A Strategy-Aware Technique for Learning Behaviors from Discrete Human Feedback

Robert Loftin
North Carolina State University
rtloftin@ncsu.edu

James MacGlashan
Department of Computer Science
Brown University
jmacglashan@gmail.com

Michael L. Littman
Department of Computer Science
Brown University
mlittman@cs.brown.edu

Matthew E. Taylor
School of Electrical Engineering and Computer Science
Washington State University
taylorm@eecs.wsu.edu

David L. Roberts
North Carolina State University
robertsd@csc.ncsu.edu

February 25, 2014

Abstract

This paper introduces two novel algorithms for learning behaviors from human-provided rewards. The primary novelty of these algorithms is that instead of treating the feedback as a numeric reward signal, they interpret feedback as a form of discrete communication that depends on both the behavior the trainer is trying to teach and the teaching strategy used by the trainer. For example, some human trainers use a lack of feedback to indicate whether actions are correct or incorrect, and interpreting this lack of feedback accurately can significantly improve learning speed. Results from user studies show that humans use a variety of training strategies in practice and both algorithms can learn a contextual bandit task faster than algorithms that treat the feedback as numeric. Additionally, simulated trainers are employed to evaluate the algorithms in both contextual bandit and sequential decision-making tasks with similar results.

Introduction

A significant body of work exists on the problem of learning from human trainers [12, 7, 3], and specifically on the problem of learning from trainer-provided feedback [6, 9]. Existing work can be grouped into two broad categories: (1) learning from demonstration, which treats inputs from human trainers as examples of some target behavior; and (2) learning from trainer-provided feedback, which models the learning problem as a reinforcement-learning task. While exciting developments have been made in both these areas, we argue neither is always an appropriate model for learning in a common paradigm of human teaching. First, providing examples of behavior is not always feasible or desirable. Second, the positive or negative feedback given by humans is not representative of a numerical reward value.

Feedback is a form of discrete communication between a trainer and a learning agent. Accordingly, that communication can be implemented using a few different *training strategies* that describe how trainers choose what feedback to give. We show how the training strategies employed by human teachers vary in the relative amount of positive and negative feedback given. For example, a trainer may choose to provide positive feedback when the learner takes a correct action, but provide no response when the learner takes an incorrect action. When the trainer employs such a strategy, the learner could interpret the lack of a response as a form of feedback in and of itself. If only negative feedback is given, then the lack of feedback is implicitly positive and *vice versa*. We report results of user studies that demonstrate human trainers using a variety of strategies, including those where a lack of feedback is meaningful.

We derive two Bayesian learning algorithms explicitly designed to model and leverage discrete feedback strategies. Our algorithms, which we refer to as *Strategy-Aware Bayesian Learning* (SABL) and *Inferring Strategy-Aware Bayesian Learning* (I-SABL), are designed to learn with fewer discrete feedbacks than existing techniques, while taking as few exploratory actions as possible. We first describe our representation of trainer strategy and the SABL policy-learning algorithm for *contextual bandit* domains. We then extend SABL to I-SABL, an algorithm that can infer an unknown strategy being followed by the trainer based on the feedback they have given. Lastly, we extend these algorithms to sequential domains.

We validate the effectiveness of these algorithms in both online user studies and experiments with simulated trainers. Results indicate that, when learning from both human and simulated trainers, our algorithms learn behaviors with fewer actions and fewer feedbacks (and hence less effort on the part of the trainers) than baseline algorithms that interpret feedback as numerical reward. We also demonstrate that the I-SABL algorithm is able to infer trainers' strategies from the feedback provided and take advantage of that knowledge to improve learning performance. Results using simulated trainers in both the contextual bandit domain and a sequential domain demonstrate the generality and robustness of SABL and I-SABL. These experiments also show that our algorithms can be adapted to sequential domains where trainers teach policies to reach goal states.

Related Work

The goal in reinforcement learning is to learn to maximize an unknown reward function. In bandit domains the learner selects among possible actions and receives a numerical reward based on the action chosen; the agent's goal is to maximize the long-term expected reward, balancing exploration to better estimate actions' true payouts with exploiting the currently estimated best action. While conceptually a simple problem, studies have shown that humans behave sub-optimally when learning in such domains [1, 2], suggesting the problem is indeed non-trivial. In a contextual bandit 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 the next state, the problem is a sequential decision problem and is typically addressed using tools from *reinforcement learning* (RL) [11].

In contrast to learning from a numerical reward signal, our work is part of a growing literature on learning from human feedback. Thomaz and Brazeal ([12]) treated human feedback as a form of guidance for an agent trying to solve a RL problem. Human feedback did not change the numerical reward from the RL problem, or the optimal policy, but improved exploration and accelerated learning. Their results show humans give reward in anticipation of good actions, instead of rewarding or punishing the agent's recent actions.

COBOT [6] was an online chat agent with the ability to learn from human agents using RL techniques. It learned how to promote and make useful discussion in a chat room, combining explicit and implicit feedback from multiple human users. The TAMER algorithm [9] has been shown to be effective for learning from human feedback in a number of task domains common in the RL research community. This algorithm is modeled after standard RL methods which learn a value function from human-delivered numerical rewards. At each time step the algorithm updates its estimate of the reward function for a state-action pair using *cumulative* reward.

On the other hand, there is a growing body of work that examines how humans can teach agents by providing demonstrations of a sequential decision task [3], or by selecting a sequence of data in a classification task [7]. More similar to our work, Knox *et al.* ([8]) examine how people want to provide feedback to learning agents: 1) there is little difference in a trainer's feedback whether they think they are providing feedback during learning or if they think they are critiquing a fixed performance; and 2) humans can reduce the amount of feedback they give over time, and forcing the learner to make mistakes can increase the rate of feedback. Our work differs because we focus on designing algorithms that can leverage how humans naturally provide feedback when teaching, not how to manipulate that feedback.

Lastly, feedback types other than numeric reward have also been explored. Heer *et al.* ([5]) describe a variety of feedback strategies employed by film directors, golf instructors, and 911 operators. These experts gave rich feedback and direction in the form of explaining consequences, querying learner understanding, using assistive aids, *etc.* That work stops short of algorithm design.

Motivation: Trainer Strategies

In our training paradigm, the learning agent takes an action and then *may* receive positive or negative feedback from the trainer. Our hypothesis is that trainers may differ in the feedback they provide, even when they are trying to teach the same behavior. For example, even in the absence of user error, when the learner takes an action that is correct, one trainer might provide an explicit positive feedback but another might provide provide no response at all.

We can classify a trainer's strategy by the cases in which they give explicit feedback. Under a *balanced feedback* strategy a trainer typically gives explicit reward for correct actions and explicit punishment for incorrect actions. Under a *reward-focused* strategy, correct actions typically get an explicit reward and incorrect actions typically get no response, while a *punishment-focused* strategy typically provides no response for correct actions and explicit punishment for incorrect actions, and an *inactive* strategy rarely gives explicit feedback of any type. Under a reward-focused strategy, the lack of feedback can be interpreted as an *implicit* negative feedback and under a punishment-focused strategy, the lack of a feedback can be interpreted as implicitly positive. Therefore, to a strategy-aware learner the lack of feedback can be as informative as explicit feedback.

Table 1: Breakdown of strategies used in the user studies

Strategy	**Number of Participants**
balanced feedback	93
reward-focused	125
punishment-focused	6
inactive	3

Table 1 shows the number of participants in our user studies who used each of the four possible strategies. Balanced feedback specifically means that the trainer gave explicit feedback to both correct and incorrect actions more than half of the time, while inactive means the trainer gave explicit feedback less than half the time in both cases. Reward-focused means that correct actions received explicit feedback more than half the time and incorrect actions received explicit feedback less than half the time, while punishment-focused is the opposite case. There are two things we note about the data in that table: 1) users employed all four strategies to some degree; and 2) a large percentage of users followed a reward-focused strategy which relied on implicit negative feedback.

Methods

To start, we represent the learning environment as a *contextual bandit* [10]. We can divide the learning process into *episodes* in which a discrete observation is generated, the learning agent takes an action, and the trainer may provide some response. Our assumption is that the trainer has an observation- or state-action mapping λ, known as a *policy*, that they wish to train the learner to follow. Here, we assume that the trainer can provide discrete feedback for each action, which can be either positive or negative. We also assume that each feedback has a fixed magnitude; that is, there are not different degrees of punishment or reward that can be given.

The SABL Algorithm

Here we present the Strategy-Aware Bayesian Learning (SABL) algorithm. SABL assumes the trainer chooses feedback to provide based only on the most recent observation and action taken. In this model, the trainer first determines if the action was consistent with the 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 $1-\mu^-$. Thus, if the learner actually takes a correct action, then 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, μ^+=0 and μ^-=0 correspond to a balanced strategy where explicit feedback is always given in response to an action, while μ^+=0 and μ^-=1 correspond to a reward-focused strategy, where only actions that are interpreted as correct receive explicit feedback. Putting these elements together, for episode t, we have a distribution over the feedback provided 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. Using this model of feedback, SABL computes a maximum likelihood estimate of the target policy λ^* given the feedback that the user has provided. Thus, the policy output by SABL is its estimate of the trainer's target policy

$$\operatorname*{argmax}_{\lambda} p(h_{1\ldots t}|\lambda^* = \lambda),$$

where h_t is the training history of actions, observations, and feedback. If a user provides multiple feedback signals before an episode ends, SABL only considers the most recent, which gives trainers a chance to correct a mistaken feedback.

I-SABL: Inferring unknown strategies

SABL will perform well when it knows the trainer's μ^+ and μ^- parameters. In practice however, an agent is unlikely to know the training strategy that a trainer uses. I-SABL benefits from the ability of the learner to infer the trainer's strategy based on partial knowledge of the target policy (assuming trainer error ϵ). If the learner knows from explicit feedback the correct action for one observation, it can infer the training strategy by looking at the history of feedback given for that observation. If, for example, more explicit feedback is given for correct actions than incorrect actions, then it is likely the trainer is reward-focused.

Under SABL's probabilistic model we can treat the unknown μ values representing the trainer's strategy as hidden parameters of the model, allowing us to marginalize over possible strategies to compute the likelihood of each possible target policy λ. Inferring-SABL, or I-SABL, computes a maximum likelihood estimate of the target policy, given the training data. Therefore, I-SABL attempts to find

$$\operatorname*{argmax}_{\lambda} \sum_{s \in S} p(h_{1\ldots t}, s|\lambda^* = \lambda),$$

where S is the set of possible training strategies (μ^+, μ^- values), $p(s)$ is uniform for all $s \in S$, and $h_{1\ldots t}$ is the training history up to the current time t.

In general, the space of possible policies will be exponential in the number of states or observations, and so algorithms for approximate inference may be used to compute the likelihood of each policy. Here, we use the Expectation Maximization algorithm [4] to compute a maximum likelihood estimate of the policy the trainer is trying to teach, and treat the unknown μ^+ and μ^- parameters as continuous, hidden variables ranging from 0 to 1. The EM update step is then

$$\lambda_{i+1} = \operatorname*{argmax}_{\lambda \in P} \int_0^1 \int_0^1 p(\mu^+, \mu^-|h, \lambda_i) \ln p(h, \mu^+, \mu^-|\lambda) d\mu^+ d\mu^-,$$

whereλ_i is the current estimate of the policy and λ_{i+1} is the new estimate of the policy. This can be simplified to maximizing the following for a policy's action for each observation o (details omitted for space):

$$\lambda_{i+1}(o) = \operatorname*{argmax}_{a \in A} \left[\alpha(\theta^{o,+}_{a,+} - \theta^{o,-}_{a,+}) + \beta\theta^{o,0}_{a,+} \right],$$

where $\theta^{o,+}$ is the number of positive feedbacks received for observation o, $\theta^{o,+}_{a,+}$ is the number of positive feedbacks where the correct action was performed given that the correct action for o is a, and $\theta^{o,-}$ and $\theta^{o,-}_{a,+}$ are analogous terms for negative feedbacks. Additionally we define

$$\alpha = \ln\left[\frac{(1-\epsilon)}{\epsilon}\right]\int_0^1\int_0^1 p(h|\mu^+,\mu^-,\lambda_i)d\mu^+d\mu^-, \text{ and}$$

$$\beta = \int_0^1\int_0^1 p(h|\mu^+,\mu^-,\lambda_i)\ln\left[\frac{(1-\epsilon)\mu^+ + \epsilon\mu^-}{\epsilon\mu^+ + (1-\epsilon)\mu^-}\right]d\mu^+d\mu^-,$$

values of a simplification of the expectation step, which can be computed once for each EM update.

Experiments

We compare SABL and I-SABL with variants of two algorithms from the literature on learning from human feedback via maximizing numerical reward. Both algorithms maintain an estimate of the expected reward associated with actions for each observation, but differ in their interpretation of no feedback. The first, denoted M$_{-0}$, is similar to TAMER [9] and ignores episodes without feedback. The second, denoted M$_{+0}$, is similar to COBOT [6] and includes episodes with zero reward — value estimates for actions will return to zero after enough episodes with no feedback. Both algorithms associate $+1$ with positive and -1 with negative feedback. Unlike SABL and I-SABL, M$_{-0}$ and M$_{+0}$ use the cumulative value of all feedback given during an episode.

User Studies

To evaluate the algorithm performance when learning from human trainers, we ran a user study in which participants trained learning agents using either SABL, I-SABL, M_{-0}, or M_{+0}, to perform a contextual bandit task. We recruited two groups of users via email, online forums, and social networks to participate in our online study: university students represented how the average computer-savvy person might train an agent, and amateur dog trainers represented participants with experience in training using discrete feedback.

Participants were asked to train an animated dog to chase rats away from a corn field. The dog was drawn at the center of the screen (Figure 1), and rats came one at a time every two seconds from three points along each of the four edges, resulting in twelve total observations. The learning algorithms were given no information about the spatial relationship between observations. The dog (learner) had four actions available: up, down, left, or right. The participants were instructed to provide rewards and/or punishments (using the keyboard) to teach the learner to move in the direction the rat is approaching from. Users decided when to terminate experiments. They were instructed to do so when they felt that the dog had learned the task sufficiently well, or if they felt that the dog would not be able to learn further.

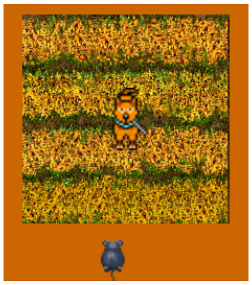

Figure 1: A screenshot of the study interface

We ran two studies with this setup, first with participants from both the dog-training forums and the university, and second with only participants from dog-training forums. The first study evaluated SABL against the M_{-0} and M_{+0} learners. The second study compared I-SABL against SABL. In the first study, 126 users participated, of which 71 completed training at least one learner and 51 completed training at least two learners. In the second study, 43 users participated, while 26 completed training at least one learner, and 18 completed training at least two learners.

The average number of steps it took each agent to reach each of a set of four pre-determined criteria was used as the performance measure. Three of the criteria were when the learner's estimate of the policy was 50%, 75%, and 100% correct. The fourth criterion was the number of steps before the user terminated the experiment. Results from the first user study show that learners using SABL tended to outperform those using M_{-0} and M_{+0}. Figure 2 shows the number of steps to reach each of the four criteria. The bars for SABL are lower than

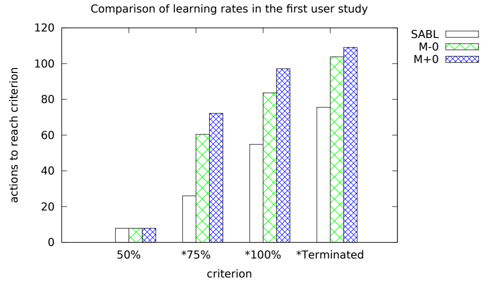

Figure 2: The average number of actions required to teach SABL, M_{-0} and M_{+0} a policy that was at least 50%, 75%, or 100% correct, and until the participants decided to terminate the session (* indicates that differences in performance were statistically significant for that column)

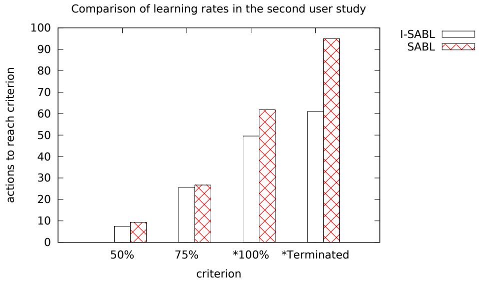

Figure 3: The average number of actions required to teach SABL and I-SABL a policy that was at least 50%, 75%, or 100% correct, and until the participants decided to terminate the session (* indicates that the difference in performance was statistically significant for that column)

their counterparts for the other algorithms, showing that on average the SABL learner took fewer steps to reach the 75%, 100%, and the user termination criteria. Unpaired two sample t-tests show that the differences between the SABL learner and the M_{-0} and M_{+0} learners, for the 75%, 100% and termination criteria, were statistically significant ($p < 0.05$). In addition, a larger percentage of sessions with the SABL learner reached 50%, 75%, and 100% policy correctness before termination than with the M_{-0} and M_{+0} learners. Pearson's χ^2 tests show that the differences between the number of times the SABL learner and the M_{-0} and M_{+0} learners reached the 100% criteria were statistically significant ($p < 0.01$), with the SABL, M_{-0} and M_{+0} learners reaching 100% correctness 53%, 17% and 19% of the time respectively.

In the second study, we compared I-SABL against SABL using the same performance criteria to test whether inferring trainers' strategies improves learning performance. Figure 3 shows the number of actions used for each algorithm to reach the criteria. Of interest here are the very small (statistically insignificant) differences between SABL and I-SABL for the 50% and 75% policy correctness criteria. The difference becomes much larger at the 100% and user-selected termination criteria, where I-SABL reaches each criteria in significantly fewer steps. This finding is expected, as improvements in learning performance for I-SABL would be most pronounced when the agent has received enough feedback for some observations to be able to infer the trainer's strategy. Unpaired t-tests show these performance differences are statistically significant, with $p = 0.01$ for the 100% and $p < 0.05$ for the termination criteria. A larger percentage of sessions with the I-SABL learner reached 50%, 75%, and 100% policy correctness before termination than with the SABL learners. Pearson's χ^2 tests show that the differences between the number of times the I-SABL learner and the SABL learner reached the 100% criteria were significant ($p < 0.01$), with the I-SABL learner reaching 100% policy correctness 50% of the time, and the SABL learner reaching it 23% of the time, respectively.

Simulated Trainer Experiments

To better understand how strategy inference allows I-SABL to outperform SABL, we ran a series of experiments, with simulated trainers in contextual bandit domains, comparing I-SABL against SABL with $\mu^+ = \mu^- = 0.1$ — an assumed balanced feedback strategy. The simulated trainer chose a target policy at random, and generated feedback using the same probabilistic model underlying SABL and I-SABL.

We tested each learning agent on tasks consisting of two, five, 10, 15 and 20 observations and two, three, or four actions. These experiments were conducted for a range of pairs of μ^+ and μ^- values for the simulated trainer. Each μ parameter was varied from 0.0 to 0.8, though we restricted experiments to cases where $\mu^- + \mu^+ \leq 1$. The trainer's error rate $\epsilon = 0.2$, matching

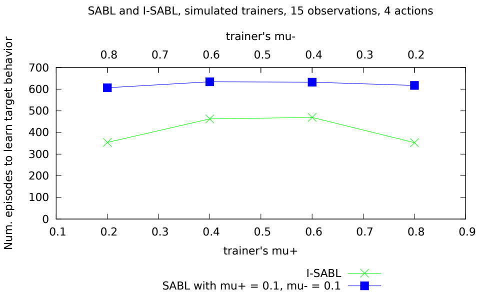

Figure 4: Plot shows the performance of I-SABL and SABL ($\mu^- = \mu^+ = 0.1$) with simulated trainers on a task with 15 observations and 4 actions. The bottom x-axis is the trainer's μ^+, the top x-axis is μ^-, and the y-axis is the number of steps taken to find the target policy.

SABL and I-SABL's assumed value. Learners in these studies took actions at random but kept an estimate of the most likely policy (given the feedback).

The results show that I-SABL is able to take advantage of information from episodes where no explicit feedback is given. Figure 4 shows two curves representing the number of steps it took the SABL and I-SABL agents to find the correct policy, for varying μ parameters. The difference in performance between I-SABL and SABL increases (in favor of I-SABL) as the trainer's μ parameters diverge from the balanced strategy that SABL assumes. Moreover, I-SABL remains competitive (relative to SABL) even when the trainer's μ parameters represent a balanced strategy.

Sequential Tasks

Experiments presented thus far apply SABL and I-SABL to contextual bandit domains. However, we can also apply these algorithms to sequential decision making domains. For efficiency, we can restrict the set of policies considered by assuming that the trainer is trying to teach a policy that is represented by a reward function from a class of reward functions defined over the domain. In this case, we can limit SABL/I-SABL to reason over optimal policies for each of those reward functions. In a grid world, for example, we can assume that the trainer is trying to teach the agent to navigate to a goal location. The use of reward functions here is a syntactic convenience, not a requirement.

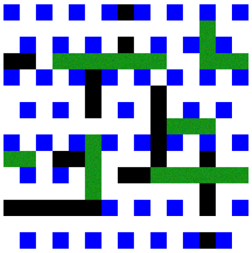

Figure 5: The sequential domain. Blue squares represent possible goal states, black squares represent obstacles of type one and grainy green squares represent obstacles of type two.

We tested SABL and I-SABL for sequential domains in a 15 by 15 grid world with a simulated trainer. The algorithm considered 48 possible goal states, as well as two special kinds of "obstacles" in the world — states the agent could move in or out of but was meant to avoid — depending on the reward function. Each of the possible reward functions returned a value of one when the agent reached the goal location, -100 when the agent entered an obstacle type that was to be avoided, and zero otherwise. There were four different obstacles conditions (no obstacles, avoid type one, avoid type two, avoid both types), resulting in $48 \times 4 = 192$ possible policies. Figure 5 shows the grid world used. Prior to applying SABL and I-SABL, these reward functions were converted to policies by solving the associated Markov Decision Process.

In this case SABL and I-SABL only considered a small, finite set of possible μ parameter combinations, representing balanced, reward-focused, and punishment-focused trainer strategies. Additionally, to leverage this simplification rather than use EM on the entire feedback history at each step, we adapted I-SABL to update its prior belief in each strategy and policy to the posterior probability distribution given by the most recent feedback and the current distribution over trainer strategies. Trainer strategies were defined by $\{\mu^+, \mu^-\} = \{0.1, 0.1\}$ for the balanced feedback strategy, $\{\mu^+, \mu^-\} = \{0.1, 0.9\}$ for the reward-focused strategy, and $\{\mu^+, \mu^-\} = \{0.9, 0.1\}$ for the punishment-focused strategy. We did not consider the inactive strategy, as it was uncommon in the user study.

Trainer's Strategy	Learning Algorithm	Identify Policy	95% Conf. Int.	# Explicit Feedbacks	95% Conf. Interval
balanced	I-SABL	44.4	±11.7	39.1	±10.4
	SABL - balanced feedback	46.7	±9.3	40.5	±8.1
	SABL - reward-focused	67.3	±21.1	60.0	±19.3
	SABL - punishment-focused	65.6	±20.6	58.1	±18.5
reward-focused	I-SABL	68.7	±20.5	54.1	±17.7
	SABL - balanced feedback	152.8	±27.9	71.4	±18.2
	SABL - reward-focused	65	±23.8	50.8	±20.4
	SABL - punishment-focused	N/A	N/A	N/A	N/A
punishment-focused	I-SABL	76.2	±25.4	14.8	±3.9
	SABL - balanced feedback	190.9	±27.3	37.4	±4.5
	SABL - reward-focused	N/A	N/A	N/A	N/A
	SABL - punishment-focused	51.3	±17.9	11.1	±2.8

Table 2: For all algorithm and simulated trainer pairs tested, the average number of steps before the agent correctly identified the intended policy as the most likely and the average number of explicit feedbacks that were provided before the intended task was identified as the most likely. "N/A" indicates that the algorithm was unable to learn the correct policy in the majority of training runs

For all strategies, $\epsilon = 0.05$.

Table 2 summarizes the results for all algorithm and trainer strategy pairs. For all simulated trainers, I-SABL and SABL using the correct feedback strategy identified the intended policy the fastest, again demonstrating that I-SABL does not suffer significantly from initial uncertainty about the trainer strategy. When the simulated trainer used a balanced-strategy, SABL using incorrect strategy assumptions performed worse, but not significantly worse. This lack of significant difference likely results from the fact that in this experiment the simulated trainer rarely failed to give explicit feedback. Regardless of their strategy assumption, SABL learners always interpret explicit feedback in the same way. However, when the trainer does not employ a balanced strategy, incorrect SABL assumptions were be more problematic. If SABL assumes a balanced feedback strategy while the trainer follows a reward-focused strategy, the policy can be learned, but more steps are needed to do so because many steps do not receive explicit feedback and are thus ignored. If SABL assumes the opposite focused feedback strategy (e.g., assuming punishment-focused when it was actually reward-focused), then the agent may never be able to learn the correct policy. Assuming the opposite focused feedback strategy likely performs so poorly because it misinterprets what a lack of feedback means. For instance, if SABL assumes a punishment-focused strategy when it's actually a reward focused strategy, it will interpret the lack of feedback when it's doing the incorrect thing as evidence that it's doing the correct thing.

An interesting fact to note in these results is how few explicit feedbacks are required for I-SABL and SABL (with the correct strategy assumption) to learn the task when the trainer employs a punishment-focused feedback strategy. This result occurs because, as the agent narrows in on the correct task, most of the actions the agent is taking are correct, which results in a lack of explicit feedback; since I-SABL (and SABL assuming a punishment-focused strategy) correctly interprets this lack of feedback as support, the lack of explicit feedback does not hinder learning.

Conclusion

Initially we argued that existing work on learning from humans which considers human input as either a demonstration or a numerical reward is not always sufficient for describing the ways in which human trainers provide feedback. We presented empirical data indicating that humans deliver discrete feedback and follow different training strategies when teaching agents — an insight at odds with existing approaches to learning from humans. To leverage the information about feedback when those strategies are modeled, we have developed two variants of a Bayesian learning algorithm, SABL and I-SABL. SABL encodes assumptions about trainer strategies using parameters describing the probability of explicit feedback given the correctness of actions, and I-SABL uses expectation maximization to infer those parameters online.

The results of our user studies and simulation experiments demonstrate effectively that the SABL and I-SABL algorithms learn in substantially fewer episodes, and with fewer feedbacks, than algorithms modeled after existing numerical-reward-maximizing algorithms from the literature. We have demonstrated this advantage even in cases where the trainer's strategy is initially unknown. Further, we have shown this approach can be applied effectively both in contextual bandit and sequential decision making domains.

References

[1] ACUNA, D., AND SCHRATER, P. Bayesian modeling of human sequential decision-making on the multi-armed bandit problem. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society* (2008), vol. 100, Washington, DC: Cognitive Science Society, pp. 200–300.

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

[3] CAKMAK, M., AND LOPES, M. Algorithmic and Human Teaching of Sequential Decision Tasks. In *Proceedings of AAAI* (2012).

[4] DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological) 39*, 1 (1977), pp. 1–38.

[5] HEER, J., GOOD, N. S., RAMIREZ, A., DAVIS, M., AND MANKOFF, J. Presiding over accidents: system direction of human action. In *Proc. of CHI* (2004). pages 463–470.

[6] ISBELL, C.L., J., SHELTON, C., KEARNS, M., SINGH, S., AND STONE, P. A social reinforcement learning agent. pp. 377 – 384. Learning agents;.

[7] KHAN, F., ZHU, X. J., AND MUTLU, B. How do humans teach: On curriculum learning and teaching dimension. In *Proceedings of NIPS* (2011), pp. 1449–1457.

[8] 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.

[9] KNOX, W. B., AND STONE, P. Interactively shaping agents via human reinforcement: The tamer framework. pp. 9 – 16.

[10] LU, T., PAL, D., AND PAL, M. Contextual multi-armed bandits. In *Proceedings of the 13th international conference on Artificial Intelligence and Statistics* (2010), Citeseer.

[11] SUTTON, R., AND BARTO, A. *Reinforcement learning: An introduction*, vol. 116. Cambridge Univ Press, 1998.

[12] THOMAZ, A. L., AND BREAZEAL, C. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Proceedings of the National Conference on Artificial Intelligence* (2006), vol. 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 1000.