Reinforcement Learning with Human Feedback

Aijun Bai

Multi-Agent Systems Lab., USTC

Nov 24, 2011

Outline

Introduction

- Autonomous *tabula rasa* learning either is intractable or takes too long in practice;
- In some domains, humans always have some valuable intuition or expertise;
- It is necessary to transfer human knowledge to learning agents to reduce learning time in such domains.

TAMER (meaning Training an Agent Manually via Evaluative Reinforcement) is a general framework for this purpose [1].

The TAMER Framework

- Assume that the human trainer is already taking each action's long-term implications into account when providing feedback;
- TAMER uses established supervised learning techniques to model a hypothetical human reinforcement function, $H : S \times A \rightarrow R$, treating the scalar human reinforcement value as a label for a state-action sample;
- To choose action in state s, a TAMER agent directly exploits the learned model $\hat{H}$ of expected reinforcement by $a = \mathrm{argmax}_a \hat{H}(s, a)$.

Algorithm

1. Observe environment state;
2. Choose an action based on $\hat{H}$ model;
3. Execute the action and observe human feedback;
4. Update $\hat{H}$ model by human feedback if any;
5. Goto Step 1.

The Experimental Domain: Tetris

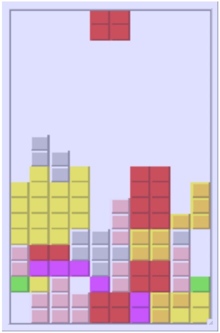

Figure: 10 × 20 Tetris

Experiments Settings:

- 22 features to describe state space;
- A linear function to approximate $\hat{H}$;
- A gradient rule to update $\hat{H}$.

Empirical Results

- By the third game, on average, performance reached an approximate peak of 65.88 lines cleared per game;
- Compared to autonomous agents, this is incredibly fast;
- As the number of training episodes increases, however, many of the autonomous agents outperform the TAMER agent.

Conclusion

The TAMER human-training framework:

1. Has a simple interface;
2. Is relatively easy to implement;
3. Can increase learning speed a lot;
4. Can not guarantee an optimal policy.

Introduction

- TAMER dose not allow human training to be combined with autonomous learning;
- This paper examines how to best combine the TAMER framework with RL (namely TAMER+RL) [2].

Specifically, this paper focuses on the scenario in which a human trainer has already trained a TAMER agent, and the learned human reinforcement function, $\hat{H}$, is available to guide a reinforcement learning agent.

The TAMER+RL Framework

1. Train a TAMER agent by human reinforcement feedback;
2. Aid the learning of a RL agent by using the knowledge of the previously trained TAMER agent.

Recall the SARSA update rule

1. $a_t = \text{argmax}_a\, Q(s_t, a)$ with probability $(1 - \varepsilon)$ or $random(A)$ with probability ε
2. $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R(s_t, a_t) + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$

Techniques for Combining TAMER and RL

The eight techniques for combining TAMER and RL:

1. $R'(s, a) = R(s, a) + weight \times \hat{H}(s, a)$;
2. $\vec{f} = [\vec{f}, \hat{H}(s, a)]$;
3. Initially train $Q(s, a)$ to approximate $constant \times \hat{H}(s, a)$;
4. $Q'(s, a) = Q(s, a) + constant \times \hat{H}(s, a)$;
5. $A' = A \cup \mathrm{argmax}_a \hat{H}(s, a)$;
6. $a = \mathrm{argmax}_a (Q(s, a) + weight \times \hat{H}(s, a))$;
7. $P(a = \mathrm{argmax}_a \hat{H}(s, a)) = p$;
8. $R'(s_t, a) = R(s_t, a) + constant \times (U(s_t) - U(s_{t-1}))$, where $U(s) = \max_a \hat{H}(s, a)$.

Success Metric

Test each combining technique both with optimistic and pessimistic initializations in the Mountain Car domain, comparing with RL (SARSA(λ)) and TAMER alone:

- **End Performance**: achieve a higher final performance than either RL or TAMER alone;
- **Cumulative Reward**: receive more reward over given episodes (500) than either RL or TAMER alone.

Empirical Results

1. Pessimistic Initializations:
   1. End Performance:
      - Improvement: Methods 1, 3, 4, 6 and 7
      - Marginal Improvement: Method 8
   2. Cumulative Reward:
      - Improvement: Methods 3, 6 and 7
2. Optimistic Initializations:
   1. End Performance:
      - Improvement: Method 1
   2. Cumulative Reward:
      - Improvement: Methods 4, 6 and 7

Comparing the combination techniques

1. Initially manipulating the model of Q correlates with poor performance: Methods 2, 3 and 5
2. Gently pushing the behavior of the learning agent toward what the TAMER agent would do and removing the influence of $\hat{H}$ slowly and smoothly correlates with good performance: Methods 1, 6 and 7

Optimistic versus Pessimistic Initialization

Observations:

1. SARSA(λ) performs best with optimistic initialization;
2. TAMER+RL almost uniformly performs best with pessimistic initialization.

Analysis:

1. Optimistic initialized Q values (including undesired actions) can only go down during learning process;
2. The only way to learn the correct Q values for undesired actions is by choosing them;
3. But the TAMER agent will not choose them in priority.

Conclusion

The TAMER+RL framework:

- Allow an agent designer to capture task knowledge from a human trainer;
- Use that knowledge to improve the performance of reinforcement learning algorithms.

Suitable domains:

- Tasks which require much exploration before discriminatory reward is received;
- Tasks in which local maximums make the best solution difficult to find;
- When the task has a noisy MDP reward signal.

References

W. B. Knox and P. Stone.
Tamer: Training an agent manually via evaluative reinforcement.
2008 7th IEEE International Conference on Development and Learning, (August):292–297, 2008.

W. B. Knox and P. Stone.
Combining manual feedback with subsequent mdp reward signals for reinforcement learning.
In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS '10, pages 5–12, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.