Reinforcement Learning with Human Feedback in Mountain Car

W. Bradley Knox
University of Texas at Austin
Department of Computer Science

Adam Setapen
Massachusetts Institute of Technology
Media Lab

Peter Stone
University of Texas at Austin
Department of Computer Science

Abstract

As computational agents are increasingly used beyond research labs, their success will depend on their ability to learn new skills and adapt to their dynamic, complex environments. If human users — without programming skills — can transfer their task knowledge to the agents, learning rates can increase dramatically, reducing costly trials. The TAMER framework guides the design of agents whose behavior can be shaped through signals of approval and disapproval, a natural form of human feedback. Whereas early work on TAMER assumed that the agent's only feedback was from the human teacher, this paper considers the scenario of an agent within a Markov decision process (MDP), receiving and simultaneously learning from both MDP reward and human reinforcement signals. Preserving MDP reward as the determinant of optimal behavior, we test two methods of combining human reinforcement and MDP reward and analyze their respective performances. Both methods create a predictive model, $\hat{H}$, of human reinforcement and use that model in different ways to augment a reinforcement learning (RL) algorithm. We additionally introduce a technique for appropriately determining the magnitude of the model's influence on the RL algorithm throughout time and the state space.

Introduction

Computational agents may soon be prevalent in society, and many of their end users will want them to learn to perform new tasks. Since some of the desired tasks will have potentially costly failures (e.g. dishwashing by a robot), we must create methods to speed learning, minimizing costly learning trials. For many of these tasks, the human user will already have significant task knowledge. Consequently, we believe we must enable non-technical users to transfer their task knowledge to the agent, reducing the cost of learning without hurting the agent's final, asymptotic behavior.

The TAMER framework guides the design of agents that learn by shaping — using signals of approval and disapproval to teach an agent a desired behavior (Knox and Stone 2009). Past work on TAMER assumes that the human provides the only evaluation signal and the agent uses that signal to learn the behavior desired by the human trainer. But for many tasks, an MDP reward signal is also available. In this paper, we ask how an agent can learn from both human reinforcement and MDP reward simultaneously, keeping MDP reward as the determinant of optimal behavior. We introduce two candidate TAMER+RL algorithms, testing the effect of human training on each at different points along the learning curve. These two algorithms comprise the first learning agents that create a predictive model of human reinforcement, $\hat{H}$, while simultaneously using its current $\hat{H}$ model to guide reinforcement learning. As a part of both algorithms, we introduce a method for appropriately determining the magnitude of the model's influence on the RL algorithm throughout time and the state space. We find that one algorithm, action biasing, consistently improves upon the RL algorithm's solo performance. Further, we note that certain potentially catastrophic flaws in $\hat{H}$ can be overcome by the TAMER+RL algorithms.

Reinforcement Learning

Here we introduce reinforcement learning (RL) with a focus on the nature of MDP reward.

We assume that the task environment is a Markov decision process specified by the tuple (S, A, T, γ, D, R). S and A are respectively the sets of possible states and actions. T is a transition function, $T : S \times A \times S \rightarrow \mathbb{R}$, which gives the probability, given a state and an action, of transitioning to another state on the next time step. γ, the discount factor, exponentially decreases the value of a future reward. D is the distribution of start states. R is a reward function, $R : S \times A \times S \rightarrow \mathbb{R}$, where the reward is a function of s_t, a, and s_{t+1}.

Reinforcement learning algorithms (see Sutton and Barto (1998)), seek to learn policies ($\pi : S \rightarrow A$) for an MDP that maximize return from each state-action pair, where $return = \sum_{t=0}^{t} E[\gamma^t R(s_t, a_t)]$. In this paper, we focus on using value-function-based RL methods, namely SARSA(λ)(Sutton and Barto 1998), to augment the TAMER-based learning that can be done directly from a human's reinforcement signal.

The MDP Reward Signal. The reward signal within an MDP is often characterized as *sparse* and *delayed*. A typical reward signal is sparse because discriminating reward is rarely received (e.g., an episodic domain in which all non-terminal transitions receive -1 reward). It is delayed because

Copyright © 2011, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

actions may not affect the immediate reward signal, instead merely building towards later feedback. From this *impoverished* informational signal, value-function-based reinforcement learning agents learn estimates of return to combat the signal's sparse and delayed character.

However, MDP reward can be described as *flawless* by definition; along with the transition function, it determines optimal behavior — the set of optimal polices that, for each state, choose the action with the highest possible return.

The TAMER Framework for Interactive Shaping

This section introduces the TAMER framework and motivates it by contrasting human reinforcement with MDP reward. TAMER circumvents the sparse and delayed nature of the MDP reward signal by replacing it with a human reward signal. Though flawed, human feedback can be exploited to learn good, if not optimal, behavior more efficiently (Knox and Stone 2009).

The TAMER Framework is an approach to the Shaping Problem, which is: given a human trainer observing an agent's behavior and delivering evaluative reinforcement signals, how should the agent be designed to leverage the human reinforcement signals to learn good behavior? A formal definition can be found in Knox and Stone (2009). For research, the Shaping Problem is restricted to domains with a predefined performance metric to allow experimental evaluation. However, shaping will also be helpful when no metric is defined, as would likely be the case with an end-user training a service robot.

The Human Reinforcement Signal

To motivate TAMER's approach to shaping, we first consider what information is contained in the human reinforcement signal and how it differs from an MDP reward signal. When a human trainer is observing an agent's behavior, he has a model of the long-term effect of that behavior. Consequently, a human reinforcement signal is *rich*, containing information about whether the targeted behavior is good or bad in the long term. Further, human reinforcement is delayed only by how long it takes the trainer to evaluate the targeted behavior and communicate the evaluation. Therefore, unlike MDP reward, human reinforcement is not sparse[1] — each reinforcement fully discriminates between approved and disapproved behavior — and it is only trivially delayed. This insight is the foundation of TAMER.

We note, though, that human reinforcement is, in general, fundamentally *flawed*. Humans make mistakes, often have low standards, get bored, and have many other imperfections, so their evaluations will likewise be imperfect.

The TAMER Approach

Following the insight above, TAMER dismisses the credit assignment problem inherent in reinforcement learning. It instead assumes human reinforcement to be fully informative about the quality of an action given the current state. TAMER uses established supervised learning techniques to model a hypothetical human reinforcement function, $H : S \times A \rightarrow \mathbb{R}$, treating the feedback value as a label for a state-action sample.[2] The TAMER framework is agnostic to the specific model and supervised learner used, leaving such decisions to the agent's designer, though we conjecture that certain model characteristics are desirable (2009).

To choose actions within some state s, a TAMER agent directly exploits the learned model $\hat{H}$ and its predictions of expected reinforcement. A greedy TAMER agent chooses actions by $a = argmax_a[\hat{H}(s,a)]$.

[1]However, human reinforcement can be sparsely delivered.

Previous TAMER Results and Conclusions

We previously (2009) implemented TAMER agents for two contrasting domains: Tetris and Mountain Car. They compared the learning curves of the TAMER agents with various autonomous agents (i.e., reinforcement learning agents). In both domains, the shaped agents strongly outperformed autonomous agents in early training sessions, quickly exhibiting qualitatively good behavior. As the number of training episodes increases, however, many of the autonomous agents surpass the performance of TAMER agents. This paper aims to combine TAMER's strong early learning with the often superior long-term learning of autonomous agents.

Simultaneous RL + TAMER

In the sections "The MDP Reward Signal" and "The Human Reinforcement Signal", we concluded that MDP reward is informationally poor yet flawless, whereas human reinforcement is rich in information yet flawed. This observation fits the aforementioned experimental results well. With a richer feedback signal, TAMER agents were able to learn much more quickly. But the signal was flawed and TAMER agents, in some cases, plateaued at lower performance levels than autonomous learning agents.

The TAMER framework does not use MDP reward. This characteristic can be a strength. It allows users to fully determine the goal behavior without defining and programming a reward function. However, it can also be a weakness. When the goal behavior is previously agreed upon and a reward function is available, a TAMER agent is ignoring valuable information in the reward signal — information which complements that found in the human reinforcement signal.

In this paper, we ask how an agent can effectively learn simultaneously from two feedback modalities — human reinforcement signals and MDP reward — in one fully integrated system. More specifically, we seek to use a frequently updated model of expected human reinforcement, $\hat{H}$, learned by a TAMER algorithm, to improve the learning of a reinforcement learning algorithm.

In previous work (Knox and Stone 2010), we examined a different learning scenario which begins with the human

[2]In domains with frequent time steps (an approximate rule for "frequent" is more than one time step per second), the reinforcement is weighted and then used as a label for multiple samples, each with one state-action pair from a small temporal window of recent time steps. We previously (2009) described this credit assignment method in detail.

training an agent using only the TAMER framework. Then, the trainer leaves, after which $\hat{H}$ is constant, and a reinforcement learning algorithm, SARSA(λ), takes over. For this paper, we will call this scenario consecutive learning. We previously tested and analyzed eight plausible methods for combining $\hat{H}$ with SARSA(λ) to improve learning. When pessimistically initializing Q, four of these eight methods were found to outperform SARSA(λ) alone in both cumulative reward and final performance level for $\hat{H}$s from two trainers of differing training ability (expert and average). These four methods also outperformed both of the corresponding TAMER-only agents, which greedily choose $argmax_a[\hat{H}]$ on each of the two unchanging $\hat{H}$ functions, in final performance level and outperformed the agent using the average $\hat{H}$ in cumulative reward. For this paper, the constraints of learning simultaneously makes one of the four techniques impractical. Two other techniques are quite similar, so we do not investigate the one that performed more poorly. In the following section, we describe the two remaining techniques adapted that are built upon in this paper for simultaneous TAMER+RL.

Simultaneously learning from human reinforcement and MDP reward presents certain challenges beyond those addressed in our previous work, in which the agent learned from the two signals in separate, consecutive learning sessions. The following subsections describe the general algorithmic contributions of TAMER+RL in the context of some of these challenges.

Combination techniques

Two of the methods that are successful for consecutive learning are adapted to perform simultaneous learning and investigated in this paper:

- $R'(s,a) = R(s,a) + (weight * \hat{H}(s,a))$. Here, the MDP reward is replaced with the sum of itself and the weighted prediction of human reinforcement. Augmenting the MDP reward signal in this fashion is called shaping rewards in the RL literature (Dorigo and Colombetti 1994; Mataric 1994), so we will call it the shaping rewards method.
- $a = argmax_a[Q(s,a) + (weight * \hat{H}(s,a))]$. The weighted prediction of human reinforcement is added to $Q(s,a)$ only during action selection. This technique biases action selection towards those actions that have higher expected human reinforcement. We will call it the action biasing method.

We discuss the $weight$ used in each technique in the section below entitled "Determining the immediate influence of $\hat{H}$".

Reward shaping in terms of the Bellman equation The goal of SARSA(λ) and many other reinforcement learning algorithms is to learn the expected return, which for a state action pair is $Q(s_0,a_0) = \sum_{t=0}^{t} E[\gamma^t R(s_t,a_t)]$. When shaping rewards by $\hat{H}$, this equation changes to $Q(s_0,a_0) = \sum_{t=0}^{t} E[\gamma^t (R(s_t,a_t)+\hat{H}(s_t,a_t))]$, which is equivalent to $Q(s_0,a_0) = \sum_{t=0}^{t} E[\gamma^t (R(s_t,a_t)] + \sum_{t=0}^{t} E[\gamma^t \hat{H}(s_t,a_t)]$ (we will leave the weight out of the equations to increase readability). Therefore, learning from shaping rewards can be seen as learning a different Q function for each signal: $Q = Q_R + Q_{\hat{H}}$.

Action biasing in terms of the Bellman equation For shaping rewards, the discount factors of the returns for the two signals are equivalent. If the discount factor of $Q_{\hat{H}}$, $\gamma_{\hat{H}}$, is zero, then $Q_{\hat{H}}(s_0,a_0) = \sum_{t=0}^{t} E[\gamma_{\hat{H}}^t \hat{H}(s_t,a_t)] = \hat{H}(s_0,a_0)]$. Therefore, $Q = Q_R + \hat{H}$. If $\hat{H}$ is known to the agent (as it is for a TAMER agent), then only Q_R needs to be learned and no part of its error will come from the known $\hat{H}$ component. Thus, greedy action selection for $\gamma_{\hat{H}} = 0$ yields $a = argmax_a[Q(s,a) + \hat{H}(s,a)]$, which is our action biasing method.

Since MDPs typically have discount factors at or near one, shaping rewards and action biasing can be seen as two extremes along a spectrum of possible $\gamma_{\hat{H}}$ values.

Determining the immediate influence of $\hat{H}$

One constraint we adopt is that the combination technique must not change the set of optimal policies in the limit, long after the human quits training. For consecutive learning, annealing the magnitude of $\hat{H}$'s influence was sufficient to achieve this constraint. However, simultaneous learning allows human trainers to insert themselves at any point of the learning process. Consequently, the influence of $\hat{H}$, which manifests as the weight in both techniques, should increase and decrease flexibly by some principled method that relies on the history of human reinforcement.

The method we developed for determining the influence of $\hat{H}$ is an adaptation of the eligibility traces often used in reinforcement learning (Sutton and Barto 1998). We will refer to it as the $\hat{H}$-eligibility module. The general idea of this $\hat{H}$-eligibility module is that we maintain a set of eligibility traces which, for any state, roughly measure the recent frequency of human feedback in that state and similar states. That measure, multiplied by a predefined constant, is used as the $weight$ term in the action biasing and shaping rewards methods. The implementation follows.

Let $\overrightarrow{e}$ be the vector of eligibility traces used to weight the influence of $\hat{H}$. Let $\overrightarrow{f}$ be the feature vector (of features that generalize across states) extracted from the current state-action pair, and let $\overrightarrow{f_n}$ be $\overrightarrow{f}$ normalized such that each element of $\overrightarrow{f_n}$ exists within the range $[0,1]$.

Our $\hat{H}$-eligibility module is designed to make $weight$ a function of $\overrightarrow{e}$, $\overrightarrow{f_n}$, and constant scaling factor c_s with range $[0,c_s]$. The guiding constraint for our design is that if $\overrightarrow{e}$ equals the vector $\overrightarrow{1}$, then the normalized dot product of $\overrightarrow{e}$ and any $\overrightarrow{f_n}$, denoted $n(\overrightarrow{e} \cdot \overrightarrow{f_n})$, should equal 1 (since it weights the influence of $\hat{H}$). To achieve this, we make $n(\overrightarrow{e} \cdot \overrightarrow{f_n}) = (\overrightarrow{e} \cdot (\overrightarrow{f_n} \,/\, \| \overrightarrow{f_n} \|_1) = (\overrightarrow{e} \cdot \overrightarrow{f_n}) \,/\, (\| \overrightarrow{f_n} \|_1) = weight \,/\, c_s$. Thus, at any time step with normalized features $\overrightarrow{f_n}$, the influence of $\hat{H}$ is calculated as

$weight = c_s(\overrightarrow{e} \cdot \overrightarrow{f})/(\| \overrightarrow{f_n} \|_1)$.

The eligibility trace is updated with $\overrightarrow{f_n}$ when human reinforcement is received. Using replacing traces, the update with reinforcement is $e_i := max(decayFactor * e_i,\ \ f_{n,i})$, where e_i and $f_{n,i}$ are the i^{th} elements of $\overrightarrow{e}$ and $\overrightarrow{f_n}$, respectively. At time steps when no human reinforcement is received, $\overrightarrow{e} := decayFactor * \overrightarrow{e}$.

Experiments

In this section we present our tests of each combination technique. As a testbed for TAMER+RL, we use an MDP called the Mountain Car domain, adapted from the version within RL-Library(Tanner and White 2009). Mountain Car consists of an agent-controlled car which starts between two hills and must go back and forth to gain enough momentum to reach the top of one of the hills. There are two continuous state variables, position and velocity, and three possible actions: $+c$ or $-c$ acceleration and no acceleration. At every transition to a non-terminal state, the agent receives -1 MDP reward, and the discount factor is one.

For our RL algorithm, we use SARSA(λ) with gradient descent over a linear model of Gaussian RBFs, which is known to perform well in Mountain Car (though more effective algorithms do exist). For TAMER+RL, we pessimistically initialize the Q function to output approximately -120 for all state-action pairs, which we found to be beneficial in our previous work on consecutive learning. We note that our goal for this paper is not to study the interaction between TAMER and different RL algorithms but rather to establish that they can be integrated effectively into one learning agent and to study the two most promising known ways for doing transfer from $\hat{H}$ to an RL algorithm. We have no reason to expect qualitatively different results with other value-function based RL algorithms, but we will investigate possible differences in future work.

To model the human reinforcement function H via TAMER during simultaneous learning, we chose k-nearest neighbors, where k is $\sqrt{d}$ for d samples, to model $H(\cdot, a)$ for each action (that is, three separate models). We set Mountain Car to run at seven time steps per second and used the credit assignment technique described in past work (2009) to account for delays in human reinforcement. The constant scale factors, described previously in the section "Determining the immediate influence of $\hat{H}$", for action biasing and reward shaping are 100.0 and 10.0, respectively. These were the largest factors shown to be successful in consecutive learning; greater size gives more influence to $\hat{H}$, making the TAMER+RL agent more responsive to the human trainer. The decay factor from the same section was 0.9998, which decreases the weight of $\hat{H}$'s influence by a factor of approximately 0.1 over 100 episodes.

In this paper, we seek to demonstrate simultaneous TAMER+RL agents that can benefit from human reinforcement from a trainer with domain expertise. More specifically, the agents should be able to benefit from training at *any* point along the learning curve. In our experiments, the agents learn for 500 episodes. Two trainers, each a coauthor of this paper with domain expertise, start training after 0, 20, and 60 episodes and stop 10 episodes later.[3] That is, SARSA(λ) learns without any TAMER-based changes until the start time. We recorded 10 sessions for each start time, combination technique, and trainer combination (trainer $T1$ or $T2$), yielding 120 10-episode sessions. Every agent in each start time group experienced the same sequence of starting states at each episode. For comparison, we also ran optimistically and pessimistically initialized SARSA(λ) agents for 50 runs of 500 episodes. To achieve a range of agent performances, each of the 50 runs had different start sequences.[4]

Results and Discussion

In this section, we summarize and discuss our experimental results. The results under both trainers were qualitatively identical, so we talk about their combined results as one. We consider both cumulative reward over all 500 episodes and final performance, as seen over the last 100 episodes, to each be measures of a technique's success. Figure 1 shows our results with respect to each of these two measures. In both, action biasing performs significantly better than either optimistically or pessimistically initialized SARSA(λ)at all start times (contrary to the appearance of the cumulative bar graph in Figure 1, the confidence interval of $T2$'s "AB, S60" training does not overlap with that of optimistic SARSA(λ)). Shaping rewards performs worse, though not always significantly so.[5]

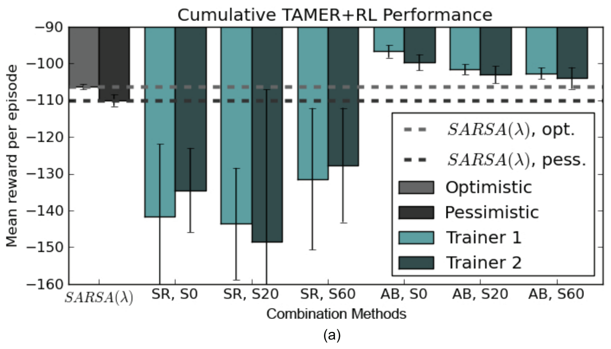
Figure 2 shows learning curves for the experimental conditions. The learning curves show that action biasing quickly improves the performance of the TAMER+RL agent upon the start of training. Shaping rewards also improves performance, but then within 50 episodes after training ends, the performance of shaping rewards plummets and does not recover for approximately 100 episodes.

The success of action biasing and the failure of shaping rewards have interesting implications. Specifically, the way that shaping rewards fails — some time after training ends — is telling. The $Q_{\hat{H}}$ that is learned during and shortly after training improves or at least does not hurt the policy, but 50 episodes later, the effect of $Q_{\hat{H}}$ is catastrophic. What occurs between those two periods is temporal difference bootstrapping on $\hat{H}$s output. Thus, our data suggests that human re-

[3]These start times are representative of three qualitative points along the curve: before learning, a mediocre policy, and a good but still imperfect policy.

[4]Because of the difference in starting states between trained agents and SARSA(λ)-only agents, the TAMER+RL agents end up performing better before training (in the 20 and 60 start time groups) than the mean of the similarly pessimistic SARSA(λ) agents. For the 20 group and 60 group, this difference in total pre-training reward impacts the mean reward over 500 episodes by -1.4 and -2.5, respectively. Adding these values to the respective means of TAMER+RL groups changes the qualitative results in only one respect: the confidence intervals of optimistic SARSA(λ) and the 60 group of action biasing begin to overlap a bit.

[5]One training session for shaping rewards was removed because the trainer could not get the agent to the goal after 2500 time steps and the session was stopped.



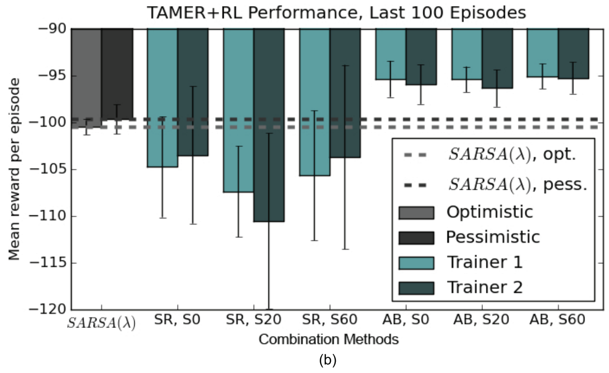


Figure 1: The mean reward received per episode over 20 training sessions of 500 episodes for both techniques — shaping rewards (SR) and action biasing (AB) — starting training at three different times. The top graph shows the mean over all 500 episodes, and the bottom graph shows the mean over the last 100 episodes, allowing us to assess final performance of each condition. The performance of SARSA(λ) under optimistic and pessimistic initialization is shown for comparison. Error bars show 95% confidence intervals, assuming that mean reward over a run is normally distributed.

inforcement should not undergo the same level of bootstrapping as MDP reward. That is, the discount factors used to define Q_R and $Q_{\hat{H}}$ should differ. In a sense, the corresponding performances of our two techniques further vindicate a large assumption of TAMER — that an agent being shaped should choose based on immediate expected reinforcement, not its return. However, it is possible that a small discount factor could aid TAMER-only learning.

For each training session, we also calculated the mean TAMER-only performance given by choosing action $argmax_a \hat{H}(s,a)$, as an exploiting TAMER agent would. Some of the TAMER-only agents performed well most of the time but looped infinitely from certain start states, never reaching an absorbing state at the goal. Considering this, the TAMER+RL agents clearly outperform the TAMER-only agents in the mean. More interesting, though, is a comparison of TAMER+RL techniques under the few training sessions that produce $\hat{H}$s that yield infinitely looping TAMER-only agents with those TAMER+RL techniques under the sessions which produce TAMER-only agents which can reach the goal. Table 1 shows that these two categories did not differ noticeably — differences from the technique used or the start time had much larger effects. Thus, through the interaction of SARSA(λ) with flawed $\hat{H}$s, the TAMER+RL algorithm was able to improve from the effective aspects of $\hat{H}$ while not being noticeably hurt by the aspects that suggest an infinitely bad policy.

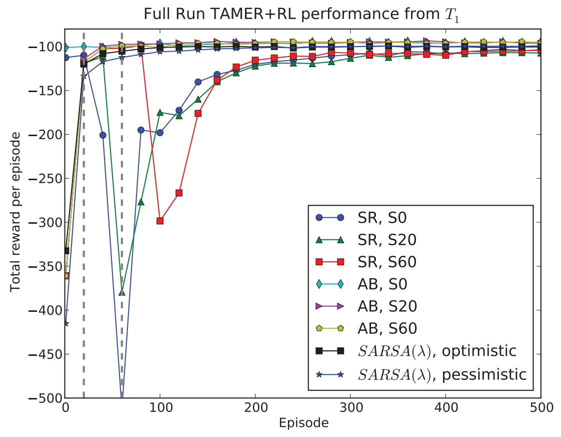


Figure 2: Learning curves (smoothed) of the mean reward per episode over 20 training sessions by trainer $T1$ for both techniques — shaping rewards (SR) and action biasing (AB) — starting training at three different times. Vertical dashed lines show the 20th and 60th episode marks.

Related work

In this section, we situate our work within previous research on transferring knowledge to a reinforcement learning agent. For a review of learning from human reinforcement without MDP reward, see the main TAMER paper shortcitekcap09-knox. For a full review of knowledge transfer to an RL agent, including methods unrelated to this paper, see our previous work (2010).

Transferring from a human

In the only other example of real-valued human reinforcement being incorporated into reinforcement learning, Thomaz & Breazeal (2006) interfaced a human trainer with a table-based Q-learning agent in a virtual kitchen environment. Their agent seeks to maximize its discounted total reward, which for any time step is the sum of human reinforcement and environmental reward. Their approach is a form of shaping rewards, differing in that Thomaz & Breazeal directly apply the human reinforcement value to the current reward (instead of modeling reinforcement and using the output of the model as supplemental reward).

Imitation learning, or programming by design, has also been used to improve reinforcement learning (Price and

Table 1: Mean TAMER+RL reward (rounded) by corresponding TAMER-only policy. Under a TAMER-only policy, goal-reaching $\hat{H}$s eventually reach an absorbing state at the goal from every tested start state, whereas infinitely looping $\hat{H}$s will never reach the goal when starting at certain states.

TAMER-only	Shaping Rewards			Action Biasing		
policy	0	20	60	0	20	60
Goal-reaching	-138	-141	-131	-98	-102	-104
Inf. Looping	-137	-162	-135	-99	-103	-103

Boutilier 2003). Another agent provides demonstrations from its policy while the agent of concern observes and learns, a technique that is related to action biasing. An advantage, though, of action biasing over demonstration from a policy is that action biasing with $\hat{H}$ can gently push the behavior of the RL agent towards the policy of a TAMER-only agent, whereas pure demonstration is all or nothing — either the demonstrator or the learning agent chooses the action. Natural language advice has also been applied to reinforcement learning (Kuhlmann et al. 2004).

Transferring from other sources

To the best of our knowledge, transfer learning for reinforcement learning typically focuses on how to use information learned in a source task to improve learning in a different target task. Our type of transfer differs: the task stays constant and we transfer from one type of task knowledge (an $\hat{H}$ function) to a different type (a Q function). Additionally, in simultaneous TAMER+RL, knowledge is transferred even before it has been fully captured.

Other forms of transfer can occur within the same task. *Shaping rewards* (Dorigo and Colombetti 1994; Mataric 1994) is changing the output of the reward function to learn the same task, as we did in Method 1. The difference of the shaped reward function and the original one can be seen as the output of a shaping function ($\hat{H}$ in our case). With a few assumptions, Ng et al. (1999) prove that such a function f, if static, must be defined as $f = \phi(s') - \phi(s)$, where $\phi : S \rightarrow \mathbb{R}$, to guarantee that shaping the reward function will not change the set of optimal policies. Our shaping rewards method anneals the output of its shaping function, avoiding Ng et al.'s theoretical constraint.

Conclusion

In previous work (Knox and Stone 2010), we started with eight plausible techniques for creating a TAMER+RL algorithm that learns form both human reinforcement and MDP reward. Four were successful in consecutive learning. Of those four, two were inappropriate for simultaneous learning. This paper examines those two remaining and finds that one of them, action biasing, clearly improves on the solo performance of the reinforcement learning algorithm. We hypothesize similar results for other domains and other incremental, temporal difference learning algorithms, and we will be interested to see the results as we expand the techniques of this paper to other tasks and algorithms.

This paper presents two novel algorithms, the first to create a predictive model of human reinforcement while simultaneously using that model to guide a reinforcement learning algorithm. Our experiments show that the combination method within one of the algorithms, action biasing, significantly improves performance of the tested reinforcement learning algorithm, SARSA(λ), in the testbed domain.

One aim of this paper is to create practical tools with which designers of reinforcement learning agents can capture human knowledge to increase the performance of their agents. With that in mind, both algorithms follow three constraints: the techniques are independent of the representation of Q and $\hat{H}$, the parameters of the RL algorithm do not change from those found to be optimal for SARSA(λ) during parameter tuning, and when the human trainer stops giving feedback, the influence of $\hat{H}$ diminishes with time.

We will continue to push forward on creating learning agents that can learn from human reinforcement, testing our findings within other domains, and examining the combination of TAMER with other reinforcement learning algorithms.

Acknowledgments

The authors would like to to thank Brian Tanner for his extensive support of RL-Glue and RL-Library. This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (IIS-0917122), ONR (N00014-09-1-0658), and the Federal Highway Administration (DTFH61-07-H-00030).

References

Dorigo, M., and Colombetti, M. 1994. Robot shaping: Developing situated agents through learning. *Artificial Intelligence* 70(2):321–370.

Knox, W. B., and Stone, P. 2009. Interactively shaping agents via human reinforcement: The tamer framework. In *The Fifth International Conference on Knowledge Capture*.

Knox, W. B., and Stone, P. 2010. Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *Proceedings of The Ninth Annual International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Kuhlmann, G.; Stone, P.; Mooney, R.; and Shavlik, J. 2004. Guiding a reinforcement learner with natural language advice: Initial results in RoboCup soccer. In *The AAAI-2004 Workshop on Supervisory Control of Learning and Adaptive Systems*.

Mataric, M. 1994. Reward functions for accelerated learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, volume 189. Citeseer.

Ng, A.; Harada, D.; and Russell, S. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. *ICML*.

Price, B., and Boutilier, C. 2003. Accelerating reinforcement learning through implicit imitation. *Journal of Artificial Intelligence Research* 19:569–629.

Sutton, R., and Barto, A. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

Tanner, B., and White, A. 2009. RL-Glue : Language-independent software for reinforcement-learning experiments. *Journal of Machine Learning Research* 10:2133–2136.

Thomaz, A., and Breazeal, C. 2006. Reinforcement Learning with Human Teachers: Evidence of Feedback and Guidance with Implications for Learning Performance. *AAAI*.