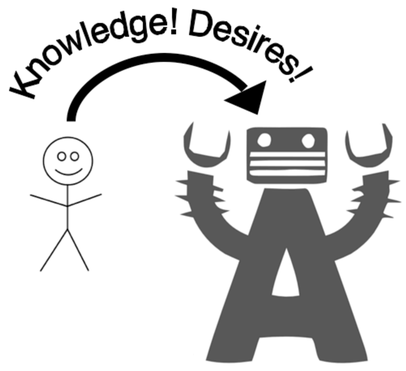
Combining Manual Feedback with Subsequent MDP Reward Signals for Reinforcement Learning

W. Bradley Knox and Peter Stone

Learning Agents Research Group
The University of Texas at Austin

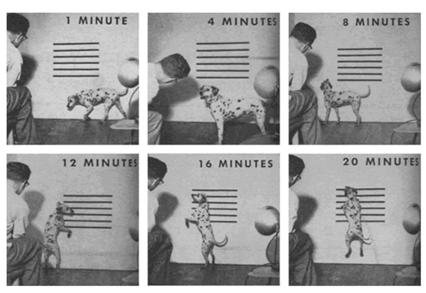
Human-teachable agents



- how to teach?
- how to learn from teaching and reinforcement learning?

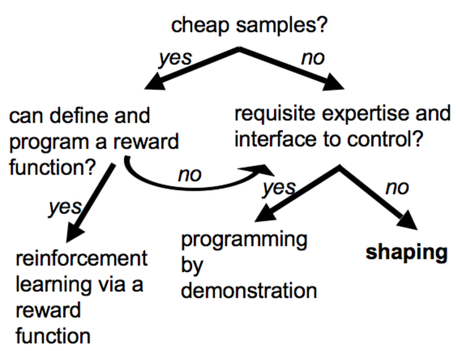
Interactive Shaping

Human trainer transfers task knowledge to an agent through signals of positive and negative reinforcement



LOOK magazine, 1952

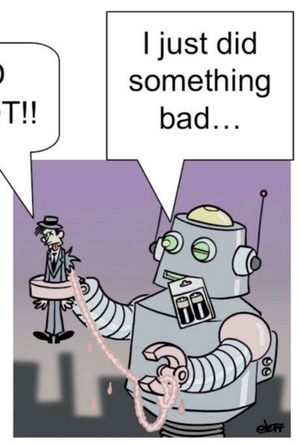
If limited to one form of knowledge transfer...



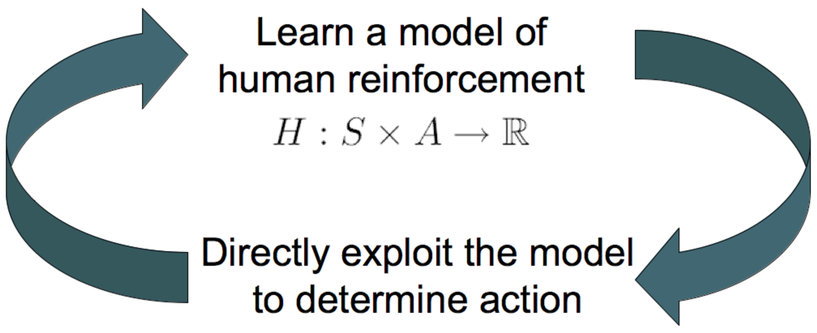
Human reinforcement

Two characteristics:
- trainer has long-term impact in mind
- small delay

Therefore, credit assignment problem is largely removed!



Teaching an Agent Manually via Evaluative Reinforcement (TAMER)



If greedy: $action = argmax_a \hat{H}(s,a)$

Knox and Stone, K-CAP 2009

TAMER Results

Compared to autonomous algorithms learning from predefined reward functions, in both test domains:

TAMER learns more quickly

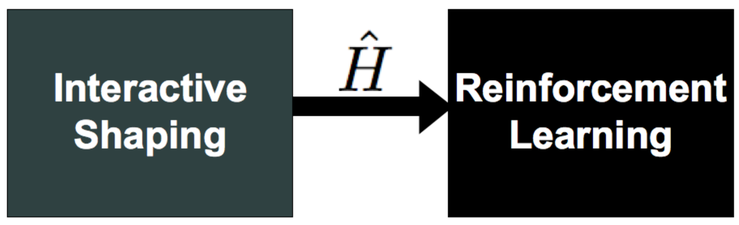
but

autonomous learners eventually equal or surpass TAMER

TAMER+RL (the AAMAS 2010 paper)

Human reinforcement: rich but flawed
MDP Reward: sparse but flawless
How to use the two signals together?

Or, more narrowly, **how can a predictive model of human reinforcement be used to aid reinforcement learning (RL)?**



Restrictions on combination techniques

1. Independent of model representations of Q and $\hat{H}$.
2. Influence of $\hat{H}$ must recede with time or repeated visits to same or similar states.
3. Parameters of RL agent stay tuned to RL-only learning.

Eight combination techniques

1. $R'(s,a) = R(s,a) + (weight * \hat{H}(s,a))$.
2. $\overrightarrow{f'} = \overrightarrow{f}.append(\hat{H}(s,a))$.
3. $Initially\ train\ Q(s,a)\ to\ approximate\ (constant * \hat{H}(s,a))$.
4. $Q'(s,a) = Q(s,a) + constant * \hat{H}(s,a)$.
5. $A' = A \cup argmax_a[\hat{H}(s,a)]$.
6. $a = argmax_a[Q(s,a) + weight * \hat{H}(s,a)]$.
7. $P(a = argmax_a[\hat{H}(s,a)]) = p.\ Otherwise\ original\ RL\ agent's\ action\ selection\ mechanism\ is\ used.$
8. $R'(s_t,a) = R(s,a) + constant * (\phi(s_t) - \phi(s_{t-1}),\ where\ \phi(s) = max_a H(s,a)$.

Experiments

- domain: Mountain Car
- RL algorithm: Sarsa(λ)
  - features: a grid of 2D Gaussian RBFs over state; one grid for each action
  - representation of Q: linear model
  - initialization of Q: both opt. and pess.
  - updates: gradient descent
- 30 runs of 500 episodes

Two predictive models used (from among 19 trainers):

$\hat{H}_1$: middling performance (9th)
$\hat{H}_2$: best performance

Definition of Success

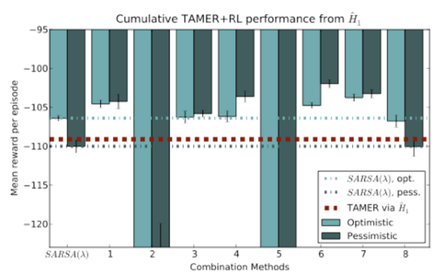
Outperforming:

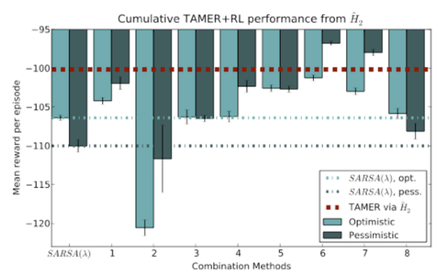
On the metrics:

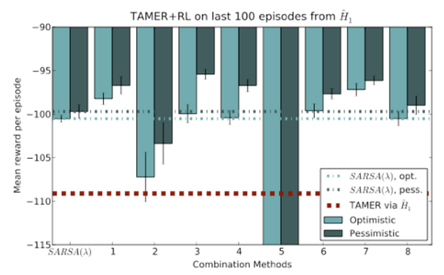
	TAMER-only	RL-only
cumulative reward	?	?
final performance	?	?

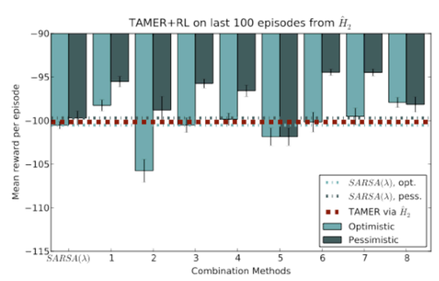
On both $\hat{H}_1$ and $\hat{H}_2$

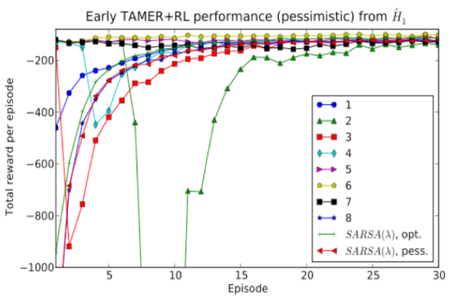
Results

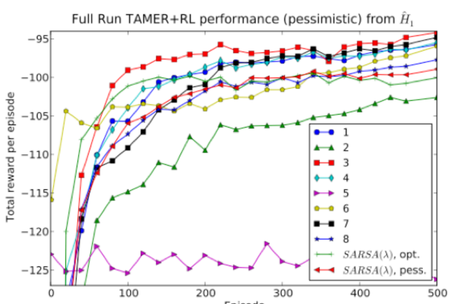












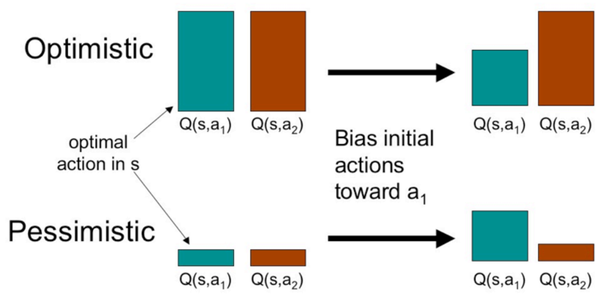
Success?

Almost: $R'(s,a) = R(s,a) + (weight * \hat{H}(s,a))$
and
$Q'(s,a) = Q(s,a) + constant * \hat{H}(s,a)$

Yes!: $a = argmax_a[Q(s,a) + weight * \hat{H}(s,a)]$.
and
$P(a = argmax_a[\hat{H}(s,a)]) = p.\ Otherwise\ original\ RL\ agent's\ action\ selection\ mechanism\ is\ used.$

Lessons

1. Pessimistic initialization works, optimistic does not



2. Biasing action selection (6th and 7th techniques) was most effective
   - better than shaping rewards (2nd technique)