}$$

$$p(f_t = f^- | o_t, a_t, \lambda^*) = \begin{cases} \epsilon(1-\mu^-), & \lambda^*(o_t) = a_t \\ (1-\epsilon)(1-\mu^-), & \lambda^*(o_t) \neq a_t, \end{cases}$$

$$p(f_t = f^0 | o_t, a_t, \lambda^*) = \begin{cases} (1-\epsilon)\mu^+ + \epsilon\mu^-, & \lambda^*(o_t) = a_t \\ \epsilon\mu^+ + (1-\epsilon)\mu^-, & \lambda^*(o_t) \neq a_t. \end{cases}$$

Here, f^+ is an explicit positive feedback, f^- is an explicit negative feedback, and f^0 represents a lack of feedback.

What is important to note about this model is that, depending on the strategy (and the corresponding μ^+ and μ^- parameters) used, the lack of feedback may be more probable for correct actions than incorrect actions, or *vice versa*. Therefore, the correct inference to make from a lack of feedback depends on the training strategy being used. This model formalizes the idea that learning depends on the training strategy being employed.

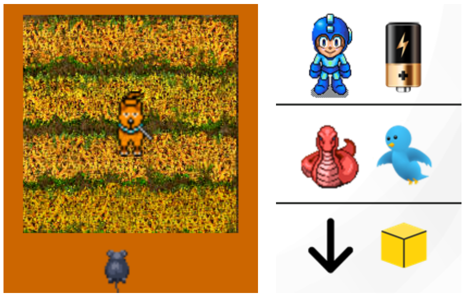
We used two learning algorithms (SABL and I-SABL) [11] that inferred the correct behavior using our feedback model, as well as two algorithms (M_{-0}, M_{+0}) based on algorithms in the literature on maximizing human feedback considered as reward [6], [8].

IV. User Studies

To assess choices of feedback strategies, we conducted three user studies in which participants trained a virtual agent to move towards objects as they approached from different sides of the screen. In the first two studies, this agent was represented by a drawing of a dog and the object to be approached as a rat, which would run away when the dog moved towards it. The third study used a number of different visual representations, to gauge the effect of the agent's appearance on the user's behavior.

In our training task, learning agent was drawn at the center of the screen, and the objects arrived once every two seconds from the edges of the screen. The objects came from three points along each of the four edges, resulting in possible 12 observations. When an object appeared, the agent moved from the center towards one of the edges. If the learner moved towards the edge from which the object was coming, that object was chased away. If the learner ran to a different edge, the object entered the field in the center and disappeared. Figure 1 shows the agent and task environment as they appeared in the first and second studies, as well as the alternative sprites used in the third study.

To train the learner to chase the objects away, users could provide reward, punishment, or no feedback. Users signaled when training was complete by pressing a button. Data for a training session was included only if it was terminated by the user signaling it was complete.



(a) Task Interface

(b) Sprites

Fig. 1: A screenshot of the study interface (a). Additional buttons that begin and end training have been cropped out. In the third study, the dog sprite could also have been a robot, a snake, or an arrow (b).

In each of the studies, users filled out a survey indicating their age, gender, education, history with dog ownership, experience in training dogs, and with which dog-training paradigms they were familiar (if any). After completing the initial survey, but before beginning training, users were taken through a tutorial, which first animated approaching objects and then instructed the user how to reward and punish the learner. After the tutorial, the users began a series of training sessions; each session was performed with a different virtual agent that learned from scratch. The user was told that each session required new training.

After each session, participants were shown a textual input box, and were asked: "Please describe the strategy you used when training the [agent] during the previous experiment. For example, when did you provide reward/punishment or when did you decide to change the task or start over (if appropriate)? Is there anything else you want to say about training the [agent]?"

A. First and Second studies

The first an second studies focused on how training strategies differed between users for a fixed training task, and on how a user's prior training experience affected their choice of strategy. As such, the learning agent in these studies was represented as a drawing of a dog, and the approaching object as a rat. In both the first and second studies, each training session used a different learning algorithm (in random order).

Participants for the first two studies were recruited from three different sources: (1) a senior-level game design class at North Carolina State University (credit was offered for participation), (2) a computer science departmental mailing list, and (3) two Internet communities focused on dog training (a Facebook group about positive-reinforcement training and a Japanese dog forum). Although the recruiting sources were the same for the first two studies, the distribution from each source was different since recruitment was performed at different times. Furthermore, different training algorithms were used in the first two studies: M_{-0}, M_{+0}, and SABL in the first; and SABL and I-SABL in the second.

B. Third study

It is possible that some users would avoid punishment when the agent appears as a dog, with which people often have positive associations and often avoid punishing in real life training. If so, and if the agent being trained was not represented as a dog, the distribution of strategies used might differ from that found when the agent does appear as a dog.

The third study addressed this question by using a number of different visual representations for the learning agent. The third study was published on Amazon's Mechanical Turk system as a set of Human Intelligence Tasks, and three separate tasks were published. Participants were recruited separately for each of these tasks, and each task had its own set of users. We had a total of 211 participants between the three tasks in the third study.

This study used a similar survey, interface, training task, and learning algorithm to thr first and second studies, but to assess how the depiction of the learning agent affects the trainers' strategies, the images used to represent the agent and the approaching object were varied. In addition to the dog/rat (Figure 1(a)), the third study used robot/battery, snake/bird, and arrow/square (Figure 1(b)) sprites, believing that these sprites would lead to varying degrees of anthropomorphisation of the agent by users.

The first task had two conditions, presented in random order to the user. The first condition had a dog chasing rats away from a field, while the second condition replaced the dog and the rat with a robot and a battery, though the mechanics of the environment were the same. The second and third Mechanical Turk tasks each had only one condition. The second task replaced the dog and rat with a snake and a bird, while the third task used an abstract arrow and a square.

It should be noted that, in the third study, workers in Mechanical Turk were compensated a base amount of \$0.25 for their participation in the study, and were offered an additional \$0.25 bonus if the agent learned to act appropriately in more than 90% of cases. This added incentive encouraged participants to take the task seriously, but may have introduced some bias in the resulting strategies.

V. Analysis of Training Strategies

Our core hypothesis was that human trainers follow a variety of strategies when teaching behaviors using feedback. As such, we characterized the distribution of different training strategies, and the factors that influenced that distribution. We were most interested in the occurrence of the R+/P− and R−/P+ strategies, as these strategies allow our model of trainer feedback to interpret meaning of the lack of feedback.

We used our probabilistic model of the training process to categorize the strategies that participants in our studies followed. As discussed previously, we group strategies into four categories: R+/P+, R+/P−, R−/P+ and R−/P−, by the conditions under which they do and do not provide explicit feedback. Specifically, we estimated the μ^+ and μ^- parameters used for each training session by computing the fraction of correct and incorrect actions that did not receive explicit feedback. The strategy for a session was classified as R+ if μ^+ was less than $\frac{1}{2}$, and R− otherwise. Similarly, the strategy for a session was classified as P+ if μ^- was less than $\frac{1}{2}$, and P− otherwise. (Recall that low μ^+ and μ^- values correspond to frequent *explicit* feedback.)

TABLE I: Frequency of strategies in the first and second studies.

	P+	P−
R+	93	125
R−	6	3

TABLE II: The number of participants beginning a training session using one strategy (rows) and ending it using another (columns) during the first study. Entries on the diagonal indicate no switch.

end
begin
	R+/P+	R+/P−	R−/P+	R−/P−

R+/P+	65	4	2	0
R+/P−	10	52	1	1
R−/P+	2	1	4	1
R−/P−	0	0	0	1

For the first and second studies we consider data from the 105 users who completed at least one experiment. Table I summarizes the distribution of training strategies from the first two studies. Recall that some participants for these studies were explicitly recruited due to their experience training dogs and they trained a learner depicted with a dog sprite (Figure 1(a)). We found that, reassuringly, the least popular strategy was R−/P−, as such a strategy gives the learner very limited feedback. Overall, the dominant strategies in these studies were R+/P− (frequent rewards, few punishments) and R+/P+ (frequent rewards and punishments).

We expected the R+/P+ strategy to be common, because the strategy represents providing as much information to the virtual dog as possible. As one participant put it, "I just punished the dog if they went to the wrong side and rewarded them when they went to the right side." We also expected to see many users using the R+/P− strategy, since it is a common dog-training paradigm. One participant explained, "I tried to Reward only. Rewarded when the dog was moving or had moved toward the rat, and provided no opportunity for Reward when the dog moved away from the rat."

Some participants changed strategies during training sessions. Table II shows how trainers in the first study changed strategies over time. There was no change in most cases (84.7%), but switching from R+/P− to R+/P+ was common (15.6%), and was often associated with improved agent performance, possibly to maintain a good behavior once it had been learned. The existence of multiple strategies and changes in strategy suggest that learning algorithms need to be aware of strategy during the training process. While existing work has addressed trainers changing their strategies by actively encouraging users to give certain types of feedback [10], it may be more effective to integrate the notion of strategy change with an overall model of trainer feedback, such as the one presented here.

A. Effects of Dog-Training Experience

As we are interested in the degree to which a participant's training experience influenced their strategy, we asked each

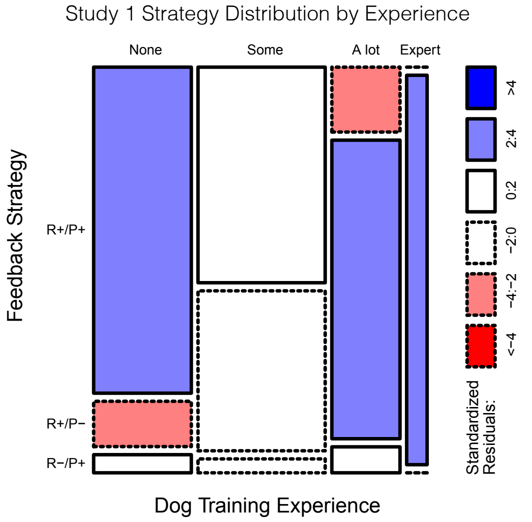


Fig. 2: A mosaic plot (generated with the R language) with Pearson residuals for strategies in the first study, grouped by dog-training experience. Users with no experience were likely to use R+/P+, users with some experience were likely to use R+/P−. Differences were 2–4 standard deviations from expected (significant with 95% confidence).

user to rate their level of experience in dog training on a four-point scale from "None" to "I am an Expert." Many participants had no experience training dogs, and those that did varied in their degree of experience.

To visualize these results, we organize the data into a contingency table and depict it as a *residual mosaic plot* (see Figure 2). There are a few important things to note about such plots. The data is organized into boxes, with one column of boxes for each value of one of the categorical variables. The order of the boxes within each column follows the set of values of the other categorical variable. The area of a box in the plot indicates the number of responses in that category. The width of each box represents, in aggregate, the probability that a response will fall into that column, regardless of which row it is in, *e.g.*, $\Pr(\text{Experience}{=}\text{Some})$.

The height of a box indicates the amount of data in that column when the value of the row is considered, *e.g.*, $\Pr(\text{Strategy} = \text{R+/P}- \quad |\text{Experience} = \text{None})$. Thus, the more asymmetric any box is, the more it deviates from the expected value; tall thin rectangles indicate more data in that entry than expected and short wide rectangles indicate fewer data in that entry than expected.

Additionally, the color of an entry indicates whether or not the rectangular shape of an entry represents a significant deviation from the expected value. A shaded entry means that the value that box represents is more than two standard deviations above (or below) the expected value, and is therefore significant with 95% confidence. If the border of the cell is solid, then the deviation is above the expected value, if it is dashed, it is below expected.

Figure 2 shows the relationship between dog-training experience and the employed feedback strategy in a mosaic plot, for participants in the first study. As a common approach to dog training is to only use positive feedback, we expected that users with dog-training experience would be more likely to use R+/P− than those without experience.

TABLE III: Breakdown of strategies used in the third study when training an agent appearing as a dog, robot, snake or arrow.

agent	R+/P+	R+/P−	R−/P+	R−/P−
dog	151	25	1	1
robot	188	21	0	4
snake	64	7	2	3
arrow	43	6	1	2

In the first study, we found that the more dog-training experience a user had, the more likely they were to use the R+/P− strategy. This relationship was found to be statistically significant at the 95% confidence level. However, this relationship did not appear as strong in the second user study in which users with at least some experience were very likely to use R+/P− (results not shown). This difference likely reflects differences in the distribution of participants between the two studies, with the second study having only four participants with no training experience.

Both the first and second studies specifically recruited participants with dog-training experience, and that choice almost certainly affected the observed frequency of different strategies. The third study, however, drew its participants from Amazon Mechanical Turk, and so should have no bias towards users with training experience.

Table III summarizes the distribution of strategies used in third study. We only report data from training sessions where at least 50% policy accuracy was achieved.[2] In this study, unlike the first two, R+/P+ strategies were much more common than R+/P− strategies. However, R+/P− strategies were still common, and still occurred much more frequently than R−/P+ or R−/P− strategies.

B. *Effect of Agent Appearance*

The third study also addressed the question of whether the appearance of the agent would affect the distribution of strategies used, either because users believed that an agent resembling a dog would respond better to strategies that are effective with real dogs, or because the appearance of an animal made users more averse to giving punishment. Recall that the third study used the same learning task as the first and second studies, but varied the sprites between a dog/rat, robot/battery, snake/bird, or arrow/box.

As shown in Table III, the distribution of strategies in the third study was relatively insensitive to the agent's appearance. Fisher's exact test shows that the number of times each of the four strategies was used was not significantly different ($p > 0.21$) between subjects training the dog and those training the robot. Similarly, we did not see differences in strategies between the snake and the arrow ($p = 0.10$).

Nonetheless, there is some evidence that the learning agent's sprite did influence trainers' choices of strategies. Consider Figure 3, which shows the distribution of dog-training experience for those trainers that used the R+/P− strategy, grouped by sprite. What is interesting to note is

[2] We exclude more data in the third study to remove participants who do the minimum amount of work to receive their compensation.

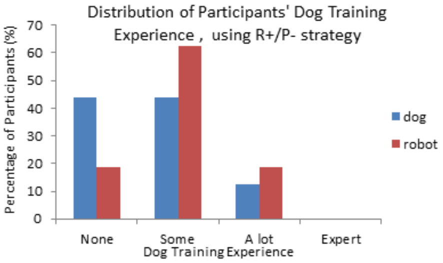


Fig. 3: A bar plot showing the distribution of participants in the third study who used the R+/P− strategy, based on their experience with dog training, grouped by the sprite they were training.

that participants with dog training experience used R+/P− in roughly equal proportion when training the dog and the robot; however, for participants without dog training experience, it appears a higher percentage used the reward-focused strategy on the dog when compared to the robot. One plausible explanation is that empathy toward the dog caused users to avoid explicit punishment, even if they were unfamiliar with dog-training techniques.

C. Trainer Mistakes

On of the main assumptions of our probabilistic model that trainers can make mistakes when providing feedback (the ϵ parameter). The results of all three studies demonstrate that trainer errors are common, and that any approach to learning from feedback must be able to recover from such errors.

Note that since we cannot know if a user made a mistake for actions that did not receive feedback, we can only estimate ϵ from cases in which explicit feedback is provided. We estimated the average ϵ for participants in the first and second studies combined to be 0.085. In the first task of the third study, where agents were represented as both dogs and as robots, the estimated average ϵ was 0.034.

The comments made by some of the participants suggest possible sources of error. One participant explained, "... i kept getting mixed up at first and hitting the wrong buttons..." , suggesting that error could be reduced with a clearer interface design and more user practice. Another user commented, "At first it got frustrating because my timing was off on the reward and punishment. That doesn't help the dog and they become afraid and stay away because they are confused." Our model does not currently account errors in the timing of feedback. This problem, however, may be mitigated by taking the weighted average of feedback over a longer time window, as in related work [8].

VI. Conclusion

Previous work on learning from human trainers has had some success treating human feedback as a numeric reward. However, the results in this paper demonstrate that the complexities of humans training strategies are not well captured by the numeric paradigm. Specifically, we showed that 1) different users adopt different training strategies when teaching an agent, and their choice of strategy can be affected by their training background; 2) feedback strategies can change over the course of a training session; and 3) there is some evidence that the feedback strategy chosen may depend on the type of agent the user believes they are training.

Taken together, these findings are a strong indication that there is a critical need to develop algorithms that explicitly model the feedback strategy a trainer uses. As a starting point, we provided a probabilistic model of human feedback to can describe the types of strategies we observed in our data. Expanding the model, and algorithms that use it, is a ripe area for new research. Accordingly, we hope this paper will raise awareness of the complexities of feedback among researchers working on learning from human teachers and engender the development of additional algorithms explicitly designed to learn from categorical feedback strategies.

Acknowledgements

This work was supported in part by NSF IIS-1149917 and NSF IIS-1319412.

References

[1] Acuna, D., and Schrater, P. Bayesian modeling of human sequential decision-making on the multi-armed bandit problem. In *Proceedings of CogSci* (2008), 200–300.

[2] Anderson, C. Ambiguity aversion in multi-armed bandit problems. *Theory and Decision 72* (2012), 15–33.

[3] Argall, B. D., Chernova, S., Veloso, M., and Browning, B. A survey of robot learning from demonstration. *Robot. Auton. Syst. 57*, 5 (May 2009), 469–483.

[4] Cakmak, M., and Lopes, M. Algorithmic and Human Teaching of Sequential Decision Tasks. In *Proceedings of AAAI* (2012).

[5] Hiby, E., Rooney, N., and Bradshaw, J. Dog training methods: their use, effectiveness and interaction with behaviour and welfare. *Animal Welfare 13*, 1 (2004), 63–70.

[6] Isbell C.L., J., Shelton, C., Kearns, M., Singh, S., and Stone, P. A social reinforcement learning agent. In *Proceedings of Autonomous Agents* (2001), 377–384.

[7] Khan, F., Zhu, X. J., and Mutlu, B. How do humans teach: On curriculum learning and teaching dimension. In *Proceedings of NIPS* (2011), 1449–1457.

[8] Knox, W., Stone, P., and Breazeal, C. Training a robot via human feedback: A case study. In *Social Robotics*, vol. 8239 of *Lecture Notes in Computer Science*. 2013, 460–470.

[9] Knox, W. B., Glass, B. D., Love, B. C., Maddox, W. T., and Stone, P. How humans teach agents - a new experimental perspective. *I. J. Social Robotics 4*, 4 (2012), 409–421.

[10] Li, G., Hung, H., Whiteson, S., and Knox, W. B. Using informative behavior to increase engagement in the TAMER framework. In *AAMAS 2013: Proceedings of the Twelfth International Joint Conference on Autonomous Agents and Multi-Agent Systems* (May 2013), 909–916.

[11] Loftin, R., MacGlashan, J., Peng, B., Taylor, M. E., Littman, M. L., Huang, J., and Roberts, D. L. A Strategy-Aware Technique for Learning Behaviors from Discrete Human Feedback. In *Proceedings of AAAI* (2014).

[12] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al. Scratch: programming for all. *Comm. of the ACM 52*, 11 (2009), 60–67.

[13] Skinner, B. F. *The behavior of organisms: An experimental analysis.* Appleton-Century, 1938.

[14] Skinner, B. F. *Science and Human Behavior.* Macmillan, 1953.

[15] Sutton, R., and Barto, A. *Reinforcement learning: An introduction*, vol. 116. Cambridge Univ Press, 1998.

[16] Thomaz, A. L., and Breazeal, C. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Proceedings of AAAI* (2006), 1000–1005.