One solution to interactive shaping

Reward from a human trainer:

– Trainer has long-term impact in mind.
  – We can consider reward a full judgment of desirability of behavior.

– Trainer can reward with small delay.
Teaching an Agent Manually via Evaluative Reinforcement (TAMER)

Learn a model of human reward

\[ H : S \times A \rightarrow \mathbb{R} \]

Directly exploit the model to determine action

If greedy:

\[ \text{action} = \arg\max_a \hat{H}(s, a) \]

ICDL 2008 and K-CAP 2009
Teaching an Agent Manually via Evaluative Reinforcement (TAMER)

\[ H : S \times A \rightarrow \mathbb{R} \]

I.e., TAMER reduces an apparent reinforcement learning problem to a supervised learning problem by setting \( \gamma = 0 \).
Teaching an Agent Manually via Evaluative Reinforcement (TAMER)

\[
H : S \times A \rightarrow \mathbb{R}
\]

\[
\hat{H} : S \times A \rightarrow \mathbb{R}
\]
TAMER in action: Tetris

Before training:

After training:

Environment courtesy of RL-Library and RL-Glue
Handling reward delay

Forward view

Backward view

\( f_{\text{delay}}(\text{time}) \)

Event

Feedback?

Event?

Feedback

\( f_{\text{delay}}(\text{time}) \)

-0.8

0

-0.2

step start

step end

time (relative to feedback)

time of feedback

Probability that a human reward signal targets a time step (relative to feedback)
TAMER success on other domains

Mountain Car
(Knox and Stone, 2009)

3 vs 2 Keepaway
(Sridharan, 2011)

Balancing Cart Pole
(Knox and Stone, 2012)

Interactive robot navigation
(Knox, Stone, and Breazeal, 2012)

Environments courtesy of RL-Library and RL-Glue (adapted)
Combination Techniques

1. $R'(s, a) = R(s, a) + (\beta \times \hat{H}(s, a))$.

2. $\vec{f} = \vec{f}.\text{append}(\hat{H}(s, a))$.

3. Initially train $Q(s, a)$ to approximate $(\beta \times \hat{H}(s, a))$.

4. $Q'(s, a) = Q(s, a) + (\beta \times \hat{H}(s, a))$.

5. $A' = A \cup \text{argmax}_a[\hat{H}(s, a)]$.

6. $Q'(s, a) = Q(s, a) + (\beta \times \hat{H}(s, a))$ only during action selection.

7. $P(a = \text{argmax}_a[\hat{H}(s, a)]) = \min(\beta, 1)$. Otherwise use base RL agent’s action selection mechanism.

8. $R'(s_t, a) = R(s, a) + (\beta \times (\phi(s_t) - \phi(s_{t-1})))$, where $\phi(s) = \max_a H(s, a)$.
Combination Techniques

1. \( R'(s, a) = R(s, a) + (\beta \times \hat{H}(s, a)) \).
2. \( \vec{f} \leftarrow \vec{f}.\text{append}(\hat{H}(s, a)) \).
3. Initially train \( Q(s, a) \) to approximate \( (\beta \times \hat{H}(s, a)) \).
4. \( Q'(s, a) = Q(s, a) + (\beta \times \hat{H}(s, a)) \).
5. \( A' = A \cup \text{argmax}_a[\hat{H}(s, a)] \).
6. \( Q'(s, a) = Q(s, a) + (\beta \times \hat{H}(s, a)) \) only during action selection.
7. \( P(a = \text{argmax}_a[\hat{H}(s, a)]) = \min(\beta, 1) \). Otherwise use base RL agent’s action selection mechanism.
8. \( R'(s_t, a) = R(s, a) + (\beta \times (\phi(s_t) - \phi(s_{t-1}))), \) where \( \phi(s) = \max_a H(s, a) \).
Combination Techniques

1. \( R'(s, a) = R(s, a) + (\beta \times \hat{H}(s, a)) \).
2. \( \vec{f}' = \vec{f}.\text{append}(\hat{H}(s, a)) \).
3. Initially train \( Q(s, a) \) to approximate \( (\beta \times \hat{H}(s, a)) \).
4. \( Q'(s, a) = Q(s, a) + (\beta \times \hat{H}(s, a)) \).
5. \( A' = \{ a \mid \arg\max_a \hat{H}(s, a) \} \).
6. \( Q'(s, a) = Q(s, a) + (\beta \times \hat{H}(s, a)) \) only during action selection.
7. \( P(a = \arg\max_a [\hat{H}(s, a)]) = \min(\beta, 1) \). Otherwise use base RL agent’s action selection mechanism.
8. \( R'(s_t, a) = R(s, a) + (\beta \times (\phi(s_t) - \phi(s_{t-1}))) \), where \( \phi(s) = \max_a H(s, a) \).
### Defining success

#### Outperforming:

<table>
<thead>
<tr>
<th>Metrics</th>
<th>TAMER-only</th>
<th>RL-only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cumulative MDP reward</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Final performance</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

On each tested $\hat{H}$
Defining success

Outperforming:

<table>
<thead>
<tr>
<th>TAMER-only</th>
<th>RL-only</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

On the metrics:
- cumulative MDP reward
- final performance

On each tested $\hat{H}$

Domains:

Sarsa($\lambda$) here
Complete successes

action biasing

\[ Q'(s, a) = Q(s, a) + (\beta \times \hat{H}(s, a)) \text{ only during action selection.} \]

and

control sharing

\[ P(a = \arg\max_a[\hat{H}(s, a)]) = \min(\beta, 1). \text{ Otherwise use base RL agent’s action selection mechanism.} \]

Outperforming:

Manipulating action selection

<table>
<thead>
<tr>
<th></th>
<th>TAMER-only</th>
<th>RL-only</th>
</tr>
</thead>
<tbody>
<tr>
<td>reward</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>final performance</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Outline

0  Background and TAMER+RL problem
1  Sequential TAMER+RL
2  Simultaneous TAMER+RL
Simultaneous TAMER+RL

Number of trials or attempts at learning

- Slow beginning
- Steep acceleration
- Plateau
Determining when and where human influences

**action biasing**

\[ Q'(s, a) = Q(s, a) + (\beta \times \hat{H}(s, a)) \text{ only during action selection.} \]

**control sharing**

\[ P(a = \arg\max_a [\hat{H}(s, a)]) = \min(\beta, 1). \text{ Otherwise use base RL agent’s action selection mechanism.} \]

**Sequential** – reduce influence of by annealing \( \beta \) as learning progresses

**Simultaneous** – influence of (as regulated by \( \beta \)) should

1. increase after training in nearby state-action space, and
2. decrease in the absence of training.
Determining when and where human influences occur.

State-action features (last step)

Traces to remember where training occurred recently

Influence level calculated from traces and features

State-action features (last step)

Traces to remember where training occurred recently

Mountain Car 1.3
E/S/T/R: 1/68/68/-1
Determining when and where human influences
Determining when and where human influences

\( \hat{H} \) Eligibility Module – qualitative characteristics

1. Scales up influence in areas of recent training
2. Slowly reduces influence in the absence of training

\[
\beta := c \mathbf{e} \cdot \left( \frac{\mathbf{f}_n}{\| \mathbf{f}_n \|_1} \right)
\]
Experiments

Mountain Car and Balancing Cart-Pole

Simultaneous TAMER+RL on Mountain Car

Simultaneous TAMER+RL on Cart Pole

Mean reward (time to goal) per episode

Mean reward (time upright) per episode

Episodes before training

Episodes before training
Experiments

Early-run simultaneous TAMER+RL on Cart Pole
Related work on learning from MDP reward and human input

• alternating stages of autonomous action and human critique (Judah et. al, 2010)
• learning from demonstration (Smart and Kaelbling, 2000; Taylor et al., 2011)
• learning options from demonstration (Subramanian et al., 2011)
• feature selection from demonstration (Cobo et al. 2011, 2012)
Human reward can be combined with MDP reward to improve upon learning from either alone.

**Manipulating action selection** – highest, most consistent gains and robust to changes in weights

**Mixing human and MDP reward in a single value function** – sometimes helps, brittle to weight values

Can learn simultaneously through an adaptation of eligibility traces