What’s in a Reward? Teaching Agents and Robots with Non-Expert Reinforcement

Matthew E. Taylor

Bei Peng
Michael L. Littman, James MacGlashan
Robert Loftin, David L. Roberts

irll.eecs.wsu.edu
Interactive Machine Learning: From Classifiers to Robotics
Brad Hayes, Ece Kamar, and Matthew E. Taylor

1. BROAD OVERVIEW OF IML
2. USING THE CROWD FOR SUPERVISED LEARNING TASKS
3. TEACHING ROBOTS TO PERFORM SEQUENTIAL TASKS
4. TRAINING AND LEARNING IN SEQUENTIAL TASKS
5. EXPERIMENTAL DESIGN & OTHER PRACTICAL CONSIDERATIONS

Suggestions? Write to: taylorm@eecs.wsu.edu

InteractiveML.net
Reinforcement Learning

- Sequential decision making problem
  \[ M = (S, A, T, R) \]
- Markov Decision Process (MDP)
  \[ \pi : S \rightarrow A \]
  \[ Q^\pi(s, a) \]
Reinforcement Learning

- Sequential decision making problem
- Markov Decision Process (MDP): 
  \[ M = (S, A, T, R) \]
  \[ \pi : S \rightarrow A \]
  \[ Q^*(s, a) \]
Speed Up Learning:
• Transfer learning
• Learning from demonstrations
• Intelligent tutoring
Learning from (human) feedback
Enable *non-expert* users to teach robots!

How can we best design an agent to learn?
Key insight: trainer evaluates behavior using model of its long-term quality

Learn a model of human reinforcement

$$H : S \times A \rightarrow \mathbb{R}$$

Directly exploit the model to determine action
Also, can combine with MDP's reward

[Knox & Stone, 2008-2015]
TAMER Learning Tetris

Initial Training

After 2 games of Training
Outline

Motivation
Learn from Rewards: SABL
Implicit Agent-Trainer Communication: LAMBDAS
FYI: Embodiment, Curricula, etc.
Example: Dog Training

Teach dog to sit & shake

Policy

“Sit” →

“Shake” →

Mapping from observations to actions

Feedback: {Bad Dog, Good Boy}

History of Evidence

Feedback history $h$


... 

Really make sense to assign numeric rewards to these?

---

Bayesian Framework

Trainer desires policy $\lambda^*$
$h_t$ is the training history at time $t$
Find MAP hypothesis of $\lambda^*$:

$$\arg\max_{\lambda} p(\lambda^* = \lambda | h_t) = \arg\max_{\lambda} p(h_t | \lambda^* = \lambda) p(\lambda^* = \lambda)$$

Prior distribution over policies
Model of training process

Strategy-Aware Bayesian Learning (SABL)

Assuming trainer feedback is given according to a probabilistic model (with known $\mu^+, \mu^-$ and $\epsilon$)

- action was correct, with error probability $\epsilon$
- withhold or give explicit feedback, with probability $\mu^+$ and $\mu^-$

Compute a maximum likelihood estimate of the target policy $\lambda$, given a training history $h$:

$$
\lambda^* = \arg\max_{\lambda} Pr[h|\lambda, \mu^+, \mu^-, \epsilon]
$$

Strategy-Aware Bayesian Learning (SABL)

To a strategy-aware learner, the lack of feedback can be as informative as explicit feedback.

No feedback?

That is not what I want, try something else!

Keep going and you will get reward eventually!

Infer Neutral

Try to learn what no-reward ($\mu^+ & \mu^-$) means.
Don’t assume they’re balanced.

Many trainers don’t use punishment.
Neutral feedback = punishment.

Some don’t use reward.
Neutral feedback = reward.

Train simulated dog to protect the field
Recruited Turkers and dog-training enthusiasts
How Humans Reward

Explicitly reward good behavior? R+
Explicitly punish bad behavior? P+
Stay consistent over time?

<table>
<thead>
<tr>
<th>begin</th>
<th>end</th>
<th>R+/P+</th>
<th>R+/P−</th>
<th>R−/P+</th>
<th>R−/P−</th>
</tr>
</thead>
<tbody>
<tr>
<td>R+/P+</td>
<td></td>
<td>65</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>R+/P−</td>
<td></td>
<td>10</td>
<td>52</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>R−/P+</td>
<td></td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>R−/P−</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

P+   | P−
---|---
R+  | 93 125
R−  | 6   3

SABL Helps *(Why else would I be talking about this?)*

SABL outperforms classical RL approaches

Comparison of learning rates in the first user study

- **SABL**
- **M-0**
- **M+0**

(Similar to TAMER)
(Similar to COBOT)

SABL Works: Can Recognize Strategies

Recognize strategies = do better than classic RL approaches

Can quickly learn from human rewards
Categorical, not numerical
Contextual bandit

Motivation
Learn from Rewards: SABL
Implicit Agent-Trainer Communication: LAMBDAS
FYI, FWIW: Embodiment, Curricula, etc.
Language Learning with Reinforcement and Punishment

An example five-room layout

Users Can Train SABL with Language

I have high confidence in my action choice.

Minimize task failures

Minimize total time

Minimize # actions

Make the best use of explicit feedback

Learning Agents Modeling Belief-Dependent Action Speeds (LAMBDAS)

- The agent’s confidence level:

\[
H = - \sum_{a \in A} Pr (a = a^* | s, F) \ln (Pr (a = a^* | s, F))
\]

- The probability distribution over optimal action:

\[
Pr (a = a^* | s, F) = \sum_{r \in R} \pi (s, a | r) Pr (r | F)
\]

\(s\): current state \quad \(a\): action taken \quad \(F\): feedback history \quad \(r\): intended reward function

Learning Agents Modeling Belief-Dependent Action Speeds (LAMBDAS)

- Map each entropy to the agent’s action execution speed

Linear model: \[ T = \theta_f + (\theta_s - \theta_f) \times H \]

Sigmoidal model: \[ T = \theta_f + \frac{\theta_s - \theta_f}{1 + e^{-10(H - \theta_f)}} \]

Threshold model: \[ T = \begin{cases} \theta_s & H > 0.1 \\ \theta_f & \text{otherwise} \end{cases} \]

\( \theta_f \): the fastest speed with highest confidence

\( \theta_s \): the slowest speed with lowest confidence

User Study

The GUI used to train the agent with given commands.

Experimental Setup

- Constant Fast Condition (0.5 seconds/step)
- Constant Slow Condition (2.0 seconds/step)
- Adaptive Condition (0.5 ~ 2.0 s/step)
  - Adaptive Condition without Instructions
  - Adaptive Condition with Instructions

60 user data on Amazon Mechanical Turk, 30 participants for each training condition

Participant Performance

- Compared to the constant slow speed agent: the adaptive speed agent can use less wall clock time but similar number of actions.

- Compared to the constant fast speed agent: the adaptive speed agent can take fewer actions but not worse wall clock time.

Our novel adaptive speed agent *dominates* different fixed speed agents on these four evaluation metrics.

More participants preferred the constant speed agents!

For participants who preferred adaptive speed agent

When should an agent behave to maximize performance metrics, even if doing so would not maximize participant satisfaction?

Conclusion for LAMBDAS

• Agent better able to learn from human feedback: adapting its speed based on its confidence

• Dominates different fixed speed agents on several evaluation metrics

Motivation
Learn from Rewards: SABL
Implicit Agent-Trainer Communication: LAMBDAS
FYI, FWIW, IMHO: Embodiment, Curricula, etc.
James MacGlashan, physical robot training

- Punishments being given
- Rewards being given
Rewards are Policy Dependent: Mark Ho

Click 'Go' to start today's training.

Punish

Bad Dog

Do Nothing

Reward

Good dog!
Rewards are Policy Dependent: Mark Ho

Case 1: Good Job!
Rewards are Policy Dependent: Mark Ho

Case 1: Good Job!

Case 2: Bad Dog!
Rewards Depend on Appearance?


http://consequentialrobotics.com/miro/
Can an agent learn to predict the benefit of transferring a policy from one task to another?

Yes!

Using learned model, agent was able to select source task that improved learning on target tasks.
A library of environments provided in a 4x4 grid.

Humans can help (relative to random curricula)

Trainer is:
Correct
Error-Prone, or
Entropy Driven

Future Work for Curriculum Design

• Allow participants to create a sequence of novel source tasks

• Show the score (# explicit feedback / # time steps needed to learn the correct reward function) of the designed curricula to motivate participants to design better ones

• Implement an algorithm that can leverage all interesting salient patterns followed by non-expert humans

What’s in a Reward?
Teaching Agents and Robots with Non-Expert Reinforcement

irll.eecs.wsu.edu
taylorm@eecs.wsu.edu